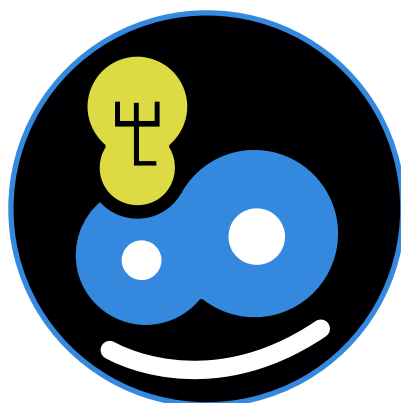




ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη πλαισίου εκπαιδευτικής εφαρμογής - Μελέτη περίπτωσης με θέμα τους Εξελικτικούς αλγορίθμους



Του φοιτητή  
Σαράντη Συράκη  
Αρ. Μητρώου: 174973

Επιβλέπων  
Κεραμόπουλος Ευκλείδης  
Καθηγητής

Θεσσαλονίκη, Ιανουάριος 2024

Τίτλος Δ.Ε. Ανάπτυξη πλαισίου εκπαιδευτικής εφαρμογής  
Μελέτη περίπτωσης με θέμα τους Εξελικτικούς αλγορίθμους

Κωδικός Δ.Ε. 22184

Ονοματεπώνυμο φοιτητή/τών Σαράντης Συράκης

Ονοματεπώνυμο εισηγητή Παναγιώτης Αδαμίδης

Ημερομηνία ανάληψης Δ.Ε. 22/3/22

Ημερομηνία περάτωσης Δ.Ε. 30/1/24

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Σαράντη Συράκη που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

*«Αφιερώνω αυτή την εργασία σε όλους του φοιτητές που αντιμετωπίζουν δυσκολίες στην κατανόηση των εξελικτικών αλγορίθμων, ελπίζοντας ότι θα τους φανεί χρήσιμη, καθώς και σε όλους του καθηγητές που θα επιθυμούσαν να τροποποιήσουν αυτό το εκπαιδευτικό εργαλείο για τον εμπλουτισμό του μαθήματος τους.»*

*Αλλά πάνω από όλα σε όσους νομίζουν ότι δεν είναι σε θέση να καταφέρουν κάτι, να μην τα παρατάνε και να θυμούνται ότι αρκεί μόνο να κάνουν το πρώτο βήμα...»*

## Πρόλογος

Η επιλογή της συγκεκριμένης εργασίας έγινε εξαιτίας του ενδιαφέροντος που αναπτύχθηκε κατά τη διδασκαλία του μαθήματος των Ευφών Συστημάτων πάνω στην ενότητα των Εξελικτικών αλγορίθμων.

Με την ενασχόληση πάνω στη συγκεκριμένη εργασία μπόρεσα να κατανοήσω καλύτερα τους εξελικτικούς αλγόριθμους, καθώς και να μπω στη διαδικασία να προσπαθήσω να μεταφέρω αυτό το ενδιαφέρον και σε άλλους, βρίσκοντας διάφορους τρόπους οπτικοποίησης κάποιων ιδιαίτερα ενδιαφερόντων τομέων τους.

Η έμπνευση για το εκπαιδευτικό κομμάτι της εφαρμογής προήλθε από μια άλλη εφαρμογή που παρουσιάστηκε στα πλαίσια του μαθήματος.

Η συγκεκριμένη εφαρμογή είχε ένα πολύ παράδειγμα εκτέλεσης ενώ εξελικτικού αλγόριθμου για το πρόβλημα της βελτιστοποίησης κοπής συγκριμένων πατρών σε περιορισμένο εμβαδόν υφάσματος.

Αυτό που μου έκανε εντύπωση είναι πόσο πιο ξεκάθαρος έγινε ο τρόπος λειτουργίας του αλγορίθμου καθώς και πόσο ενδιαφέρον ήταν η συνεχής αλλαγή των στατιστικών μετρήσεων από γενιά σε γενιά, αλλά γενικότερα η χρήση μιας εφαρμογής εντός της τάξης για τη μεταλαμπάδευση γνώσης.

Ο δεύτερος λόγος ήταν για αυξήσω το ποσοστό των γνώσεων μου πάνω στις τεχνολογίες WEB (HTML, CSS, JavaScript, κτλ) που αποτελούν την πιο χρησιμοποιημένη κολεκτίβα εργαλείων για την ανάπτυξη εφαρμογών τη σημερινή εποχή.

Είναι προφανές ότι αυτές οι τεχνολογίες έχουν καθιερώσει τη θέση τους στην εργαλειοθήκη των περισσότερων προγραμματιστών καθώς είναι σε θέση να δημιουργούν εξαιρετικής ποιότητας εφαρμογές, που μπορούν να αξιοποιηθούν άμεσα από πάρα πολλούς χρήστες με τη χρήση ενός περιηγητή ιστού (web browser), μια κλάση προγραμμάτων που οι λειτουργείς τους είναι άμεσα γνωστές στους περισσότερους σημερινούς χρήστες.

## Περίληψη

Σκοπός της τρέχουσας διπλωματικής εργασίας είναι η ανάπτυξη μιας εκπαιδευτικής εφαρμογής, με μελέτη περίπτωσης (case study) τους Εξελικτικούς Αλγόριθμους. Αυτή η εφαρμογή χρησιμοποιεί διαδραστικές τεχνικές για να ενισχύσει την εμπέδωση των γνώσεων που αποκτώνται από τη διδασκαλία με φυσική παρουσία (παράδοση μαθήματος, συζήτηση, ερωτήσεις και επίλυση προβλημάτων). Η εφαρμογή παρέχει ένα περιβάλλον στο οποίο ο χρήστης μπορεί να αλληλεπιδρά με παραμετροποιήσιμες και οπτικοποιημένες ασκήσεις που διδάσκονται κατά τη διάρκεια του μαθήματος και είναι διαθέσιμες στα αρχεία ύλης που χρησιμοποιούνται στο μάθημα καθώς και τη δυνατότητα ανασκόπησης της απαιτούμενης ύλης του μαθήματος ανάλογα με την άσκηση που παρουσιάζεται. Πολλές από αυτές τις ασκήσεις παρέχουν διάφορους στόχους που πρέπει να ολοκληρώσει ο χρήστης, γεγονός που ενισχύει την παιχνιδιοποίηση της εφαρμογής. Επίσης, παρέχεται έλεγχος των αποκτηθέντων γνώσεων με τη χρήση κουίζ. Αυτά τα κουίζ μπορούν να έχουν ερωτήσεις πολλαπλής επιλογής, σωστού-λάθους, καθώς και ερωτήσεις ελεύθερου κειμένου για την αξιολόγηση του χρήστη στα θέματα που έχει μελετήσει. Επιπλέον, όλα τα αρχεία ύλης εμπλουτίζονται με υπερσυνδέσμους προς εξωτερικούς ιστότοπους που σχετίζονται άμεσα με την ύλη του μαθήματος, καθώς και με τη δυνατότητα καταγραφής σημειώσεων για κάθε ένα από τα έγγραφα. Η εφαρμογή παρέχει ένα σύστημα ειδοποιήσεων των συνδεδεμένων χρηστών από τον διδάσκοντα καθηγητή καθώς και διάφορες ρυθμίσεις, επιτρέποντάς την εξατομίκευση της στις προτιμήσεις του εκάστοτε χρήστη. Επιπροσθέτως, έχει γίνει προσπάθεια τυποποίησης της εφαρμογής ώστε να μπορεί να προσαρμοστεί σε άλλα μαθήματα. Ουσιαστικά, αυτή η εφαρμογή έχει σχεδιαστεί για να παρέχει ένα πλαίσιο πάνω στο οποίο κάποιος διδάσκων θα μπορούσε να αναπτύξει τη δική του εκπαιδευτική εφαρμογή για το μάθημα που τον ενδιαφέρει. Η επιλογή των εξελικτικών αλγορίθμων έγινε λόγω του προσωπικού ενδιαφέροντος που αναπτύχθηκε για το συγκεκριμένο γνωστικό αντικείμενο κατά την παρακολούθηση του αντίστοιχου μαθήματος.

# Development of an educational application framework – A case study in Evolutionary Algorithms

Sarantis Syrakis

## **Abstract**

This thesis aims to develop an educational application, using Evolutionary Algorithms as a case study. The application employs interactive techniques to enhance the consolidation of knowledge acquired through face-to-face teaching (including lesson delivery, discussions, questions and problem solving). It offers an environment where users can interact with customizable and visualized exercises covered in the course, available both in the material files used in the class and during lessons. Users can also review the necessary course material based on the presented exercise. Many of these exercises present various goals for the user to complete, enhancing gamification within the app. Additionally, the application features knowledge assessments through quizzes, including multiple-choice, true-false, and free-text questions to evaluate users on the studied topics. Moreover, all material files are enriched with hyperlinks to external websites directly related to the course content, and users can record notes for each document. The application includes a notification system for connected users from the teacher or trainer and various settings, allowing for personalization according to each user's preferences. Furthermore, efforts have been made to standardize the application for adaptability to other courses. Essentially, this app is designed to serve as a framework, empowering teachers to develop their own educational apps on subjects of interest. The choice of evolutionary algorithms was made due to the personal interest developed for the specific subject while attending the respective course.

## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω από τα βάθη της καρδιάς μου τη μητέρα μου και τον πατέρα μου, για όλη τους τη βοήθεια και τη συμπαράσταση κατά τη διάρκεια εκπόνησης της συγκεκριμένης εργασίας καθώς και τον καθηγητή μου, τον κύριο Παναγιώτη Αδαμίδη, για την πολύτιμη βοήθεια, τις συμβουλές και τις υποδείξεις που μου έκανε, καθώς χωρίς αυτές δε θα μπορούσε να ολοκληρωθεί η διπλωματική εργασία.

# Περιεχόμενα

|                                            |      |
|--------------------------------------------|------|
| Πρόλογος.....                              | iv   |
| Περίληψη.....                              | v    |
| Abstract.....                              | vi   |
| Ευχαριστίες.....                           | vii  |
| Περιεχόμενα.....                           | viii |
| Κατάλογος Σχημάτων.....                    | xii  |
| Συντομογραφίες.....                        | xv   |
| Κεφάλαιο 1ο: Εισαγωγή.....                 | 1    |
| Κεφάλαιο 2ο: Θεωρητικό Πλαίσιο.....        | 2    |
| 2.1 Εισαγωγή.....                          | 2    |
| 2.2 Εκπαιδευτικές εφαρμογές.....           | 2    |
| 2.3 Εξελικτικοί αλγόριθμοι.....            | 4    |
| 2.4 Επίλογος.....                          | 5    |
| Κεφάλαιο 3ο: Τεχνολογίες και Εργαλεία..... | 6    |
| 3.1 Εισαγωγή.....                          | 6    |
| 3.2 JavaScript.....                        | 6    |
| 3.2.1 DOM.....                             | 6    |
| 3.2.2 Fetch API.....                       | 7    |
| 3.2.3 LocalStorage.....                    | 7    |
| 3.2.4 Clipboard API.....                   | 7    |
| 3.2.5 Canvas API.....                      | 7    |
| 3.2.6 Web Workers API.....                 | 8    |
| 3.2.7 Service Worker API.....              | 8    |
| 3.2.8 Web Components.....                  | 8    |
| 3.3 Bootstrap.....                         | 8    |
| 3.4 Google Material Icons.....             | 9    |
| 3.5 HTML.....                              | 9    |
| 3.6 JSON.....                              | 9    |
| 3.7 Plotly.js.....                         | 10   |
| 3.8 Suneditor.....                         | 10   |
| 3.9 PDF.js.....                            | 10   |
| 3.10 Node.js.....                          | 11   |

|                                                               |    |
|---------------------------------------------------------------|----|
| 3.11 Express.js.....                                          | 11 |
| 3.12 Socket.io.....                                           | 11 |
| 3.13 Ιστοσελίδες περιεχομένου.....                            | 11 |
| 3.13.1 SVG Repo.....                                          | 12 |
| 3.13.2 Zapsplat.....                                          | 12 |
| 3.14 Visual Studio Code.....                                  | 12 |
| 3.15 Inkscape.....                                            | 13 |
| 3.16 Επίλογος.....                                            | 13 |
| Κεφάλαιο 4ο: Ανάλυση Μεθοδολογίας και Συστημάτων.....         | 14 |
| 4.1 Εισαγωγή.....                                             | 14 |
| 4.2 Βασική αρχιτεκτονική.....                                 | 14 |
| 4.3 Διαίρει και βασίλευε.....                                 | 14 |
| 4.4 JavaScript Modules.....                                   | 15 |
| 4.5 Περιγραφή δομής βασικού συστήματος.....                   | 18 |
| 4.6 Fetcher.....                                              | 18 |
| 4.7 View Components.....                                      | 19 |
| 4.8 AudioManager.....                                         | 20 |
| 4.9 ModalManager.....                                         | 21 |
| 4.10 ToastManager.....                                        | 23 |
| 4.11 LoadingScreen.....                                       | 24 |
| 4.12 Cdn Files.....                                           | 24 |
| 4.13 Content Files.....                                       | 25 |
| 4.14 AuxiliaryScripts.....                                    | 25 |
| 4.15 Επίλογος.....                                            | 26 |
| Κεφάλαιο 5ο: Παρουσίαση Εφαρμογής.....                        | 27 |
| 5.1 Εισαγωγή.....                                             | 27 |
| 5.2 Αρχική οθόνη και τρόπος λειτουργίας των views.....        | 27 |
| 5.2.1 Συμπληρωματικές λεπτομέρειες.....                       | 32 |
| 5.3 Σύστημα βοήθειας.....                                     | 33 |
| 5.3.1 Συμπληρωματικές λεπτομέρειες.....                       | 34 |
| 5.4 Έγγραφα ύλης και οι συμπληρωματικές τους λειτουργίες..... | 37 |
| 5.4.1 Εισαγωγή.....                                           | 37 |
| 5.4.2 View Υλικού Μελέτης (Study Material).....               | 37 |

|                                                                          |    |
|--------------------------------------------------------------------------|----|
| 5.4.3 View Εμφάνισης Εγγράφου.....                                       | 40 |
| 5.4.4 Συμπληρωματικές λεπτομέρειες.....                                  | 47 |
| 5.5 Ασκήσεις εφαρμογής.....                                              | 50 |
| 5.5.1 Εισαγωγή.....                                                      | 50 |
| 5.5.2 View εμφάνισης ασκήσεων (Exercises).....                           | 51 |
| 5.5.3 Διευκρίνιση αρχιτεκτονικής ασκήσεων.....                           | 52 |
| 5.5.4 Τρόπος λειτουργίας των στόχων των ασκήσεων.....                    | 53 |
| 5.5.5 Παρουσίαση ασκήσεων.....                                           | 55 |
| 5.6 Κουίζ Εφαρμογής.....                                                 | 60 |
| 5.6.1 Εισαγωγή.....                                                      | 60 |
| 5.6.2 View εμφάνισης κουίζ (Quizzes).....                                | 60 |
| 5.6.3 View Προσπάθειας κουίζ.....                                        | 63 |
| 5.6.4 Συμπληρωματικές λεπτομέρειες.....                                  | 69 |
| 5.7 View Ρυθμίσεων εφαρμογής (Settings).....                             | 75 |
| 5.7.1 Θέμα εφαρμογής (Application Theme).....                            | 75 |
| 5.7.2 Ρυθμίσεις ήχου (Volume Control).....                               | 76 |
| 5.7.3 Ρυθμίσεις αυτόματης αποθήκευσης σημειώσεων (Editor Auto Save)..... | 76 |
| 5.7.4 Σημειώσεις καθηγητή (Teacher Notes).....                           | 76 |
| 5.7.5 Λειτουργία υποστήριξης εκτός σύνδεσης (Offline Support Mode).....  | 76 |
| 5.7.6 Ρυθμίσεις των Modals (Application Modals).....                     | 78 |
| 5.7.7 Ρυθμίσεις των Toasts (Application Toasts).....                     | 78 |
| 5.7.8 Διαχείριση δεδομένων εφαρμογής.....                                | 79 |
| 5.7.9 Επαναφορά ρυθμίσεων (Reset Settings To Default).....               | 79 |
| 5.7.10 Συμπληρωματικές λεπτομέρειες.....                                 | 79 |
| 5.8 View Ειδοποιήσεων εφαρμογής (Notifications).....                     | 80 |
| 5.8.1 Συμπληρωματικές λεπτομέρειες.....                                  | 81 |
| 5.9 View Παρουσίασης τεχνολογιών (About).....                            | 82 |
| 5.9.1 Λογότυπο εφαρμογής.....                                            | 83 |
| 5.9.2 Συμπληρωματικές λεπτομέρειες.....                                  | 83 |
| 5.10 Επίλογος.....                                                       | 84 |
| Κεφάλαιο 6ο: Λεπτομέρειες Εξυπηρετητή και Εγκατάστασης.....              | 85 |
| 6.1 Εισαγωγή.....                                                        | 85 |
| 6.2 Ο εξυπηρετητής.....                                                  | 85 |

|                                                                 |    |
|-----------------------------------------------------------------|----|
| 6.2.1 Δομή του αρχείου index.js.....                            | 85 |
| 6.3 Τρόποι εγκατάστασης.....                                    | 89 |
| 6.3.1 Τοπική εγκατάσταση.....                                   | 89 |
| 6.3.2 Προώθηση της τοπικής εγκατάστασης.....                    | 89 |
| 6.3.3 Χρήση ιδιωτικού εξυπηρετητή.....                          | 89 |
| 6.3.4 Πλατφόρμες ανάρτησης εφαρμογών Node.js.....               | 89 |
| 6.4 Επίλογος.....                                               | 90 |
| Κεφάλαιο 7ο: Συμπεράσματα και Προτάσεις Βελτίωσης.....          | 91 |
| 7.1 Υποστήριξη πολλαπλών γλωσσών.....                           | 91 |
| 7.2 Προσθήκη περισσότερων τύπων ερωτήσεων.....                  | 91 |
| 7.3 Δυνατότητα τυποποίησης των στόχων.....                      | 92 |
| 7.4 Υποστήριξη μεγαλύτερου φάσματος συσκευών.....               | 92 |
| 7.5 Προσθήκη περαιτέρω μεθόδων απόδοσης ύλης.....               | 92 |
| 7.6 Προσθήκη μεθόδων οργάνωσης σημειώσεων.....                  | 92 |
| 7.7 Μείωση εξάρτησης από τη βιβλιοθήκη Bootstrap.....           | 93 |
| 7.8 Συμμετοχή χρηστών στη δημιουργία κουίζ.....                 | 93 |
| 7.9 Προσθήκη παιχνιδιών σχετικών με το αντικείμενο μελέτης..... | 93 |
| BIBΛΙΟΓΡΑΦΙΑ.....                                               | 94 |

## Κατάλογος Σχημάτων

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| Σχήμα 2.1: Έσοδα εκπαιδευτικών εφαρμογών από 2017-2022 (σε δισεκατομμύρια ευρώ)..... | 3  |
| Σχήμα 2.2: Χρήστες εκπαιδευτικών εφαρμογών από το 2017-2020 (σε εκατομμύρια).....    | 3  |
| Σχήμα 4.1: Το export εμφανίζεται στην αρχή μιας συνάρτησης.....                      | 16 |
| Σχήμα 4.2: Γίνεται μόνο μια εγγραφή export στο τέλος του αρχείου.....                | 16 |
| Σχήμα 4.3: Δομή των import statements που εμφανίζονται στην αρχή του αρχείου.....    | 17 |
| Σχήμα 4.4: Συνάρτηση που κάνει χρήση δυναμικού import.....                           | 17 |
| Σχήμα 4.5: Εμφάνιση ενός Modal.....                                                  | 21 |
| Σχήμα 4.6: Εμφάνιση ενός Toast.....                                                  | 23 |
| Σχήμα 4.7: Εμφάνιση οθόνης φόρτωσης.....                                             | 24 |
| Σχήμα 5.1: Αρχική οθόνη εφαρμογής.....                                               | 27 |
| Σχήμα 5.2: Μενού πλοήγησης εφαρμογής.....                                            | 28 |
| Σχήμα 5.3: Εμφάνιση ενός view με τα κουμπιά πλοήγησης.....                           | 29 |
| Σχήμα 5.4: Μενού επιλογής ανοιχτών view.....                                         | 30 |
| Σχήμα 5.5: Κουμπιά διαχωρισμού οθόνης και βοήθειας.....                              | 30 |
| Σχήμα 5.6: Οριζόντιος διαχωρισμός οθόνης.....                                        | 31 |
| Σχήμα 5.7: Κατακόρυφος διαχωρισμός οθόνης.....                                       | 31 |
| Σχήμα 5.8: Κουμπιά έξτρα λειτουργιών διαχωρισμού οθόνης.....                         | 32 |
| Σχήμα 5.9: Εμφάνιση modal επιλογής view για τη λήψη βοήθειας.....                    | 34 |
| Σχήμα 5.10: Τμήμα του αρχείου help_definitions.json.....                             | 35 |
| Σχήμα 5.11: Παράδειγμα αρχείου βοήθειας.....                                         | 36 |
| Σχήμα 5.12: Προβολή του view που εμφανίζει τα αρχεία ύλης.....                       | 37 |
| Σχήμα 5.13: Τμήμα του αρχείου pdf_definitions.json.....                              | 38 |
| Σχήμα 5.14: Τμήμα του αρχείου socket-definitions.js.....                             | 39 |
| Σχήμα 5.15: Προβολή ενός εγγράφου ύλης.....                                          | 40 |
| Σχήμα 5.16: Μενού επιλογής εγγράφου.....                                             | 41 |
| Σχήμα 5.17: Μενού επιλογών εγγράφου.....                                             | 41 |
| Σχήμα 5.18: Μενού επιλογών ενός link.....                                            | 42 |
| Σχήμα 5.19: Μενού δημιουργίας σημειώσεων.....                                        | 42 |
| Σχήμα 5.20: Εμφάνιση view προβολής   επεξεργασίας σημείωσης.....                     | 43 |
| Σχήμα 5.21: Γκρουπ προσαρμοσμένων λειτουργιών.....                                   | 44 |

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| Σχήμα 5.22: Εμφάνιση κάρτας ανάκτησης σημειώσεων.....                   | 45 |
| Σχήμα 5.23: Εμφάνιση view ανάκτησης σημειώσεων.....                     | 45 |
| Σχήμα 5.24: Μενού δημιουργίας σελιδοδεικτών.....                        | 46 |
| Σχήμα 5.25: Εμφάνιση μιας εγγραφής link.....                            | 47 |
| Σχήμα 5.26: Εμφάνιση εγγραφής σημείωσης καθηγητή.....                   | 50 |
| Σχήμα 5.27: Εμφάνιση view ασκήσεων.....                                 | 51 |
| Σχήμα 5.28: Εμφάνιση προτύπου ασκήσεων.....                             | 52 |
| Σχήμα 5.29: Εμφάνιση στόχων μιας άσκησης.....                           | 53 |
| Σχήμα 5.30: Εμφάνιση ολοκληρωμένου στόχου.....                          | 54 |
| Σχήμα 5.31: Εγγραφή πεδίου ενός αρχείου στόχων.....                     | 54 |
| Σχήμα 5.32: Εμφάνιση view πρώτης άσκησης.....                           | 55 |
| Σχήμα 5.33: Εμφάνιση view δεύτερης άσκησης.....                         | 56 |
| Σχήμα 5.34: Εμφάνιση view τρίτης άσκησης.....                           | 57 |
| Σχήμα 5.35: Εμφάνιση view τέταρτης άσκησης.....                         | 58 |
| Σχήμα 5.36: Εμφάνιση view πέμπτης άσκησης.....                          | 59 |
| Σχήμα 5.37: Εμφάνιση view διαθέσιμων κουίζ.....                         | 60 |
| Σχήμα 5.38: Εμφάνιση modal πληροφοριών κουίζ.....                       | 61 |
| Σχήμα 5.39: Εμφάνιση modal στατιστικών.....                             | 62 |
| Σχήμα 5.40: Τμήμα αρχείου quiz_definitions.json.....                    | 62 |
| Σχήμα 5.41: Εμφάνιση view προσπάθειας κουίζ.....                        | 63 |
| Σχήμα 5.42: Εμφάνιση γραμμής ποσοστού συμπλήρωσης κουίζ.....            | 64 |
| Σχήμα 5.43: Εμφάνιση ερώτησης σωστού-λάθους.....                        | 64 |
| Σχήμα 5.44: Εμφάνιση μιας σωστά απαντημένης ερώτησης σωστού-λάθους..... | 65 |
| Σχήμα 5.45: Είδος προβολής σε στιλ καρουζελ.....                        | 66 |
| Σχήμα 5.46: Είδος προβολής πολλαπλών σελίδων.....                       | 67 |
| Σχήμα 5.47: Εμφάνιση διαθέσιμου χρόνου κουίζ.....                       | 67 |
| Σχήμα 5.48: Εμφάνιση modal τελικού αποτελέσματος κουίζ.....             | 68 |
| Σχήμα 5.49: Εμφάνιση σύνταξης μιας ερώτησης σωστού-λάθους.....          | 70 |
| Σχήμα 5.50: Εμφάνιση δομής ενός κουίζ.....                              | 71 |
| Σχήμα 5.51: Έξτρα πεδία ερώτησης με κουμπιά τσεκαρίσματος.....          | 72 |
| Σχήμα 5.52: Εμφάνιση τμήματος του view των ρυθμίσεων.....               | 75 |
| Σχήμα 5.53: Εμφάνιση εφαρμογής με το σκουρόχρωμο θέμα.....              | 75 |

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| Σχήμα 5.54: Περιεχόμενα αρχείου service-worker-initialization.js..... | 77 |
| Σχήμα 5.55: Περιεχόμενα αρχείου sw.js.....                            | 78 |
| Σχήμα 5.56: Προβολή view ειδοποιήσεων.....                            | 80 |
| Σχήμα 5.57: Εμφάνισή νέας ειδοποίησης.....                            | 81 |
| Σχήμα 5.58: Εμφάνιση του αριθμού των ειδοποιήσεων στο μενού.....      | 81 |
| Σχήμα 5.59: Εμφάνιση κάρτας τεχνολογίας.....                          | 82 |
| Σχήμα 5.60: Λογότυπο εφαρμογής.....                                   | 83 |
| Σχήμα 5.61: Εγγραφή του αρχείου about.json.....                       | 83 |
| Σχήμα 6.1: Πρώτο τμήμα αρχείου index.js.....                          | 85 |
| Σχήμα 6.2: Εγγραφές αρχείων προς παρακολούθηση.....                   | 86 |
| Σχήμα 6.3: Ορισμός κάποιων διαδρομών στο αρχείο index.js.....         | 87 |
| Σχήμα 6.4: Εμφάνιση αρχικής οθόνης /teacher.....                      | 88 |

## Συντομογραφίες

|       |                              |
|-------|------------------------------|
| Δ.Ε.  | Διπλωματική Εργασία          |
| ΔΙΠΑΕ | Διεθνές Πανεπιστήμιο Ελλάδος |
| Π.Ε.  | Πτυχιακή Εργασία             |

## Κεφάλαιο 1ο: Εισαγωγή

Η τεχνολογία αγγίζει κάθε πτυχή της ζωής μας. Με αυτή την πανταχού παρουσία της έχει καταφέρει να αλλάξει τον τρόπο που σκεφτόμαστε και ενεργούμε σε πάρα πολλούς τομείς της ζωής μας. Ένας από αυτούς τους τομείς είναι η εκπαίδευση. Η εκπαίδευση ήταν ανέκαθεν ανεκτίμητης αξίας για τη σύνθεση μιας κοινωνίας από σκεπτόμενους και ενσυναίσθητους ανθρώπους, με τη δημιουργία ώριμων και ενεργών πολιτών, ιδιαίτερα τώρα που υπάρχει τεράστιος όγκος πληροφοριών που οφείλει να γνωρίζει κάποιος.

Για αυτόν τον λόγο είναι απαραίτητος ο εκσυγχρονισμός των μεθόδων που χρησιμοποιούνται σήμερα στην εκπαίδευση. Η υιοθέτηση όμως νέων μεθόδων και εργαλείων στην εκπαίδευση δε συμβαίνει με τους ίδιους ρυθμούς όπως σε άλλους τομείς.

Πολλές φορές οι ίδιοι οι καθηγητές δεν είναι πρόθυμοι και έτοιμοι να κάνουν αυτή την αλλαγή, και κάποιες άλλες τα εκπαιδευτικά ιδρύματα δε θέτουν ως προτεραιότητα τον εκσυγχρονισμό των εκπαιδευτικών τους εργαλείων και πρακτικών για την παροχή καλύτερων και πιο διαδραστικών εμπειριών εκπαίδευσης [1], [2].

Με στόχο την αντιμετώπιση κάποιων από τα προαναφερθέντα προβλήματα έγινε η ανάπτυξη της εν λόγω εφαρμογής. Η δημιουργία ενός πλαισίου ανοιχτού κώδικα που θα επιτρέπει την εύκολη και λειτουργική προσαρμογή υπάρχουσας ύλης σε ένα εργαλείο που θα είναι άμεσα προσβάσιμο από έναν απλό υπολογιστή με σύνδεση στο διαδίκτυο ήταν ένας από τους κύριους στόχους της εφαρμογής.

Εκτός αυτού η δυνατότητα ελέγχου των αποκτηθέντων γνώσεων των μαθητών με τη χρήση κουίζ και η παράθεση οπτικοποιημένων ασκήσεων που θα επιτρέπουν στον χρήστη να κάνει καλύτερη αφομοίωση της ύλης του μαθήματος ήταν εξίσου σημαντικές στη συμπερίληψη.

Η εφαρμογή είχε ως κύριο γνώμονα την εύκολη παραμετροποίηση όλων των διαθέσιμων λειτουργιών που προσφέρει, έτσι ώστε να μπορεί να είναι όσο το δυνατόν πιο προσαρμόσιμη για να μπορέσει να ανταπεξέλθει στις απαιτήσεις των χρηστών της.

Ο λόγος που έγινε η επιλογή των εξελικτικών αλγορίθμων ως περίπτωση μελέτης είναι γιατί εμφανίζουν εξαιρετικό ενδιαφέρον ως αντικείμενο και έχουν εφαρμογή σε ευρύ πεδίο γνωστικών αντικειμένων και προβλημάτων. Γενικά εφαρμόζονται σε οποιαδήποτε πεδίο έχει ανάγκη αναζήτησης βέλτιστης λύσης (βελτιστοποίηση).

Τα παραδοτέα της εργασίας είναι η εφαρμογή σε μορφή εκτελέσιμου αρχείου καθώς και όλος ο πηγαίος κώδικας που δημιουργήθηκε κατά τη διάρκεια εκπόνησής της.

Στο τρέχον κείμενο παρατίθεται μία παρουσίαση των δυνατοτήτων της εφαρμογής μαζί με τις τεχνολογίες που χρησιμοποιεί καθώς και μία μορφή εγχειριδίου για την προσαρμογή της στις εκάστοτε απαιτήσεις των χρηστών της.

Συγκεκριμένα, στα επόμενα κεφάλαια ξεκινώντας από το δεύτερο, γίνεται η ανάλυση του θεωρητικού πλαισίου, δηλαδή μια σύντομη επεξήγηση των εξελικτικών αλγορίθμων, των μεθόδων παραδοσιακής διδασκαλίας τους και τα αποτελέσματα αυτών καθώς και τους λόγους που έγινε η επιλογή των κουίζ ως κύριος λόγος εξέτασης των αποκτηθέντων γνώσεων.

Στο τρίτο κεφάλαιο γίνεται η παράθεση όλων των εργαλείων και τεχνολογιών που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Ακολούθως στο τέταρτο κεφάλαιο γίνεται η μελέτη της μεθοδολογίας και των συστημάτων έτσι ώστε να προετοιμαστεί το έδαφος για το επόμενο κεφάλαιο, το πέμπτο, όπου γίνεται η παρουσίαση της εφαρμογής από τη μεριά του χρήστη. Στο έκτο κεφάλαιο αναφέρονται τα τμήματα του κώδικα που εκτελεί ο εξυπηρετητής της εφαρμογής καθώς και τον τρόπο δοκιμής της εφαρμογής σε τοπικό σύστημα. Τέλος, στο έβδομο κεφάλαιο αναφέρονται τα συμπεράσματα και κάποιες προτάσεις βελτίωσης της εφαρμογής.

## Κεφάλαιο 2ο: Θεωρητικό Πλαίσιο

### 2.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται η παράθεση κάποιων χαρακτηριστικών που χαρακτηρίζουν τις εκπαιδευτικές εφαρμογές καθώς και μια συνοπτική παρουσίαση μιας πολύ διάσημης εφαρμογής.

Εκτός αυτού γίνεται μια σύντομη περιγραφή του τρόπου λειτουργίας ενός γενετικού αλγόριθμου καθώς και κάποιες ιδιαίτερες πτυχές του που έχουν να κάνουν με τον τρόπο που διδάσκονται καθώς και με τον τρόπο που πρέπει να χρησιμοποιούνται.

### 2.2 Εκπαιδευτικές εφαρμογές

Τη σήμερον ημέρα η χρήση της τεχνολογίας είναι αναπόσπαστο κομμάτι της πραγματικότητας μας.

Ο πότε η σταδιακή εμφάνιση της και στους τομείς που έχουν να κάνουν με την εκπαίδευση είναι φυσικό επακόλουθο. Αλλά ακόμα και στις αρχές του εικοστού αιώνα η χρήση της τότε υπάρχουσας τεχνολογία για εκπαιδευτικούς σκοπούς δεν έπαυε.

Το 1943 φτιάχτηκε ένα σύστημα συνθετικού εκπαιδευτή ραντάρ τύπου 19, που είχε ως σκοπό τη δημιουργία προσομοιωμένων δεδομένων οργάνων, έτσι ώστε οι πιλότοι να μπορούσαν να εκπαιδευτούν κατάλληλα πριν την πραγματική του επιβίβαση σε πραγματικό αεροπλάνο [3].

Βέβαια, επιστρέφοντας στη σήμερον εποχή, εξαιτίας του απρόβλεπτου παράγοντα της πανδημίας που εμφανίστηκε, έπρεπε να γίνει η ραγδαία υιοθέτηση εκπαιδευτικών εφαρμογών και άλλων παρόμοιων εργαλείων για να μπορέσει να συνεχιστεί η διαδικασία της εκπαίδευσης έστω και εξ αποστάσεως με τη χρήση εφαρμογών τηλεδιάσκεψης.

Λόγο αυτής της ξαφνικής μαζικής υιοθέτησης έγινε κατανοητό σε όλους πόσο σημαντικές είναι αυτού του είδους οι εφαρμογές. Χάρη στη νέα προσοχή που δόθηκε από το ευρύ κοινό προκάλεσε ραγδαίες καινοτομίες καθώς και αύξηση των διαθέσιμων εφαρμογών.

Όλα αυτά μπορούν να παρατηρηθούν στο πόσα έσοδα έφεραν οι εκπαιδευτικές εφαρμογές το έτος 2022 καθώς και την αύξηση του αριθμού των χρηστών τέτοιων εφαρμογών που αυξήθηκε από 185 εκατομμύρια το 2019 στα 270 εκατομμύρια το 2020 [4].

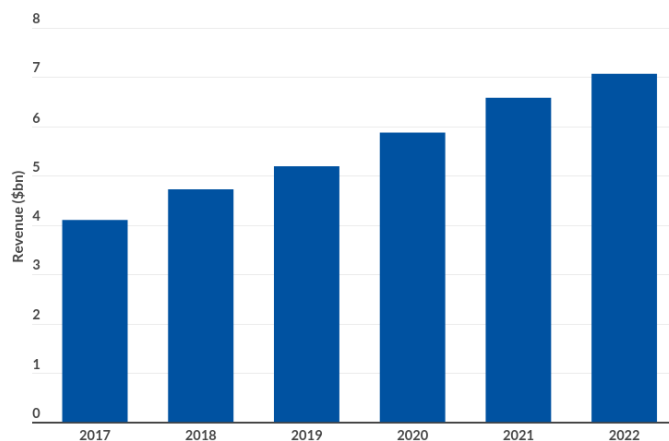
Τα γραφήματα που εμφανίζονται στο Σχήμα 2.1 και στο Σχήμα 2.2 δείχνουν την αυξητική πορεία των προαναφερθέντων στοιχείων.

Ένας από τους κύριους λόγους που υπάρχει τέτοιου είδους άνθηση στον τομέα των εκπαιδευτικών εφαρμογών είναι εξαιτίας του γεγονότος ότι υπάρχει τεράστια χρήση των smartphone τη σήμερον εποχή.

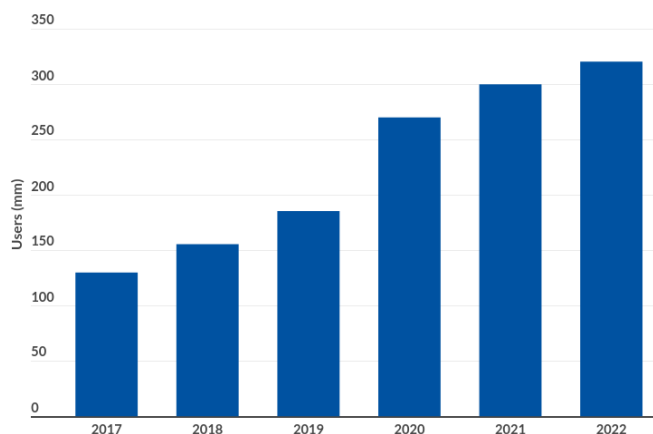
Πιθανολογείτε ότι υπάρχουν πιο πολλές κινητές συσκευές (smartphones, tablets, κτλ) από ότι υπάρχουν άτομα στον πλανήτη. Λόγου αυτού του γεγονότος έχουν γίνει η κύρια πλατφόρμα μέσω της οποία δουλεύουν ένα μεγάλο πλήθος εκπαιδευτικών εφαρμογών. Το γεγονός ότι είναι διαθέσιμα οποιαδήποτε στιγμή επιθυμούν οι χρήστες κάνει και τις ίδιες τις εφαρμογές διαθέσιμες. Εκτός αυτού το γεγονός πως αυτές οι συσκευές είναι σχεδόν πάντα συνδεδεμένες με το διαδίκτυο επιτρέπει στους χρήστες να έχουν άμεση πρόσβαση με μια πληθώρα εκπαιδευτικών πόρων. Όταν υπάρχει αυτή

## Κεφάλαιο 2ο:

η διαθεσιμότητα είναι πολύ πιο εύκολο για τον χρήστη να διαλέξει ο ίδιος που και πότε θέλει να αλληλεπιδράσει με την εφαρμογή, χωρίς να χρειάζεται να νοιάζεται για τυχόν χρονοδιάγραμμα που υπάρχουν στα εκπαιδευτικά ιδρύματα. Μπορεί επίσης να έχει καταγραφή της προόδου του, έχοντας έτσι τη δυνατότητα να προσαρμόσει καλύτερα τον χρόνο που πρέπει να αφιερώσει στην εφαρμογή αλλά και να έχει μια οπτική του πως προοδεύει μέσα στον χρόνο.



*Σχήμα 2.1: Έσοδα εκπαιδευτικών εφαρμογών από 2017-2022 (σε δισεκατομμύρια ευρώ)*



*Σχήμα 2.2: Χρήστες εκπαιδευτικών εφαρμογών από το 2017-2020 (σε εκατομμύρια)*

Έτσι μπορεί να γνωρίζει σε πια θέματα πρέπει να δώσει περαιτέρω προσοχή. Βέβαια, όλα αυτά τα προτερήματα δεν περιορίζονται μόνο σε εκπαιδευτικές εφαρμογές για smartphone αλλά υπάρχουν στις περισσότερες εφαρμογές αυτού του τύπου.

Επίσης, εκτός αυτών ο διαδραστικός χαρακτήρας σε συνδυασμό με τη συχνή χρήση της ιδέας του gamification (ονομάζεται μια στρατηγική επαύξηση συστημάτων με λειτουργίες που εμφανίζονται σε παιχνίδια [π.χ. βαθμολόγηση, ανταγωνισμός με άλλους, κανόνες παιχνιδιού] για την απόκτηση των

θετικών οφελών που χαρακτηρίζουν τα παιχνίδια [5]) που έχουν αυτές οι εφαρμογές τις κάνει ιδιαίτερα ευχάριστες ως προς τη χρήση τους, πράγμα που έχει ως αποτέλεσμα την αύξηση της ικανοποίησης που λαμβάνει ένας χρήστης και την περαιτέρω ενθάρρυνση του να συνεχίσει να τις χρησιμοποιεί. [6], [7]

Ένα παράδειγμα μιας τέτοιας εφαρμογής είναι το Duolingo. Είναι η μεγαλύτερη εκπαιδευτική εφαρμογή στον κόσμο ως προς τον αριθμό λήψεων της που ανέρχεται στις 16.3 εκατομμύρια τον Σεπτέμβριο του 2023 [8].

Σκοπό έχει την εκμάθηση ξένων γλωσσών με τρόπο που δίνει έμφαση στην παροχή εξατομικευμένης εκπαίδευσης για κάθε χρήστη ενώ παράλληλα στοχεύει στο να κάνει τη διαδικασία μάθησης όσο πιο ευχάριστη γίνεται [9].

Παρατηρήθηκε πως με τη χρήση του Duolingo υπήρξε μια στατιστικά σημαντική βελτίωση στις ικανότητες χρήσης της εν λόγω γλώσσας καθώς και μεγάλη ικανοποίηση με τη διαδικασία διεκπεραίωσης [10].

Επίσης, εφαρμογές ειδικού σκοπού φαίνεται να μπορούν να βοηθήσουν άτομα τα οποία πάσχουν από ήπια νοητική υστέρηση [11].

## 2.3 Εξελικτικοί αλγόριθμοι

Είναι στοχαστικοί αλγόριθμοι που χρησιμοποιούν ως μηχανισμό αναζήτησης λύσεων κάποια φυσικά φαινόμενα όπως τη γενετική κληρονομικότητα και τη δαρβινική διαμάχη για επιβίωση, που έχει ως αποτέλεσμα τη σταδιακή οδήγηση του πληθυσμού σε ολικά βέλτιστα.

Αυτό που κάνουν είναι να μιμούνται τις αρχές της φυσικής εξέλιξης. Αυτό ξεκινά με την κωδικοποίηση μιας λύσης του προβλήματος σε μια δομή δεδομένου που έχει χαρακτηριστικά χρωμοσώματος. Στην αρχή γίνεται η αρχικοποίηση του πληθυσμού με (τυπικά τυχαία) χρωμοσώματα.

Έπειτα γίνεται η αξιολόγηση των λύσεων βάση τη συνάρτηση αξιολόγησης, που είναι κατάλληλα σχεδιασμένη και προσαρμοσμένη το εκάστοτε πρόβλημα. Αυτή η συνάρτηση δίνει σε όλα τα χρωμοσώματα μια τιμή η οποία ορίζει το πόσο καλές θεωρούνται οι λύσεις που αναπαραστούν. Έπειτα λαμβάνει μέρος η επιλογή ενός τμήματος του πληθυσμού (υπάρχουν διάφοροι τρόποι που μπορεί να γίνει αυτή η επιλογή και η κάθε μια πρέπει να εξεταστεί για το πόσο κατάλληλη είναι για τη λύση του προβλήματος) που θα έχουν τον ρόλο των γονέων και θα είναι υπεύθυνοι για τη δημιουργία του νέου πληθυσμού. Εφόσον γίνει η επιλογή των γονέων τότε εφαρμόζεται πάνω στα χρωμοσώματα ένας τελεστής ανασυνδιασμού έτσι ώστε να προκύψουν απόγονοι που θα έχουν κληρονομήσει κάποιο μέρος του χρωμοσώματος τους και από τους δύο γονείς. Μετά μπορεί να γίνει μια τυχαία μετάλλαξη σε ένα ή περισσότερα από τα χρωμοσώματα έτσι ώστε να μειωθεί η σύγκληση του πληθυσμού σε κάποιο τοπικό μέγιστο ή ελάχιστο, πράγμα που μπορεί να προκαλέσει τη μην εύρεση μια πιθανότατα καλύτερης λύσης. Έπειτα λαμβάνει μέρος η μερική ή ολική αντικατάσταση του παλιού πληθυσμού με τον νέο. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί μια αρκετά ικανοποιητική λύση ή μέχρι να περάσει ένας συγκεκριμένος αριθμός επανάληψης όλων των προαναφερθέντων βημάτων ή όταν περάσουν κάποιες επαναλήψεις χωρίς να υπάρχει σημαντική βελτίωση στον μέσο όρο ποιότητας των χρωμοσωμάτων [12].

## Κεφάλαιο 2ο:

Οι γενετικοί αλγόριθμοι έχουν την ικανότητα να βρίσκουν σχεδόν βέλτιστες λύσεις μέσα σε λογικά χρονικά πλαίσια σε αντίθεση με παραδοσιακές μεθόδους που πρέπει να ελέγξουνε όλες τις πιθανές λύσεις.

Υπάρχουν δύο βασικά σχεδιαστικά ζητήματα που πρέπει να λαμβάνει κανείς υπόψη όταν σχεδιάζει έναν γενετικό αλγόριθμο : ποια είναι η κατάλληλη αναπαράσταση χρωμοσώματος και ποια είναι η αρμόζουσα συνάρτηση αξιολόγησης για το πρόβλημα που μελετάται. Με τη χρήση διαδραστικών πειραμάτων στη διαδικασία μάθησης των εξελικτικών αλγορίθμων, οι μαθητές μπορούν να έχουν μια πολύ πιο ενεργή συμμετοχή στο αντικείμενο, προκαλώντας έτσι τη σύνθεση νέας καθώς και την ενδυνάμωση της δικιάς του εσωτερικής γνώσης. Επίσης, η βαθύτερη ενασχόληση με τους εξελικτικούς αλγόριθμους έδωσε στους μαθητές έναν μεγάλο βαθμό εκτίμηση για την ικανότητα τους να λύνουν εξαιρετικά περίπλοκα προβλήματα σε σχετικά μικρό χρονικό διάστημα με μεγάλη επιδεξία. Όπως είπε και ο Donald Ervin Knuth "πρέπει να δει κανείς έναν αλγόριθμο για να τον πιστέψει" [13].

Πρέπει να γίνει κατανοητό ότι παρόλο που οι γενετικοί αλγόριθμοι έχουν την ικανότητα να προσφέρουν καλές λύσεις σε πολλά προβλήματα, υπάρχουν και κάποια στα οποία δεν μπορούν να ανταπεξέλθουν.

Επίσης, μεγάλο ρόλο παίζουν και τα κριτήρια που διαλέγουμε για να κρίνουμε την επίδοση ενός γενετικού αλγορίθμου ανάλογα με το αποτέλεσμα που προσδοκούμε. Μπορεί πχ ένας αλγόριθμος να δίνει πολύ καλές λύσεις αλλά να χρειάζεται πολύ χρόνο για να τις παράξει ενώ ένας άλλος δεν παράγει το ίδιο καλές λύσεις αλλά μπορεί να τις παράξει εξαιρετικά γρήγορα. Αξίζει να σημειωθεί ότι σε μεγάλο βαθμό οι γενετικοί αλγόριθμοι είναι τέχνη παρά επιστήμη, καθώς μεγάλο μέρος τους βασίζεται πάνω στην εμπειρία του σχεδιαστή να διαλέξει τους κατάλληλους τελεστές αναπαραγωγής, τελεστές μετάλλαξης κτλ καθώς να φροντίσει τη διαμόρφωση του αλγορίθμου [14]. Μια ενδιαφέρουσα μέθοδος διδασκαλίας γενετικών αλγορίθμων ήταν με τη χρήση ιαπωνικών παζλ.

Η χρήση τέτοιων μεθόδων οπτικοποίησης έχουν δείξει ότι ευνοούν την κατανόηση πολλών εννοιών όπως την κωδικοποίηση, τους γενετικούς τελεστές, τους μεμετικούς αλγόριθμους κτλ σε μεγαλύτερο βαθμό σε σχέση με τις παραδοσιακές μεθόδους εκμάθησης, καθώς ένα παζλ είναι εύκολο στην κατανόηση αλλά απαιτητικό ως προς τη λύση [15].

Το γεγονός ότι τις περισσότερες φορές η διδασκαλία των γενετικών αλγορίθμων γίνεται μόνο σε θεωρητικό πλαίσιο φαίνεται να έχει αρνητική επίπτωση στην κατανόηση τους από τους μαθητές ιδικά σε μαθητές πληροφορικής [16].

## 2.4 Επίλογος

Σε αυτό το κεφάλαιο έλαβε μέρος η παρουσίαση των πιο κύριων χαρακτηριστικών των εκπαιδευτικών εφαρμογών καθώς και μια σύντομη περιγραφή του τρόπου λειτουργίας ενός γενετικού αλγορίθμου και κάποιων ιδιαιτεροτήτων ως προς τις δυνατότητες και τον τρόπο χρήσης τους.

## Κεφάλαιο 3ο: Τεχνολογίες και Εργαλεία

### 3.1 Εισαγωγή

Στο συγκεκριμένο κεφάλαιο γίνεται μια αρχική απαρίθμηση όλων των εργαλείων και τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη της συγκεκριμένης εφαρμογής καθώς και αναφορές σε ιστοσελίδες που χρησιμοποιήθηκαν για την παροχή εικόνων και ηχητικών αρχείων.

Για κάθε μια εγγραφή, αναφέρεται συνοπτικά (όπου αυτό αρμόζει) ο συγκεκριμένος λόγος επιλογής του/της εργαλείου/τεχνολογίας. Περαιτέρω λεπτομέρειες ως προς τον ακριβή τρόπο χρήσης τους, καθώς και της βαρύτητας των γίνεται στα επόμενα κεφάλαια.

Για κάθε εργαλείο/τεχνολογία παρέχεται σύνδεσμος (στον τίτλο του υποκεφαλαίου) που ανατρέχει στην εκάστοτε ιστοσελίδα όπου μπορεί να το/την κατεβάσει κάποιος, η εάν αυτό δεν είναι δυνατόν (πχ. η JavaScript είναι ενσωματωμένη με τους περιηγητές ιστού) μια ιστοσελίδα που παρέχει περαιτέρω πληροφορίες.

### 3.2 [JavaScript](#)

Επρόκειτο για μία γλώσσα προγραμματισμού που αποτελεί τον ακρογωνιαίο λίθο όσο αφορά την ανάπτυξη όλων των τύπων εφαρμογών που υπάρχουν στο διαδίκτυο από την πλευρά του πελάτη αλλά και του εξυπηρετητή με τη χρήση εργαλείων που επιτρέπουν την εκτέλεση της σε περιβάλλοντα εκτός ενός κοινού περιηγητή ιστού.[17]

Είναι μια γλώσσα που επιτρέπει την ανάπτυξη περίπλοκων λειτουργιών μέσα σε ιστοσελίδες. Επιτρέπει την προσθήκη διαδραστικών στοιχείων μέσα σε μια ιστοσελίδα, κάνοντας την πολλή πιο χρήσιμη από μια κλασική στατική ιστοσελίδα.

Πρακτικά είναι κύρια γλώσσα για την προσθήκη τέτοιων λειτουργιών οπότε η χρήση της ήταν αναγκαία. Επίσης, η JavaScript παρέχει ένα μεγάλο αριθμό από διαθέσιμα Web APIs, που μας επιτρέπουν να διευκολύνουμε ραγδαία την ανάπτυξη των εφαρμογών μας. Στις ακόλουθες υποενότητες γίνεται η αναφορά όλων αυτών που χρησιμοποιήθηκαν στην παρούσα εφαρμογή.

#### 3.2.1 [DOM](#)

Το **Document Object Model (DOM)** είναι υπεύθυνο για τη διασύνδεση των ιστοσελίδων με τα αντίστοιχα script με το να αντιπροσωπεύει τη δομή ενός κειμένου – όπως πχ τη HTML αναπαράσταση στη μνήμη.

Το DOM αντιπροσωπεύει ένα κείμενο με τη μορφή ενός δένδρου. Κάθε κλαδί αυτού του δένδρου τερματίζει σε ένα κόμβο, όπου αυτός ο κόμβος απαρτίζεται από αντικείμενα.

Η μέθοδο που παρέχει το DOM επιτρέπουν την προγραμματιστική πρόσβαση στο δένδρο, επιτρέποντας τις αλλαγές στη δομή, το στιλ αλλά και το περιεχόμενο του.[18]

Εν ολίγης είναι το εργαλείο που μας επιτρέπει να κάνουμε όλες τις αλλαγές στα στοιχεία της διεπαφής της εφαρμογής (να αλλάξουμε το κείμενο σε ένα κουμπί, να κρύψουμε ένα μενού λόγο μιας ρύθμισης κτλ).

Κεφάλαιο 3ο:

### 3.2.2 [Fetch API](#)

Μας παρέχει ένα API με το οποίο μπορούμε να κάνουμε αιτήσεις για διάφορους πόρους που είναι αποθηκευμένοι σε έναν εξυπηρετητή (πχ αρχεία HTML, JavaScript, εικόνες ήχους κτλ).[19]

Ο τρόπος που δουλεύει εσωτερικά είναι με τη χρήση των Promises, που πρόκειται για αντικείμενα της JavaScript που αντιπροσωπεύουν την ενδεχόμενη επιτυχία (η αποτυχία) μια ασύγχρονης ενέργειας που πιστεύει κάποια τιμή.

Ο λόγος δημιουργίας αυτού του API ήταν για να αντικαταστήσει ένα πολύ παλιότερο, αυτό των XMLHttpRequest γιατί ήταν δύσκολο, καθώς απαιτούσε τη σύνταξη μεγάλου αριθμού κώδικα για τη χρήση του και ως κύριο δομικό στοιχείο είχε τα callbacks (πρόκειται για το πέρασμα συναρτήσεων ως παραμέτρους σε άλλες συναρτήσεις), που είχε ως άμεσο αποτέλεσμα να προκαλεί τη συγγραφή κώδικα που ήταν δύσκολος στην κατανόηση αλλά και στην αποσφαλμάτωση.

### 3.2.3 [LocalStorage](#)

Είναι μια τοπική βάση δεδομένων στην οποία μπορούμε να αποθηκεύσουμε δεδομένα με τη μορφή κλειδί-τιμή (key-value pair) σε μορφή που μοιάζει με αυτή που χρησιμοποιείται σε αρχεία JSON.

Τα δεδομένα αφορούν μόνο τη συγκεκριμένη ιστοσελίδα που έχουν δημιουργηθεί και δεν είναι προσβάσιμα από άλλες σελίδες.

Επίσης, τα δεδομένα που αποθηκεύονται εδώ δε χάνονται, ακόμα και εάν ο χρήστης κλείσει την ιστοσελίδα, πράγμα που είναι ιδιαίτερα σημαντικό για την αποθήκευση ρυθμίσεων καθώς και άλλων δεδομένων που δημιουργούνται κατά τη διάρκεια χρήσης της εφαρμογής χωρίς να χρειάζεται να τα στείλουμε στον εξυπηρετητή.

Έχει χωρητικότητα 5mb, που μπορεί να μη φαίνεται επαρκής για τις ανάγκες μιας εφαρμογής, όμως το γεγονός ότι αποθηκεύουμε κείμενο σημαίνει ότι μπορεί να έχουμε 1250 σελίδες γεμάτες κείμενο, κάτι που είναι πρακτικά αδύνατο να αγγίζουμε με τις απαιτήσεις της συγκεκριμένης εφαρμογής.[20]

### 3.2.4 [Clipboard API](#)

“Το Clipboard API παρέχει τη δυνατότητα απάντησης σε εντολές του προχείρου (αποκοπή, αντιγραφή και επικόλληση) καθώς και ασύγχρονη ανάγνωση και εγγραφή στο πρόχειρο του συστήματος.”[21]

Χρησιμοποιήθηκε για να παρέχει ένα μηχανισμό προσηνής αποθήκευσης και μεταφοράς δεδομένων με τη μορφή κειμένου.

### 3.2.5 [Canvas API](#)

“Το Canvas API παρέχει ένα μέσο για τη σχεδίαση γραφικών μέσω JavaScript και του στοιχείου HTML <canvas>. Μεταξύ άλλων, μπορεί να χρησιμοποιηθεί για κινούμενα σχέδια, γραφικά παιχνιδιών, οπτικοποίηση δεδομένων, χειρισμό φωτογραφιών και επεξεργασία βίντεο σε πραγματικό χρόνο.

Το Canvas API εστιάζει σε μεγάλο βαθμό στα γραφικά 2D.”[22]

Έγινε η χρήση του για τη δημιουργία πολλών οπτικοποιήσεων ασκήσεων, καθώς λόγω κάποιων ιδιαιτεροτήτων τους δεν μπορούσαν να δομηθούν με μόνο χρήση HTML.

### 3.2.6 Web Workers API

“Οι Web Workers καθιστούν δυνατή την εκτέλεση μιας λειτουργίας ενός script σε ένα νήμα ξεχωριστό από το κύριο νήμα εκτέλεσης μιας εφαρμογής Ιστού. Το πλεονέκτημα αυτού είναι ότι η επίπονη επεξεργασία μπορεί να εκτελεστεί σε ξεχωριστό νήμα, επιτρέποντας στο κύριο νήμα (συνήθως το UI) να τρέχει χωρίς να μπλοκάρεται/επιβραδύνεται.”[23]

Έχουν χρησιμοποιηθεί για την εκτέλεση διάφορων εργασιών στο παρασκήνιο (την αίτηση πόρων από τον εξυπηρετητή που ενδέχεται να απαιτούν περαιτέρω επεξεργασία).

### 3.2.7 Service Worker API

“Οι Service Workers λειτουργούν ουσιαστικά ως διακομιστές μεσολάβησης που βρίσκονται μεταξύ των εφαρμογών Ιστού, του προγράμματος περιήγησης και του δικτύου (όταν είναι διαθέσιμοι). Προορίζονται, μεταξύ άλλων, να επιτρέπουν τη δημιουργία αποτελεσματικών εμπειριών εκτός σύνδεσης, να παρακολουθούν αιτήματα δικτύου και να λαμβάνουν τις κατάλληλες ενέργειες με βάση το εάν το δίκτυο είναι διαθέσιμο και να ενημερώνουν τα στοιχεία που βρίσκονται στον διακομιστή.”[24]

Ο κύριος λόγος χρήσης τους ήταν για να επιτρέψουν στην εφαρμογή να μπορεί να λειτουργεί χωρίς σύνδεση στον εξυπηρετητή, εφόσον βέβαια έχουν γίνει όλες οι απαραίτητες προετοιμασίες (η επεξήγηση τους γίνεται στην υποενότητα 5.7.5).

### 3.2.8 Web Components

Τα Web Components είναι μια κολεκτίβα διάφορων τεχνολογιών που επιτρέπουν τη δημιουργία επαναχρησιμοποιήσιμων προσαρμοσμένων στοιχείων, με όλοι τους τη λειτουργικότητα ενθυλακωμένη ξεχωριστά από τον υπόλοιπο κώδικα.[25]

Γίνεται εκτιμής της χρήσης των Web Components στην εφαρμογή, και για αυτόν τον λόγο θα αφιερωθεί μια ολόκληρη υποενότητα (4.9) που θα δίνει περαιτέρω πληροφορίες για το πως ακριβώς χρησιμοποιούνται καθώς και τα πλεονέκτημα που προσφέρουν.

## 3.3 Bootstrap

Η Bootstrap είναι μια βιβλιοθήκη ανοιχτού κώδικα που έχει σχεδιαστεί για τη δημιουργία ευέλικτων και “ανταποκρίσιμων” εφαρμογών, χρησιμοποιώντας τις τεχνολογίες HTML, CSS και JavaScript.

Με τον όρο “ανοιχτού κώδικα” αναφερόμαστε στο γεγονός ότι ο οποιοσδήποτε έχει το δικαίωμα να διαβάσει τον πηγαίο κώδικα της βιβλιοθήκης καθώς και να κάνει οιαδήποτε τροποποίηση ή επαύξηση επιθυμεί.[26]

Με τον όρο “ανταποκρίσιμη” (responsive) εννοούμε τη δυνατότητα της εφαρμογής να προσαρμόζει δυναμικά τον τρόπο παρουσίασης της στην εκάστοτε συσκευή.[27]

Παρέχει στον χρήστη ένα μεγάλο αριθμό έτοιμων συστατικών για τη διευκόλυνση του στον τομέα της σχεδίασης ιστοσελίδων και web εφαρμογών.

Προσθέτει μία αίσθηση ομοιομορφίας στην ιστοσελίδα/εφαρμογή που τη χρησιμοποιεί, κάνοντας τη να φαίνεται ανώτερης ποιότητας.

Κεφάλαιο 3ο:

Ο λόγος που χρησιμοποιήθηκε ήταν για να επιταχύνει τον ρυθμό ανάπτυξης του UI (User Interface) [28] της εφαρμογής, καθώς και να απαλλάξει τον προγραμματιστή από τον φόρτο όλων των επιλογών που προαπαιτεί η εκ νέου σχεδίαση ενός UI.

### 3.4 [Google Material Icons](#)

Είναι μια συλλογή εικονιδίων που έχουν φτιαχτεί από την Google και ως βασικό σκοπό έχουν να συνεχούν όμορφα με το material design, που είναι μια σειρά σχεδιαστικών προτύπων που έχουν δημιουργηθεί επίσης από την Google και ως σκοπό έχουν την παροχή ενός προτύπου για κάποιον που επιθυμεί να αναπτύξει μια εφαρμογή, χωρίς όμως να χρειάζεται να σχεδιάσει από την αρχή όλη την αλληλεπίδραση ανάμεσα στην εφαρμογή και στον χρήστη.[29]

Η εφαρμογές που χρησιμοποιούν αυτό το πρότυπο φαίνονται εξαιρετικά οικίες στους χρήστες, γιατί μοιάζουν στις εφαρμογές που παρέχει η ίδια η Google.

Παρόλο το γεγονός ότι δεν έχει γίνει χρήση του material design στην εφαρμογή, τα εικονίδια είναι ιδιαίτερα προσαρμόσιμα και σε άλλα σχεδιαστικά πρότυπα, όπως αυτά της Bootstrap, και επίσης λόγω των τεράστιων αριθμών χρηστών των εφαρμογών της Google, η σημασιολογική σημασία των εικονιδίων είναι άμεσα κατανοητή από τους περισσότερους, αποφεύγοντας έτσι την ανάγκη εκμάθησης νέων εικονιδίων.

### 3.5 [HTML](#)

Η HTML είναι μια γλώσσα τύπου markup που χρησιμοποιείτε για τη σύνταξη εγγράφων που προορίζονται για εμφάνιση σε ένα περιηγητή ιστού (web browser).

Καθορίζει την πρωταρχική δομή του περιεχομένου μιας ιστοσελίδας καθώς και τη σημασιολογική σημασία όλων των στοιχείων που την απαρτίζουν.

Τα συγκεκριμένα στοιχεία ονομάζονται HTML elements, και είναι τα δομικά κομμάτια που απαρτίζουν ένα HTML έγγραφο.[30]

Χρησιμοποιήθηκε στην εφαρμογή για τον απλούστατο λόγο ότι δεν μπορεί να υπάρξει μια ιστοσελίδα/εφαρμογή web χωρίς τη χρήση της. Είναι πρακτικά ο θεμελιώδης λίθος πάνω στον οποίον στηρίζονται ένας μεγάλος αριθμός των τεχνολογιών που χρησιμοποιήθηκαν.

### 3.6 [JSON](#)

“Το **JSON** (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων.

Είναι εύκολο για τους ανθρώπους να το διαβάσουν και γράψουν. Είναι εύκολο για τις μηχανές να το αναλύσουν (parse) και να το παράγουν (generate).

Βασισμένο πάνω σε ένα υποσύνολο της γλώσσας [προγραμματισμού JavaScript](#), Standard ECMA-262 Έκδοση 3η - Δεκέμβριος 1999.

Το JSON είναι ένα πρότυπο κειμένου το οποίο είναι τελείως ανεξάρτητο από γλώσσες προγραμματισμού αλλά χρησιμοποιεί πρακτικές (conventions) οι οποίες είναι γνωστές στους προγραμματιστές της οικογένειας προγραμματισμού C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python, και πολλών άλλων. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων.”[31]

Ο λόγος που χρησιμοποιήθηκε στην εφαρμογή είναι ότι παρόλο το γεγονός ότι είναι ανεξάρτητο από γλώσσες προγραμματισμού επειδή βασίζεται στα πρότυπα γραφής αντικειμένων JavaScript, είναι εξαιρετικά εύκολο να γίνει ή αμφίδρομη μετατροπή από JSON σε αντικείμενα JavaScript Object και το αντίστροφο. Εφόσον η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η JavaScript, ήταν προφανής η επιλογή.

### 3.7 [Plotly.js](#)

“Το Plotly.js είναι μια αυτόνομη βιβλιοθήκη οπτικοποίησης δεδομένων JavaScript και τροφοδοτεί επίσης τις λειτουργικές μονάδες Python και R που ονομάζονται plotly σε αυτά τα αντίστοιχα οικοσυστήματα (αναφέρονται ως Plotly.py και Plotly.R).

Το Plotly.js μπορεί να χρησιμοποιηθεί για την παραγωγή δεκάδων τύπων γραφημάτων και απεικονίσεων, συμπεριλαμβανομένων στατιστικών γραφημάτων, τρισδιάστατων γραφημάτων, επιστημονικών διαγραμμάτων, χαρτών SVG και πλακιδίων, οικονομικών διαγραμμάτων και άλλων.”[32]

Η συγκεκριμένη βιβλιοθήκη χρησιμοποιήθηκε για να δημιουργούν όλα τα γραφήματα της εφαρμογής.

Το API που παρέχει η βιβλιοθήκη είναι εξαιρετικά εκτενές, καθώς επιτρέπει την εξατομίκευση σχεδόν όλων των πιθανών πτυχών ενός γραφήματος. Στα γραφήματα της εφαρμογής χρησιμοποιήθηκαν διάφορες ρυθμίσεις που απέκρυπταν κάποιες λειτουργίες που δεν ήταν απαραίτητες για τον τύπο γραφημάτων που δημιουργήθηκαν.

### 3.8 [Suneditor](#)

Ο Suneditor είναι ένα πρόγραμμα συγγραφής κειμένου γραμμένο σε “καθαρή” (χωρίς τη χρήση άλλων επικουρικών βιβλιοθηκών) JavaScript με υποστήριξη για όλους του μοντέρνους περιηγητές ιστού.

Βασίζεται στο πρότυπο κειμενογράφων WYSIWYG (what you see is what you get | αυτό που βλέπετε είναι αυτό που παίρνετε) που σημαίνει ότι το πώς φαίνεται το περιεχόμενο του κειμένου κατά τη διαδικασία της συγγραφής, θα είναι σε συμφωνία με το πως θα φαίνεται σε μια διεπαφή χρήσης, ιστοσελίδα, παρουσίαση κτλ.

Το παρών πρόγραμμα χρησιμοποιήθηκε για να προσφέρει τη δυνατότητα συγγραφής σημειώσεων εντός της εφαρμογής. Επιπρόσθετα έχουν γίνει αναπροσαρμογές στον τρόπο εμφάνισης κάποιων επιλογών, καθώς και προσθήκες για την καλύτερη ομογενοποίηση του με την υπόλοιπη εφαρμογή.

### 3.9 [PDF.js](#)

“Το PDF.js είναι ένα πρόγραμμα προβολής φορητής μορφής εγγράφου (PDF) που έχει δημιουργηθεί με HTML5. Το PDF.js βασίζεται στην κοινότητα και υποστηρίζεται από τη Mozilla. Στόχος μας είναι να δημιουργήσουμε μια πλατφόρμα γενικής χρήσης, βασισμένη σε πρότυπα ιστού για ανάλυση και απόδοση PDF.”[33]

Καθώς η βασική μορφή ύλης στην εφαρμογή είναι τα αρχεία PDF είναι απαραίτητη η δυνατότητα προβολής τους μέσα σε ένα περιηγητή ιστού.

## Κεφάλαιο 3ο:

Μια σημαντική παρατήρησή είναι ότι το παρών πρόγραμμα συμπεριλαμβάνεται σε πολλούς περιηγητές ιστού, οπότε ποιος ο λόγος της ξεχωριστής προσθήκης του;

Ο λόγος είναι για τη δυνατότητα προσαρμογής κάποιων λειτουργιών του καθώς και την προσθήκη νέων που είναι σχετικές με τα διάφορα άλλα κομμάτια της εφαρμογής.

### 3.10 [Node.js](#)

Είναι ανοιχτού κωδικά[26], ασύγχρονο και event-driven περιβάλλον runtime της JavaScript που μπορεί να τρέξει σε πληθώρα λειτουργικών συστημάτων όπως Windows, MacOS, Linux και άλλα.

Με τον όρο runtime, εννοούμε ένα σύστημα το οποίο είναι παρών και στον υπολογιστή που εγγράφηκε μια εφαρμογή καθώς και στον υπολογιστή που επρόκειτο να εκτελεστεί η συγκεκριμένη εφαρμογή.

Σκοπός του είναι να επιτρέπει την εκτέλεση προγραμμάτων JavaScript σε περιβάλλοντα εκτός από αυτά ενός περιηγητή ιστού καθώς και τη δημιουργία επεκτάσιμων εφαρμογών δικτύωσης.

Έγινε η χρήση του για την ευκολία που παρέχει στον προγραμματιστή στην ανάπτυξη εφαρμογών δικτύωσης λόγω του τεράστιου αριθμού διαθέσιμων πακέτων επέκτασης, αλλά κατά κύριο λόγο το γεγονός ότι δεν απαιτείτε η χρήση άλλης γλώσσας προγραμματισμού εκτός της JavaScript, πράγμα που συμβάλει στην καλύτερη συνοχή του κώδικα της εφαρμογής.

### 3.11 [Express.js](#)

“Το Express είναι ένα ελάχιστο και ευέλικτο πλαίσιο εφαρμογών ιστού Node.js που παρέχει ένα ισχυρό σύνολο δυνατοτήτων για εφαρμογές ιστού και για κινητές συσκευές.

Με μια πληθώρα μεθόδων χρησιμότητας HTTP και ενδιαμέσου λογισμικού στη διάθεσή σας, η δημιουργία ενός ισχυρού API είναι γρήγορη και εύκολη.”[34]

Ο λόγος που χρησιμοποιήθηκε είναι για την ευκολότερη και γρηγορότερη δημιουργία του εξυπηρετητή της εφαρμογής, καθώς και για τη διαχείριση των διάφορων restful api που παρέχει η εφαρμογή για τον χειρισμό των διάφορων λειτουργιών της.

### 3.12 [Socket.io](#)

“Είναι μια βιβλιοθήκη για τη σχεδίαση και ανάπτυξη εφαρμογών πραγματικού χρόνου με τη δυνατότητα αμφίδρομης επικοινωνίας ανάμεσα στον πελάτη και τον εξυπηρετητή. Είναι εξίσου πρωτόκολλο, οπότε επιτρέπει σε διάφορες άλλες υλοποιήσεις του πρωτοκόλλου να επικοινωνούν μεταξύ τους.

Η βασική υλοποίηση του αποτελείται από δυο κομμάτια, αυτό της εφαρμογής που τρέχει σε έναν περιηγητή ιστού και σε αυτή που τρέχει σε έναν εξυπηρετητή ιστού.”[35]

Ο λόγος που χρησιμοποιήθηκε είναι για να υποστηρίξει τη δυνατότητα αποστολής ειδοποιήσεων στους χρήστες της εφαρμογής καθώς και για την άμεση ενημέρωσή τους για αλλαγές που έχουν γίνει στα αρχεία περιεχομένου.

### 3.13 **Ιστοσελίδες περιεχομένου**

Από τις συγκεκριμένες ιστοσελίδες χρησιμοποιήθηκαν εικόνες αλλά και ηχητικά αρχεία.

Οι συγκεκριμένες ιστοσελίδες επιλέχθηκαν γιατί παρέχουν δωρεάν όλα τα αρχεία με άδειες που επιτρέπουν στον κάθε ένα να τα χρησιμοποιεί για διάφορους σκοπούς, είτε προσωπικούς είτε εμπορικούς.

### 3.13.1 [SVG Repo](#)

Το SVG Repo παρέχει τη δυνατότητα της λήψης δωρεάν εικόνων SVG για οποιοδήποτε σενάριο χρήσης και παρέχει εργαλεία που επιτρέπουν την αναζήτηση μέσα σε μια βάση που περιέχει περισσότερες από 500.000 εικόνες.[36]

Η συγκεκριμένη ιστοσελίδα παρέχει εικόνες τύπου SVG, που είναι εικονίδια που χρησιμοποιούν μαθηματικές σχέσεις και συναρτήσεις για να παράξουν μια εικόνα.[37]

Για αυτό τον λόγο είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν την καλή προσαρμογή του περιεχομένου τους σε διάφορο τύπο συσκευών.

Επιπρόσθετα το γεγονός ότι δε χρειάζεται να αποθηκεύσουν διακριτά πληροφορίες για το κάθε pixel μιας εικόνας, τις κάνει πολύ πιο μικρές σε μέγεθος. Αυτοί είναι και οι λόγοι που επιλέχτηκε ο συγκεκριμένος τύπος εικόνας.

### 3.13.2 [Zapsplat](#)

Επρόκειτο για μια πλατφόρμα που επιτρέπει τη δωρεάν λήψη διαφόρων ηχητικών αρχείων από μια μεγάλη γκάμα επιλογών.[38]

Ο ήχος είναι ιδιαίτερα σημαντικός, καθώς μας επιτρέπει να επικοινωνήσουμε ιδέες στους χρήστες της εφαρμογής χωρίς να χρειάζεται να το αναφέρουμε κάπου ρητά.

Ε αν παραδείγματος χάριν θα θέλαμε να δηλώσουμε στον χρήστη ότι μια ενεργεία που εκτελεί δεν είναι επιθυμητή, μπορούμε να παίξουμε ένα ηχητικό αρχείο που θα ήταν δυσάρεστο στην ακοή έτσι ώστε να τον ειδοποιούσε ότι είναι ανώφελο να συνεχίζουν να εκτελούν τη συγκεκριμένη ενέργεια.

Η εάν θέλουμε να επικροτήσουμε τον χρήστη για μια σωστή του πράξη μπορούμε να παίξουμε ένα ηχητικό αρχείο που θα ήταν ιδιαίτερα ευχάριστο στην ακοή.

Μπορούμε δηλαδή να παρέχουμε ένα είδος ανάδρασης για κάθε ενέργεια που θεωρούμε αρκετά σημαντική για να τη συνοδεύσουμε με ηχητικό αρχείο, καθώς και να κάνουμε την εφαρμογή πιο προσιτή σε άτομα που μπορεί να έχουν προβλήματα όρασης πχ.

Η επιλογή των ηχητικών αρχείων πρέπει να γίνει με σύνεση και σωστή έρευνα, καθώς ο ήχος έχει τη δυνατότητα να αλλάζει την ψυχική κατάσταση του χρήστη.[39]

## 3.14 [Visual Studio Code](#)

“Το Visual Studio Code, κοινώς γνωστό και ως VS Code, είναι ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα που αναπτύχθηκε από τη Microsoft για Windows, Linux και macOS.

Οι δυνατότητες περιλαμβάνουν υποστήριξη για εντοπισμό σφαλμάτων, επισήμανση σύνταξης, έξυπνη συμπλήρωση κώδικα, αποσπάσματα, ανακατασκευή κώδικα και ενσωματωμένο Git.

Οι χρήστες μπορούν να αλλάξουν το θέμα, τις συντομεύσεις πληκτρολογίου, τις προτιμήσεις και να εγκαταστήσουν επεκτάσεις που προσθέτουν λειτουργικότητα.”[40], [41]

## Κεφάλαιο 3ο:

Έχει εξαιρετική υποστήριξη για τη JavaScript, και οι επεκτάσεις που μπορούν να προστεθούν είναι εξαιρετικά χρήσιμες.

Συγκεκριμένα το intellisense που παρέχει επιτρέπει στον προγραμματιστή να έχει άμεσα ανταπόκριση για τυχόν τυπογραφικά λάθη που μπορεί να έχει κάνει, καθώς και το auto-code-complition που επιτρέπει την αυτόματη εισαγωγή τυποποιημένων κομματιών κώδικα όπως for-loops, while-loops, switch-statements. Αυτές οι λειτουργίες επιτρέπουν την πιο γρήγορη ανάπτυξη εφαρμογών, αλλά και τη μείωση των λαθών.

### 3.15 [Inkscape](#)

“Το Inkscape είναι ένα δωρεάν και ανοιχτού κώδικα πρόγραμμα επεξεργασίας γραφικών για GNU/Linux, Windows και macOS.

Προσφέρει ένα πλούσιο σύνολο χαρακτηριστικών και χρησιμοποιείται ευρέως τόσο για καλλιτεχνικές όσο και για τεχνικές εικονογραφήσεις, όπως κινούμενα σχέδια, clip art, λογότυπα, τυπογραφία, διαγράμματα και διαγράμματα ροής.

Χρησιμοποιεί διανυσματικά (Vector) γραφικά για να επιτρέψει ευκρινείς εκτυπώσεις και αποδόσεις σε απεριόριστη ανάλυση και δε δεσμεύεται σε σταθερό αριθμό pixel όπως τα γραφικά ράστερ.

Το Inkscape χρησιμοποιεί την τυποποιημένη μορφή αρχείου SVG ως κύρια μορφή, η οποία υποστηρίζεται από πολλές άλλες εφαρμογές, συμπεριλαμβανομένων των προγραμμάτων περιήγησης ιστού.”[42]

Αυτό το εργαλείο χρησιμοποιήθηκε για τη δημιουργία του λογότυπου της εφαρμογής καθώς και την τροποποίηση διάφορων εικόνων SVG όπου αυτό κρίθηκε απαραίτητο.

### 3.16 Επίλογος

Σε αυτό το κεφάλαιο έγινε η παράθεση των εργαλείων και των τεχνολογιών που χρησιμοποιήθηκαν για την εκπόνηση της εφαρμογής καθώς και ο λόγος της εκάστοτε επιλογής τους. Με την ανάπτυξη των επόμενων κεφαλαίων ο τρόπος λειτουργία τους γίνεται πολύ πιο ξεκάθαρος.

## Κεφάλαιο 4ο: Ανάλυση Μεθοδολογίας και Συστημάτων

### 4.1 Εισαγωγή

Στο συγκεκριμένο κεφάλαιο γίνεται μια ανάλυση της γενικής δομής που έχει χρησιμοποιηθεί στην εφαρμογή (πχ ο τρόπος διασύνδεσης όλων των τμημάτων που την απαρτίζουν) καθώς και η παράθεση κάποιων ιδιαίτερα σημαντικών συστημάτων που έχουν χρησιμοποιηθεί εκτεταμένα μέσα στην εφαρμογή.

Εφόσον γίνει η επεξήγηση όλων αυτών των τμημάτων στην παρούσα ενότητα θα θεωρούνται πως πλέον είναι γνωστά και κατανοητά. Στην επόμενη ενότητα γίνεται η πλήρη παρουσίαση όλων των υπόλοιπων τμημάτων που συγκροτούν την εφαρμογή.

### 4.2 Βασική αρχιτεκτονική

Η εφαρμογή θεωρείται ως SPA (Single Page Application) γιατί σε αντίθεση με μια συμβατική εφαρμογή ιστού, δε χρειάζεστε να αλλάζει την τρέχουσα σελίδα για να εκτελέσει μια λειτουργία.

Αυτό που κάνει είναι να εκτελεί αιτήματα στον εξυπηρετητή που σερβίρει την εφαρμογή ως επακόλουθο μιας ενέργειας του χρήστη, και έπειτα να προσαρμόζει δυναμικά τη διεπαφή της για να ανταποκριθεί στις αλλαγές που πρέπει να συμβούν.

Αυτό έχει το πλεονέκτημα ότι δε χρειάζεστε να γίνει ανανέωση της σελίδας κάθε φορά που ο χρήστης εκτελεί μια ενεργεία και κάνει την εφαρμογή να φαίνεται ότι είναι πιο συμβατική με μια εφαρμογή που είναι τοπικά εγκατεστημένη, δηλαδή δε μοιάζει με μια απλή ιστοσελίδα.

Επίσης, αυτή η αρχιτεκτονική επιλογή κάνει την εφαρμογή να είναι CSR (Client Side Rendered), που σημαίνει ότι η οντότητα που είναι υπεύθυνη για την απόδοση της διεπαφής χρήστη είναι ο ίδιος ο περιήγητης ιστού και όχι ο εξυπηρετητής, όπως σε μια συμβατική εφαρμογή ιστού.

Αυτό που κάνει ο εξυπηρετητής όταν δέχεται ένα αίτημα σύνδεσης από ένα χρήστη είναι να του στείλει την αρχική σελίδα HTML, που ως συνήθως λόγω συμβάσεων είναι η index.html (το ίδιο ισχύει και στην παρούσα εφαρμογή).

Αυτή η σελίδα έχει τον βασικό σκελετό της εφαρμογής αλλά ουσιαστικά δεν έχει περιεχόμενο.

Αυτό που περιέχει όμως είναι τα απαραίτητα script που μόλις φορτωθούν (κάνοντας τις απαραίτητες αιτήσεις στον εξυπηρετητή) θα κάνουν όλες τις απαραίτητες αλλαγές και προσθήκες στην αρχική σελίδα για να γίνει η σωστή αρχικοποίηση της.

### 4.3 Διαιρεί και βασίλευε

Τα αρχεία που έχουν σχέση με το κομμάτι της εφαρμογής που αλληλεπιδρά ο απλός χρήστης (client side code) βρίσκονται μέσα στον φάκελο app.

Στην εφαρμογή σχεδόν όλα τα συστήματα έχουν χωριστεί σε φακέλους που αντιπροσωπεύουν το τι είδους σύστημα είναι. Χρησιμοποιούν την ονομαστική σύμβαση “ΟνομαΒασικήςΟντότηταςManager”

Πχ PdfManager, NoteManager

## Κεφάλαιο 4ο:

Εκτός αυτού τα ίδια τα συστήματα απαρτίζονται από άλλα υποσυστήματα με τη μορφή αρχείων JavaScript.

Ο λόγος που έγινε η επιλογή της συγκεκριμένης τμηματοποίησης του κώδικα είναι γιατί επιτρέπει την καλύτερη σχεδίαση και ανάπτυξη όλων των συστημάτων, φροντίζοντας ότι το κάθε σύστημα και υποσύστημα κάνει ακριβώς αυτό που πρέπει και τίποτα άλλο.

Με αυτόν τον τρόπο αποτρέπεται η σχεδίαση δομών που χειρίζονται πάρα πολλές εργασίες, πράγμα που τις κάνει δύσχρηστες και επιρρεπείς στα σφάλματα. Είναι πολύ πιο εύκολο να διορθωθούν λάθη σε ένα σύστημα όταν αυτό είναι σχετικά μικρού μεγέθους και αναλαμβάνει ένα συγκεκριμένο αριθμό αρμοδιοτήτων.

Επίσης, ένα άλλο θετικό αποτέλεσμα αυτής της πρακτικής είναι η μεγάλη ευκολία που παρέχει όσων αφορά την επαναχρησιμοποίηση κωδικά, που είναι ιδιαίτερα θεμιτή ειδικά σε μεγάλης κλίμακας εφαρμογές, όπου γίνεται μια συνεχής προσπάθεια μείωσης του ποσοστού κώδικα που χρησιμοποιείται.

### 4.4 [JavaScript Modules](#)

Λόγο του μεγάλου αριθμού κώδικα που απαιτούν οι περισσότερες εφαρμογές σήμερα ήταν απαραίτητη η δημιουργία ενός τρόπου που θα επέτρεπε την τμηματοποίηση του κώδικα σε μικρότερα και πιο διαχειρίσιμα κομμάτια.[43]

Αυτό το πρόβλημα λύνουν τα JavaScript Modules, που έχουν πλήρη υποστήριξη από τους περισσότερους συμβατικούς περιηγητές ιστού που χρησιμοποιούνται σήμερα.

Η συγκεκριμένη τεχνολογία είναι αυτή που επιτρέπει σε όλα τα συστήματα της εφαρμογής να επικοινωνούν μεταξύ τους.

Ο τρόπος που δουλεύουν είναι ο εξής:

Καταρχάς πρέπει να υπάρχει ένας τρόπος για τον προγραμματιστή να δηλώσει ότι επιθυμεί να κάνει ένα κομμάτι κώδικα διαθέσιμο και σε άλλα μέρη της εφαρμογής.

Ο τρόπος που το κάνει αυτό είναι με το export statement. Ένα αρχείο JavaScript που έχει κάποιο export statement λέγεται module.

Όλες οι μεταβλητές που μπορεί να δηλωθούν μέσα σε ένα module είναι private και δεν μπορούν να διαβαστούν από τα άλλα τμήματα κωδικά που του χρησιμοποιούν, εκτός και αν γίνει export αυτών των μεταβλητών.

Αυτό μας επιτρέπει να έχουμε global μεταβλητές εντός του αρχείου χωρίς να μολύνουμε τα άλλα αρχεία που το χρησιμοποιούν.

Επίσης, ο περιηγητής ιστού φορτώνει ένα module μια μόνο φορά, οπότε εάν χρησιμοποιείται από πολλά άλλα κομμάτια της εφαρμογής δε σπαταλώνται πόροι για την επαναφόρτωση του, και επίσης ότι αλλαγές έχει υποστεί στον τρόπο λειτουργίας του είναι ορατές από όλα τα κομμάτια που το έχουν κάνει import. Υπάρχουν διάφοροι τρόποι που μπορεί να κάνει κανείς export.

Μπορεί να βάλει το πρόθεμα export μπροστά από τη συνάρτηση (Σχήμα 4.1), κλάση ή οποιαδήποτε άλλο στοιχείο του επιτρέπει η JavaScript να κάνει export.

```
export function arrayShuffle(array) {
  let current_index = array.length;
  let random_index = 0;

  while (current_index !== 0) {
    random_index = Math.floor(Math.random() * current_index);
    current_index--;
    [array[current_index], array[random_index]] = [
      array[random_index],
      array[current_index],
    ];
  }

  return array;
}
```

Σχήμα 4.1: Το export εμφανίζεται στην αρχή μιας συνάρτησης

```
export {
  updateQuizDefinitions,
  generateQuizzesLockedState,
  getQuizDefinitions,
  getQuizJson,
  getLockedState,
  unlockQuiz,
  getQuizStats,
  updatePassed,
  updateFailed,
  updateCanceled,
  getQuizOverride,
  setQuizOverride,
};
```

Σχήμα 4.2: Γίνεται μόνο μια εγγραφή export στο τέλος του αρχείου

Επίσης, μπορεί να επισημάνει στο τέλος του αρχείου όλα τα στοιχεία που επιθυμεί να κάνει export (Σχήμα 4.2).

## Κεφάλαιο 4ο:

Κάποια πράγμα του πρέπει να σημειωθούν είναι ότι οποιαδήποτε στοιχείο (συνάρτηση, κλάση κτλ) δεν είναι δηλωμένο με ένα από τους δυο τρόπους δεν είναι προσβάσιμο από άλλο σημείο της εφαρμογής αλλά μπορεί να χρησιμοποιηθεί από άλλα στοιχεία που είναι δηλωμένα μέσα στο ίδιο το αρχείο.

Το δεύτερο σκέλος είναι η χρήση αυτών των στοιχείων που έχουν δηλωθεί από άλλα κομμάτια της εφαρμογής, που αυτό γίνεται με τη χρήση του import statement. Υπάρχουν δυο βασικοί τύποι import, τα στατικά και τα δυναμικά.

Τα στατικά import (Σχήμα 4.3) πρέπει να εμφανίζονται στην αρχή του αρχείου, φορτώνονται στην αρχή μια φορά και έχουν την εξής δομή.

```
import { createModal } from "../ModalManager/modal-manager.js";
import { createToast } from "../ToastManager/toast-manager.js";
import {
  closeView,
  disableViewManager,
  enableViewManager,
  updateQuizView,
} from "../AuxiliaryScripts/view-manager.js";
```

Σχήμα 4.3: Δομή των import statements που εμφανίζονται στην αρχή του αρχείου

Μπορούμε να έχουμε παραπάνω από ένα statement και μπορούμε να κάνουμε ένα ή πολλά imports για το κάθε ένα.

Στο τέλος του κάθε ένα statement αναγράφεται το path στο οποίο βρίσκεται το αρχείο που περιέχει τα αιτούμενα imports.

Τα δυναμικά imports (Σχήμα 4.4) μπορούν να εμφανίζονται σε οποιαδήποτε σημείο μέσα σε ένα αρχείο και έχουν διαφορετική σύνταξη με τα στατικά.

```
async updateToastAnimation() {
  this.settings.toast.animation = this.elements.toastAnimation.checked;
  await import("../SettingsManagers/toast-settings-manager.js").then(
    (module) => {
      module.setAnimation(this.settings.toast.animation);
    }
  );
}
```

Σχήμα 4.4: Συνάρτηση που κάνει χρήση δυναμικού import

Όπως και στα στατικά είναι απαραίτητο να αναφερθούμε στο σωστό αρχείο αλλά εδώ δεν αναγράφουμε το όνομα του στοιχείου που κάνουμε import αλλά μας δίνεται ένα αντικείμενο το οποίο τα περιέχει.

Ο λόγος που χρησιμοποιήθηκαν τα δυναμικά `import` είναι γιατί κάποιες φορές δεν είναι απαραίτητη η φόρτωση ενός `module` με το ξεκίνημα της εφαρμογής, οπότε έτσι γλυτώνουμε την άσκοπη σπατάλη της επεξεργαστικής ισχύς του υπολογιστή του χρήστη καθώς και τις ανούσιες αιτήσεις στον εξυπηρετητή που σερβίρει την εφαρμογή.

#### 4.5 Περιγραφή δομής βασικού συστήματος

Ένα μεγάλος αριθμός των συστημάτων που έχουν δημιουργηθεί για την εφαρμογή έχουν κάποιες αρχιτεκτονικές ομοιότητες μεταξύ τους. Ένα βασικό συστατικό ενός συστήματος είναι ο `manager`.

Σε αυτό το αρχείο υπάρχουν όλες οι συναρτήσεις που είναι υπεύθυνες για τη λήψη των απαιτούμενων αρχείων καθώς και τη διεκπεραίωση της επικοινωνίας ανάμεσα στο σύστημα και την `localStorage`.

Μετά υπάρχουν τα `components`, τα οποία είναι αυτά που υλοποιούν τη γραφική διεπαφή ενός συστήματος παρέχοντας διάφορα γραφικά στοιχεία όπως κουμπιά, λίστες, γραφήματα κτλ, καθώς και τις απαραίτητες συναρτήσεις που κάνουν την αντιστοίχιση ανάμεσα στις ενέργειες του χρήστη και τη διεκπεραίωση τους από την εφαρμογή.

Τα `components` χρησιμοποιούν τον `manager` για να έχουν πρόσβαση σε αρχεία και την `localStorage`.

Έπειτα κάποιες φορές υπάρχουν και `container-components`, τα οποία απαρτίζονται από άλλα `components` και επιτρέπουν την καλύτερη οργάνωση όλων αυτών των συστατικών καθώς και επικοινωνία ανάμεσα στα `components` με ένα τρόπο που επιτρέπει την ευκολότερη διαχείριση.

Βέβαια, υπάρχουν και συστήματα τα οποία δεν υλοποιούν γραφική διεπαφή, αλλά είναι πολύ σπάνια στη συγκεκριμένη εφαρμογή. Στις επόμενες υποενότητες γίνεται μια πιο πλήρης ανάλυση ενός τέτοιου συστήματος.

#### 4.6 Fetcher

Το συγκεκριμένο σύστημα είναι μια ειδίκευση του `fetch api`, που πιο συγκεκριμένα μας επιτρέπει να αιτούμε αρχεία `JSON` και απλού κειμένου από τον εξυπηρετητή.

Στην εφαρμογή γίνεται ιδιαίτερα εκτενής χρήση τέτοιων αρχείων για τη διεκπεραίωση των κούιζ, υπερσυνδέσμων και άλλων συστημάτων που αναφέρονται στις ακόλουθες ενότητες.

Το σύστημα είναι χωρισμένο σε δύο μέλη, των `fetcher` και τον `fetcher-worker`.

Ο `fetcher` είναι υπεύθυνος για την καταγραφή, οργάνωση των αιτήσεων που γίνονται μέσω των συναρτήσεων που παρέχει στα υπόλοιπα συστήματα. Υπάρχουν μέσα του δομές που αποθηκεύουν προσωρινά όλα τα αιτήματα μέχρι αυτά να διεκπεραιωθούν.

Μόλις λάβει μέρος η διεκπεραίωση τότε διαγράφονται αυτόματα, επιτρέποντας μας έτσι να μπορούμε να κάνουμε πολλές διαφορετικές αιτήσεις ταυτόχρονα.

Έχει 4 βασικές συναρτήσεις που χωρίζονται σε δύο κατηγορίες, αυτές που χειρίζονται τα αρχεία `JSON` και αυτές του απλού κειμένου.

Θα αναφερθούμε σε αυτές που χειρίζονται `JSON`, καθώς έχουν παρόμοια υλοποίηση με αυτές του απλού κειμένου.

Η συνάρτηση `fetchJson(dataUrl,callback)` δέχεται ως παραμέτρους ένα `string` που αντιπροσωπεύει την τοποθεσία ενός αρχείου `JSON` στον εξυπηρετητή και μια συνάρτηση τύπου `callback` που θα

## Κεφάλαιο 4ο:

εκτελεστεί μόλις γίνει η λήψη του συγκεκριμένου αρχείου έχοντας περασμένο μέσα της το περιεχόμενο του αρχείου σε μορφή αντικείμενου JavaScript.

Η συνάρτηση `returnPlainJson(dataUrl)` δέχεται ως παράμετρο ένα string που αντιπροσωπεύει την τοποθεσία ενός αρχείου JSON στον εξυπηρετητή και επιστρέφει τα δεδομένα του αρχείου σε μορφή αντικείμενου JavaScript σε όποιον την κάλεσε.

Ο λόγος ύπαρξης της είναι γιατί πολλές φορές πρέπει να επεξεργαστούμε τα δεδομένα πριν αυτά μεταφερθούν σε μια συνάρτηση. Αξίζει να σημειωθεί ότι εδώ γίνεται η χρήση της `async` μορφής σύνταξης συναρτήσεων.

Με αυτόν τον τρόπο μπορούμε να γράψουμε ασύγχρονο κώδικα σε μορφή που είναι παρόμοια με αυτή ενός σύγχρονου, κάνοντας πιο εύκολη την κατανόηση καθώς και αποσφαλμάτωση του κώδικα.

Με τον όρο ασύγχρονο κώδικα εννοούμε κώδικα που ο χρόνος εκτέλεσης του δεν είναι προβλέψιμος εξαρχής.

Πολλές από τις λειτουργίες που αιτούμαστε να κάνουμε είναι ασύγχρονης φύσης, και εάν προσπαθούσαμε να εκτελέσουμε αυτές τις λειτουργίες με σύγχρονο τρόπο θα προκαλούσαν το "πάγωμα" της διεπαφής χρήστη, πράγμα που πρέπει να αποφεύγουμε πάση θυσία.

Ο `fetcher-worker` είναι υπεύθυνος για τη διεκπεραίωση των αιτήσεων.

Αυτό που κάνει είναι με τις πληροφορίες που έχει λάβει για μια αίτηση από τον `fetcher` καλεί τον εξυπηρετητή να του στείλει το αιτούμενο αρχείο.

Εάν αυτό το αρχείο υπάρχει ο εξυπηρετητής το στέλνει πίσω στον `fetcher-worker` όπου μετά αυτός ειδοποιεί τον `fetch` ότι έχει διεκπεραιωθεί ή αίτηση.

Σε περίπτωση που δεν υπάρχει το αιτούμενο αρχείο επιστρέφει ένα κωδικό σφάλματος μην εύρεσης του αρχείου που μπορεί να χρησιμοποιηθεί για ειδοποιηθεί ο χρήστης ανάλογα.

Η υλοποίηση του `fetcher-worker` γίνεται με τη χρήση του `Worker API`, που μας επιτρέπει να εκτελούμε όλες τις αιτήσεις σε άλλο νήμα του υπολογιστή, μην μπλοκάροντας έτσι το βασικό νήμα που τρέχει η εφαρμογή κάνοντας τη να μη σταματά όλες τις εκτελούμενες λειτουργίες κάθε φορά που θα πραγματοποιούσαμε μια αίτηση, πράγμα που θα έκανε την εφαρμογή πολύ αργή και εκνευριστική ως προς τη χρήση της.

### 4.7 View Components

Στην εφαρμογή οτιδήποτε δεν είναι η μπάρα πλοήγησης, το πλευρικό μενού, `toast` ή `modal` είναι ένα `ViewComponent` (ή σελίδα). Τα `ViewComponents` χωρίζονται σε δύο κατηγορίες, αυτές που είναι σελίδες προορισμού και αυτές που είναι ιδιικής λειτουργίας.

Οι περισσότερες σελίδες προορισμού είναι πιο γενικές και έχουν απλά μια λίστα από επιλογές, όπως τα διαθέσιμα αρχεία ύλης, τις ασκήσεις, τα κουίζ που οδηγούν σε σελίδες ιδιικής λειτουργίας, όμως υπάρχουν και εξαιρέσεις, όπως η σελίδα των ρυθμίσεων.

Είναι ο τρόπος που παρουσιάζονται και οργανώνονται όλα τα στοιχεία της διεπαφής. Σε αυτόν τον φάκελο περιέχονται όλες οι σελίδες προορισμού. Είναι αυτές οι σελίδες που μπορούν να προσπελαστούν από το πλευρικό μενού, που εξηγείται στην ενότητα 5.2.

## 4.8 AudioManager

Αυτό το σύστημα είναι υπεύθυνο για τη διαχείριση του ήχου στην εφαρμογή. Ο τρόπος που δουλεύει είναι ο εξής. Μέσα στον φάκελο του συστήματος υπάρχει ένας υποφάκελλος με όνομα AudioFiles.

Όπως προϋποθέτει το όνομα, μέσα σε αυτόν τον φάκελο υπάρχουν όλα τα διαθέσιμα ηχητικά αρχεία.

Το αρχείο audio-manager.js είναι αυτό που υλοποιεί το σύστημα. Μέσα στο αρχείο υπάρχει ένα αντικείμενο που κάνει την αντιστοιχία ενός ονόματος οντότητας με το όνομα ενός αρχείου.

Αν δηλαδή μέσα στον φάκελο AudioFiles υπάρχει ένα αρχείο test.mp3 τότε δημιουργείται μια εγγραφή του τύπου "testSound" : "test.mp3".

Με αυτόν τον τρόπο μπορούμε να δώσουμε οποιαδήποτε όνομα επιθυμούμε χωρίς να χιάζεται να αλλάζουμε το όνομα του αρχείου. Εκτός αυτού, υπάρχουν και δύο μεταβλητές που ορίζουν τη λειτουργία του.

Η μια εξ αυτών είναι η muteStatus, που ορίζει εάν υπάρχει σίγαση του ήχου η όχι ανάλογα με το αν είναι true ή false. Η άλλη μεταβλητή είναι η volumeNumber, που ορίζει την ένταση με την οποία θα ακουστούν τα ηχητικά αρχεία.

Έπειτα υπάρχει μια συνάρτηση με όνομα playSound(fileName), όπου εφόσον ο ήχος είναι ενεργοποιημένος (δεν έχουμε απενεργοποιήσει τον audio-manager κάνοντας τον mute) μέσα στην εφαρμογή και η παράμετρος της συνάρτησης υπάρχει ως εγγραφή (πχ testSound) τότε παίζει το συγκεκριμένο αρχείο ήχου.

Εδώ πρέπει να αναφέρουμε ότι πολλά από τα συστήματα που έχουν υλοποιηθεί παρέχουν τη δυνατότητα παραμετροποίησης της λειτουργίας τους μέσω διάφορων ρυθμίσεων που είναι προσαρμοσμένες στο εκάστοτε σύστημα.

Ο τρόπος που το κάνουν αυτό είναι χρησιμοποιώντας κάποια άλλα συστήματα που έχουν την ευθύνη της αποθήκευσης των ρυθμίσεων στην localStorage καθώς και τον άμεσο συγχρονισμό τους. Αυτά τα συστήματα βρίσκονται μέσα στον φάκελο SettingsManagers. Για κάθε ένα σύστημα που επιτρέπει την παραμετροποίηση του υπάρχει και ο αντίστοιχος "systemName"-settings-manager.js.

Αυτό που κάνει ένας settings-manager είναι να αρχικοποιεί τις διάφορες μεταβλητές που χρησιμοποιούνται για την υλοποίηση των εκάστοτε ρυθμίσεων μέσα στην localStorage εάν δεν υπάρχουν, για να μπορούμε να τις έχουμε την επόμενη φορά που ανοίγει η εφαρμογή, και να παρέχει συναρτήσεις που επιτρέπουν τις αλλαγές αυτών των μεταβλητών.

Ένα πρόβλημα που έπρεπε να λυθεί ήταν ο συγχρονισμός των ρυθμίσεων ανάμεσα στην localStorage και στο σύστημα που λάμβαναν μέρος οι αλλαγές.

Μέσα σε κάθε settings-manager υπάρχει μια συνάρτηση loadSettings, όπου αυτή επιστρέφει όλες τις μεταβλητές που είναι αποθηκευμένες στην localStorage και έχουν σχέση με το εκάστοτε σύστημα.

Σε κάθε σύστημα που κάνει χρήση ενός settings-manager υπάρχει μια συνάρτηση changeSettings, που αυτό που κάνει είναι να καλεί την loadSettings και να ανανεώνει τις μεταβλητές της με αυτές που είναι αποθηκευμένες στην localStorage.

Όταν κάνουμε μια αλλαγή σε μια ρύθμιση με τη χρήση των συναρτήσεων που μας παρέχει ο settings-manager τότε καλείται μια συνάρτηση που λέγεται updateSettings.

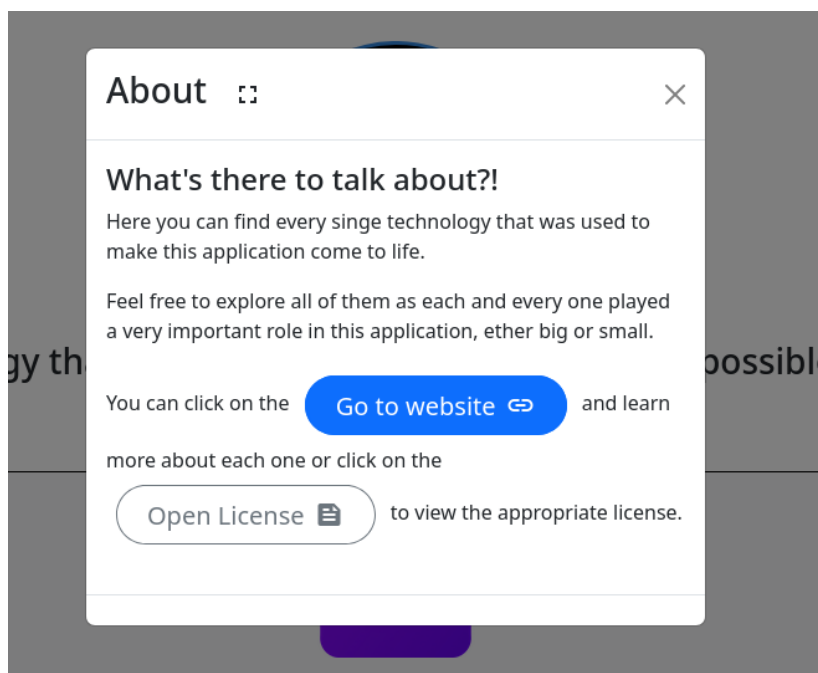
Κεφάλαιο 4ο:

Αυτή η συνάρτηση συγχρονίζει την localStorage με όλες τις αλλαγές που έχουν συμβεί και μετά καλεί την changeSettings του συστήματος που τον χρησιμοποιεί. Με αυτόν τον τρόπο η localStorage και το σύστημα είναι πάντα συγκεκριμένα, χωρίς να απαιτείται η επανεκκίνηση της εφαρμογής.

## 4.9 ModalManager

Στην εφαρμογή γίνεται εκτενής χρήση των modal, για αυτό και κρίθηκε απαραίτητη η δημιουργία ενός συστήματος που θα επέτρεπε την οργάνωση και τη διευκόλυνση ως προς τη χρήση τους.

Ένα modal (Σχήμα 4.5) είναι ένα μικρό παράθυρο προσκηνίου που έχει ως σκοπό να τραβήξει την προσοχή του χρήστη για μια ενέργεια που απαιτεί την έγκριση του ή για την παρουσίαση πληροφοριών που πρέπει να γίνουν άμεσα ορατές σε αυτόν.



Σχήμα 4.5: Εμφάνιση ενός Modal

Τα modal παρέχονται από την Bootstrap, απλά έχει γίνει η επαύξηση και εξατομίκευση τους με καινούριες λειτουργίες.

Τα modal της εφαρμογής είναι χωρισμένα σε δύο τμήματα, αυτό του manager και αυτό του component που στην πραγματικότητα είναι ένα web component. Ο τρόπος που δουλεύει ένα web component είναι ο εξής. Πρώτον πρέπει να γίνει η κληρονόμηση όλων των απαραίτητων λειτουργιών ενός απλού HTML Element. Αυτό γίνεται με το να κάνουμε τη δική μας κλάση να κάνει extend την κλάση HTMLElement.

Έπειτα δημιουργούμε τον constructor, και μέσα σε αυτόν καλούμε τη super συνάρτηση για να κληρονομήσουμε όλες τις μεταβλητές και τις μεθόδους από τη super class, και ορίζουμε όσες μεταβλητές χρειάζονται για τη λειτουργία του web component.

Στην προκριμένη περίπτωση ορίζουμε μια μεταβλητή που δείχνει εάν το modal είναι σε πλήρη οθόνη ή όχι. Μετά ορίζουμε τη συνάρτηση `connectedCallback`, η οποία καλείται κάθε φορά που γίνεται η εμφάνιση του web component (συγκεκριμένα όταν αυτό συνδέεται στο DOM), και μέσα της ορίζεται το τι θα εμφανίζει στη διεπαφή.

Επίσης, εδώ γίνεται η ένωση των γραφικών στοιχείων με τις συναρτήσεις που αντιπροσωπεύουν καθώς και τυχόν τροποποιήσεις που πρέπει να γίνουν στη διεπαφή του web component.

Στην προκριμένη περίπτωση εάν είναι ενεργοποιημένη η δυνατότητα της πλήρους οθόνης τότε συνδέουμε το κουμπί με την κατάλληλη συνάρτηση, αλλιώς δεν κάνουμε τη σύνδεση και κρύβουμε το κουμπί.

Έπειτα ορίζουμε τη συνάρτηση `disconnectedCallback`, που καλείται κάθε φορά που το web component αφαιρείται από το DOM.

Δεν είναι απαραίτητη η συμπερίληψη της, αλλά μας δίνει τη δυνατότητα να εκτελέσουμε όποιες άλλες ενέργειες θέλουμε πριν καταστραφεί το web component.

Τέλος, πρέπει να δηλώσουμε το web component ώστε να είναι αναγνωρίσιμο από το DOM και επίσης πρέπει να κάνουμε export το web component για να μπορεί να χρησιμοποιηθεί από τα υπόλοιπα συστήματα.

Ο manager είναι αυτός που παρέχει τη μέθοδο δημιουργίας ενός modal μέσω της συνάρτησης `createModal`, η οποία έχει δέχεται αυτές τις παραμέτρους:

- `title` : ορίζει τον τίτλο του modal (είναι προαιρετικός).
- `body` : ορίζει το κυρίως σώμα του modal, που μπορεί να είναι απλό κείμενο ή HTML markup.
- `footer` : ορίζει το περιεχόμενο αμέσως από κάτω του σώματος (είναι προαιρετικός).

Για τα `title`, `body`, `footer` μπορούμε να δώσουμε τις παραμέτρους σαν string ή σαν αντικείμενα DOM.

- `soundFileName` : ορίζει τον ήχο που ακουστεί μόλις εμφανιστεί το modal με βάσει τις εγγραφές του `AudioManager` (είναι προαιρετικό).
- `optionsOverride` : ορίζει τυχόν αλλαγές ως προς τον βασικό τρόπο λειτουργίας (πχ τρόπο εμφάνισης) που θέλουμε να έχουμε για ένα modal που είναι διαφορετικές από τις προκαθορισμένες (είναι προαιρετικό).
- `fullScreenEnabled` : ορίζει εάν μπορεί να μεγιστοποιηθεί το modal, που για κάποια modal ενδέχεται να είναι αρκετά χρήσιμο εάν έχουν πολύ περιεχόμενο (είναι προαιρετικό).

Επίσης, είναι υπεύθυνος για τη διαχείριση της εμφάνιση πολλαπλών modal.

Εάν δύο η παραπάνω modal είναι ενεργά, εμφανίζεται μόνο το πιο πρόσφατο.

Μόλις επέλθει το κλείσιμο του πρώτου modal τότε ο manager εμφανίζει το επόμενο αυτόματα, μέχρι να μην υπάρχουν άλλα modal για εμφάνιση.

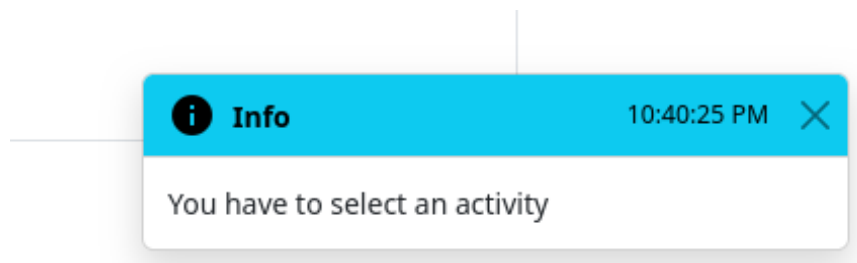
Με παρόμοιο τρόπο έχουν υλοποιηθεί και τα υπόλοιπα συστήματα που κάνουν χρήση των web component, φυσικά έχοντας πολλές έξτρα συναρτήσεις και μεταβλητές ανάλογα με την πολυπλοκότητα του.

#### 4.10 ToastManager

Τα toast (Σχήμα 4.6) είναι ένας πολύ εύχρηστος τρόπος για να μεταφέρουμε στον χρήστη την κατάσταση του συστήματος όταν ολοκληρώνεται μια ενέργεια.

Με τα toast ειδοποιούμε τον χρήστη όταν μια ενέργεια ολοκληρώθηκε με επιτυχία ή με σφάλμα.

Μπορούμε να τον προειδοποιήσουμε για μια ενέργεια που ενδέχεται να είναι επικίνδυνη ή να του παράσχουμε πληροφορίες πριν τη διένεξη μιας ενέργειας χωρίς απαραίτητα να διακόψουμε οτιδήποτε έκανε ήδη.



Σχήμα 4.6: Εμφάνιση ενός Toast

Τα toast παρέχονται από την Bootstrap, αλλά έχει γίνει η περαιτέρω εξατομίκευση τους.

Στην εφαρμογή υπάρχουν τέσσερα είδη toast :

- info : χρησιμοποιούνται για την ενημέρωση του χρήστη για μια ενέργεια.
- success : χρησιμοποιούνται για να δηλώνουν την επιτυχή διένεξη μιας ενέργειας.
- warning : χρησιμοποιούνται για την προειδοποίηση του χρήστη για μια πιθανά επικίνδυνη ενέργεια.
- error : χρησιμοποιούνται για την ενημέρωση του χρήστη για κάποιο σφάλμα που συνέβη κατά της εκτέλεση μιας ενέργειας.

Οι διαφορές μεταξύ τους είναι ο τίτλος καθώς και το χρώμα αυτού (τα χρώματα έχουν επιλεγεί κατάλληλα έτσι ώστε να είναι τα αναμενόμενα που χρησιμοποιούνται στις περισσότερες διεπαφές χρήστη), όπως και επίσης ο ήχος (εξίσου επιλεγμένος με τα προαναφερθέντα κριτήρια) που ακούγεται κατά την εμφάνιση τους, δεδομένο ότι έχει ενεργοποιηθεί αυτή τους η λειτουργία.

Ο manager είναι αυτός που παρέχει τη μέθοδο δημιουργίας ενός toast μέσω της συνάρτησης createToast, η οποία έχει δέχεται αυτές τις παραμέτρους:

- mode : ορίζει το τι τύπου toast θα είναι (μπορεί να είναι μια από τις τέσσερις επιλογές)
- text : ορίζει το κείμενο που θα εμφανίσει το toast όταν γίνει ορατό.
- soundOption : ορίζει το αν θα ακουστεί ο προκαθορισμένος ήχος του toast η όχι με βάση το αν η παράμετρος είναι true ή false.
- optionsOverride : ορίζει τυχόν αλλαγές ως προς τον βασικό τρόπο λειτουργίας (πχ τρόπο εμφάνισης) που θέλουμε να έχουμε για ένα toast που είναι διαφορετικές από τις προκαθορισμένες (είναι προαιρετικό).

Επίσης, είναι υπεύθυνος για την τοποθέτηση των toasts στην κάτω δεξιά γωνιά της εφαρμογής, καθώς εκεί είναι η πιο σύνηθες σημείο τοποθέτησης αυτού του στοιχείου διεπαφής.

#### 4.11 LoadingScreen

Σε αυτόν τον φάκελο υπάρχει η υλοποίηση μας απλής οθόνης φόρτωσης.

Όταν μια ενέργεια ενδέχεται να χρειαστεί αρκετή ώρα να τελειώσει ή ο χρόνος διεκπεραίωσης της είναι άγνωστος τότε είναι καλό να μπορούμε να επικοινωνήσουμε το συγκεκριμένο γεγονός στον χρήστη, για να μην έχει την εντύπωση ότι η εφαρμογή δεν ανταποκρίνεται.

Αυτό μπορεί να γίνει εύκολα με μια οθόνη φόρτωσης (Σχήμα 4.7).



Σχήμα 4.7: Εμφάνιση οθόνης φόρτωσης

Πολύ απλά υπάρχουν δύο μέθοδοι, `createLoadingScreen(targetElement)` και `removeLoadingScreen(targetElement)`, οι οποίες δέχονται ως παράμετρο το HTML στοιχείο που θα επικαλύψουν, και στη συνέχεια δημιουργούν και αντίστοιχα καταστρέφουν την οθόνη φόρτωσης.

Η οθόνη δημιουργείται όταν ξεκινάει η εν λόγω ενέργεια και καταστρέφεται όταν πια έχει λάβει μέρος η ολοκλήρωση της.

#### 4.12 Cdn Files

Στον φάκελο αυτό είναι αποθηκευμένες όλες οι εξωτερικές βιβλιοθήκες (δεν είναι μέρος της απλής JavaScript), τεχνολογίες καθώς και άλλα απαραίτητα για τη διεκπεραίωση των λειτουργιών τους αρχεία, όπως πχ τα αρχεία τύπου CSS της Bootstrap για τον ορισμό των κανόνων εμφάνισης όλων των στοιχείων διεπαφής.

Κανονικά όλα αυτά τα αρχεία βρίσκονται σε διάφορα CDN (Content Delivery Networks) και για να γίνει η λήψη τους θα χειρίζονταν η επικοινωνία με πολλούς εξωτερικούς εξυπηρετητές, πράγμα που θα έκανε τη φόρτωση της αρχικής σελίδας πολύ αργή, που έχει ως άμεσο αποτέλεσμα τον εκνευρισμό του χρήστη.

Κεφάλαιο 4ο:

Έχοντας ένα αντίγραφο τους στον εξυπηρετητή που σερβίρει την εφαρμογή, μειώνουμε ραγδαία τον χρόνο φόρτωσης, καθώς μπορεί να λάβει μέρος η βελτιστοποίηση των αιτήσεων εντός του εξυπηρετητή.

Εκτός αυτού, για κάποιες βιβλιοθήκες ήταν απαραίτητη η αντιγραφή τους στον εξυπηρετητή καθώς κάποιες από τις λειτουργίες τους δε θα ήταν διαθέσιμες.

#### 4.13 Content Files

Σε αυτόν τον φάκελο είναι αποθηκευμένα όλα τα αρχεία που έχουν σχέση με το περιεχόμενο της εφαρμογής. Εδώ βρίσκονται όλες τις ασκήσεις, κούιζ, σημειώσεις, αρχεία βοήθειας, αρχεία ύλης καθώς και άλλοι τύποι αρχείων που βοηθούν την οργάνωση και διαχείριση του περιεχομένου της εφαρμογής.

Ο λόγος που έχουν τοποθετηθεί μέσα σε αυτόν τον φάκελο είναι για να διευκολυνθεί η διαδικασία της εύρεσης και της επεξεργασίας τους. Ο καθηγητής που κάνει χρήση της εφαρμογής χρειάζεται πρακτικά να κάνει αλλαγές στα αρχεία μόνο αυτού του φακέλου, χωρίς να χρειάζεται να επεμβαίνει στα αρχεία κώδικα της υπόλοιπης εφαρμογής, πράγμα που θα έκανε τέτοιες επεμβάσεις ιδιαίτερα δύσκολοι.

Περαιτέρω λεπτομέρειες ως προς τη δομή των υποφακέλων και των αρχείων δίνονται στα επόμενα κεφάλαια.

Στις ακόλουθες ενότητες όταν γίνεται αναφορά σε φακέλους που στο όνομα τους περιέχουν το “ContentFiles” σημαίνει ότι βρίσκονται μέσα σε αυτόν τον φάκελο.

#### 4.14 AuxiliaryScripts

Σε αυτόν τον φάκελο υπάρχουν πολλά βοηθητικά script τα οποία δεν μπορούν να κατηγοριοποιηθούν ως συστήματα αλλά δεν παύουν να είναι ιδιαίτερα σημαντικά για τη διεκπεραίωση πάρα πολλών λειτουργιών της εφαρμογής. Πολλά από αυτά αναλύονται σε επόμενες ενότητες. Σε αυτή την ενότητα θα αναλυθούν δύο. Το πρώτο είναι το αρχείο `utils.js`.

Μέσα σε αυτό υπάρχει ένας μεγάλος αριθμός από γενικές (δεν έχουν εξωτερικές απαιτήσεις και λειτουργούν πάνω σε απλές δομές δεδομένων της JavaScript) βοηθητικές συναρτήσεις που χρησιμοποιούνται σχεδόν από κάθε ένα από τα υπόλοιπα συστήματα.

Αξίζει να σημειωθεί ότι οι συναρτήσεις, λόγω της γενικότητας του μπορούν να χρησιμοποιηθούν στην ανάπτυξη οποιασδήποτε άλλης εφαρμογής JavaScript χωρίς καμία τροποποίηση σε αυτές.

Μέσα στο σύνολο όλων αυτών των συναρτήσεων ξεχωρίζουν οι εξής :

`stringToNode(nodeString)` : αυτή η συνάρτηση έχει τη δυνατότητα να μετατρέπει την παράμετρο `nodeString` που περιγράφει στοιχεία HTML στα αντίστοιχα αντικείμενα DOM που μπορούμε να χρησιμοποιήσουμε εντός της JavaScript.

Με αυτόν τον τρόπο αποφεύγουμε τη συγγραφή όλου του κώδικα που θα χρειαζόταν για την τυπική περιγραφή τους, καθώς θα έπρεπε να έχουμε το λιγότερο 3 γραμμές κώδικα για τον ορισμό ενός μόνο στοιχείου, χωρίς να λαμβάνουμε υπόψιν τυχόν έξτρα αλλαγές που μπορεί να είναι απαραίτητες για το κάθε ένα.

`stringToStyleSheetNode(styleSheetString)` : η συνάρτηση αυτή μας επιτρέπει να μετατρέπουμε την παράμετρο `styleSheetString` που περιγράφει τους κανόνες εμφάνισης στοιχείων HTML με CSS, σε ένα αντικείμενο που μπορεί να χρησιμοποιηθεί άμεσα από τη JavaScript.

Αλλάζοντας προγραμματιστικά κάθε απαραίτητη ιδιότητα CSS ενός στοιχείου θα ήταν απίστευτα χρονοβόρα αλλά και ανελαστική, καθώς η κάθε αλλαγή θα επηρέαζε μόνο ένα στοιχείο, ενώ ένας κανόνας μπορεί να επηρεάσει την εμφάνιση πολλών στοιχείων μαζί.

Το άλλο απλό script που θα αναφερθεί είναι το `single-instance.js`.

Με την προσθήκη αυτού του script στο αρχείο `index.html`, μπορούμε να αποτρέψουμε το άνοιγμα της εφαρμογής για δεύτερη φορά, καθώς η παράλληλη εκτέλεση των δύο μπορεί να προκαλέσει αποσυγχρονισμό των ρυθμίσεων καθώς και απρόβλεπτα σφάλματα.

Ο τρόπος που το κάνει αυτό είναι ελέγχοντας την ύπαρξη μιας μεταβλητής στην `localStorage`.

Εάν αυτή η μεταβλητή δεν υπάρχει σημαίνει ότι η εφαρμογή εκτελείται για πρώτη φορά, εάν όμως υπάρχει ήδη σημαίνει ότι εκτελείται για δεύτερη φορά και αυτό που γίνεται είναι ο παύση φόρτωσης της εφαρμογής και η εμφάνιση ενός προειδοποιητικού μηνύματος.

Τώρα εάν κάποιος επιθυμεί να αφαιρέσει αυτή τη λειτουργία το μόνο που πρέπει να κάνει είναι να αφαιρέσει το script από το αρχείο `index.html`.

#### 4.15 Επίλογος

Στη συγκεκριμένη ενότητα έγινε η επεξήγηση του τρόπου οργάνωσης της εφαρμογής καθώς και πολλών συστημάτων που χρησιμοποιούνται σε μεγάλο βαθμό για την υλοποίηση της.

Πρέπει να σημειωθεί ότι η παράθεση αυτών των συστημάτων είναι σχετικά συνοπτική, καθώς απαιτείται η παρουσίαση των υπόλοιπων συστημάτων για να γίνει πλήρως κατανοητός ο ευρύτερος τρόπος λειτουργίας και αλληλοσύνδεσης τους. Αυτό λαμβάνει μέρος στην επόμενη ενότητα.

## Κεφάλαιο 5ο: Παρουσίαση Εφαρμογής

### 5.1 Εισαγωγή

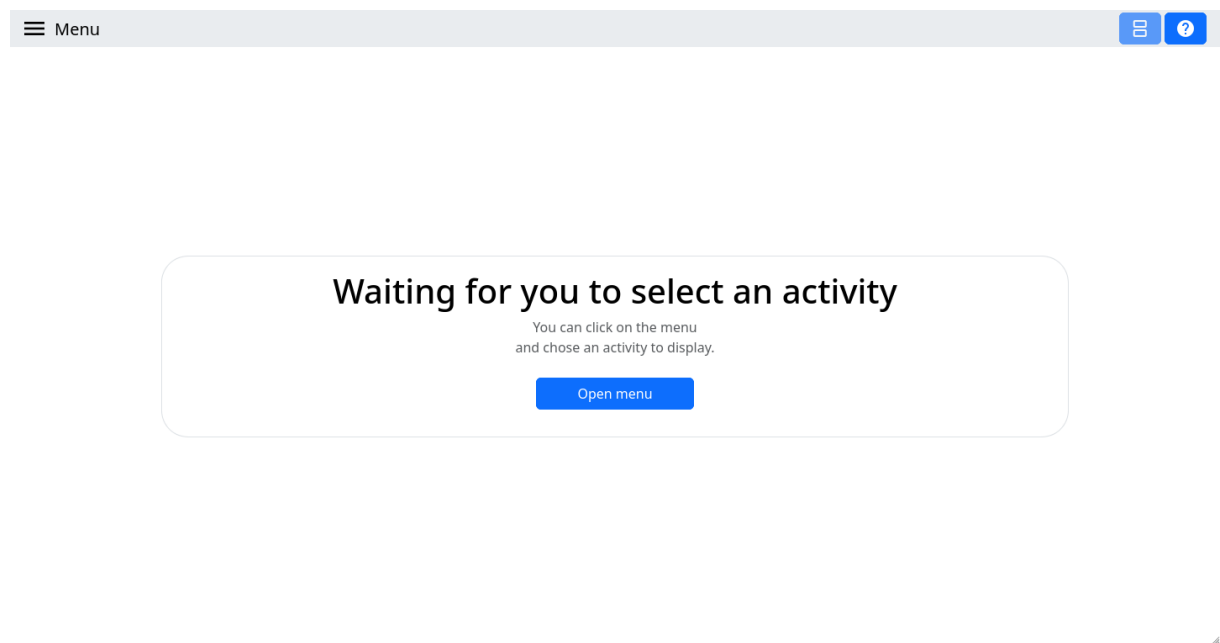
Σε αυτή την ενότητα λαμβάνει μέρος η περιγραφή των υπόλοιπων συστημάτων καθώς και οι παρουσίαση όλων των λειτουργιών που πραγματοποιούν.

Θα δοθεί περισσότερη έμφαση στα χαρακτηριστικά (features) που προσφέρει η εφαρμογή καθώς και στον τρόπο προσαρμογής τους από τον χρήστη αλλά και τον καθηγητή παρά στην εξαντλητική ανάλυση τους από τη μεριά του κώδικα που χρειάστηκαν για να υλοποιηθούν.

Βεβαίως, όπου κριθεί απαραίτητη η παράθεση κομματιών κώδικα καθώς και η παρουσίαση κάποιων σημαντικών συναρτήσεων που χρησιμοποιούνται σε μεγάλο βαθμό στην εφαρμογή τότε θα πραγματοποιούν οι ανάλογες ενέργειες επεξήγησης τους.

### 5.2 Αρχική οθόνη και τρόπος λειτουργίας των views

Η αρχική οθόνη που βλέπει ο χρήστης είναι αυτή που φαίνεται στο Σχήμα 5.1.



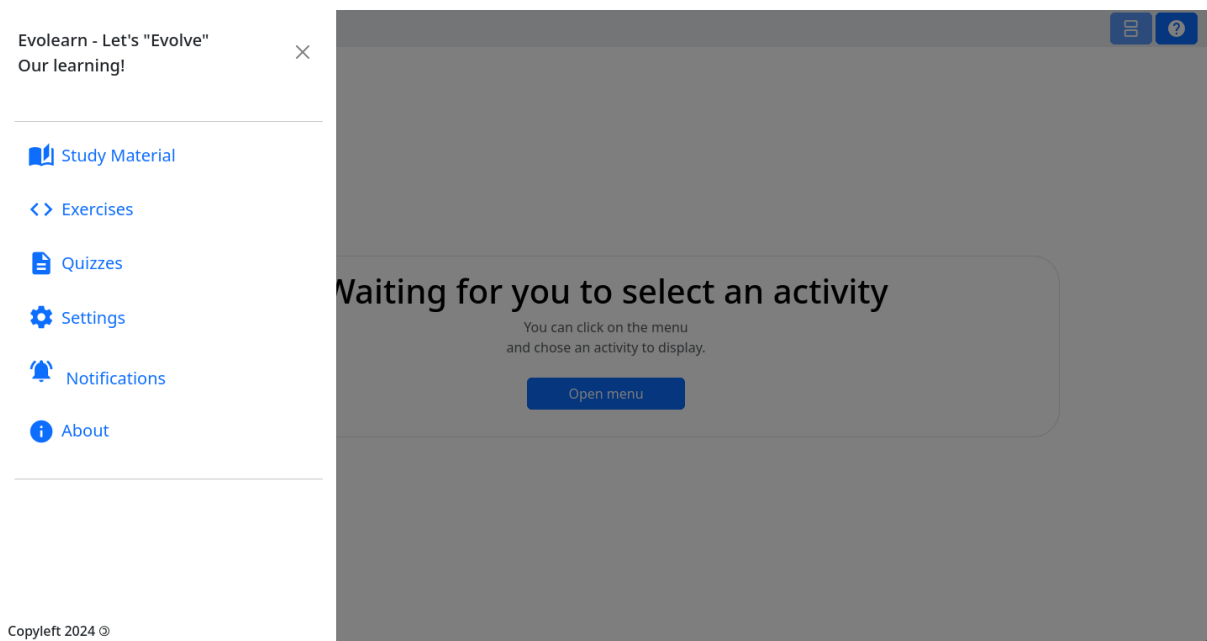
Σχήμα 5.1: Αρχική οθόνη εφαρμογής

Η συγκριμένη οθόνη ωθεί τον χρήστη να επιλέξει μια δραστηριότητα. Οι δραστηριότητες είναι ο τρόπος που γίνεται ο λογικός διαχωρισμός των τμημάτων της διεπαφής της εφαρμογής.

Το πρώτο πράγμα που κάνει ο χρήστης είναι να πατήσει το μπλε κουμπί στο κέντρο της οθόνης ή να χρησιμοποιήσει το κουμπί που βρίσκεται στην πάνω αριστερή γωνία της οθόνης.

Με αυτή την ενέργεια θα ανοίξει το μενού πλοήγησης που φαίνεται στο Σχήμα 5.2.

## Κεφάλαιο 5ο:



Σχήμα 5.2: Μενού πλοήγησης εφαρμογής

Το μενού αποτελεί το βασικό εργαλείο πλοήγησης της εφαρμογής. Από εδώ ο χρήστης μπορεί να μεταφερθεί σε όλα τα βασικά views (δραστηριότητες) που προσφέρει η εφαρμογή.

Αξίζει να σημειωθεί ότι υπάρχει σχεδόν αντιστοιχία ένα προς ένα με τα αρχεία που βρίσκονται μέσα στον φάκελο ViewComponents που αναφέρθηκε στο προηγούμενο κεφάλαιο και με τις διαθέσιμες επιλογές του μενού, καθώς εκεί είναι που γίνεται η υλοποίησή τους.

Η τοποθεσία που βρίσκεται του μενού (στην αριστερή πλευρά) επιλέχτηκε γιατί είναι η πιο σύνηθες τοποθέτηση ενός τέτοιου είδους μενού, κάνει καλύτερη χρήση του διαθέσιμου χώρου καθώς δεν είναι “επίμονο” (μόλις γίνει η επιλογή ενός view, το μενού κρύβεται αυτόματα) και διευκολύνει τη σάρωση, καθώς οι περισσότεροι ξεκινάν από την πάνω αριστερή γωνία και κατεβαίνουν προς τα κάτω όταν σαρώνουν μια σελίδα για πρώτη φορά.[44]

Κάνοντας κλικ πάνω σε ένα αντικείμενο του μενού μεταφερόμαστε στην αντίστοιχη δραστηριότητα (οθόνη, view). Μόλις λάβει μέρος αυτή η ενέργεια το κεντρικό τμήμα της οθόνης θα αντικατασταθεί από το επιλεγμένο view και θα εμφανιστούν μια σειρά από κουμπιά στο πάνω αριστερά μέρος της οθόνης όπως φαίνεται στο Σχήμα 5.3

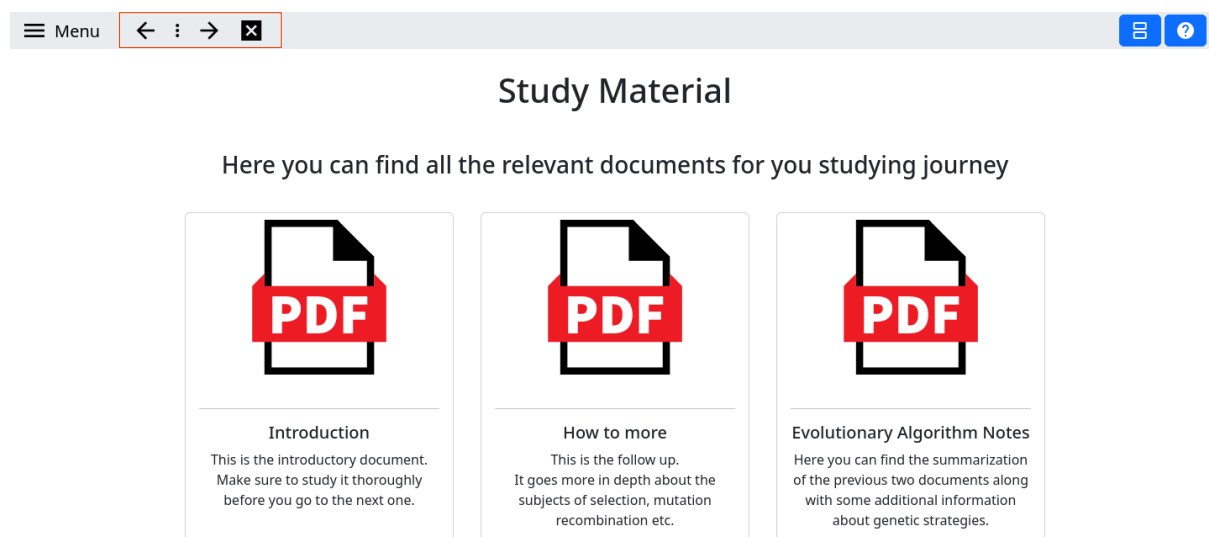
Με κάθε αλλαγή view γίνεται και η αντίστοιχη αλλαγή στον τίτλο της καρτέλας του περιηγητή ιστού αλλά και του URL της σελίδας.

Τα URL μορφοποιούνται με έναν συγκεκριμένο τρόπο έτσι ώστε να μπορούν να χρησιμοποιηθούν για την πλοήγηση σε όλα τα διαθέσιμα view της εφαρμογής.

Ένα τέτοιο URL έχει την ακόλουθη μορφή [<http://localhost:3000/?view=pdfView>].

Στην παράμετρο ?view= δίνουμε το όνομα που είναι εσωτερικά δηλωμένο και κάνοντας φόρτωση της σελίδας θα βρεθούμε αυτόματα στο view που όριζε το URL παρακάμπτοντας την αρχική οθόνη.

Αυτή η λειτουργία είναι ιδιαίτερα χρήσιμη για κάποιον που ενδεχομένως να αντιμετωπίζει κάποιο πρόβλημα σε ένα μέρος της εφαρμογής και θα ήθελε να μοιραστεί με κάποιον το ακριβές view στο οποίο αντιμετωπίζει το πρόβλημα (πχ εάν δεν καταλαβαίνει κάτι σε μια σελίδα ύλης μπορεί να στείλει ένα link στον καθηγητή με τη σελίδα, έτσι ώστε μόλις την ανοίξει ο καθηγητής στον περιηγητή ιστού του να πλοηγηθεί αυτόματα στην εν λόγω σελίδα).



Σχήμα 5.3: Εμφάνιση ενός view με τα κουμπιά πλοήγησης

Προς το παρόν δε χρειάζεστε να δοθεί σημασία στα views που απεικονίζονται στις εικόνες καθώς θα εξηγηθούν στις επόμενες υποενότητες.

Η εφαρμογή έχει τη δυνατότητα να έχει πολλαπλά views ανοιχτά την ίδια στιγμή.

Αυτό είναι χρήσιμο γιατί μας επιτρέπει να έχουμε πολλές παράλληλα εκτελούμενες εργασίες και να μπορούμε να εναλλασσόμαστε ανάμεσα σε αυτές.

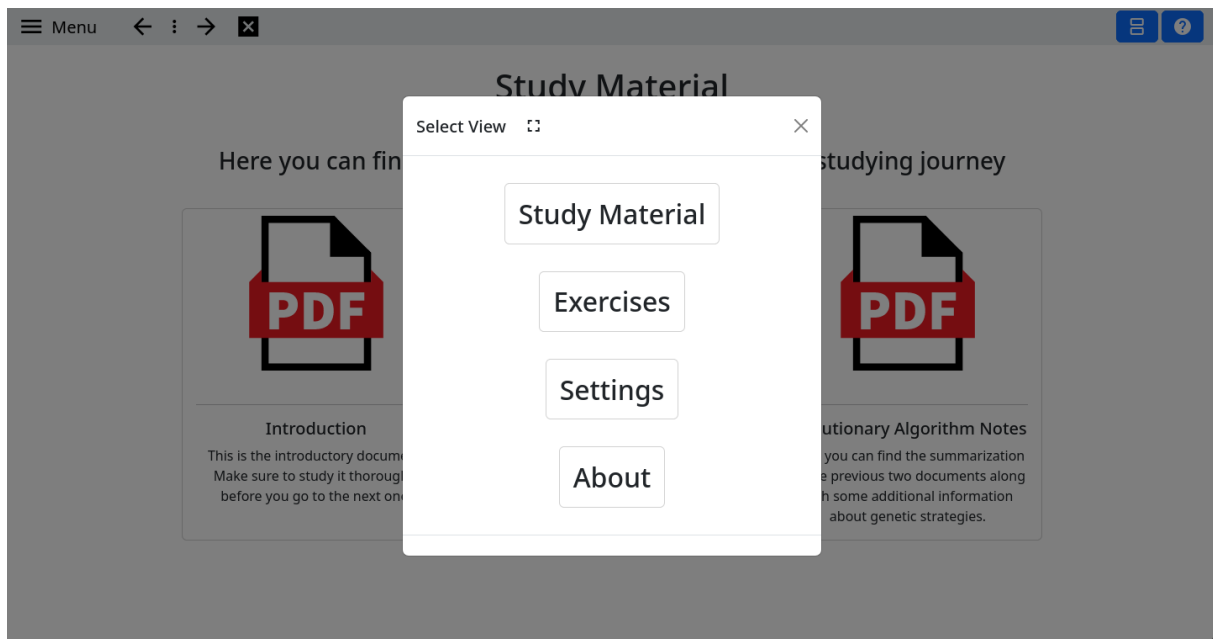
Μπορούμε πχ να ξεκινήσουμε την εκτέλεση μιας άσκησης και όσο περιμένουμε για την ολοκλήρωση της, να διαβάσουμε την ύλη που αφορά την ίδια την άσκηση.

Με τα βέλη (τα κουμπιά που δείχνουν βέλη με αριστερή και δεξιά κατεύθυνση) μπορούμε να πλοηγηθούμε στο προηγούμενο ή στο επόμενο view.

Τα views αποθηκεύονται εσωτερικά σαν μια ουρά, οπότε μόλις φτάσουμε στο τελευταίο view, τότε αν πάμε στο επόμενο θα βρεθούμε πάλι στο αρχικό (φυσικά ισχύει και το αντίθετο).

Εκτός από αυτόν τον τρόπο πλοήγησης έχουμε τη δυνατότητα να δούμε όλα τα ανοιχτά views με τη σειρά που ανοίχτηκαν πατώντας το κουμπί που βρίσκεται ανάμεσα στα βέλη, όπως φαίνεται στο Σχήμα 5.4.

## Κεφάλαιο 5ο:



Σχήμα 5.4: Μενού επιλογής ανοιχτών view

Με αυτόν τον τρόπο μπορούμε να πάμε στο επιθυμητό view χωρίς να χρειάζεται να κάνουμε πολλαπλά κλικ στα κουμπιά βέλη, προσπερνώντας έτσι τη σειρά που είναι χρήσιμο όταν υπάρχουν πολλά ανοιχτά views.

Επίσης, πρέπει να αναφερθεί ότι για τα περισσότερα view, εάν η εφαρμογή ανιχνεύσει ότι το αιτούμενο view είναι ήδη ανοιχτό δεν ανοίγει και δεύτερο αντίγραφο αλλά φέρνει στο προσκήνιο το ήδη υπάρχον.

Καθώς τα views είναι τα θεμελιώδη κομμάτια της εφαρμογής κρίθηκε άκρος απαραίτητη η υλοποίηση διαφόρων μεθόδων για την πλοήγηση μεταξύ τους. Εκτός από τα προαναφερθέντα κουμπιά υπάρχει και αυτό που εκτελεί το κλείσιμο ενός view.

Πατώντας το, το view που βρίσκεται στο προσκήνιο διαγράφεται από τη διεπαφή και τυχόν ανυποθήκευτες αλλαγές καταστρέφονται. Εάν κλείσουν όλα τα ανοικτά views, τότε η εφαρμογή επιστρέφει στην αρχική οθόνη (Σχήμα 5.1).

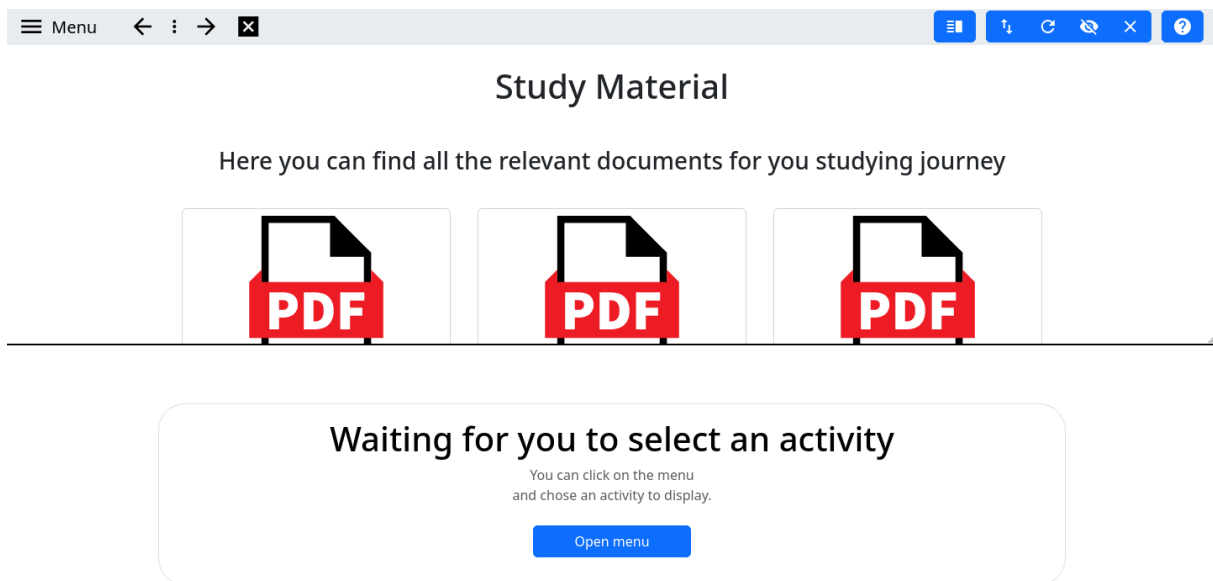
Εκτός από την αλλαγή ανάμεσα στα views υπάρχει και η δυνατότητα της παράθεσης δυο view στην οθόνη την ίδια στιγμή.

Αυτό μπορεί να πραγματοποιηθεί με το πάτημα του αριστερού κουμπιού που φαίνεται στο Σχήμα 5.5.



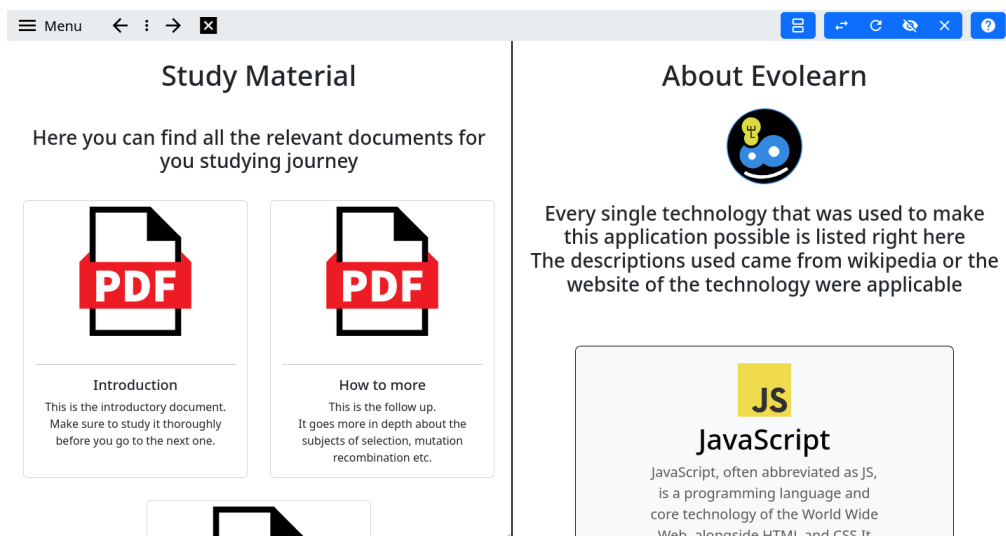
Σχήμα 5.5: Κουμπιά διαχωρισμού οθόνης και βοήθειας

Μόλις γίνει το πάτημα αυτού του κουμπιού η οθόνη φαίνεται κάπως έτσι (Σχήμα 5.6).



Σχήμα 5.6: Οριζόντιος διαχωρισμός οθόνης

Τώρα μπορούμε να παρατηρήσουμε ότι η οθόνη έχει χωριστεί στη μέση οριζόντια, στο κάτω μέρος έχουμε ένα παρόμοιο μήνυμα με αυτό της αρχικής οθόνης και έχουν προστεθεί έξτρα κουμπιά στην μπάρα πλοήγησης.



Σχήμα 5.7: Κατακόρυφος διαχωρισμός οθόνης

Κάθε νέο view που ανοίγει τώρα θα πηγαίνει στο κάτω μέρος της οθόνης χωρίς να αντικαθιστά αυτό που βρίσκεται στο πάνω μέρος.

## Κεφάλαιο 5ο:

Πρέπει να σημειωθεί ότι με την ενεργοποίηση αυτής της λειτουργίας δεν κρατούνται πολλαπλά views στο κάτω μέρος αλλά κάθε καινούριο αντικαθιστά το παλιό.

Οι έξτρα λειτουργίες θα αναφερθούν με τη σειρά των κουμπιών που εμφανίζονται στο Σχήμα 5.8 με κατάταξη από τα αριστερά προς τα δεξιά.

Το πρώτο κουμπί κάνει τον κατακόρυφο διαχωρισμό της οθόνης (Σχήμα 5.7). Το δεύτερο κάνει ανταλλαγή των ανοιχτών views είτε κατακόρυφα είτε οριζόντια, ανάλογα με την επιλογή που ισχύει τη δεδομένη στιγμή (αναφέρεται ότι το κουμπί αλλάζει χρώμα για να δηλώσει ότι έχει επέλθει αλλαγή στον τρόπο εμφάνισης των views).

Το τρίτο αναιρεί τυχόν αλλαγές έχουν γίνει ως προς την αλλαγή μεγέθους του διαχωρισμού των views (στο τέλος της διαχωριστικής μπάρας ανάμεσα στα ανοιχτά views υπάρχει ένα τρυγονάκι που επιτρέπει την αλλαγή του μοιράσματος της οθόνης).



Σχήμα 5.8: Κουμπιά έξτρα λειτουργιών διαχωρισμού οθόνης

Το τέταρτο κουμπί κρύβει το προσωρινό view (προσωρινό view θεωρείται αυτό που ανοίγει μετά την ενεργοποίηση του διαχωρισμού οθόνης), χωρίς όμως να το καταστρέφει.

Εάν πατηθεί το πρώτο κουμπί τότε γίνεται η επαναφορά του view με τις εξατομικεύσεις που είχε πριν κρυφτεί. Το πέμπτο κουμπί καταστρέφει το προσωρινό view και επιστρέφει η οθόνη στην κατάσταση που ήταν στο Σχήμα 5.6.

### 5.2.1 Συμπληρωματικές λεπτομέρειες

Για όλα τα views της εφαρμογής είναι υπεύθυνη μία οντότητα που λέγεται view-manager.js και βρίσκεται στον φάκελο AuxiliaryScripts.

Σε συνεργασία με το αρχείο index.html, θέτουν την υποδομή για όλες τις άλλες λειτουργίες της εφαρμογής.

Το αρχείο index.html περιέχει όλα τα γραφικά στοιχεία της αρχικής διεπαφής χρήστη καθώς και κάποια ενσωματωμένα script που επιτρέπουν τη μετατροπή των URL που αναφέρθηκαν στην προηγούμενη υποενότητα σε οδηγίες που μπορεί να καταλάβει ο view-manager.

Μέσα σε κάθε view υπάρχουν δυο εσωτερικές μεταβλητές η documentTitle και η urlString.

Με τη χρήση της documentTitle ο view-manager αλλάζει τον τίτλο της καρτέλας σε αυτόν που ορίζει το τρέχον view και με το urlString αλλάζει το URL αντίστοιχα.

Υπάρχουν κάποια views που έχουν πιο σύνθετες παραμέτρους στο URL που συνθέτουν για να μπορούν να προσφέρουν πιο στοχευμένη στον χρήστη (πχ με τη χρήση ενός εξειδικευμένου URL μπορούμε να ανοίξουμε ένα συγκεκριμένο έγγραφο και να πλοηγηθούμε σε μια συγκεκριμένη σελίδα εντός αυτού το εγγράφου).

Εσωτερικά του view-manager υπάρχει μια λίστα με εγγραφές που αντιστοιχούν τα διάφορα αρχεία που υλοποιούν τα views με τα αντίστοιχα ονόματα που χρησιμοποιούνται εσωτερικά στην εφαρμογή.

Επιπρόσθετα είναι αυτός που κάνει χρήση όλων των κουμπιών που βρίσκονται στο αρχείο index.html και τους προσθέτει όλη τη λειτουργικότητα που αναφέρθηκε.

Μέσα του υπάρχουν διάφορες συναρτήσεις που καθιστούν δυνατή την υλοποίηση όλων των προαναφερθέντων λειτουργιών, καθώς και την παραμετροποίηση του ίδιου του view-manager (όπως τη δυνατότητα απενεργοποίησης/ενεργοποίησης του όποτε αυτό κριθεί απαραίτητο από κάποιο άλλο σύστημα) όμως μια εξ αυτών αξίζει μια ελαφρός βαθύτερη ματιά.

Η εν λόγω συνάρτηση ονομάζεται `changeView(viewName,...args)` και είναι αυτή που καλείται κάθε φορά που ξεκινάει η διαδικασία εμφάνισης ενός view, καθώς και η μεταφορά από ένα view σε κάποιο άλλο.

Μπορεί να θεωρηθεί ως “βοηθητική” καθώς η ίδια απλά αποφασίζει με πια σειρά για γίνει το κάλεσμα άλλων εσωτερικών συναρτήσεων αλλά πρακτικά είναι η μόνη που γίνεται διαθέσιμη από τον view-manager για την αλλαγή των view από άλλα συστήματα.

Με τη χρήση αυτής της συνάρτησης μας δίνεται η δυνατότητα να πλοηγηθούμε προγραμματιστικά σε όποιο view επιθυμούμε ανεξάρτητα από πιο σημείο της εφαρμογής καλείται.

Έτσι μπορούμε να έχουμε άμεση συσχέτιση των διάφορων view μεταξύ τους (πχ μπορούμε να πάμε από μια άσκηση σε ένα έγγραφο ύλης και από το έγγραφο ύλης σε ένα κουίζ)

Η πρώτη παράμετρος της είναι το εσωτερικά δηλωμένο όνομα του view που επιθυμούμε να δημιουργήσουμε ή να φέρουμε στο προσκήνιο εάν είναι απλά κριμένο.

Η δεύτερη παράμετρος από την άλλη είναι μεταβλητού μεγέθους και μπορεί να μην έχει καν τιμή.

Κάποια views βρίσκονται σε θέση να επεξεργαστούν παραμέτρους πριν την τελική εμφάνιση τους, αυτοί οι παράμετροι είναι που δηλώνονται στη δεύτερη παράμετρο σαν μια λίστα από τιμές.

Έπειτα γίνονται διάφοροι έλεγχοι (πχ για το αν είναι καν ενεργοποιημένος ο view-manager, και σε περίπτωση που δεν είναι ενεργοποιημένος σταματά η περαιτέρω διαδικασία εμφάνισης ενός view) για το εάν είναι το view ήδη φορτωμένο και απλά πρέπει να έρθει στο προσκήνιο καθώς και εάν το view που προσπαθούμε να δημιουργήσουμε υπάρχει, και εάν δεν υπάρχει να εμφανιστεί το ανάλογο σφάλμα με τη χρήση ενός toast.

Εφόσον στεφθούν με επιτυχία όλοι οι έλεγχοι τότε λαμβάνει μέρος η αίτηση του view από τον εξυπηρετητή και μόλις γίνει η λήψη του εμφανίζεται στην οθόνη.

Το γεγονός ότι στην εφαρμογή φορτώνονται τα views μόνο όταν το επιθυμεί ο χρήστης κάνει τους αρχικούς χρόνους φόρτωσης της εφαρμογής πολύ μικρότερους σε σχέση με αυτούς που θα ήταν εάν φορτώναμε όλα τα views κάθε φορά που ξεκινούσε η εφαρμογή.

### 5.3 Σύστημα βοήθειας

Επειδή η εφαρμογή απαρτίζεται από πολλά views που το κάθε ένα έχει διαφορετικούς κανόνες και τρόπους λειτουργίας, μπορεί να γίνει δύσκολη η απομνημόνευση των διάφορων λειτουργιών αλλά και η πλήρη αξιοποίηση τους χωρίς λίγη βοήθεια.

## Κεφάλαιο 5ο:

Για αυτόν τον λόγο έχει γίνει η υλοποίηση ενός σχετικά απλού αλλά εξαιρετικά ευέλικτου συστήματος παροχής βοήθειας.

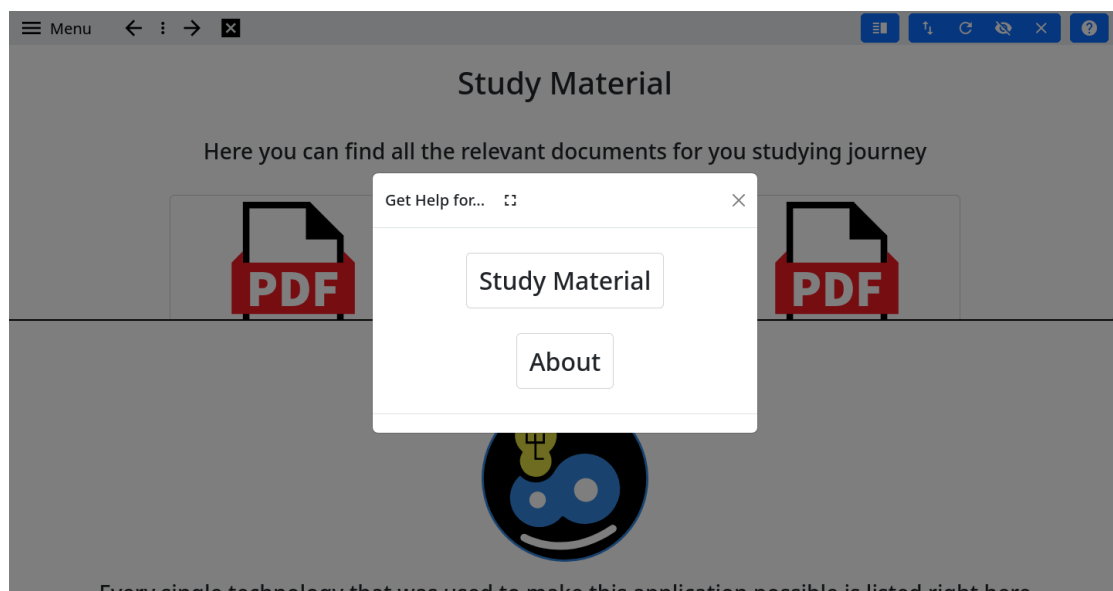
Με το πάτημα του δεξιού κουμπιού που φαίνεται στο Σχήμα 5.5, ανοίγει ένα modal το οποίο παρέχει βοήθεια στον χρήστη ανεξέταστα σε πιο view είναι.

Το κουμπί αυτό είναι πάντα διαθέσιμο και ανάλογα με το πιο view είναι στο προσκήνιο εμφανίζει αυτόματα την ανάλογη βοήθεια.

Ο λόγος που σχεδιάστηκε έτσι είναι για να παρέχει μια αίσθηση ασφάλειας και σιγουριάς, διότι ανάρτητα σε πιο view βρίσκεται ο χρήστης, ξέρει ότι πάντα θα είναι διαθέσιμο το κουμπί βοήθειας για να λύσει τυχόν απορίες που ενδέχεται να έχει.

Επίσης, το modal παρέχει τη δυνατότητα της μεγιστοποίησης του, στην περίπτωση που υπάρχει σημαντικός όγκος περιεχομένου προς εμφάνιση.

Σε περίπτωση που είναι ενεργός ο διαχωρισμός οθόνης τότε εμφανίζεται ένα modal που δίνει στον χρήστη τη δυνατότητα να επιλέξει για πιο από τα δύο view θα ήθελε να λάβει βοήθεια, όπως φαίνεται στο Σχήμα 5.9



Σχήμα 5.9: Εμφάνιση modal επιλογής view για τη λήψη βοήθειας

Επίσης, το modal αυτό αλλάζει τη σειρά και τη στοίχιση των διαθέσιμων επιλογών ανάλογα με το εάν ο διαχωρισμός της οθόνης είναι οριζόντιος ή κατακόρυφος ή εάν έχει λάβει μέρος ανταλλαγή των views από δεξιά προς τα αριστερά ή από πάνω προς τα κάτω.

### 5.3.1 Συμπληρωματικές λεπτομέρειες

Το συγκεκριμένο σύστημα υλοποιείται μέσα στον φάκελο HelpManager. Μέσα τον φάκελο HelpContentFiles, υπάρχουν δύο υποφακέλοι, ExerciseHelpFiles και ViewHelpFiles καθώς και ένα αρχείο που λέγεται help\_definitions.json.

Μέσα στο αρχείο `help_definitions.json` (Σχήμα 5.10), υπάρχουν όλες οι εγγραφές που αντιστοιχούν ένα view με ένα αρχείο `htm` (και το όνομα που εμφανίζεται στο modal επιλογής βοήθειας όταν υπάρχουν δύο views ανοιχτά) που υπάρχει μέσα στους προαναφερθέντες φακέλους.

```
"pdf": {
  "fileName": "pdf.htm",
  "title": "Pdf Viewer"
},
"exercise": {
  "title": "Exercise"
},
"quiz": {
  "fileName": "quiz.htm",
  "title": "Quiz"
},
"quizReview": {
  "fileName": "quiz_review.htm",
  "title": "Quiz Review"
},
},
```

Σχήμα 5.10: Τμήμα του αρχείου `help_definitions.json`

Εσωτερικά ο `help-manager` γνωρίζει ανάλογα με το τι είδος view είναι να αντιστοιχίσει και το κατάλληλο αρχείο `htm`.

Ο λόγος που έχει λάβει μέρος ο διαχωρισμός των αρχείων σε δύο φακέλους είναι γιατί οι ασκήσεις της εφαρμογής, οι οποίες θα παρουσιαστούν στην υποενότητα 5.5, χρειάζονται εξατομικευμένα αρχεία βοήθειας, καθώς μπορούν να έχουν τεράστιες αλλαγές ή μία από την άλλη ως προς τις λειτουργίες τους.

Αυτό όμως δεν ισχύει για τα άλλα views, καθώς αυτά έχουν πολύ συγκεκριμένες λειτουργίες και επιλογές που είναι ανεξάρτητες με το περιεχόμενο που διαχειρίζονται (στο Σχήμα 5.10 η εγγραφή `exercise` δεν έχει αντιστοιχία με αρχείο).

Επιλέχθηκαν τα αρχεία τύπου `htm` (είναι αρχεία HTML απλά έχουν άλλο όνομα, για να επιτευχθεί η διάκριση τους από τα άλλα αρχεία HTML της εφαρμογής), γιατί είναι εξαιρετικά ευέλικτα, καθώς επιτρέπουν την εισαγωγή πληθώρας πολυμέσων (βίντεο, εικόνες, ήχους, άλλες ιστοσελίδες κτλ) και την αντιγραφή κομματιών της διεπαφής.

Τα αρχεία έχουν την εξής δομή.

Το κάθε ένα απαρτίζεται από δύο ή τρεις ενότητες (ανάλογα με την προτίμηση του συγγραφέα) και ο διαχωρισμός γίνεται με τη χρήση αυτής της σειράς χαρακτήρων [~~~~~] (Σχήμα 5.11).

Η πρώτη ενότητα αντιπροσωπεύει τον τίτλο του modal, η δεύτερη αντιπροσωπεύει το κυρίως περιεχόμενο και η τρίτη το υποσέλιδο του modal.

## Κεφάλαιο 5ο:

```
1  <h3>About</h3>
2  ~~~~~
3  <h4>What's there to talk about?!</h4>
4  <p>
5    Here you can find every singe technology that was used to make this
6    application come to life.
7  </p>
8  <p>
9    Feel free to explore all of them as each and every one played a very important
10   role in this application, ether big or small.
11 </p>
12 <p>
13   You can click on the
14   <span
15     ><button
16       class="website_button d-inline-flex align-items-center btn btn-primary
17       btn-lg px-4 rounded-pill m-2"
18       type="button"
19       >
20       Go to website <span class="material-icons ps-2">link</span>
21     </button></span>
22   and learn more about each one or click on the
23   <span
24     ><button
25       class="license_button d-inline-flex align-items-center btn
26       btn-outline-secondary btn-lg px-4 rounded-pill m-2"
27       type="button"
28       >
29       Open License <span class="material-icons ps-2">feed</span>
30     </button></span>
31   to view the appropriate license.
32 </p>
```

Σχήμα 5.11: Παράδειγμα αρχείου βοήθειας

Έτσι δίνονται πολλές επιλογές στον συγγραφέα αυτών των αρχείων για το πως καλύτερα να προσφέρει βοήθεια στους χρήστες της εφαρμογής (πχ μπορεί να περιγράψει τη λειτουργία ενός κουμπιού και μετά να συμπεριλάβει το ίδιο το κουμπί μέσα στην περιγραφή, δημιουργώντας έτσι μια άμεση σχέση ανάμεσα στη λειτουργία και στο στοιχείο της διεπαφής που την υλοποιεί).

Λόγο του δυναμικού χαρακτήρα αυτών των αρχείων θα μπορούσαν να χρησιμοποιηθούν και για άλλους λόγους.

Θα μπορούσε να γίνει παράθεση υπερσυνδέσμων σε ιστοσελίδες που επεξηγούν μια έννοια σε κάποια άσκηση.

Μια άλλη χρήση τους θα μπορούσε να είναι η καταγραφή ερωτήσεων από τους μαθητές στην παράδοση του μαθήματος και η μεταγενέστερη προσθήκη τους στην εφαρμογή με την κατάλληλη εξήγηση από τον καθηγητή.

Έτσι, σταδιακά θα μπορούσε να αναπτυχθεί μια είδους βάση δεδομένων ερωτοαπαντήσεων που θα είχαν πρόσβαση οι χρήστες της εφαρμογής μέσω των βοηθειών.

## 5.4 Έγγραφα ύλης και οι συμπληρωματικές τους λειτουργίες

### 5.4.1 Εισαγωγή

Στην παρούσα υποενότητα γίνεται η παρουσίαση των views και των συστημάτων που διαχειρίζονται την οργάνωση, προβολή και επέκταση των εγγράφων ύλης.

Λόγο της αλληλοσύνδεσης των συστημάτων, σε κάποια σημεία του κειμένου γίνεται η αναφορά σε ήδη αναφερθέντα συστήματα, με στόχο την πλήρη επεξήγηση τους με τις νέες πληροφορίες που θα αναφερθούν στην εκάστοτε υποενότητα.

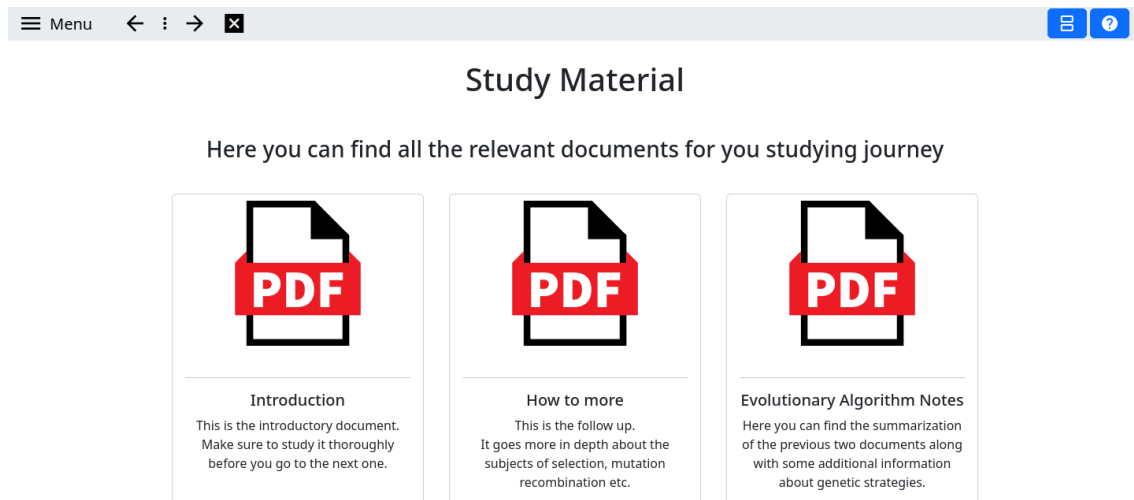
### 5.4.2 View Υλικού Μελέτης (Study Material)

Η παράδοση της ύλης στην εφαρμογή γίνεται με τη χρήση αρχείων PDF.

Στο συγκεκριμένο view γίνεται η παράθεση τίτλου και σύντομης περιγραφής για το κάθε ένα από τα εν λόγω έγγραφα με τη χρήση μιας κάρτας (Σχήμα 5.12).

Ανάλογα με τον αριθμό των εγγράφων το view προσαρμόζει τον τρόπο παρουσίασης του έτσι ώστε να κάνει όσο το δυνατόν καλύτερη χρήση του διαθέσιμου χώρου.

Κάνοντας κλικ πάνω σε μία κάρτα, ο χρήστης μεταφέρεται σε ένα view που προβάλλει το έγγραφο για ανάγνωση.



Σχήμα 5.12: Προβολή του view που εμφανίζει τα αρχεία ύλης

#### 5.4.2.1 Συμπληρωματικές λεπτομέρειες

Το συγκεκριμένο view υλοποιείται στο αρχείο pdf-view.js.

Εκεί είναι που λαμβάνει μέρος η δημιουργία των καρτών που αναφέρθηκαν προηγουμένως.

Ο τρόπος του γνωρίζει πια είναι τα διαθέσιμα έγγραφα καθώς και τους τίτλους και τις περιγραφές τους είναι ο εξής.

Κεφάλαιο 5ο:

Μέσα στον φάκελο PdfContentFiles υπάρχει ένας υποφάκελλος με όνομα PdfFiles και ένα αρχείο ονόματος pdf\_definitions.json.

Μέσα στο αρχείο αυτό υπάρχουν εγγραφές που έχουν τη μορφή που φαίνεται στο Σχήμα 5.13

```
{
  "_1": {
    "selectorName": "Introduction",
    "fileName": "EA_Introduction.pdf",
    "description": "This is the introductory document.\nMake sure to study it thoroughly before you go to the next one."
  },
  "_2": {
    "selectorName": "How to more",
    "fileName": "EA_Howto_more.pdf",
    "description": "This is the follow up.\nIt goes more in depth about the subjects of selection, mutation recombination etc."
  },
  "_3": {
    "selectorName": "Evolutionary Algorithm Notes",
    "fileName": "EA_notes.pdf",
    "description": "Here you can find the summarization of the previous two documents along with some additional information about genetic strategies."
  }
}
```

Σχήμα 5.13: Τμήμα του αρχείου pdf\_definitions.json

Τα πεδία με τον χαρακτήρα “\_” ακολουθούμενα από έναν αριθμό αποτελούν τους τα id των εγγράφων. Κάθε ένα πρέπει να είναι μοναδικό και να αποτελείται μόνο από ένα “\_” και ένα νούμερο αορίστου μεγέθους.

Εάν γίνει αλλαγή του id χωρίς να αλλάξουν τα εσωτερικά πεδία, τότε θεωρείται για την εφαρμογή ως καινούριο έγγραφο, παρόλο που έχει τα ίδια χαρακτηριστικά.

Δεν είναι ανάγκη τα νούμερα να είναι σε αύξουσα σειρά, ούτε το ένα να είναι το επόμενο του άλλου.

Επίσης, οι εγγραφές μπορούν να εμφανίζονται με οποιαδήποτε σειρά επιθυμεί ο συγγραφέας του αρχείου.

Εντός της κάθε εγγραφής υπάρχουν τρία πεδία :

- selectorName : αποτελεί τον τίτλο του εγγράφου που εμφανίζεται στην κάρτα.
- fileName : αποτελεί το όνομα ενός αρχείου που βρίσκεται στον φάκελλο PdfFiles. Εκεί είναι που βρίσκονται τα αρχεία PDF και με την παροχή του ονόματος τους μπορεί να γίνει η λογική συσχέτιση τους με το id και τις υπόλοιπες πληροφορίες που αναγράφονται για αυτά (τίτλος, περιγραφή).
- description : παρέχει την περιγραφή του κάθε εγγράφου που εμφανίζεται στην κάρτα.

Η εφαρμογή έχει τη δυνατότητα να ενημερώνει τον χρήστη όταν παρατηρείται κάποια αλλαγή στο αρχείο αυτό.

Ο τρόπος που το κάνει αυτό είναι με τη χρήση της βιβλιοθήκης Socket.io.

Με τη χρήση αυτής της βιβλιοθήκης μπορούμε να έχουμε μια αμφίδρομη επικοινωνία με τον εξυπηρετητή, δηλαδή μπορούμε να στείλουμε αλλά και να δεχτούμε μηνύματα από αυτόν.

Ο εξυπηρετητής παρακολουθεί το αρχείο για τυχόν αλλαγές, ενώ παράλληλα έχει ανοιχτό και ένα socket (ένα κανάλι στο οποίο μπορεί να συνδεθεί ένας χρήστης για την εδραίωση της αμφίδρομης επικοινωνίας), πάνω στο οποίο συνδέονται όλοι οι ενεργοί χρήστες (χρησιμοποιούν την εφαρμογή τη δεδομένη στιγμή).

Μόλις ο εξυπηρετητής ανιχνεύσει μια αλλαγή στέλνει σε όλους του συνδεδεμένους χρήστες ένα “μήνυμα πρόθεσης” (intent).

Μέσα στον φάκελο AuxiliaryScripts υπάρχει ένα αρχείο socket-definitions.js.

Εκεί είναι που γίνεται η διεκπεραίωση του συγκεκριμένου intent καθώς και όλων των άλλων συστημάτων που κάνουν χρήση αυτού του μηχανισμού.

Στο Σχήμα 5.14 φαίνεται ένα απόσπασμα αυτού του αρχείου.

```
let socket;
const URL = "http://localhost:3000";

socket = io.connect(URL, { autoConnect: true, reconnection: false });

socket.on("pdfDefinitionsChange", async (_) => {
  console.log("Pdf Definitions changed");
  await import("../PdfManager/pdf-state-manager.js").then((module) => {
    module.updatePdfFileDefinitions();
  });
});
```

Σχήμα 5.14: Τμήμα του αρχείου socket-definitions.js

Αυτό που κάνει είναι να εγκαθιδρύει την επικοινωνία με τον εξυπηρετητή καθώς και να ορίσει ένα χειριστή για το intent.

Μόλις η εφαρμογή λάβει το intent από τον εξυπηρετητή τότε εκτελεί τις απαραίτητες ενέργειες για πραγματοποιηθεί η ανανέωση όλων των αποθηκευμένων εγγράφων καθώς και ο επανασχεδιασμός των καρτών του view, έτσι ο χρήστης έχει άμεση πρόσβαση στις πιο ενημερωμένες εκδόσεις των εγγράφων.

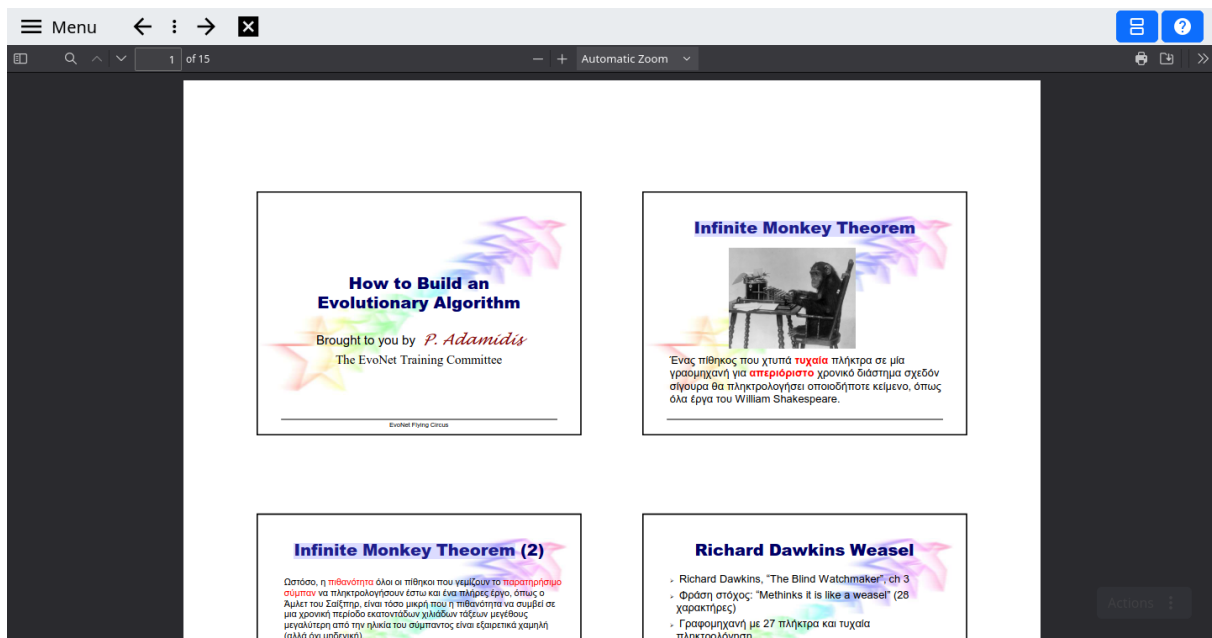
Όλη αυτή η διαδικασία λήψης μηνυμάτων από τον εξυπηρετητή και η εκτέλεση μιας σειράς ενεργειών στην εφαρμογή χρησιμοποιείται και σε άλλα κομμάτια της εφαρμογής, οπότε δε γίνεται περαιτέρω αναφορά στο πως δουλεύουν στο μέλλον.

Κεφάλαιο 5ο:

Ότι έχει να κάνει με τη χρήση των socket στην εφαρμογή είναι ορισμένο στο αρχείο socket-definitions.js.

### 5.4.3 View Εμφάνιση Εγγράφου

Μόλις ο χρήστης κάνει κλικ σε μια κάρτα τότε θα μεταφερθεί σε ένα view που μοιάζει με αυτό στο Σχήμα 5.15



Σχήμα 5.15: Προβολή ενός εγγράφου ύλης

Εδώ μπορεί να διαβάσει το επιλεγμένο έγγραφο καθώς και να πραγματοποιήσει “ενέργειες”.

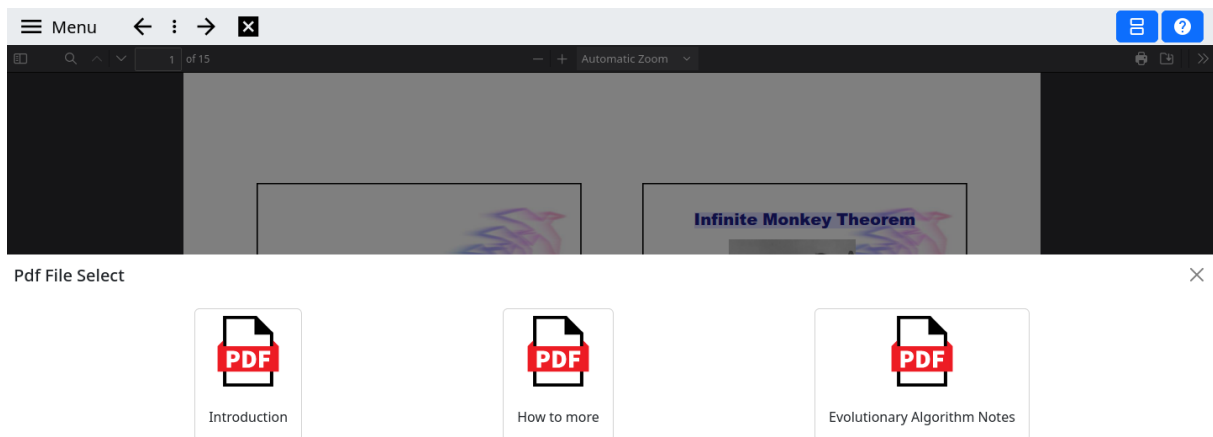
Εκτός από τις δεδομένες λειτουργίες που προσφέρει ένα συμβατικό πρόγραμμα προβολής αρχείων PDF, το συγκεκριμένο πρόγραμμα προβολής που χρησιμοποιείται στην εφαρμογή (PDF.js) έχει προσauξηθεί με επιπλέον λειτουργίες, με τη μορφή “ενεργειών” που μπορούν να προσπελαστούν κάνοντας κλικ στο κάτω δεξιό κουμπί με το όνομα “Actions”.

Μετά το πάτημα αυτού του κουμπιού θα εμφανιστεί το μενού επιλογών που φαίνεται στο Σχήμα 5.17. Οι επιλογές θα εξηγηθούν με τη σειρά που εμφανίζονται, ξεκινώντας από επάνω προς τα κάτω.

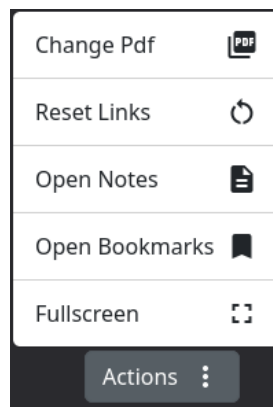
#### 5.4.3.1 Αλλαγή εγγράφου

Η πρώτη επιλογή δίνει τη δυνατότητα στον χρήστη να αλλάξει το έγγραφο που παρουσιάζεται χωρίς να πρέπει να πάει στο προηγούμενο view επιλογής εγγράφου.

Κάνοντας κλικ πάνω στο “Change Pdf” θα εμφανιστεί το ακόλουθο μενού (Σχήμα 5.16), που έχει ακριβώς τις ίδιες επιλογές που έχει το view επιλογής εγγράφου (χωρίς τις έξτρα πληροφορίες όπως την περιγραφή), και είναι πάντα συγχρονισμένο με την κατάσταση των εγγράφων της εφαρμογής.



Σχήμα 5.16: Μενού επιλογής εγγράφου



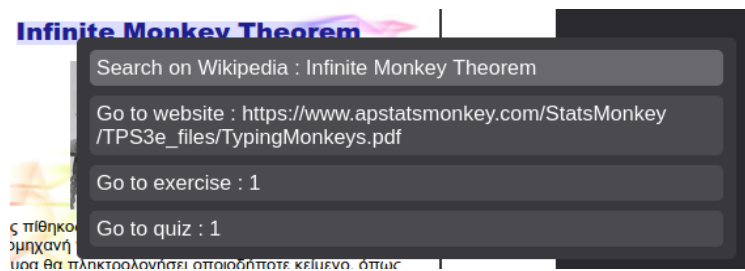
Σχήμα 5.17: Μενού επιλογών εγγράφου

Κάνοντας κλικ πάνω σε μια από τις διαθέσιμες επιλογές, θα πραγματοποιηθεί η αλλαγή του εγγράφου χωρίς να αλλάξει το τρέχον view.

#### 5.4.3.2 Σύνδεσμοι εγγράφου

Στο Σχήμα 5.15 μπορεί κανείς να διακρίνει ότι ο τίτλος “Infinite Monkey Theorem” είναι περιμετρικά χρωματισμένος με ένα γαλάζιο χρώμα. Αυτό δείχνει την ύπαρξη ενός συνδέσμου (link), που είναι εξατομικευμένος πάνω στην ίδια την εφαρμογή και παρέχει τη δυνατότητα πλοήγησης του χρήστη σε άλλα μέρη της καθώς και στο άνοιγμα νέων παραθύρων που έχουν σχέση με το περιεχόμενο του επιλεγμένου κειμένου που έχει το link. Κάνοντας κλικ σε οποιαδήποτε σημείο εντός του link, θα ανοίξει το μενού που φαίνεται στο Σχήμα 5.18

## Κεφάλαιο 5ο:



Σχήμα 5.18: Μενού επιλογών ενός link

Τα link της εφαρμογής παρέχουν τις εξής λειτουργίες :

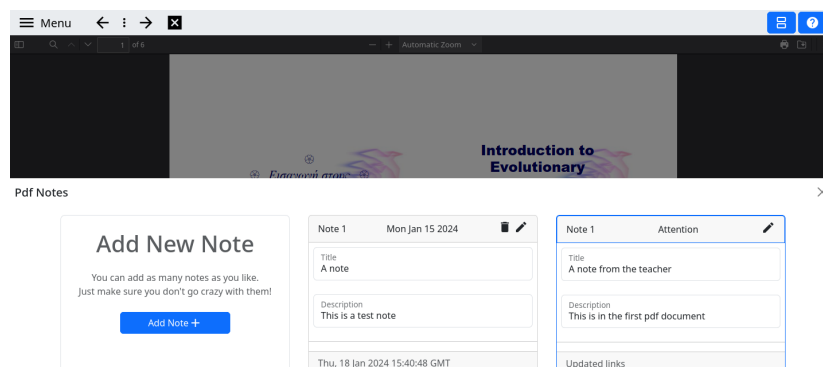
- Τη δυνατότητα ανοίγματος σε υπάρχον λίκμα στην ιστοσελίδα της Βικιπέδιας.
- Τη δυνατότητα ανοίγματος εξωτερικής ιστοσελίδας.
- Μεταφορά σε μια υπάρχουσα άσκηση εντός της εφαρμογής.
- Εκκίνηση υπάρχον κουίζ εντός της εφαρμογής.

Σε περίπτωση που υπάρχει λανθασμένη εγγραφή στον ορισμό του link (5.4.4.2), τότε ενημερώνεται ο χρήστης με τη χρήση ενός toast. Λόγο κάποιων τεχνικών δυσκολιών με την ενσωμάτωση του προγράμματος PDF.js, έπρεπε να δοθεί στον χρήστη η δυνατότητα της επαναφόρτωσης των link ενός εγγράφου. Αυτή η επαναφόρτωση μπορεί να γίνει κάνοντας κλικ στο “Reset Links” του μενού που εμφανίζεται στο Σχήμα 5.17.

### 5.4.3.3 Σημειώσεις εγγράφου

Η εφαρμογή προσφέρει τη δυνατότητα στον χρήστη να προσθέτει σημειώσεις σε ένα έγγραφο ύλης.

Το πρώτο βήμα για τη δημιουργία μίας σημείωσης είναι το άνοιγμα του μενού δημιουργίας σημειώσεων, που μπορεί να γίνει με το πάτημα του κουμπιού “Open Notes” του μενού που εμφανίζεται στο Σχήμα 5.17. Μόλις λάβει μέρος αυτή η ενέργεια, θα ανοίξει το μενού που φαίνεται στο Σχήμα 5.19.



Σχήμα 5.19: Μενού δημιουργίας σημειώσεων

Εδώ ο χρήστης κάνοντας κλικ στο μπλε κουμπί με όνομα “Add Note +”, μπορεί να προσθέσει μια καινούρια σημείωση που θα έχει τη μορφή αυτών στο Σχήμα 5.19.

Εκτός από τις σημειώσεις χρήστη υπάρχουν και οι σημειώσεις του καθηγητή, που έχουν ένα γαλάζιο περίβλημα για να είναι άμεσα διαχωρίσιμες με μια απλή ματιά.

Οι σημειώσεις του καθηγητή δεν μπορούν να διαγραφούν αλλά και να τροποποιηθούν, όμως η εφαρμογή προσφέρει τη δυνατότητα στον χρήστη να τις κρύψει αν αυτός το επιθυμεί με τη χρήση μιας ρύθμισης (η ρύθμιση αυτή βρίσκεται στην υποενότητα 5.7.4).

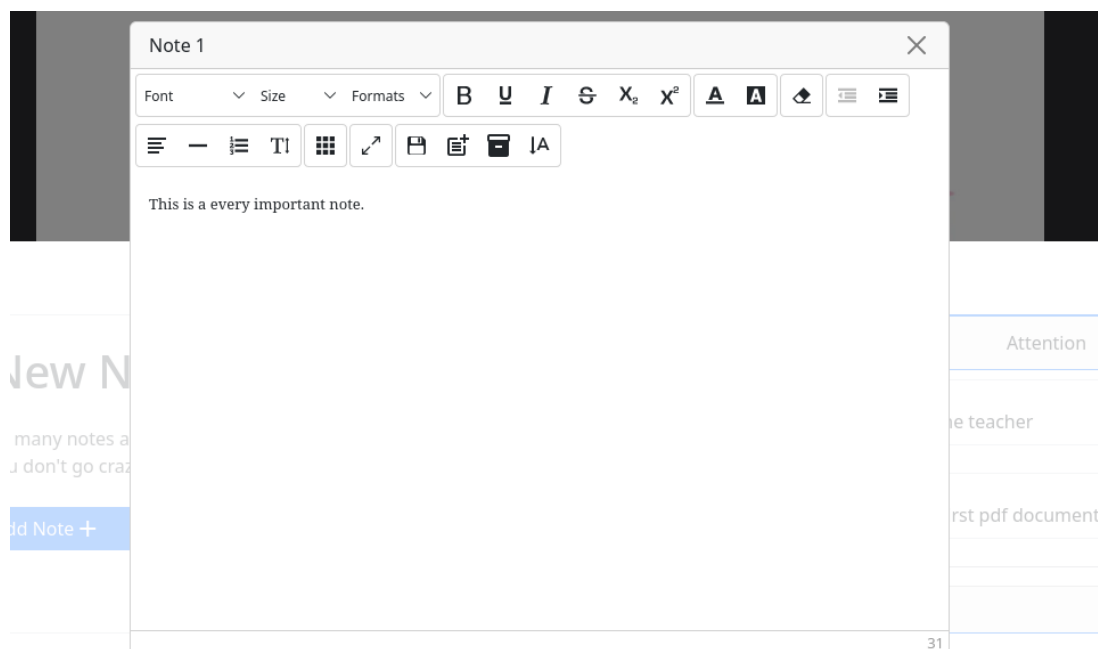
Όλες οι σημειώσεις αναγράφουν πάνω τους την ημερομηνία δημιουργίας τους, καθώς και την ημερομηνία της τελευταίας τροποποίησής τους, κάνοντας εύκολη την εύρεση της πιο πρόσφατης σημείωσης.

Εκτός αυτού έχουν δύο επεξεργάσιμα πεδία που ορίζουν τον τίτλο της σημείωσης καθώς και μια σύντομη περιγραφή της.

Οποιαδήποτε τροποποίηση σε ένα από αυτά τα πεδία αποθηκεύεται αυτόματα από την εφαρμογή.

Συμπληρωματικά μια σημείωση μπορεί να διαγραφεί πατώντας το εικονίδιο που παρουσιάζει ένα κάδο στην πάνω δεξιά γωνία μιας σημείωσης.

Η προβολή και η τροποποίηση μιας σημείωσης αποτελεί την ίδια ενέργεια στην εφαρμογή και μπορεί να πραγματοποιηθεί πατώντας το εικονίδιο που παρουσιάζει ένα μολύβι στην πάνω δεξιά γωνία μιας σημείωσης.



Σχήμα 5.20: Εμφάνιση view προβολής | επεξεργασίας σημείωσης

Εκτελώντας αυτή την ενέργεια ο χρήστης μεταφέρεται στο view που παρουσιάζεται στο Σχήμα 5.20.

Σε αυτό το view γίνεται χρήση του προγράμματος Suneditor.js (στη συνέχεια αποκαλείται απλά editor) για την παροχή των λειτουργιών επεξεργασίας κειμένου.

## Κεφάλαιο 5ο:

Το πρόγραμμα αυτό παρέχει πολλές επιλογές μορφοποίησης κειμένου η οποίες είναι παρόμοιες με αυτές που προσφέρονται από όλα τα συμβατικά προγράμματα τέτοιου τύπου, οπότε δε γίνεται διεξοδική ανάλυση τους.

Ένα γκρουπ επιλογών που αξίζουν μια περαιτέρω ανάλυση είναι αυτές που έχουν προστεθεί σε αυτό το πρόγραμμα για την επίτευξη μιας πιο αρμονικής συνεργασίας με την εφαρμογή.

Το γκρουπ αυτών των επιλογών φαίνεται στο Σχήμα 5.21



*Σχήμα 5.21: Γκρουπ προσαρμοσμένων λειτουργιών*

Η πρώτη επιλογή (η σειρά ξεκινά από αριστερά προς τα δεξιά) αφορά την αποθήκευση των αλλαγών που έχει υποστεί μια σημείωση.

Εάν ο χρήστης δεν κάνει αποθήκευση των αλλαγών, μόλις κλείσει το view αυτές θα χαθούν.

Αν για τον χρήστη αυτή η λειτουργία είναι ανεπιθύμητη τότε μπορεί να ενεργοποιήσει την αυτόματη αποθήκευση των αλλαγών κάνοντας κλικ στην τέταρτη επιλογή.

Μόλις συμβεί αυτό το εικονίδιο γίνεται γαλάζιο, που σηματοδοτεί ότι η αυτόματη αποθήκευση είναι ενεργή. Τώρα όταν συμβαίνει μια αλλαγή δεν είναι απαραίτητη η χειροκίνητη αποθήκευση.

Πρέπει να σημειωθεί ότι η συγκεκριμένη λειτουργία παραμένει ενεργεί όσο δημιουργούμε, διαγράφουμε και επεξεργαζόμαστε σημειώσεις.

Μόλις πάει να ισχύει αυτό, τότε η ρύθμιση επανέρχεται στην αρχική της κατάσταση, δηλαδή παύει να είναι ενεργή η αυτόματη αποθήκευση.

Σε περίπτωση που ο χρήστης θέλει να έχει συνέχεια αυτή τη λειτουργία τότε μπορεί να την ενεργοποιήσει στις ρυθμίσεις. Η δεύτερη επιλογή επιτρέπει την προσθήκη εξωτερικού αντιγραμμένου κειμένου.

Μόλις γίνει κλικ πάνω της, τότε θα εμφανιστεί ένα modal μέσα στο οποίο μπορούμε να εναποθέσουμε το αντιγραμμένο κείμενο και έπειτα πατώντας το κουμπί “Import” να το ενσωματώσουμε μέσα στον editor.

Από την άλλη πλευρά η τρίτη επιλογή κάνει ακριβώς το αντίθετο, δηλαδή αντιγράφει όλο το διαθέσιμο κείμενο της σημείωσης καθώς και τη μορφοποίηση του στο clipboard του υπολογιστή.

Έτσι μπορεί να γίνει η αποθήκευση του κειμένου σε ένα αρχείο html (έτσι ώστε να έχει τη σωστή μορφοποίηση).

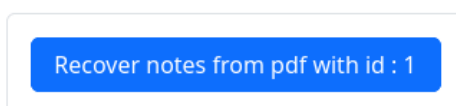
Σε συνδυασμό με τη δεύτερη επιλογή προσθήκης κειμένου μπορεί να υπάρξει ένας τρόπος διαμοιρασμού των σημειώσεων ανάμεσα στους μαθητές, με την απλή μεταφορά αυτού του αντιγραμμένου κειμένου (πχ με τη χρήση email ή μιας υπηρεσίας κοινωνικής δικτύωσης).

Τέλος, σε περίπτωση όπου αναπάντεχα αφαιρεθεί ένα έγγραφο ύλης ενώ ο χρήστης είχε αποθηκευμένες σημειώσεις μέσα σε αυτό τότε η εφαρμογή του δίνει τρόπο να τις ανακτήσει.

Μόλις γίνει αυτό το view που φαίνεται στο Σχήμα 5.12 θα έχει και μία καινούρια κάρτα που θα μοιάζει με αυτή στο Σχήμα 5.22.

Κάνοντας κλικ πάνω σε μία από τις επιλογές που ενδέχεται να έχει (εάν έχουν αφαιρεθεί περισσότερα από ένα έγγραφα) θα μεταφερθεί σε ένα view το οποίο θα μοιάζει με αυτό στο Σχήμα 5.23.

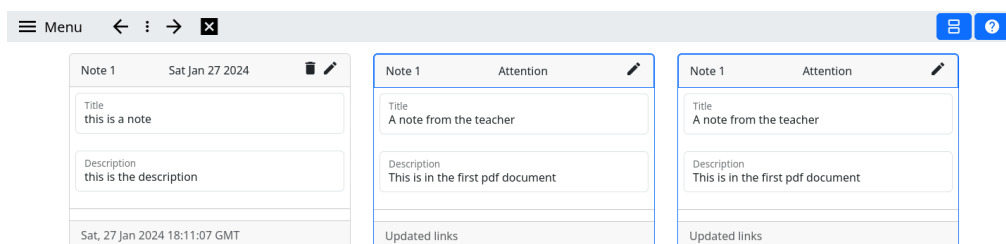
Από αυτό το view μπορεί να αντιγράψει όλες τις σημειώσεις του σε καινούρια αρχεία η ακόμα και να τις προσθέσει σε κάποιο άλλο έγγραφο ύλης.



*Σχήμα 5.22: Εμφάνιση κάρτας ανάκτησης σημειώσεων*

Μπορεί επίσης να τις αφήσει εδώ για όσο χρόνο θέλει χωρίς αυτές να διαγραφτούν.

Σε περίπτωση που ο χρήστης προβεί σε διαγραφή όλων των σημειώσεων ή γίνει η επαναφορά του αρχείου από τον καθηγητή τότε αυτό το view δε θα είναι πια προσβάσιμο γιατί δεν υπάρχει λόγος ύπαρξής του.



*Σχήμα 5.23: Εμφάνιση view ανάκτησης σημειώσεων*

#### 5.4.3.4 Σελιδοδείκτες εγγράφου

Μια εξίσου χρήσιμη λειτουργία που προσφέρει η εφαρμογή είναι η δημιουργία σελιδοδεικτών, για τη διευκόλυνση της πλοήγησης εγγράφων που έχουν πολλά σημεία ενδιαφέροντος.

Το πρώτο βήμα για τη δημιουργία ενός σελιδοδείκτη είναι το άνοιγμα του μενού δημιουργίας σελιδοδεικτών, που μπορεί να γίνει με το πάτημα του κουμπιού “Open Bookmarks” του μενού που εμφανίζεται στο Σχήμα 5.17.

Κεφάλαιο 5ο:

Εδώ ο χρήστης κάνοντας κλικ στο μπλε κουμπί με όνομα “Add Bookmark +”, μπορεί να προσθέσει έναν καινούριο σελιδοδείκτη που θα έχει τη μορφή αυτών στο Σχήμα 5.24.

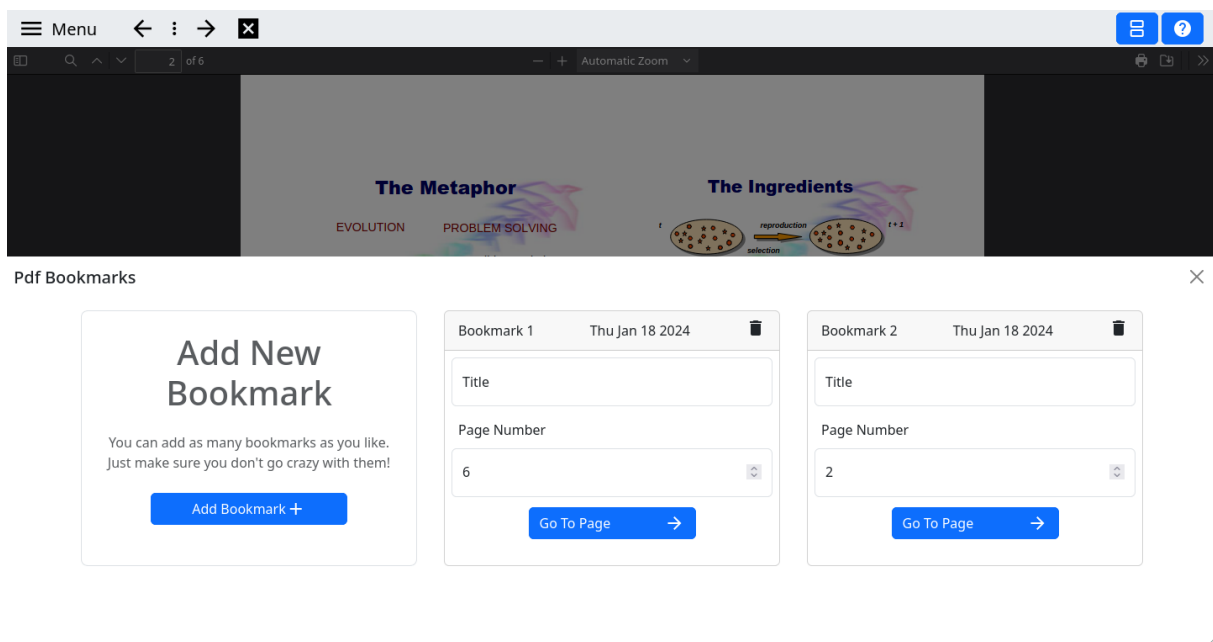
Όλοι οι σελιδοδείκτες αναγράφουν πάνω τους την ημερομηνία δημιουργίας τους.

Εκτός αυτού έχουν δύο επεξεργάσιμα πεδία που ορίζουν τον τίτλο του σελιδοδείκτη καθώς και τη σελίδα πάνω στην οποία αυτός αναφέρεται.

Με την αρχική δημιουργία ενός σελιδοδείκτη αυτός έχει ως αριθμό σελίδας τον αριθμό της τρέχουσας σελίδας.

Οποιαδήποτε τροποποίηση σε ένα από αυτά τα πεδία αποθηκεύεται αυτόματα από την εφαρμογή.

Πατώντας πάνω στο κουμπί “Go to page”, τότε η εφαρμογή θα φέρει στο προσκήνιο τη σελίδα που αναγράφεται στον σελιδοδείκτη.



Σχήμα 5.24: Μενού δημιουργίας σελιδοδεικτών

Συμπληρωματικά ένας σελιδοδείκτης μπορεί να διαγραφεί πατώντας το εικονίδιο που παρουσιάζει ένα κάδο στην πάνω δεξιά γωνία του σελιδοδείκτη.

Εάν γίνει η αλλαγή της σελίδας σε μια η οποία δεν υπάρχει τότε ο χρήστης θα ειδοποιηθεί για αυτό το σφάλμα με ένα toast.

#### 5.4.3.5 Πλήρης οθόνη

Η τελευταία επιλογή του μενού που φαίνεται στο Σχήμα 5.17 είναι αυτή της πλήρης οθόνης.

Με τη χρήση αυτής της επιλογής μεγιστοποιεί το εν λόγω έγγραφο χωρίς να χρειάζεται να γίνει χρήση του πληκτρολογίου.

Μόλις ο χρήστης επιθυμήσει να τερματίσει την πλήρη οθόνη μπορεί να πατήσει το κουμπί με όνομα “Exit FullScreen”, που βρίσκεται στο κάτω δεξιό μέρος της οθόνης, για να επανέλθει στην κανονική προβολή εγγράφου.

#### 5.4.4 Συμπληρωματικές λεπτομέρειες

Ότι έχει σχέση με το σύστημα προβολής των αρχείων PDF βρίσκεται στον φάκελο PdfManager.

Πιο συγκεκριμένα το αρχείο pdf-state-manager.js είναι υπεύθυνο για την εύρεση των αρχείων PDF καθώς και των αρχείων link που συσχετίζονται με αυτά. Το αρχείο pdf-component.js (στο εξής αναφέρεται ως component) είναι υπεύθυνο για την προβολή των εγγράφων καθώς και την οργάνωση όλων των άλλων υποσυστημάτων που συνεργάζονται μεταξύ τους για να υλοποιήσουν τις λειτουργίες που αναφέρθηκαν.

Ουσιαστικά αυτό το view αποτελεί ένα περικάλυμμα για το πρόγραμμα PDF.js, που έχει ως σκοπό την επαύξηση των λειτουργιών του.

Οι επαυξήσεις αυτές έχουν τη μορφή των link που ενσωματώνονται μέσα στο αρχείο και των “Actions” που φαίνονται στο Σχήμα 5.17

##### 5.4.4.1 Αλλαγή εγγράφου

Ο μηχανισμός λειτουργίας του είναι σχεδόν ίδιος με αυτών που αναφέρθηκε στην υποενότητα 5.4.2.1, ως προς τον τρόπο οργάνωσης των εγγράφων και τη χρήση των id.

Απλά στο component πρέπει να γίνει η επαναφορά του τίτλου της καρτέλας και του URL χειροκίνητα καθώς και να ξεκινήσουν εκ νέου οι διαδικασίες παρακολούθησης της τρέχουσας σελίδας για λόγους που θα εξηγηθούν στις επόμενες υποενότητες.

##### 5.4.4.2 Σύνδεσμοι εγγράφου

Οι σύνδεσμοι δουλεύουν με τον ακόλουθο τρόπο.

Για κάθε ένα αρχείο PDF στον φάκελο PdfFiles μπορεί να υπάρχει και ένα αρχείο .json που έχει ακριβώς το ίδιο όνομα και διαφέρει μόνο στην επέκταση αρχείου (πχ αρχείο.pdf και το αντίστοιχο αρχείο.json)

```
{
  "1": [
    {
      "text": "Infinite Monkey Theorem",
      "occurrences": "all",
      "span_links": {
        "wikipedia": ["Infinite Monkey Theorem"],
        "url": ["https://www.apstatsmonkey.com/StatsMonkey/TPS3e_files/TypingMonkeys.pdf"],
        "exercise": [1],
        "quiz": [1]
      }
    }
  ]
}
```

Σχήμα 5.25: Εμφάνιση μιας εγγραφής link

Η ύπαρξη ενός τέτοιου αρχείου σηματοδοτεί στην εφαρμογή την ύπαρξη link για το έγγραφο στο οποίο αναφέρεται ο τίτλος του.

## Κεφάλαιο 5ο:

Ένα τέτοιο αρχείο αποτελείται από εγγραφές που φαίνονται στο Σχήμα 5.25

Μια εγγραφή ξεκινάει από τη σελίδα στην οποία αναφέρεται (στην προκειμένη περίπτωση είναι η πρώτη γιατί έχει το “1” ως κλειδί) και πρέπει να είναι μοναδική στο αρχείο, αλλά μπορούμε να έχουμε όσες σελίδες θέλουμε εφόσον αυτές έχουν ανταπόκριση και στο εκάστοτε έγγραφο.

Έπειτα ορίζεται ένας πίνακας μέσα στον οποίο μπορούν να υπάρχουν παραπάνω από μία εγγραφές για το τι κείμενο επιθυμούμε να μετατρέψουμε σε link.

Η δομή αυτών πεδίων των εσωτερικών εγγραφών είναι η εξής :

- text : περιγράφει το κείμενο που θέλουμε να μετατρέψουμε σε link (πρέπει να υπάρχει στη σελίδα που αναφερόμαστε αλλιώς θα προκύψει σφάλμα)
- occurrences (μπορεί να είναι αριθμός ή all) : εάν το κείμενο που έχουμε διαλέξει υπάρχει παραπάνω από μια φορά στη σελίδα τότε μπορούμε να διαλέξουμε πια εμφάνιση του θα θέλαμε να μετατρέψουμε σε link βάζοντας τον κατάλληλο αριθμό (η αρίθμηση ξεκινά από το 1), μπορεί όμως να γίνει και η επιλογή όλων των εμφανίσεων βάζοντας το all σαν string, όπως φαίνεται στο Σχήμα 5.25
- span\_links : είναι ένα αντικείμενο που μέσα του έχει τα ακόλουθα πεδία (το κάθε ένα αντιπροσωπεύει ένα πινάκα και μπορεί να έχει παραπάνω από μια τιμή) :
  1. wikipedia : τα string που βρίσκονται μέσα στον πίνακα μετατρέπονται σε συνδέσμους σε λήματα του Βικιπέδια.
  2. url : τα string που βρίσκονται μέσα στον πίνακα πρέπει να είναι έγκυρα URL.
  3. exercise : οι τιμές πρέπει να είναι αριθμοί και να αντιπροσωπεύουν υπάρχουσες ασκήσεις, αλλιώς ειδοποιείται ο χρήστης για το σφάλμα με ένα toast.
  4. quiz : οι τιμές πρέπει να είναι αριθμοί και να αντιπροσωπεύουν υπάρχουσα κουίζ, αλλιώς ειδοποιείται ο χρήστης για το σφάλμα με ένα toast.

Οι αριθμοί για τα πεδία exercise και quiz πρακτικά αναφέρονται στα id που είναι δηλωμένα εντός της εφαρμογής για αυτές τις οντότητες, για τις οποίες θα λάβει μέρος η ανάλυση σε επόμενες υποενότητες.

Το component εσωτερικά έχει μηχανισμούς για την παρακολούθηση της τρέχουσας σελίδα εμφανίζεται στο έγγραφο.

Εκτός αυτού έχει γίνει και η προσθήκη του span-links-component.js μέσα στο PDF.js, που υλοποιεί το μενού που εμφανίζεται στο Σχήμα 5.18

Ουσιαστικά εδώ επεκτείνουμε ένα ήδη υπάρχον στοιχείο της HTML, το span με έξτρα λειτουργίες.

Μόλις ανιχνευτεί ότι η τρέχουσα σελίδα έχει και εγγραφή στο αρχείο που περιέχει τα link τότε λαμβάνει μέρος η συγκεκριμένη σειρά ενεργειών :

- Πρώτα σκανάρονται όλοι οι χαρακτήρες της σελίδας και χωρίζονται σε μεμονωμένες λέξεις σε επίπεδο στοιχείων span (επεμβαίνουμε στον κώδικα της σελίδας δυναμικά και χωρίζουμε τα μεγαλύτερα span σε μικρότερα μέχρι να έχουμε μεμονωμένες λέξεις).

- Έπειτα γίνεται η εύρεση όλων των span που απαρτίζουν τη φράση που επιθυμούμε να μετατρέψουμε σε link, και η τοποθέτηση τους μέσα σε ένα πίνακα.
- Μετά γίνεται η αντικατάσταση τους από τα αντίστοιχα span-links-components στα οποία προστίθενται όλα τα υπόλοιπα στοιχεία του πεδίου span\_links της εγγραφής έτσι ώστε να μπορεί να λάβει μέρος η δημιουργία των επιλογών.

Αυτή η διαδικασία επαναλαμβάνεται για τα υπόλοιπα link που μπορεί να έχει μια σελίδα.

Το “reset” που πρέπει να δέχονται τα link περιοδικά προκύπτει από το γεγονός ότι το PDF.js για να εξοικονομήσει πόρους του συστήματος δεν κρατά φορτωμένο όλο το αρχείο στη μνήμη αλλά μόνο τις πιο πρόσφατες σελίδες.

Το γεγονός αυτό προκαλεί τον επανασχεδιασμό των σελίδων που έχουν καταστραφεί, αυτό έχει ως αποτέλεσμα την ανάγκη για την εκ νέου πρόσθεση των link που ενδεχομένως να είχαν καταστραφεί.

#### 5.4.4.3 Σημειώσεις εγγράφου

Το αρχεία που υλοποιούν το σύστημα των σημειώσεων βρίσκονται στον φάκελο NoteManager.

Η πιο κύρια οντότητα του συστήματος είναι ο note-manager.js και είναι υπεύθυνος για την αποθήκευση, ανάγνωση και τροποποίηση όλων των αποθηκευμένων στην localStorage σημειώσεων καθώς και την παροχή μεθόδων που επιτρέπουν σε άλλες οντότητες του συστήματος να επενεργήσουν πάνω στις σημειώσεις.

Το note-container-component.js είναι αυτό που οργανώνει της σημειώσεις ως προς τη σειρά εμφάνισης τους, παρέχει τη λειτουργία δημιουργίας καινούριας σημείωσης με τη χρήση του note-manager.js και λειτουργεί ως η γέφυρα ανάμεσα στις σημειώσεις και στον editor.

Ο editor των σημειώσεων υλοποιείται στο αρχείο editor-component.js και εσωτερικά κάνει χρήση του προγράμματος Suneditor.js καθώς και του note-manager.js για να μπορεί να αποθηκεύει και να διαβάζει τις σημειώσεις.

Οι δύο ειδών σημειώσεις, του χρήστη και του καθηγητή υλοποιούνται στα αντίστοιχα αρχεία note-component.js και teacher-note-component.js.

Εσωτερικά έχουν μεθόδους που χρησιμοποιώντας τον note-manager.js λειτουργούν ως γέφυρα για τη διεκπεραίωση των λειτουργιών της αυτόματης αποθήκευσης αλλαγής ονόματος, ενημέρωση χρόνου τροποποίησης, διαγραφή κτλ.

Το teacher-note-component.js αποτελεί μια τροποποίηση του note-component.js, αλλάζοντας των τρόπο άντλησης των σημειώσεων, το χρώμα εμφάνισης και την αφαίρεση της λειτουργίας διαγραφής.

Οι σημειώσεις του καθηγητή αντλούνται από τον φάκελο NoteContentFiles/PdfNotes.

Για να γίνει η αντιστοιχία του αρχείου με ένα έγγραφο ο τίτλος του πρέπει να έχει την εξής μορφή [pdf\_note[id του εκάστοτε εγγράφου].json].

Εσωτερικά αυτό αρχείο πρέπει να περιέχει ένα πίνακα, που μπορεί να έχει διάφορα αντικείμενα που περιγράφουν μια σημείωση.

Τα πεδία αυτά είναι όλα string και μπορούν να έχουν όποια τιμή καλύπτει καλύτερα τις ανάγκες του καθηγητή (τα πεδία αυτά είναι title, description, content, creationDate, lastModificationDate και εμφανίζονται στο Σχήμα 5.26).

```
{
  "title": "A note from the teacher",
  "description": "This is in the first pdf document",
  "content": "<h1><u><strong>This is the title</strong></u><br></h1><p>This is the first sentence.<br></p><p>This is a math expression :&nbsp; x<sup>2</sup>+ 1 = y<br></p><p><table class='se-table-size-100'><tbody><tr><td><div>1<br></div></td><td><div>a<br></div></td><td><div>b<br></div></td></tr><tr><td><div>1<br></div></td><td><div>a<br></div></td><td><div>b<br></div></td></tr></tbody></table><h1 style='text-align: center;'><span style='background-color: rgb(0, 85, 255);'><span style='color: rgb(255, 0, 0);'><del>This title is centered</del></span></span><br></h1><hr class='_se_solid on'><p>This is a list:<br></p><ol><li>Item 1<br></li><li>Item 2<br></li><li>Item 3<br></li></ol>",
  "creationDate": "Attention",
  "lastModificationDate": "Updated links"
}
```

Σχήμα 5.26: Εμφάνιση εγγραφής σημείωσης καθηγητή

#### 5.4.4.4 Ανάκτηση σημειώσεων

Το view που φαίνεται στο Σχήμα 5.23 υλοποιείται στο αρχείο note-recovery-container-component.js και ουσιαστικά πρόκειται για μια ελαφρός τροποποιημένη έκδοση του note-container-component.js που αφαιρεί τη δυνατότητα προσθήκης νέων σημειώσεων και τροποποιεί τον τρόπο διαγραφής των προς ανάκτηση σημειώσεων.

Ο τρόπος που εφαρμογή γνωρίζει όταν ένα έγγραφο ύλης δεν είναι πια προσβάσιμο είναι με το να κρατάει ένα αντίγραφο από την προηγούμενη κατάσταση των εγγράφων.

Μόλις ανιχνεύσει ότι υπάρχει κάποια αλλαγή τότε κοιτάζει πιο έγγραφο λείπει και προβαίνει στις κατάλληλες ενέργειες για να εκκίνηση το περιβάλλον ανάκτησης.

#### 5.4.4.5 Σελιδοδείκτες εγγράφου

Τα αρχεία που υλοποιούν το σύστημα των σελιδοδεικτών βρίσκεται στον φάκελο BookmarkManager.

Η οντότητα που είναι υπεύθυνη για τη δημιουργία, τροποποίηση και διαγραφή των σελιδοδεικτών είναι ο bookmark-manager.js.

Ο ίδιος ο σελιδοδείκτης υλοποιείται με τη χρήση του αρχείου bookmark-component.js, που είναι υπεύθυνο για την απεικόνιση του στην οθόνη κάνοντας χρήση των μεθόδων που παρέχει ο bookmark-manager.js, για την επίτευξη των λειτουργιών αυτόματης αποθήκευσης αλλαγών, διαγραφής αλλά και του έλεγχου ορθότητας του αριθμού σελίδας.

Το bookmark-container-component.js είναι αυτό που παρέχει το περιβάλλον πάνω στο οποίο εμφανίζονται οι σελιδοδείκτες καθώς και τη δυνατότητα δημιουργίας καινούριου, με τη χρήση του bookmark-manager.js.

Στην πραγματικότητα οι σελιδοδείκτες αποτελούν τροποποιημένες σημειώσεις.

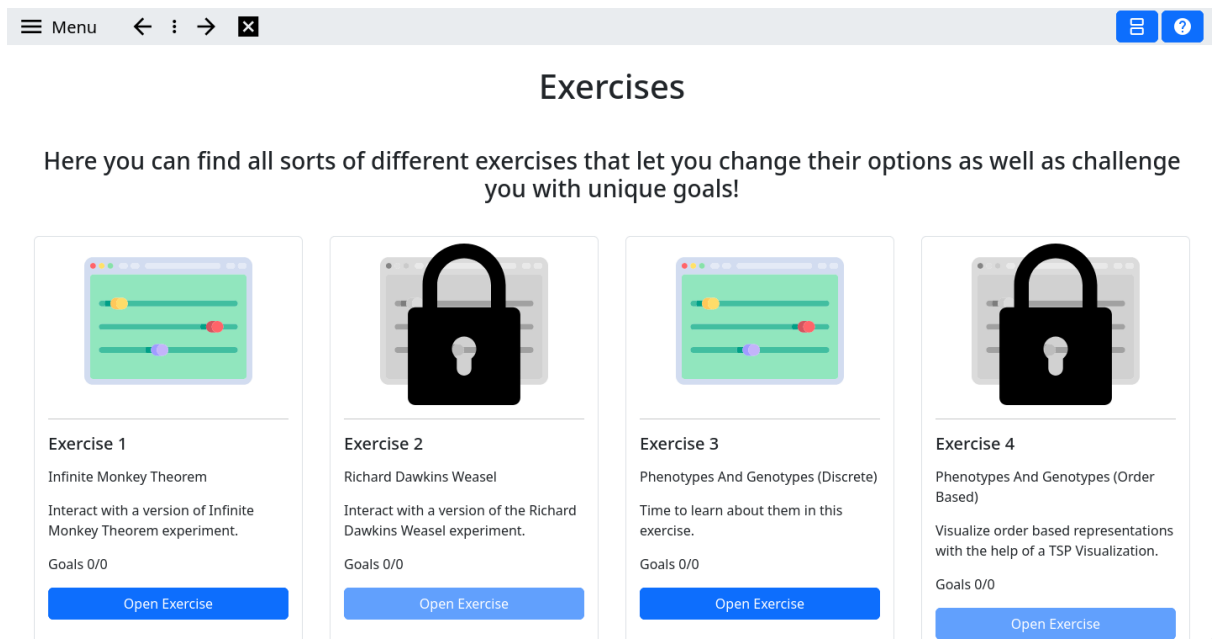
### 5.5 Ασκήσεις εφαρμογής

#### 5.5.1 Εισαγωγή

Σε αυτή την ενότητα γίνεται η παρουσίαση του τρόπου λειτουργίας των ασκήσεων που έχει η εφαρμογή καθώς και μια συνοπτική αναφορά στα αρχεία που υλοποιούν τις εκάστοτε λειτουργίες.

### 5.5.2 View εμφάνισης ασκήσεων (Exercises)

Όλες οι διαθέσιμες ασκήσεις της εφαρμογής εμφανίζονται με τον τρόπο που φαίνεται στο Σχήμα 5.27.



Σχήμα 5.27: Εμφάνιση view ασκήσεων

Η κάθε μια κάρτα που αντιπροσωπεύει μια άσκηση έχει ένα τον αριθμό αύξοντα αριθμό της άσκησης ως προς τη σειρά που εμφανίζονται, έναν τίτλο καθώς και μια σύντομη περιγραφή της.

Εκτός αυτών στο κάτω αριστερό μέρος της έχει και δύο αριθμούς χωρισμένους με το σύμβολο / που δείχνει πόσοι στόχοι της άσκησης έχουν συμπληρωθεί σε σχέση με τον συνολικό αριθμό των στόχων.

Οι στόχοι αναλύονται περαιτέρω στην υποενότητα 5.5.4.

Τέλος, κάποιες από τις ασκήσεις παρουσιάζουν ένα λουκέτο, που δηλώνει ότι δεν είναι ακόμα προσβάσιμες γιατί δεν έχει λάβει μέρος η ολοκλήρωση των στόχων της προηγούμενης άσκησης.

#### 5.5.2.1 Συμπληρωματικές λεπτομέρειες

Το συγκεκριμένο view υλοποιείται στο αρχείο `exercise-view.js`

Ο τρόπος που εφαρμογή γνωρίζει ποιες και που είναι η διαθέσιμες ασκήσεις έχει πολλές ομοιότητες με αυτόν που παρουσιάστηκε στο 5.4.2.1 με τα έγγραφα ύλης.

Μέσα στον φάκελο `ExerciseContentFiles` υπάρχει ένα αρχείο με όνομα `exercise_definitions.json`, το οποίο έχει εξαιρετικά παρόμοια δομή με το αρχείο που χρησιμοποιείτε για τα έγγραφα.

Πιο συγκεκριμένα έχουν τον ίδιους κανόνες ως προς τη σύνταξη του `id` και μοιράζονται τα πεδία του τίτλου και της περιγραφής (επίσης με κάθε αλλαγή του υπάρχει η άμεση ανάδραση στην πλευρά του χρήστη όπως γίνεται και με τα άλλα αρχεία τέτοιου τύπου).

Οι διαφορές τους είναι ότι η συσχέτιση της άσκησης με το `id` γίνεται με διαφορετικό τρόπο και όχι με τη χρήση του πεδίου `fileName` και επίσης η ύπαρξη ενός έξτρα πεδίου.

## Κεφάλαιο 5ο:

Αυτό το πεδίο είναι το `lockedState`, και ορίζει το αν μια άσκηση ξεκινά για τον χρήστη ως κλειδωμένη ή ξεκλειδωτή ανάλογα με το αν η τιμή είναι 0 ή 1 (το πεδίο είναι προαιρετικό και η απουσία του σηματοδοτεί το κλείδωμα της άσκησης).

Εάν γίνει μελλοντική αλλαγή μιας άσκησης από ξεκλειδωτή σε κλειδωμένη από τον καθηγητή ενώ ο χρήστης την είχε ήδη ξεκλειδώσει προηγούμενος, τότε αυτή η αλλαγή απορρίπτεται από την πλευρά αυτού του χρήστη έτσι ώστε να μη χάσει την πρόοδο του.

Ο τρόπος που γίνεται η συσχέτιση της άσκησης με μια εγγραφή είναι με το να υπάρχει το `id` της εγγραφής στο όνομα της εκάστοτε άσκησης.

Πιο συγκεκριμένα αν μια εγγραφή ορίζει ως `id` το “\_123” τότε εάν στον φάκελο που υπάρχουν υλοποιημένες οι ασκήσεις (αυτός ο φάκελος είναι ο `Exercises` που βρίσκεται μέσα στον φάκελο `ExerciseContentFiles`) υπάρχει μια άσκηση με όνομα αρχείου “`exercise_123.js`”, τότε γίνεται αυτόματα η συσχέτιση (τα ονόματα των ασκήσεων πρέπει να ακολουθούν αυτό το πρότυπο).

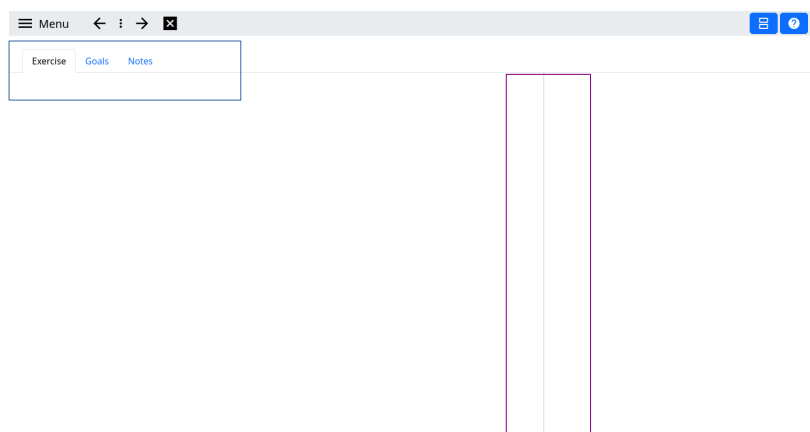
### 5.5.3 Διευκρίνιση αρχιτεκτονικής ασκήσεων

Πριν γίνει η παρουσίαση των ασκήσεων είναι χρήσιμο να πραγματοποιήσει η ανάλυση του προτύπου πάνω στο οποίο δουλεύουν και έχουν όλες του κοινό. Αυτό πρότυπο παρέχει μια οργάνωση της διεπαφής που έχουν στη διάθεση τους οι ασκήσεις. Η διεπαφή αυτού του προτύπου φαίνεται στο Σχήμα 5.28.

Η διεπαφή χωρίζεται σε τρεις καρτέλες (είναι μαρκαρισμένες με μπλε πλαίσιο στο σχήμα προαναφερθέν σχήμα) και η αρχική οθόνη σε δύο καθέτως χωρισμένα κομμάτια (η διαχωριστική τους γραμμή είναι μαρκαρισμένη με μοβ πλαίσιο στο προαναφερθέν σχήμα).

Στην αρχική οθόνη το αριστερό μέρος της ορίζεται για τα οπτικά στοιχεία που χρησιμοποιεί η άσκηση και το δεξί μέρος για όλες τις ρυθμίσεις. Οι δεύτερη από τις τρεις καρτέλες δείχνει του στόχους που μπορεί να υπάρχουν για μια άσκηση και θα αναλυθεί σε μεγαλύτερο βάθος στην υποενότητα 5.5.4.

Η τρίτη και τελευταία επιτρέπει τη σύνταξη σημειώσεων και λειτουργεί με ακριβώς τον ίδιο τρόπο με αυτόν που αναφέρεται στην υποενότητα 5.4.3.2 (οι σημειώσεις του καθηγητή τοποθετούνται στον φάκελο `NoteContentFiles/ExerciseNotes`). Τα κύρια αρχεία που υλοποιούν αυτή τη διεπαφή βρίσκονται μέσα στον φάκελο `ExerciseManager` και είναι το `exercise-viewer-component.js` και το `exercise-template-component.js`.



Σχήμα 5.28: Εμφάνιση προτύπου ασκήσεων

Το `exercise-viewer-component.js` υλοποιεί τη γενικότερη διεπαφή δηλαδή είναι υπεύθυνο για τη διαχείριση των καρτελών και της αλλαγής ανάμεσα τους.

Αυτό που κάνει είναι χρησιμοποιεί κάποια άλλα `view` της εφαρμογής, όπως το `view` των σημειώσεων και να τους παρέχει ένα τρόπο απόκρυψης όταν δεν είναι ανάγκη να προβάλλονται.

Το `exercise-template-component.js` είναι αυτό που γεμίζει την κεντρική οθόνη και παρέχει τον διαχωρισμό της δεξιάς με την αριστερή πλευρά.

Όλες οι ασκήσεις πρέπει να κάνουν χρήση αυτού του αρχείο επεκτείνοντας το και χρησιμοποιώντας δύο εσωτερικές μεταβλητές του.

Η μία μεταβλητή είναι η `exerciseView`, η οποία παρέχει πρόσβαση στην αριστερή πλευρά της οθόνης και η άλλη είναι η `exerciseOptions` που παρέχει πρόσβαση στο δεξί μέρος.

Με αυτήν εδώ την πρόσβαση ο προγραμματιστής είναι σε θέση να ορίσει όποια στοιχεία διεπαφής επιθυμεί με ένα τρόπο που παρέχει εύκολη οργάνωση.

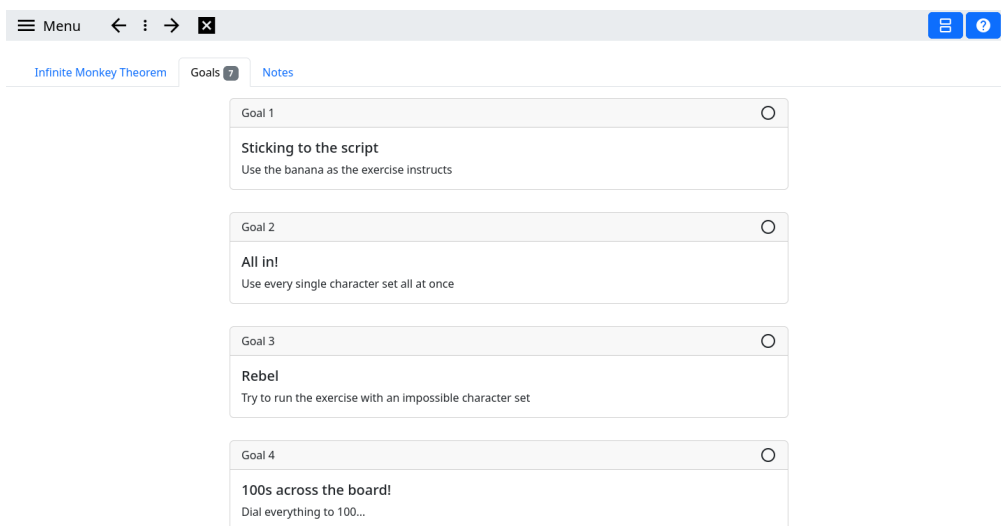
### 5.5.4 Τρόπος λειτουργίας των στόχων των ασκήσεων

Για κάθε άσκηση υπάρχει η δυνατότητα προσθήκης διάφορων στόχων που πρέπει να εκπληρωθούν για να ξεκλειδωθεί η επόμενη στη σειρά άσκηση (εάν για μια άσκηση δεν έχουν δηλωθεί στόχοι τότε η επόμενη ξεκλειδώνει αυτόματα).

Η προσθήκη των στόχων έχει ως σκοπό να ωθήσει τον χρήστη στο να κάνει πλήρη χρήση όλων των διαθέσιμων ρυθμίσεων που προσφέρει μια άσκηση αλλά και να του διδάξει πράγματα για το θέμα που παρουσιάζει η άσκηση με έναν τρόπο παραπλήσιο σε αυτόν ενός παιχνιδιού, κάνοντας έτσι τη διαδικασία μάθησης πιο ευχάριστη.

Ο χρήστης μπορεί να έχει πρόσβαση στους στόχους κάνοντας πάνω στην καρτέλα “Goals” (η καρτέλα επίσης δείχνει τον αριθμό των στόχων που δεν έχουν ολοκληρωθεί).

Εκεί γίνεται η παράθεση όλων των στόχων με τον τρόπο που φαίνεται στο Σχήμα 5.29.



Σχήμα 5.29: Εμφάνιση στόχων μιας άσκησης

## Κεφάλαιο 5ο:

Εδώ φαίνονται όλοι οι στόχοι μιας άσκησης, όπου ο κάθε ένας έχει έναν αύξοντα αριθμό που δείχνει τη σειρά εμφάνισης του καθώς και έναν τίτλο και μια σύντομη περιγραφή τους.

Το ποιοι είναι οι στόχοι, τι τίτλο και περιγραφή έχουν είναι καθαρά αυθαίρετα και μπορούν να διαφέρουν από τη μία άσκηση στην άλλη.

Όταν λάβουν μέρος οι ενέργειες που απαιτούνται για την επίτευξη ενός στόχου, τότε ο χρήστη ειδοποιείτε με ένα toast και ο στόχος αλλάζει την εμφάνιση του και μειώνεται κατά ένα ο αριθμός της καρτέλας (γίνεται σαν αυτόν που φαίνεται στο Σχήμα 5.30)

Μόλις λάβει μέρος η ολοκλήρωση όλων των στόχων τότε αυτόματα ξεκλειδώνεται η επόμενη άσκηση που έχει ως αποτέλεσμα την ενεργοποίηση του κουμπιού έναρξης της και την αφαίρεση του λουκέτου.

### 5.5.4.1 Συμπληρωματικές λεπτομέρειες

Όλα τα αρχεία που έχουν να κάνουν με την εμφάνιση αλλά και την οργάνωση των στόχων των ασκήσεων είναι στον φάκελο GoalManager. Το view που φαίνεται στο Σχήμα 5.29 υλοποιείται στο αρχείο goal-container-component.js και είναι υπεύθυνο για την παρουσίαση των στόχων, την ενημέρωση του αριθμού των ανεκπλήρωτων στόχων στην καρτέλα αλλά και την παροχή της μεθόδου ενεργοποίησης ενός στόχου.



Σχήμα 5.30: Εμφάνιση ολοκληρωμένου στόχου

Ο κάθε στόχος είναι και ξεχωριστό component που υλοποιείτε με τη βοήθεια του αρχείου goal-component.js που ρυθμίζει τον τρόπο εμφάνισης ενός στόχου και έχει μια μέθοδο εσωτερικά που προκαλεί την αλλαγή του χρώματός του όταν λάβει μέρος η ολοκλήρωση του.

Αυτός που είναι υπεύθυνος για την αποθήκευση των ποιών στόχων έχουν ολοκληρωθεί καθώς και τη γενική διαχείριση των αρχείων που τους ορίζουν είναι ο goal-manager.js.

Το ποιοι στόχοι αντιστοιχούν σε ποια άσκηση δουλεύει με τον εξής τρόπο. Μέσα στον φάκελο ExerciseContentFiles υπάρχουν αρχεία JSON που έχουν ακριβώς το ίδιο όνομα (εκτός του τμήματος της επέκτασης) με τις αντίστοιχες ασκήσεις του υπάρχουν στον φάκελο Exercises.

```
"_1": {  
  "title": "Sticking to the script",  
  "description": "Use the banana as the exercise instructs"  
},
```

Σχήμα 5.31: Εγγραφή πεδίου ενός αρχείου στόχων

Όταν λοιπόν υπάρχει μια άσκηση και ένα αρχείο JSON (δεδομένου ότι έχουν και τα δύο σωστή δομή) με το ίδιο όνομα, τότε αυτή η άσκηση έχει και στόχους. Μια εγγραφή ενός στόχου έχει τη μορφή που φαίνεται στο Σχήμα 5.31 και περιέχει ένα id (εδώ είναι το \_1 και πρέπει να είναι μοναδικό για κάθε στόχο στο αρχείο) έναν τίτλο και μια περιγραφή.

Πρέπει να σημειωθεί ότι η στόχοι υλοποιούνται από τον ίδιο τον προγραμματιστή της άσκησης, η εγγραφές σε αυτά τα αρχεία υπάρχουν για να περιγράψουν τους στόχους στον χρήστη, για το πως θα τους φέρει εις πέρας.

## 5.5.5 Παρουσίαση ασκήσεων

### 5.5.5.1 Άσκηση 1

Η πρώτη άσκηση διαπραγματεύεται το [Θεώρημα των απείρων πιθήκων](#) για το οποίο λέει ότι ένας πίθηκος αν είχε άπειρο χρόνο θα μπορούσε να γράψει όλα τα έργα του Ουίλιαμ Σαίξπηρ με μια γραφομηχανή.

Το view της άσκησης φαίνεται στο Σχήμα 5.32.

Σχήμα 5.32: Εμφάνιση view πρώτης άσκησης

Στη συγκεκριμένη άσκηση ο χρήστης μπορεί να δώσει ένα string ως είσοδο και να δει πόση ώρα χρειάζεται για να βρεθεί αυτό μέσα από τη διαδικασία της παραγωγής τυχαίων string ίδιου μήκους.

Μπορεί να ορίσει τι είδους χαρακτήρες θέλει να συμπεριλαμβάνονται στα τυχαία string που δημιουργούνται καθώς και να διαλέξει τα κριτήρια τερματισμού της αναζήτησης (μπορεί να εκτελείται η άσκηση μέχρι να βρεθεί το ζητούμενο string ή να σταματήσει η διαδικασία μετά από ένα συγκεκριμένο αριθμό προσπαθειών που μπορεί να ορίσει ο χρήστης).

Έκτος αυτών μπορεί εάν το θέλει μπορεί να κρατά ένα είδους log κάποιων προσπαθειών (υπάρχουν διάφορες επιλογές για το πόσες εγγραφές κρατάει), όπου αυτές εμφανίζονται στην κάτω αριστερή γωνία του view.

Συμπληρωματικά μπορεί να αλλάξει των χρόνο που απαιτείται για τη δημιουργία ενός καινούριου string σε ms.

## Κεφάλαιο 5ο:

Τέλος, στο κάτω μέρος του view υπάρχουν τρία κουμπιά που ελέγχουν την εκτέλεση της άσκησης.

Το πρώτο κουμπί με όνομα “Start” ξεκινά τη διαδικασία εύρεσης του string με την επαναληπτική διαδικασία δημιουργίας τυχαίων string μέχρι να βρεθεί το αρχικό (εάν φυσικά έχει συμπληρωθεί το string προς εύρεση και το σετ χαρακτήρων που επιλέχθηκαν για τη δημιουργία των string μπορούν να δημιουργήσουν και το αρχικό.

Το δεύτερο κουμπί με όνομα “Stop” σταματά την εκτέλεση της άσκηση προσωρινά (ο χρήστης μπορεί να πατήσει το “Start” και να συνεχιστεί η εκτέλεση από το σημείο που διακόπηκε).

Τέλος, το κουμπί “Reset” επαναφέρει όλη την άσκηση στην αρχική της κατάσταση.

Στο τέλος όλης της διαδικασίας εύρεσης ο χρήστης ενημερώνεται με toast για το αν τελικά βρέθηκε το string ή όχι.

### 5.5.5.2 Άσκηση 2

Η δεύτερη άσκηση διαπραγματεύεται το πείραμα του [Richard Dawkins](#) το οποίο λέει πως η κινητήριος δύναμη των εξελικτικών συστημάτων που είναι η τυχαία αλλαγή σε συνδυασμό με μια μην τυχαία συσσώρευση λαθών είναι διαφορετική από την τυχειότητα.

Richard Dawkins Weasel

Goals Notes

Iteration: 154

Target: test

Previous Fitness: 75%

Current Fitness: 100%

teet

test

| Iteration | Generated String | Fitness Percentage |
|-----------|------------------|--------------------|
| 154       | teet             | 100%               |
| 153       | t et             | 50%                |
| 152       | t et             | 50%                |

Type the text that you wish to find

test

Select what characters you want in the character set

☒ a-z

☐ A-Z

☐ 0 - 9

Select the termination method of the exercise

Until the string is found

Select how many string records to keep in the table

1000

Iteration Speed Delay : 0ms

Start Stop Reset

Σχήμα 5.33: Εμφάνιση view δεύτερης άσκησης

Η άσκηση αυτή χρησιμοποιεί την πρώτη ως βάση, για αυτόν τον λόγο έχει και τις ίδιες ρυθμίσεις όμως διαφέρει στον τρόπο που κάνει την αναζήτηση του string.

Αυτό που κάνει είναι να δημιουργεί ένα τυχαίο string όπως και στην πρώτη άσκηση αλλά έπειτα αντί να δημιουργεί εκ νέου ένα άλλο string αντιγράφει το πρώτο string με κάποιο λάθος (διαλέγει ένα τυχαίο χαρακτήρα μέσα στο string και τον αντικαθιστά με έναν άλλον τυχαίο).

Μετά την αντιγραφή γίνεται ένας έλεγχος για το πόσο όμοιο είναι το νέο string με το string στόχο (προσμετρούνται οι σωστοί χαρακτήρες, που θεωρούνται αυτοί που είναι στην ίδια θέση με αυτούς

του στόχου και με τη διαίρεση τους με το σύνολο των χαρακτήρων βγαίνει ένα ποσοστό καταλληλότητας) και αν το ποσοστό καταλληλότητας είναι μεγαλύτερο από το προηγούμενο string τότε αυτό αντικαθίσταται από το νέο.

Η διαδικασία αυτή επαναλαμβάνεται μέχρι να βρεθεί το string στόχος.

Η άσκηση επίσης έχει διαφορετικές εγγραφές στο log που φαίνεται στην κάτω αριστερή γωνία στο Σχήμα 5.33.

Για κάθε μια εγγραφή φαίνεται ποιος χαρακτήρα αντικαταστάθηκε (έχει κόκκινο πλαίσιο) και με ποιόν (έχει πράσινο πλαίσιο) καθώς και το ποσοστό καταλληλότητας που έφερε η συγκεκριμένη αλλαγή.

Κάθε φορά που μία αλλαγή φέρει καλύτερο ποσοστό από την τρέχουσα καλύτερη τότε αυτή αποκτά ένα γαλάζιο χρώμα πλαισίου.

Αξίζει να σημειωθεί ότι η σειρά καταγραφής του log είναι ανάποδη (η πιο πρόσφατη αλλαγή είναι το πάνω μέρος του πίνακα για να είναι πιο εύκολη η ανάγνωση του).

Στη μέση του view υπάρχουν δύο κουτιά, εκ των οποίων το πρώτο δείχνει το προηγούμενο string με το καλύτερο ποσοστό καταλληλότητας και το δεύτερο δείχνει το τρέχον string με τη μεγαλύτερη τιμή καταλληλότητας.

### 5.5.5.3 Άσκηση 3

Αυτή η άσκηση έχει σκοπό να δείξει τη συσχέτιση που μπορεί να υπάρχει ανάμεσα σε ένα φαινότυπο και έναν γονότυπο για διακριτές αναπαραστάσεις.

Phenotypes And Genotypes (Discrete) Goals Notes

1 1 0 1 1

$$1*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 27$$

Select the phenotype representation you want  
Integer

Select the size of the genotype : 5

Insert your own genotype or modify the existing one  
11011

Generate Genotype Reset

Σχήμα 5.34: Εμφάνιση view τρίτης άσκησης

Το view της άσκησης φαίνεται στο Σχήμα 5.34.

## Κεφάλαιο 5ο:

Εδώ ο χρήστης μπορεί να αλλάξει τον φαινότυπο και να δει τι αλλαγές προκαλεί αυτό στο αριστερό μέρος του view.

Μπορεί επίσης να αλλάξει το πόσα γονίδια χρησιμοποιούνται για τον γονότυπο οι και να εισάγει τα δικά του με τη μορφή ενός string που αποτελείται μόνο από του χαρακτήρες 0 και 1.

Όταν έχει προβεί στην αλλαγή του μεγέθους του γονότυπου ή έχει εισάγει τον δικό του τότε κάνοντας κλικ στο κουμπί “Generate Genotype” τότε η εφαρμογή θα επεξεργαστεί τις ρυθμίσεις που έχει εισάγει ο χρήστης και θα ανανεώσει κατάλληλα το αριστερό μέρος του view.

Με το κουμπί “Reset” μπορεί να επαναφέρει την άσκηση στις αρχικές τις ρυθμίσεις.

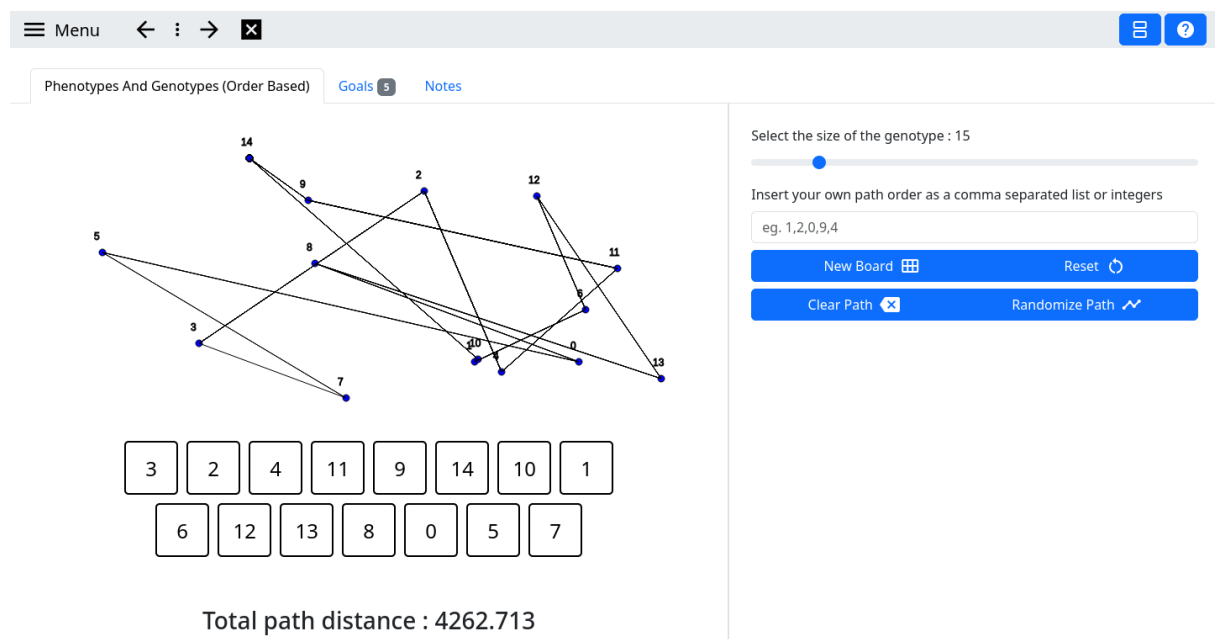
Υπάρχει και μια έξτρα λειτουργία εάν ο χρήστης ορίσει τον τύπο του φαινότυπου σαν πραγματικό αριθμό.

Μόλις γίνει αυτό έχει μετά τη δυνατότητα να εισάγει μια μαθηματική φόρμουλα που μπορεί να έχει διάφορους συνδυασμούς αριθμών, μαθηματικών συμβόλων (όπως +, -, \*, /) αλλά και τη μεταβλητή X η οποία είναι η αναπαράσταση του γονότυπου σαν έναν ακέραιο θετικό αριθμό.

### 5.5.5.4 Άσκηση 4

Η άσκηση λειτουργία ως ένα εργαλείο οπτικοποίησης των γραφημάτων που δημιουργούνται για το πρόβλημα του πλανόδιου πωλητή ([TSP](#)), που εξετάζει ποια μπορεί να είναι η καλύτερη σειρά μεταβίβασης από μια πόλη σε μια άλλη μόνο μια φορά και την επιστροφή στην αρχική με τον στόχο της ελαχιστοποίησης της απόστασης που πρέπει να διανυθεί ανάμεσα στις πόλεις.

Εκτός αυτού παρέχει και μια συσχέτιση ανάμεσα σε ένα φαινότυπο και έναν γονότυπο για αναπαραστάσεις που έχει σημασία η σειρά των γονιδίων μέσα στον γονότυπο.



Σχήμα 5.35: Εμφάνιση view τέταρτης άσκησης

Το view της άσκησης φαίνεται στο Σχήμα 5.35

Στην άσκηση αυτή ο χρήστης μπορεί να διαλέξει τον αριθμό των πόλεων που θα εμφανιστούν στο γράφημα. Εκτός αυτού μπορεί κάνοντας κλικ στο κουμπί “New Board” να δημιουργήσει μια νέα τυχαία αναδιάταξη των πόλεων που επέλεξε.

Με κάθε νέα διάταξη φαίνεται στο αριστερό μέρος του view το αντίστοιχο παραγόμενο γράφημα, που δείχνει μια τυχαία δημιουργούμενη σειρά που ακολουθείτε από τον “πωλητή”.

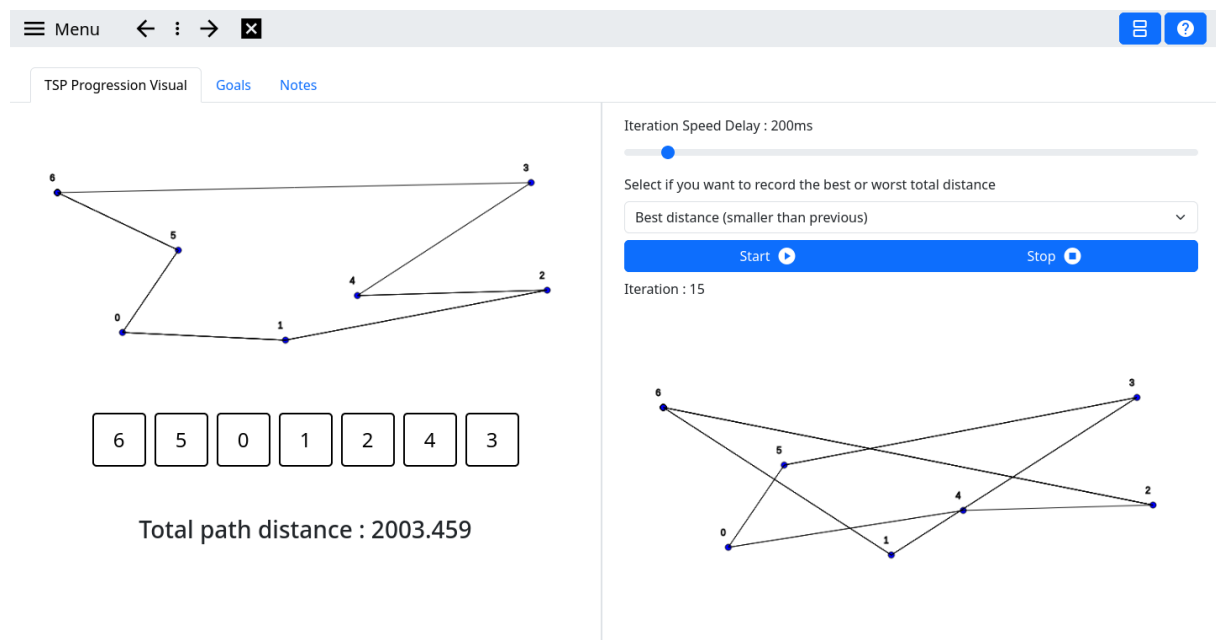
Ακριβός κάτω του γραφήματος υπάρχει η γονιδιακή αναπαράσταση της σειρά με την οποία ο “πωλητής” επισκέφτηκε τις πόλεις καθώς και η συνολική διανυόμενη απόσταση.

Ο χρήστης αν δεν είναι ικανοποιημένος με την παραγόμενη σειρά τότε μπορεί να δημιουργήσει μια νέα κάνοντας κλικ στο κουμπί “Randomize Path”. Με αυτή τη λειτουργία κρατάτε η τοποθέτηση των πόλεων ίδια και απλά αλλάζουν όλα που έχουν να κάνουν με τη σειρά.

Εκτός αυτού ο χρήστης μπορεί να σχεδιάσει τη δική του διαδρομή, κάνοντας πρώτα τον καθαρισμό της τρέχουσα κάνοντας κλικ στο “Clear Path”, και έπειτα παρέχοντας ένα string από αριθμούς που είναι τα νούμερα των πόλεων χωρισμένες με κόμμα (πχ 4,3,1,2,0). Τέλος, το κουμπί “Reset” επαναφέρει τις ρυθμίσεις στις αρχικές τους τιμές.

#### 5.5.5.5 Άσκηση 5

Η πέμπτη και τελευταία άσκηση αποτελεί μια τροποποίηση της προαναφερθείσας, παρέχοντας τη δυνατότητα της οπτικοποίησης των αλλαγών που λαμβάνουν μέρος στην εύρεση της μικρότερης ή της μεγαλύτερης δυνατής απόστασης για μια κατάταξη πόλεων.



Σχήμα 5.36: Εμφάνιση view πέμπτης άσκησης

Το view της άσκησης φαίνεται στο Σχήμα 5.36

Οι λειτουργίες που παρέχει λειτουργούν με ακριβός τον ίδιο τρόπο όπως αυτές της προηγούμενης άσκησης απλά σε αυτήν έχει προστεθεί η δυνατότητα καταγραφής στιγμιotypών από τις διάφορες φάσεις εκτέλεσης της άσκησης.

Κεφάλαιο 5ο:

Πιο συγκεκριμένα ο χρήστης διαλέγει πόσο γρήγορα θα ήθελε η εφαρμογή να παράγει νέους συνδυασμούς σειράς των πόλεων σε ms.

Έπειτα διαλέγοντας ποιος είναι το τελικός στόχος (μπορεί να είναι η μεγιστοποίηση ή η ελαχιστοποίηση της στοχευμένη απόστασης) μπορεί να κάνει κλικ στο “Start”.

Με αυτή την ενέργεια η εφαρμογή θα δημιουργεί κάθε φορά καινούριους συνδυασμούς σειράς των πόλεων, και ανάλογα την επιλογή του στόχου καταγράφει στο κάτω δεξί μέρος του view την τρέχουσα καλύτερη σειρά αλλά και τη συνολική απόσταση που αυτή παράγει.

Έτσι μπορεί να δει σε πραγματικό χρόνο το γράφημα να γίνεται πιο αραιό ή πιο πυκνό, ανάλογα με το εάν ο στόχος είναι μεγιστοποίηση ή η ελαχιστοποίηση της διανυόμενης απόστασης.

#### 5.5.5.6 Αρχεία οπτικοποίησης

Οι οπτικοποίηση των γραφημάτων TSP (για την απεικόνιση γίνεται χρήση του Canvas API) και του γονότυπου που χρησιμοποιείται σε πολλές από τις ασκήσεις υλοποιούνται στα αρχεία tsp.js και basic-genotype.js αντίστοιχα που βρίσκονται στον φάκελο EvoGen/Visualizations. Οι υλοποιήσεις αυτών των οπτικοποιήσεων παρέχουν μια πληθώρα μεθόδων που επιτρέπουν την προσαρμογή τους σε ένα μεγάλο φάσμα εφαρμογών.

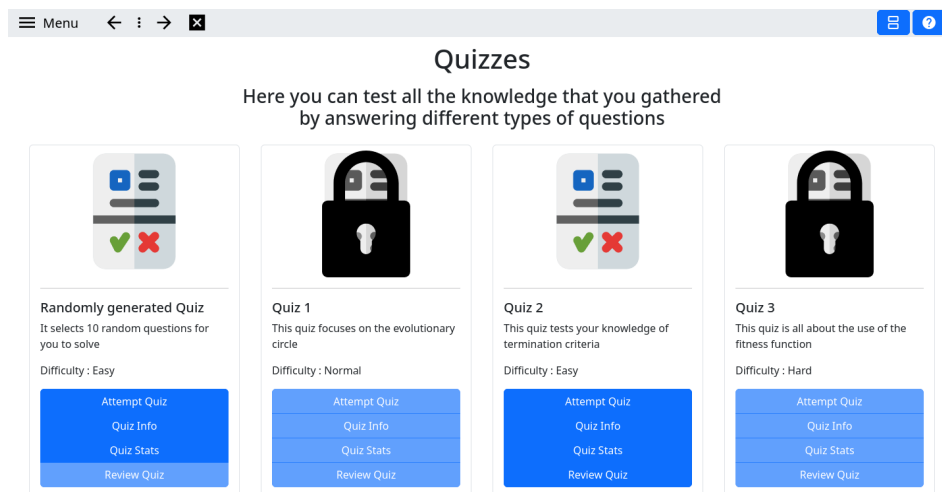
### 5.6 Κουίζ Εφαρμογής

#### 5.6.1 Εισαγωγή

Στη συγκεκριμένη υποενότητα γίνεται η παρουσίαση του τρόπου λειτουργίας των κουίζ που παρέχει η εφαρμογή. Τον τρόπο σύνταξης, τροποποίησης και προσαρμογής τους στις εκάστοτε ανάγκες που καλούνται να καλύψουν.

#### 5.6.2 View εμφάνισης κουίζ (Quizzes)

Τα κουίζ της εφαρμογής εμφανίζονται με τον τρόπο που φαίνεται στο Σχήμα 5.37, και το κάθε ένα παρέχει μια λίστα από διαθέσιμες λειτουργίες. Γίνεται η συνοπτική παρουσίαση αυτών των λειτουργιών, όμως για την πλήρη κατανόηση τους απαιτούνται πληροφορίες που θα εμφανιστούν στις επόμενες υποενότητες.



Σχήμα 5.37: Εμφάνιση view διαθέσιμων κουίζ

Η κάθε κάρτα που αντιπροσωπεύει ένα κουίζ περιέχει έναν τίτλο, μια περιγραφή και μία ένδειξη της εκάστοτε δυσκολίας που παρουσιάζει ως προς την επίλυση του. Κάποιες από αυτές τις κάρτες έχουν ένα λουκέτο και τις επιλογές του απενεργοποιημένες γιατί δεν έχουν ξεκλειδωθεί ακόμα από τον χρήστη. Για κάθε κουίζ ορίζεται το σκορ που απαιτείτε για να ξεκλειδώσει όλες τις δικές του λειτουργίες καθώς και το σκορ που χρειάζεται για να ξεκλειδωθεί το επόμενο στη σειρά κουίζ.

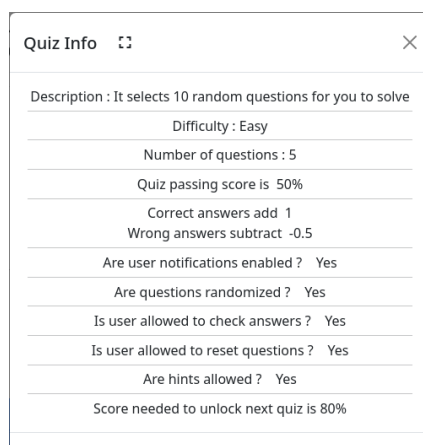
Τώρα η πρώτη επιλογή με όνομα “Attempt Quiz” μιας κάρτας είναι αυτή που κάνει τη μεταβίβαση στο περιβάλλον προσπάθειας ενός κουίζ και θα αναλυθεί σε βάθος στην υποενότητα 5.6.3. Η δεύτερη επιλογή με όνομα “Quiz Info” εμφανίζει με τη χρήση ενός modal (Σχήμα 5.38) όλες τις πληροφορίες που χρειάζεται ο χρήστης να γνωρίζει πριν το άνοιγμα του κουίζ. Οι επιλογές που παρουσιάζονται θα αναλυθούν στην υποενότητα 5.6.4.2. Τώρα η τρίτη επιλογή, που έχει όνομα “Quiz Stats” παρέχει τη δυνατότητα εμφάνισης στατιστικών για το συγκεκριμένο κουίζ.

Πιο συγκεκριμένα εμφανίζει τον αριθμό των επιτυχημένων, αποτυχημένων, και ακυρωμένων προσπαθειών του κουίζ καθώς και των μέσω όρο του αθροίσματος των επιτυχημένων και των αποτυχημένων (Σχήμα 5.39). Εκτός αυτού παρέχει ένα γράφημα με όλες αυτές τις προσπάθειες εμφανιζόμενες με χρονολογική σειρά.

Η τέταρτη και τελευταία επιλογή είναι αυτή με το όνομα “Review Quiz” που επιτρέπει στον χρήστη να εισέλθει σε ένα περιβάλλον που μπορεί να βρει όλες τις ερωτήσεις του κουίζ απαντημένες.

Αυτή η λειτουργία επιτρέπει την πιο διεξοδική μελέτη του κουίζ, δεδομένο φυσικά ότι έχει πετύχει ένα προβιβάσιμο σκορ για να την ξεκλειδώσει.

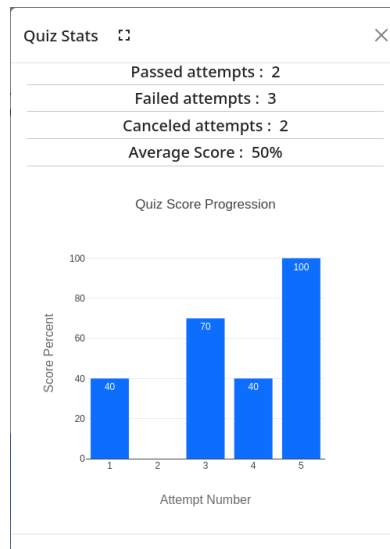
#### 5.6.2.1 Συμπληρωματικές λεπτομέρειες



Σχήμα 5.38: Εμφάνιση modal πληροφοριών κουίζ

Το αρχείο που είναι υπεύθυνο για την καταγραφή των διαθέσιμων κουίζ είναι το quiz\_definitions.json και βρίσκεται στον φάκελο QuizContentFiles, και το αρχείο που διεκπεραιώνει τη δημιουργία των καρτών που ορίζονται στον προαναφερθέν φάκελο είναι το quiz-view.js.

## Κεφάλαιο 5ο:



Σχήμα 5.39: Εμφάνιση modal στατιστικών

Η μορφοποίηση του αρχείου φαίνεται στο Σχήμα 5.40 και ο τρόπος λειτουργίας των πεδίων είναι ο ίδιος που αναφέρεται στην υποενότητα 5.5.2.1.

Ο φάκελος που περιέχει τα κουίζ είναι αυτός με το όνομα Quizzes και η αντιστοίχιση των εγγραφών με το ανάλογο κουίζ είναι ίδια με αυτή των ασκήσεων.

```
{
  "_1": {
    "title": "Randomly generated Quiz",
    "description": "It selects 10 random questions for you to solve",
    "difficulty": "Easy",
    "lockedState": 0
  },
  "_2": {
    "title": "Quiz 1",
    "description": "This quiz focuses on the evolutionary circle",
    "difficulty": "Normal",
    "lockedState": 0
  },
  "_3": {
    "title": "Quiz 2",
    "description": "This quiz tests your knowledge of termination criteria",
    "difficulty": "Easy",
    "lockedState": 1
  },
  "_4": {
    "title": "Quiz 3",
    "description": "This quiz is all about the use of the fitness function",
    "difficulty": "Hard"
  }
}
```

Σχήμα 5.40: Τμήμα αρχείου quiz\_definitions.json

Οι μόνες διαφορές είναι ότι όλα τα πεδία εκτός του lockedState είναι απλά string και δεν επηρεάζουν κάποια λειτουργία, αλλά μπορούν να χρησιμοποιηθούν με όποιον τρόπο επιθυμεί ο καθηγητής.

Το lockedState των κουίζ διαφέρει λίγο με αυτό των ασκήσεων γιατί έχει μια παραπάνω τιμή που μπορεί να δεχτεί.

Όταν το `lockedState` είναι 0 τότε όλες οι λειτουργίες του κουίζ είναι κλειδωμένες και εμφανίζεται το χαρακτηριστικό λουκέτο, εκτός και αν ο χρήστης το έχει λύσει με επιτυχία οπότε αγνοείται οποιαδήποτε μεταγενέστερη αλλαγή του από τον καθηγητή.

Πρέπει να σημειωθεί ότι εάν το κουίζ είναι το πρώτο που εμφανίζεται τότε είναι πάντα ξεκλειδωτο και αγνοεί την τιμή 0 του `lockedState`.

Εάν έχει τιμή 1 είναι πλήρως ξεκλειδωμένο ενώ εάν έχει τιμή 2 έχει κλειδωμένη μόνο την επιλογή “Review Quiz”.

### 5.6.3 View Προσπάθειας κουίζ

Όταν ο χρήστης κάνει κλικ πάνω στο κουμπί “Attempt Quiz” τότε μεταφέρεται σε ένα view που μοιάζει με αυτό του φαίνεται στο Σχήμα 5.41.

Μόλις λάβει μέρος η έναρξη του κουίζ ο χρήστης δεν μπορεί να πλοηγηθεί σε κανένα άλλο view της εφαρμογής μέχρι να λάβει τέλος η προσπάθεια.

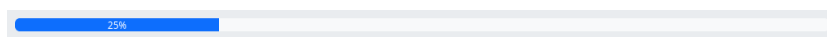
The screenshot displays a quiz interface titled "Randomly generated Quiz". It features a top navigation bar with a "Menu" button and a language selector set to "En". The main content area contains two questions, each with a "Clear Selections" button and a "Check Question" button. Question 1 asks "Which one of these options is correct ?" and "What is the first step in the evolutionary cycle", with radio button options: Selection, Mutation, and Recombination. Question 2 asks "Check the correct answer or answers?" and "Which of these steps is part of the evolutionary cycle?", with checkbox options: Selection, Recombination, and Exchange. At the bottom, there are buttons for "Cancel And Exit" and "Finish Quiz".

Σχήμα 5.41: Εμφάνιση view προσπάθειας κουίζ

Στην πάνω αριστερή μεριά της μπάρας εμφανίζεται ο τίτλος του κουίζ, και κάνοντας κλικ επάνω του θα εμφανιστεί το modal που φαίνεται στο Σχήμα 5.38. Στη μέση της μπάρας εμφανίζεται η γραμμή ποσοστού συμπλήρωσης του κουίζ.

Κάθε φορά που ο χρήστης απαντάει σε μια ερώτηση αυτή η γραμμή μεγαλώνει, και δείχνει το ποσοστό του κουίζ που έχει απαντηθεί (Σχήμα 5.42), ανεξάρτητα εάν η ερωτήσεις είναι απαντημένες σωστά ή όχι.

Η γραμμή δε θα αλλάξει ποσοστό εάν ο χρήστης αλλάξει την επιλογή του για μια ερώτηση γιατί αυτή θεωρείται απαντημένη.



Σχήμα 5.42: Εμφάνιση γραμμής ποσοστού συμπλήρωσης κουίζ

Στην πάνω δεξιά γωνία της μπάρας υπάρχει η επιλογή αλλαγής της γλώσσας των ερωτήσεων καθώς και κάποιων άλλων κομματιών της συγκεκριμένης διεπαφής (οι διαθέσιμες επιλογές είναι τα ελληνικά και τα αγγλικά). Στη μέση της οθόνης βρίσκονται οι ερωτήσεις του κουίζ.

### 5.6.3.1 Ανάλυση ερωτήσεων

Η εφαρμογή παρέχει τη δυνατότητα δημιουργίας τεσσάρων διαφορετικών ειδών ερωτήσεων.

Αρκετές από τις λειτουργίες του είναι παρόμοιες οπότε γίνεται η πλήρη αναφορά τους πάνω σε μια ερώτηση και όλες οι διαφορές θα αναλυθούν ξεχωριστά. Η ανάλυση γίνεται με τον πιο απλό τύπο ερώτησης, αυτόν του σωστού-λάθους.

Μια τέτοια ερώτηση εμφανίζεται στο Σχήμα 5.43.

Στην πάνω αριστερή γωνία αναγράφεται ο αριθμός της ερώτησης ανάλογα με τη σειρά εμφάνισης της.

Αντικριστά του αριθμού ερώτησης υπάρχει το κουμπί Hint, το οποίο παρέχει μια μικρή βοήθεια στον χρήστη για την ερώτηση.

Η ύπαρξη του hint δεν είναι αναγκαστική, και σε περίπτωση που δεν υπάρχει δηλωμένο στην ερώτηση τότε στη θέση του εμφανίζεται ένα ενημερωτικό κείμενο που αναφέρει την απουσία του.

Εκτός αυτού τα hints μπορούν να απενεργοποιηθούν τελείως από τον καθηγητή, που αυτό απενεργοποιεί το κουμπί.

A screenshot of a quiz question interface. At the top, it says 'Question 3' next to a globe icon. The question text is 'Is this question False or True ?' followed by 'Does selection appear before replacement ?'. There are two radio buttons: 'False' and 'True'. At the bottom, there are two buttons: 'Clear Selections' with an 'X' icon and 'Check Question' with a checkmark icon.

Σχήμα 5.43: Εμφάνιση ερώτησης σωστού-λάθους

Στη μέση της ερώτησης υπάρχουν δύο κουμπιά επιλογής του σωστού-λάθους (True-False).

Μόλις γίνει το πάτημα ενός από τα δυο τότε το χρώμα της ερώτησης αλλάζει σε μπλε για να δείξει ότι ο χρήστης έχει αλληλεπιδράσει με την ερώτηση.

Τώρα στο κάτω δεξιό μέρος της ερώτησης υπάρχουν δύο κουμπιά το “Clear Selections” και το “Check Question”.

Το πρώτο επαναφέρει την ερώτηση στην αρχική της κατάσταση, δηλαδή αναιρεί όποια επιλογή υπήρχε, αφαιρεί το μπλε χρώμα της ερώτησης και προκαλεί αλλαγή στη γραμμή προόδου μειώνοντας τη.

Το δεύτερο επιτρέπει τον έλεγχο της ορθότητας μιας απαντημένης ερώτησης (εάν δεν έχει απαντηθεί ο χρήστης ενημερώνεται με ένα toast). Μόλις γίνει αυτό η ερώτηση κλειδώνει και δεν επιτρέπονται περαιτέρω αλλαγές της.

Ενδέχεται να έχουν απενεργοποιηθεί οι λειτουργίες ελέγχου της ερώτησης ή και της επαναφοράς της από τον καθηγητή. Σε αυτή την περίπτωση τα κουμπιά εξαφανίζονται από την ερώτηση.

Εάν η απάντηση είναι σωστή τότε το χρώμα της ερώτησης γίνεται πράσινο όπως φαίνεται στο Σχήμα 5.44, εάν είναι λάθος τότε γίνεται κόκκινο.

Αν για οποιοδήποτε λόγο μια ερώτηση παραμένει αναπάντητη στο τέλος του κουίζ τότε αυτή αποκτά χρώμα ανοιχτό γαλάζιο.

Όταν λάβει μέρος ο έλεγχος μιας ερώτησης τότε εκτός από την αλλαγή στο χρώμα γίνεται και η εμφάνιση του πεδίου επεξήγησης της.

Σε αυτό το πεδίο, εάν είναι ορισμένο μέσα στην ερώτηση εμφανίζεται μια σύντομη επεξήγηση για τον λόγο που η ερώτηση έχει τη συγκεκριμένη απάντηση (εάν δεν είναι ορισμένο εμφανίζεται ενημερωτικό κείμενο). Εκτός αυτού μπορεί και να περιέχει και μια αναφορά στο κείμενο στο οποίο αναφέρεται η ερώτηση.

Έτσι κάνοντας κλικ πάνω στο “Explanation Reference” ή το αντίστοιχο κείμενο που εμφανίζεται με την αλλαγή της γλώσσας στα ελληνικά, ο χρήστης μεταφέρεται στη συγκεκριμένη σελίδα του έγγραφου ύλης που υπάρχει η αναφορά. Ανάλογα με τον τρόπο σύνταξης αυτής της αναφοράς μπορεί να γίνει και η επισήμανση συγκεκριμένου κομματιού κειμένου μέσα στο έγγραφο.

Σχήμα 5.44: Εμφάνιση μιας σωστά απαντημένης ερώτησης σωστού-λάθους

Οι υπόλοιποι τύποι ερωτήσεων διαφέρουν ως προς τα κουμπιά επιλογής που περιέχουν και την τρόπο που γίνεται ο έλεγχος της ορθότητας τους.

Οι ερωτήσεις πολλαπλής επιλογής μοιάζουν με αυτές του σωστού-λάθους απλά έχουν περισσότερες από δύο διαθέσιμες επιλογές.

Έπειτα υπάρχουν και οι ερωτήσεις με κουμπιά τσεκαρίσματος (check box), στις οποίες επιτρέπεται η επιλογή πολλαπλών απαντήσεων.

## Κεφάλαιο 5ο:

Τέλος, οι ερωτήσεις ελεύθερου κειμένου επιτρέπουν την εισαγωγή κειμένου ως απάντηση σε μια ερώτηση.

Για την καλύτερη ευχρηστία, κάθε απάντηση που δίνεται προσαρμόζεται κατάλληλα έτσι ώστε να είναι όσο το δυνατόν πιο ανεκτική στα τυχόν ορθογραφικά λάθη (πχ γίνεται η αφαίρεση των τόνων, κενών, προσαρμογή κεφαλαίων και μικρών γραμμάτων).

### 5.6.3.2 Τρόποι εμφάνισης ερωτήσεων

Η εφαρμογή παρέχει τη δυνατότητα αλλαγής του τρόπου εμφάνισης των ερωτήσεων ανάλογα με τις απαιτήσεις του καθηγητή. Το πρώτο είδος προβολής είναι αυτό που φαίνεται στο Σχήμα 5.41, που απλά γίνεται η παράθεση όλων των ερωτήσεων.

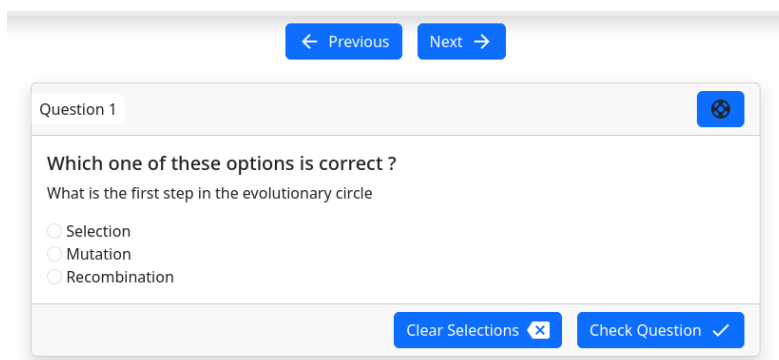
Το δεύτερο είδος φαίνεται στο Σχήμα 5.45 και παρουσιάζει μόνο μια ερώτηση κάθε φορά και είναι εύχρηστο όταν ο αριθμός των ερωτήσεων είναι σχετικά μικρός.

Έχει δύο κουμπιά, το “Previous” και το “Next” τα οποία είναι υπεύθυνα για την πλοήγηση από τη μια ερώτηση στην άλλη.

Εάν ο χρήστης φτάσει στην τελευταία ερώτηση και προσπαθήσει να πλοηγηθεί στην επόμενη, θα επιστρέψει πάλι στην αρχική.

Παρόμοια συμπεριφορά έχει και η μετάβαση από την πρώτη ερώτηση στην προηγούμενη (πηγαίνει στην τελευταία).

Έχει δηλαδή τη συμπεριφορά ενός συμβατικού στοιχείου διεπαφής τύπου καρουζέλ.



Σχήμα 5.45: Είδος προβολής σε στιλ καρουζέλ

Το τρίτο και τελευταίο είδος προβολής είναι σχετικά πιο περίπλοκο σε σχέση με τα προηγούμενα και προορίζεται όταν ο αριθμός των ερωτήσεων προς προβολή είναι μεγάλος (Σχήμα 5.46).

Σε αυτό το είδος προβολής στο κάτω μεσαίο μέρος της οθόνης εμφανίζεται μια σειρά από αριθμημένα κουμπιά. Το πόσα κουμπιά εμφανίζονται εξαρτάται από τον αριθμό διαίρεσης που επιλέγει ο καθηγητής για τις ερωτήσεις.

Πχ. Εάν υπάρχουν τριάντα ερωτήσεις μέσα σε ένα κουίζ και διαλέξει να εμφανίζει κάθε φορά μόνο πέντε ερωτήσεις, τότε θα υπάρχουν έξι διαθέσιμα κουμπιά, όπου το κάθε ένα αλλάζει το γκρουπ των εμφανιζόμενων ερωτήσεων.

Question 1

Which one of these options is correct ?  
What is the first step in the evolutionary cycle

☐ Selection  
☐ Mutation  
☐ Recombination

Clear Selections ✕ Check Question ✓

Question 2

Check the correct answer or answers?  
Which of these steps are part of the evolutionary cycle?

☐ Selection  
☐ Recombination  
☐ Exchange

Clear Selections ✕ Check Question ✓

1 2 3 4

Σχήμα 5.46: Είδος προβολής πολλαπλών σελίδων

### 5.6.3.3 Χρήση χρονομέτρου

Για την αύξηση της δυσκολίας ενός κουίζ ο καθηγητής μπορεί να κάνει χρήση της λειτουργίας χρονομέτρου.

Όταν αυτή η λειτουργία είναι ενεργή στην πάνω δεξιά γωνία που της μπάρα του view που έχει τον τίτλο του κουίζ εμφανίζεται ο διαθέσιμος χρόνος ολοκλήρωσης της προσπάθειας του κουίζ όπως φαίνεται στο Σχήμα 5.47.

En ▾ Time Remaining -- 04 : 47

Σχήμα 5.47: Εμφάνιση διαθέσιμου χρόνου κουίζ

Μόλις λήξει ο διαθέσιμος χρόνος τότε το κουίζ περνάει αυτόματα στη φάση έλεγχου χωρίς να μπορεί ο χρήστης να κάνει κάποια αλλαγή στις απαντήσεις που έδωσε. Πριν το πέρας του χρόνου υπάρχει η δυνατότητα ενημέρωσης του χρήστη για τον διαθέσιμο χρόνο σε δύο φάσεις.

Πρώτα για το πέρας του μισού χρόνου (πχ εάν το κουίζ ορίζει έξι λεπτά για την ολοκλήρωση τότε ο χρήστης λαμβάνει ένα ενημερωτικό toast στα τρία λεπτά), και έπειτα όταν μένει μόνο ένα λεπτό διαθέσιμο, ο διαθέσιμος χρόνος αναβοσβήνει με ένα κόκκινο χρώμα για να τραβήξει την προσοχή του χρήστη.

Κεφάλαιο 5ο:

#### 5.6.3.4 Ολοκλήρωση του κουίζ

Στο κάτω μέρος του view που εμφανίζεται στο Σχήμα 5.41 υπάρχουν δύο κουμπιά, το πρώτο είναι το “Cancel End Exit” και το δεύτερο είναι το “Finish Quiz”.

Με το πρώτο ο χρήστης μπορεί να ακυρώσει την τρέχουσα προσπάθεια του εφόσον το επιβεβαιώσει πατώντας το “Yes” στο modal που εμφανίζεται.

Με την ακύρωση του κουίζ ο χρήστης επιστρέφει στο προηγούμενο ανοιχτό view που είχε φορτωμένο η εφαρμογή και δεν υπάρχει καμία καταμέτρηση του βαθμού του στα στατιστικά που κρατούνται για τα κουίζ. Το δεύτερο κουμπί ο χρήστης μπορεί να ολοκληρώσει την προσπάθεια του.

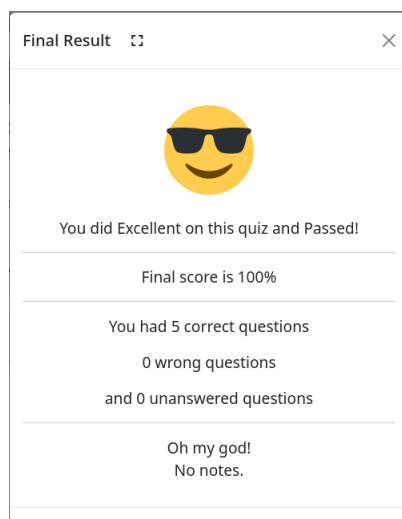
Μόλις γίνει το πάτημα αυτού του κουμπιού εμφανίζεται ένα modal επιβεβαίωσης, και σε περίπτωση που έχει αναπάντητες ερωτήσεις του προτείνει να της ρίξει μια καλύτερη ματιά.

Εάν ο χρήστης είναι σίγουρος πως επιθυμεί να τελειώσει το κουίζ, τότε πατώντας το “Yes” θα εμφανιστεί μια οθόνη που θα μοιάζει με αυτή στο Σχήμα 5.48.

Με τη χρήση αυτού το modal ο χρήστης μπορεί να δει πως τα πήγε στο κουίζ, δηλαδή να δει με μια ματιά αν πήρε τον απαραίτητο βαθμό για να θεωρηθεί η προσπάθεια του επιτυχής, το σκορ του σαν ποσοστό καθώς και μια αρίθμηση των σωστών, λάθους και αναπάντητων ερωτήσεων που είχε.

Εκτός αυτού εμφανίζεται και μια φαταούλα καθώς και ένα μήνυμα που σκοπό έχει να ενθαρρύνει τον χρήστη να τα πάει καλύτερα ή να των επιβραβεύσει αν τα πήγα καλά.

Το μήνυμα που εμφανίζεται είναι κάθε φορά διαφορετικό, οπότε ο χρήστης ενθαρρύνεται να επαναλάβει το κουίζ για να δει όλα τα διαθέσιμα μηνύματα που εμφανίζονται.



Σχήμα 5.48: Εμφάνιση modal τελικού αποτελέσματος κουίζ

Μόλις ο χρήστης κλείσει το modal είναι σε θέση να δει όλες του τις απαντήσεις, ποιες είναι σωστές και ποιες είναι λάθος.

Με την ολοκλήρωση ενός κουίζ η εφαρμογή προβαίνει σε κάποιες ενέργειες ανάλογα με το σκορ που κατάφερε να αποκτήσει ο χρήστης. Το κάθε κουίζ έχει δύο ειδών σκορ, αυτό του ξεκλειδώματος του ίδιου του κουίζ και το σκορ ξεκλειδώματος του επόμενου στη σειρά κουίζ.

Αυτά τα δύο σκορ μπορούν να είναι διαφορετικά μεταξύ τους, επιβραβεύοντας έτσι τον χρήστη για μια καλή προσπάθεια αλλά και παράλληλα προτρέποντας τον στο να τα πάει καλύτερα για να ξεκλειδώσει το επόμενο (δεδομένου ότι ο πρώτος βαθμός είναι μικρότερος από τον δεύτερο).

Με το ξεκλείδωμα του ίδιου του κουίζ εννοείται η ενεργοποίηση της τελευταίας λειτουργίας που φαίνεται στο Σχήμα 5.37, δηλαδή τη λειτουργία “Review Quiz”.

Αυτή η λειτουργία ουσιαστικά χρησιμοποιεί το ήδη υπάρχον περιβάλλον προσπάθειας ενός κουίζ που φαίνεται στο Σχήμα 5.41 (κάνοντας τις κατάλληλες προσαρμογές όπως την αφαίρεση του χρονικού περιθωρίου εάν υπάρχει και τη διαγραφή των κουμπιών ακύρωσης και ολοκλήρωσης του κουίζ) σε συνδυασμό με την αυτόματη λύση των ερωτήσεων για να παρέχει τη δυνατότητα σε ένα χρήστη που πείρε έναν προβιβάσιμο βαθμό να δει τις λύσεις του κουίζ και να διορθώσει τυχόν λάθη στην επόμενη προσπάθεια.

## 5.6.4 Συμπληρωματικές λεπτομέρειες

### 5.6.4.1 Αρχείο βάσης ερωτήσεων

Ο κύριος τρόπος που δουλεύουν τα κουίζ είναι χάρη στην ύπαρξη ενός αρχείου με όνομα `quiz_master_file.json` που βρίσκεται στον φάκελο `QuizContentFiles/Quizzes`.

Αυτό το αρχείο είναι σαν μια βάση δεδομένων που περιέχει όλες τις ερωτήσεις που μπορούν να εμφανιστούν σε κάποιο κουίζ. Προτού γίνει η παρουσία του τρόπου σύνταξης ενός κουίζ θα αναλυθεί ο τρόπος σύνταξης των ερωτήσεων που μπορούν να το απαρτίσουν. Ως γενική δομή χωρίζει όλες τις ερωτήσεις σε κατηγορίες, και η κάθε μία κατηγορία μπορεί να έχει έναν μεταβλητό αριθμό ερωτήσεων.

Η ονομασία των κατηγοριών είναι αυθαίρετη και μπορεί να αλλάξει για να καλύπτει καλύτερα τις ανάγκες του καθηγητή. Όσον αφορά τις ερωτήσεις αυτές έχουν πολλά πεδία τα οποία είναι κοινά μεταξύ τους, ενώ κάποια άλλα είναι διαφορετικά.

Τα κοινά πεδία που έχουν είναι αυτά που παρουσιάζονται παρακάτω και τα περισσότερα από αυτά είναι αντικείμενα που έχουν δύο εγγραφές, μια `el` και μια `en` που παρέχουν τις μεταφράσεις για την αλλαγή της γλώσσας των κουίζ στα ελληνικά και στα αγγλικά αντίστοιχα (είναι αντικείμενο εάν εμφανίζεται το `obj` στην παρένθεση).

Εκτός αυτού στα πεδία αναγράφεται εάν αυτά είναι προαιρετικά η όχι:

- `title` (υποχρεωτικό | `obj`) : ορίζει τον τίτλο που εμφανίζεται στην εκάστοτε ερώτηση.
- `hint` (προαιρετικό | `obj`) : ορίζει ένα βοηθητικό μήνυμα για την ερώτηση.
- `explanation` (προαιρετικό | `obj`) : ορίζει την επεξήγηση της ερώτησης.
- `explanation_reference` (προαιρετικό) : ορίζει σε πιο έγγραφο ύλης βρίσκεται η επεξήγηση της ερώτησης. Ο τρόπος που το κάνει αυτό είναι με τον ορισμό ενός πίνακα που έχει τα εξής πεδία :

## Κεφάλαιο 5ο:

1. Αριθμός εγγράφου : είναι ο αριθμός που έχει το id που αναλύεται στην ενότητα 5.4.2.1.
2. Κείμενο προς επισήμανση : είναι το κείμενο που επιχειρεί να βρει η εφαρμογή και να επισημάνει αλλάζοντας το χρώμα του σε κίτρινο.
3. Αριθμός σελίδας : η σελίδα στην οποία βρίσκεται το κείμενο προς επισήμανση.
4. Αριθμός επιλογή κειμένου : στην περίπτωση που το κείμενο προς επισήμανση εμφανίζεται παραπάνω από μια φορά μπορεί ο συγγραφέας της ερώτησης να διαλέξει πια εμφάνιση του επιθυμεί να επισημάνει.

```
"q1": {
  "title": {
    "el": "Η επιλογή εμφανίζεται πριν την αντικατάσταση ?",
    "en": "Does selection appear before replacement ?"
  },
  "hint": {
    "el": "Σκέψου την σειρά που πρέπει να λάβουν μέρος τα βήματα ενός πληθυσμού",
    "en": "Think about the order that a population's steps should take"
  },
  "explanation": {
    "el": "Ναι γιατί πρέπει πρώτα να λάβει μέρος η επιλογή των αρχικών μελών του πληθυσμού πριν μπορέσει να γίνει η αντικατάσταση τους με τους νέους",
    "en": "Yes, because the selection of the original members of the population must first take place before they can be replaced with the new ones"
  },
  "explanation_reference": [1, "The Evolutionary Cycle", 3, 1],
  "type": "true_false",
  "answer": 1
},
```

Σχήμα 5.49: Εμφάνιση σύνταξης μιας ερώτησης σωστού-λάθους

Εκτός αυτών των πεδίων η κάθε ερώτηση πρέπει να έχει και ένα μοναδικό id, παρόμοια με αυτά που έχουν ήδη παρουσιαστεί (το id μπορεί να είναι κάτι αυθαίρετο ανάλογα με τις ανάγκες του καθηγητή).

Για να γίνουν πιο κατανοητές όλες αυτές οι έννοιες στο Σχήμα 5.49 φαίνεται πως είναι μια πλήρης (έχει όλα τα πεδία συμπληρωμένα) ερώτηση σωστού-λάθους με id “q1”.

Υπάρχουν κάποια πεδία που μοιράζονται ανάμεσα στις ερωτήσεις αλλά έχουν διαφορετικές τιμές και σύνταξη ανάλογα με την ερώτηση, για αυτόν τον λόγο γίνεται η παρουσίαση τους ξεχωριστά για κάθε ένα τύπο.

Οι ερωτήσεις σωστού λάθους έχουν τα ακόλουθα έξτρα πεδία :

- type: πρέπει να είναι πάντα το string “true\_false” όπως φαίνεται και στο Σχήμα 5.49.

answer : ορίζει την απάντηση και πρέπει να είναι ο αριθμός 1 ή 0 για το σωστό και το λάθος αντίστοιχα.

Οι ερωτήσεις πολλαπλής επιλογής έχουν τα ακόλουθα έξτρα πεδία :

- type: πρέπει να είναι πάντα το string “multiple\_choice”.
- options : ορίζει τις διαθέσιμες επιλογές απάντησης και είναι ένα αντικείμενο που έχει δύο εγγραφές, μια el και μια en για να μπορεί η ερώτηση να μεταφράζεται από τη μια γλώσσα στην άλλη. Η κάθε μια εγγραφή περιέχει έναν πίνακα από strings που ορίζουν τις επιλογές για την κάθε γλώσσα.

- **answer**: ορίζει την απάντηση και πρέπει να είναι ένα πίνακας που έχει μόνο έναν αριθμό που λειτουργεί ως δείκτης πάνω στους πίνακες που έχει το **options** (η αρίθμηση του δείκτη των στοιχείων του πίνακα ξεκινά από το 0).

Οι ερωτήσεις με κουμπιά τσεκαρίσματος έχουν τα ακόλουθα έξτρα πεδία (Σχήμα 5.51) :

- **type**: πρέπει να είναι πάντα το string “checkbox\_select”.
- **options** : ορίζει τις διαθέσιμες επιλογές απάντησης και είναι ένα αντικείμενο που έχει δύο εγγραφές, μια **el** και μια **en** για να μπορεί η ερώτηση να μεταφράζεται από τη μια γλώσσα στην άλλη. Η κάθε μια εγγραφή περιέχει έναν πίνακα από strings που ορίζουν τις επιλογές για την κάθε γλώσσα.
- **answer**: ορίζει την ή τις απαντήσεις και πρέπει να είναι ένας πίνακας που μπορεί να περιέχει έναν αριθμό ή και παραπάνω, εφόσον αυτοί δεν ξεπερνούν των αριθμό όλων των επιλογών και λειτουργούν ως δείκτες πάνω στους πίνακες που έχει το **options** (η αρίθμηση των δεικτών των στοιχείων του πίνακα ξεκινά από το 0). Πρέπει να υπάρχει μόνο ένας δείκτης για το κάθε στοιχείο του πίνακα.

Οι ερωτήσεις ελεύθερου κειμένου έχουν τα ακόλουθα έξτρα πεδία :

- **type**: πρέπει να είναι πάντα το string “input”.
- **answer** : ορίζει τη σωστή απάντηση και είναι ένα αντικείμενο που έχει δύο εγγραφές, μια **el** και μια **en** για να μπορεί η ερώτηση να μεταφράζεται από τη μια γλώσσα στην άλλη. Η κάθε μια εγγραφή περιέχει ένα string που ορίζει την απάντηση στην εκάστοτε γλώσσα.

#### 5.6.4.2 Δομή αρχείου ορισμού κουίζ

Ο τρόπος που ορίζεται ένα κουίζ είναι ο εξής.

Μέσα στον φάκελο QuizContentFiles/Quizzes εκτός του αρχείου που περιέχει όλες τις ερωτήσεις (5.6.4.1) υπάρχουν και τα αρχεία που έχουν ονόματα όπως [quiz\_[νούμερο].json].

Αυτά είναι τα κουίζ και η εσωτερική τους δομή έχει την εξής εμφάνιση που φαίνεται στο Σχήμα 5.50

Ένα τέτοιο αρχείο κουίζ χωρίζεται σε δύο μέλη, το **info** και το **configuration**.

```
{
  "info": {
    "title": "Quiz 1",
    "description": "This is a mix of selection, initialization and evolutionary circle questions",
    "difficulty": "Normal"
  },
  "configuration": {
    "show": "all",
    "split_number": 4,
    "passing_score": 50,
    "quiz_unlock": 80,
    "negative_scoring": [0.5, 1],
    "show_hints": true,
    "allow_answer_checking": true,
    "allow_question_reset": true,
    "random_order": false,
    "timer": 0,
    "notify_user": true,
    "questions": [
      "evolutionary_circle",
      {
        "selection": ["q2", "q1"]
      },
      "initialization"
    ]
  }
}
```

Σχήμα 5.50: Εμφάνιση δομής ενός κουίζ

## Κεφάλαιο 5ο:

Τα πεδία του info είναι αυτά που εμφανίζονται μέσα στις κάρτες που παρουσιάζουν τα διαθέσιμα κουίζ στο Σχήμα 5.37 και έχουν τα εξής πεδία :

- title : ορίζει τον τίτλο του κουίζ.
- description : ορίζει την περιγραφή του κουίζ.
- difficulty : ορίζει τη δυσκολία που θεωρεί ότι έχει από τον καθηγητή.

```
"options": {  
  "el": ["Επιλογή", "Ανασυνδυασμός", "Ανταλλαγή"],  
  "en": ["Selection", "Recombination", "Exchange"]  
},  
"type": "checkbox_select",  
"answer": [0, 1]
```

*Σχήμα 5.51: Έξτρα πεδία ερώτησης με κουμπιά τσεκαρίσματος*

Όλα τα πεδία είναι string και μπορούν να χρησιμοποιηθούν για οποιαδήποτε άλλο σκοπό παράθεσης πληροφοριών για ένα κουίζ, πέρα από την προκαθορισμένη λειτουργία τους.

Το δεύτερο σκέλος του αρχείου, δηλαδή το configuration είναι αυτό που ορίζει όντος το κουίζ και έχει τα εξής πεδία (στην παρένθεση ορίζεται τι τύπου τιμής είναι το κάθε πεδίο) :

- show (string) : ορίζει τον τρόπο εμφάνισης ενός κουίζ όπως παρουσιάστηκε στο (5.6.3.2) και μπορεί να έχει τις ακόλουθες τιμές :
  1. all : εμφανίζει όλες τις ερωτήσεις σε μία οθόνη όπως φαίνεται στο Σχήμα 5.41.
  2. one : εμφανίζει μια ερώτηση κάθε φορά όπως φαίνεται στο Σχήμα 5.45.
  3. split : εμφανίζει έναν προκαθορισμένο αριθμό ερωτήσεων όπως φαίνεται Σχήμα 5.46. Το πόσες ερωτήσεις θα εμφανίσει σε κάθε φορά ορίζεται από το split\_number.
- split\_number (number) : ορίζει πόσες ερωτήσεις θα εμφανίζονται κάθε φορά όταν το πεδίο show έχει τιμή split. Εάν ο αριθμός που δηλωθεί είναι 1 τότε η εφαρμογή αλλάζει τον τρόπο προβολής στο one και εάν είναι μεγαλύτερος από τον συνολικό αριθμό διαθέσιμων ερωτήσεων μέσα στο κουίζ τότε ο τρόπος προβολής αλλάζει σε all. Το συγκεκριμένο πεδίο αγνοείται όταν το show έχει τιμές all και one.
- passing\_score (number) : ορίζει το σκορ που απαιτείται για να ξεκλειδωθεί το ίδιο το κουίζ. Πρέπει να είναι τιμή από 0 έως 100.
- quiz\_unlock : ορίζει το σκορ που απαιτείται για να ξεκλειδωθεί η το επόμενο κουίζ στη σειρά. Πρέπει να είναι τιμή από 0 έως 100.
- negative\_scoring (array): ορίζει τον τρόπο που καθορίζεται η τελική βαθμολογία. Είναι ένας πίνακας με δύο αριθμούς, ο πρώτος ορίζει πόσες μονάδες αφαιρεί μια λάθος απάντηση και ο δεύτερος αριθμός ορίζει το πόσες μονάδες δίνει μια σωστή.
- show\_hints (boolean) : ορίζει το εάν επιτρέπεται η εμφάνιση βοηθητικών μηνυμάτων της κάθε ερώτησης. Εάν είναι true τότε επιτρέπεται, ενώ αν είναι false απαγορεύεται.

- `allow_answer_checking` (boolean) : ορίζει το εάν επιτρέπεται ο έλεγχος των ερωτήσεων πριν τη λήξη του κουίζ. Εάν είναι `true` τότε επιτρέπεται, ενώ αν είναι `false` απαγορεύεται.
- `allow_question_reset` (boolean) : ορίζει το εάν επιτρέπεται η επαναφορά της ερώτησης στην αρχική κατάσταση. Εάν είναι `true` τότε επιτρέπεται, ενώ αν είναι `false` απαγορεύεται. Πρέπει να σημειωθεί ότι η χρήση αυτής της επιλογής ενδέχεται να κάνει το κουίζ πολύ δύσκολο οπότε πρέπει η χρήση της να είναι φειδωλή.
- `random_order` (boolean) : ορίζει το εάν κάθε φορά που ανοίγει το κουίζ οι ερωτήσεις θα εμφανίζονται με τυχαία σειρά. Εάν είναι `true` τότε εμφανίζονται με τυχαία σειρά, ενώ αν είναι `false` δεν εμφανίζονται.
- `timer` (number) : ορίζει το εάν θα υπάρχει χρονόμετρο στο κουίζ. Μπορεί να είναι οποιαδήποτε θετικός ακέραιος αριθμός που ορίζει τα διαθέσιμα λεπτά προσπάθειας. Με τη χρήση του αριθμού 0 γίνεται η απενεργοποίηση του χρονόμετρου.
- `notify_user` (boolean) : ορίζει το εάν είναι ενεργή η λειτουργία ειδοποίησης του χρήστη για το πέρας του διαθέσιμου χρόνου όταν είναι ενεργό το χρονόμετρο.
- `questions` (array ή number) : ορίζει ποιες ερωτήσεις θα έχει το κουίζ. Προηγουμένως αναφέρθηκε (5.6.4.1) ότι οι ερωτήσεις χωρίζονται σε κατηγορίες για την καλύτερη οργάνωση τους στο αρχείο `quiz_master_file.json`. Εδώ λοιπόν γίνεται η χρήση αυτών των κατηγοριών. Πχ Εάν είχαμε τρεις κατηγορίες `c1`, `c2`, `c3` τότε θα μπορούσα με να συμπεριλάβουμε όλες τις ερωτήσεις των κατηγοριών που θέλουμε βάζοντας ως τιμή στο πεδίο `["c2","c3"]`. Η σειρά μπορεί να είναι αυθαίρετη, δεν υπάρχει ανάγκη χρήσης όλων των κατηγοριών και σε περίπτωση που γίνει κάποιο λάθος στην παράθεση των κατηγοριών η εφαρμογή το αγνοεί και επιχειρεί να διαβάσει την επόμενη. Εάν κατά αυτή τη διαδικασία δεν μπορέσει να δημιουργήσει το κουίζ τότε ειδοποιεί τον χρήστη με ένα `toast`. Εκτός αυτού μπορεί να γίνει και η παράθεση συγκεκριμένων ερωτήσεων από μια κατηγορία με την εξής σύνταξη `{ "c1" : [ "q1", "q2" ] }`, όπου το `c1` είναι η κατηγορία και μέσα της ορίζεται ένας πίνακας με τα `id` των ερωτήσεων σαν `string` που θέλουμε να χρησιμοποιήσουμε. Πρέπει να σημειωθεί ότι μπορεί να γίνει συνδυασμός των δύο αυτών μεθόδων παράθεσης ερωτήσεων. Τέλος, μπορεί στο πεδίο απλά να γίνει η παράθεση ενός θετικού μη μηδενικού ακεραίου για τη δημιουργία ενός κουίζ με τύχες ερωτήσεις από όλες τις κατηγορίες.

#### 5.6.4.3 Συνοπτική παράθεση αρχείων υλοποίησης

Όλα τα αρχεία που έχουν να κάνουν με τη διεκπεραίωση των προαναφερθέντων λειτουργιών βρίσκονται μέσα στον φάκελο `QuizManager`.

Το αρχείο `quiz-manager.js` είναι υπεύθυνο για τη δημιουργία των κουίζ από τα αρχεία ορισμού που εξετάστηκαν στην προηγούμενη υποενότητα, για την αποθήκευση των `score` και των στατιστικών και τη διαχείριση των κουίζ ως προς τα πια είναι κλειδωμένα και ποια πρέπει να ξεκλειδωθούν με την ολοκλήρωση ενός κουίζ.

Το αρχείο `quiz-question-components.js` είναι αυτό που υλοποιεί όλους του τύπους ερωτήσεων που έχει διαθέσιμους η εφαρμογή και κάνει εκτεταμένη χρήση της ιδέας της προγραμματιστικής κληρονομικότητας, καθώς όλες βασίζονται σε ένα γενικό πρότυπο ερώτησης που παρέχει όλες τις βασικές λειτουργίες που έχουν όλες τους.

## Κεφάλαιο 5ο:

Έπειτα γίνεται η επαύξηση αυτού του προτύπου με καινούριες λειτουργίες που είναι κατάλληλα προσαρμοσμένες για την κάθε ερώτηση.

Με αυτόν τον τρόπο μπορεί να μειωθεί σημαντικά το ποσοστό κώδικα που απαιτείτε για την υλοποίηση τους.

Στο αρχείο `quiz-viewer-component.js` υλοποιείται το πιο κύριο κομμάτι τον κουίζ που είναι το `view` προσπάθειας (Σχήμα 5.41). Είναι αυτό που ενώνει όλα σχεδόν όλα τα υπόλοιπα κομμάτια αυτού του συστήματος.

Το αρχείο `quiz-review-component.js` είναι αυτό που παρέχει τη δυνατότητα ανασκόπησης ενός ξεκλειδωμένου κουίζ.

Ο τρόπος που λειτουργεί είναι με το να επεκτείνει το `quiz-viewer-component` αλλάζοντας τον τρόπο λειτουργίας του έτσι ώστε ανεξάρτητα με τις ρυθμίσεις που μπορεί να έχει ένα κουίζ, να μπορεί να τις αγνοεί και να εφαρμόζει του δικούς του κανόνες εμφάνισης και κάνοντας χρήση της μην προσβάσιμη από τον χρήστη δυνατότητας αυτόματης απάντησης των ερωτήσεων.

Εκτός αυτών των αρχείων υπάρχουν και δύο φάκελοι οι οποίοι περιέχουν αρχεία τα οποία υλοποιούν πολλές από τις λειτουργίες που χρησιμοποιούνται από το `quiz-viewer-component`.

Μέσα στον φάκελο `Renderers` υπάρχουν τρία αρχεία τα οποία υλοποιούν τους διαφορετικούς τρόπους παρουσίασης των ερωτήσεων στα κουίζ.

Το αρχείο `render-all.js` κάνει την εμφάνιση όλων των ερωτήσεων, ενώ το αρχείο `render-split.js` χρησιμοποιεί το `render-all.js` εσωτερικά απλά έχει έναν αλγόριθμο που αποφασίζει ποιες ερωτήσεις να κρύψει από τον χρήστη, σε συνδυασμό με την προσθήκη των κουμπιών πλοήγησης ανάμεσα στις ερωτήσεις.

Το αρχείο `render-one.js` εμφανίζει μια ερώτηση τη φορά και κάνει χρήση του `carrousel component` που παρέχεται από την `Bootstrap`. Μέσα στον φάκελο `Utils` υπάρχουν τα αρχεία που έχουν να κάνουν με τις βοηθητικές λειτουργίες που υπάρχουν για τα κουίζ.

Το αρχείο `final-results.js` είναι υπεύθυνο για την υλοποίηση του `modal` που εμφανίζεται όταν ο χρήστης τελειώσει με την προσπάθεια ενός κουίζ (Σχήμα 5.48).

Το αρχείο `info.js` υλοποιεί το `modal` που εμφανίζεται όταν ο χρήστης ζητά πληροφορίες για ένα κουίζ (Σχήμα 5.38).

Αυτό που κάνει ουσιαστικά είναι να μετατρέπει τα πεδία ορισμού (Σχήμα 5.50) του κουίζ σε κάτι που είναι εύκολα κατανοητό.

Το αρχείο `stats.js` είναι αυτό που σε συνεργασία με τον `quiz-manager` εμφανίζει όλα τα στατιστικά στοιχεία που είναι διαθέσιμα για τα κουίζ (Σχήμα 5.39).

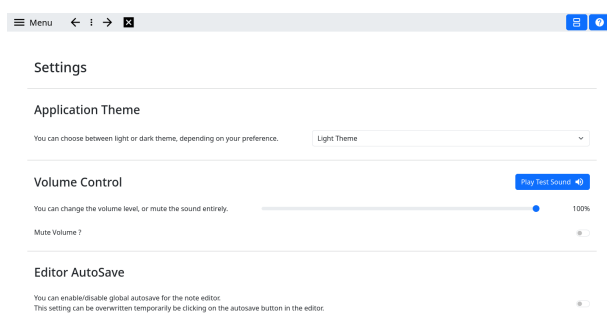
Ο τρόπος που κάνει την εμφάνιση των γραφημάτων είναι με τη χρήση ενός αρχείου που βρίσκεται στον φάκελο `Plotters` (ο φάκελος βρίσκεται εκτός του φακέλου των κουίζ) και έχει όνομα `quiz-stats-plotter.js` και χρησιμοποιεί τη βιβλιοθήκη `Plotly.js` για τη δημιουργία των γραφημάτων μπάρας.

Τέλος, το αρχείο `timer.js` είναι αυτό που δίνει τη δυνατότητα σε ένα κουίζ να μπορεί να έχει χρονόμετρο.

Αυτό που κάνει ουσιαστικά είναι να περιμένει μέχρι το πέρας του χρόνου και μετά να εκτελεί την ίδια διαδικασία που θα έκανε ο χρήστης για να τελειώσει το κουίζ.

## 5.7 View Ρυθμίσεων εφαρμογής (Settings)

Για επιτευχθεί η καλύτερη προσαρμογή της εφαρμογής στις ανάγκες και απαιτήσεις του εκάστοτε χρήστη, κρίθηκε απαραίτητη η δυνατότητα τροποποίησης των διάφορων λειτουργιών που παρέχει με τη χρήση ενός συστήματος ρυθμίσεων.

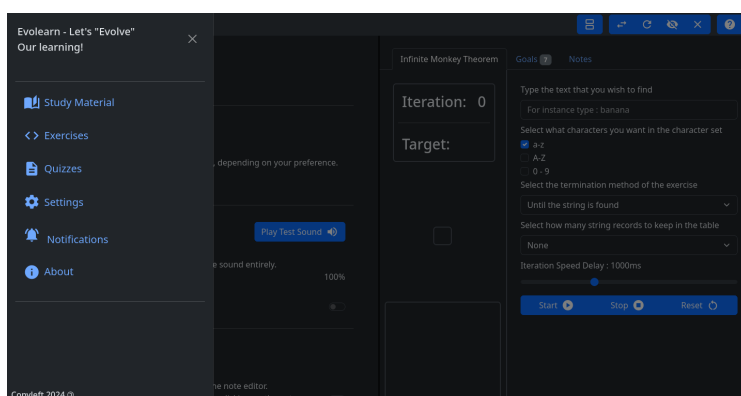


Σχήμα 5.52: Εμφάνιση τμήματος του view των ρυθμίσεων

Στα ακόλουθα υποκεφάλαια γίνεται η επεξήγηση των διάφορων ρυθμίσεων καθώς και η παράθεση εξτρά υλικού όπου αυτό είναι απαραίτητο για την καλύτερη κατανόηση της ρύθμισης. Το τμήμα του view των ρυθμίσεων φαίνεται στο Σχήμα 5.52.

### 5.7.1 Θέμα εφαρμογής (Application Theme)

Αυτή η ρύθμιση αλλάζει όλη την εμφάνιση της εφαρμογής από ανοιχτόχρωμα χρώματα σε σκουρόχρωμα χρώματα και το αντίθετο. Για την προσθήκη του σκουρόχρωμου θέματος απαιτήθηκε η αναπροσαρμογή αρκετών κομματιών της εφαρμογής (ιδιαίτερα αυτών που δεν είχαν τη λειτουργία) αλλά είναι μια ρύθμιση η οποία είναι αναμενόμενη στις σύγχρονες εφαρμογές ιστού, εξ ου και η προσθήκη της. Η προκαθορισμένη ρύθμιση είναι αυτή του ανοιχτόχρωμου θέματος (στο Σχήμα 5.53 φαίνεται και το σκουρόχρωμο θέμα).



Σχήμα 5.53: Εμφάνιση εφαρμογής με το σκουρόχρωμο θέμα

### 5.7.2 Ρυθμίσεις ήχου (Volume Control)

Η συγκεκριμένη ρύθμιση έχει να κάνει με την ένταση την οποία ακούγονται όλοι οι ήχοι στην εφαρμογή. Υπάρχει μια μπάρα η οποία έχει ως ελάχιστη τιμή έντασης το 0% και ως μέγιστη το 100%.

Η προκαθορισμένη ρύθμιση είναι στο 100% αλλά ο χρήστης μπορεί να την κυλίσει σε όποια τιμή επιθυμεί.

Εκτός αυτού υπάρχει και η επιλογή της ολικής απενεργοποίησης του ήχου ενεργοποιώντας την επιλογή “Mute Volume ?” (με την ενεργοποίηση της απενεργοποιείται η μπάρα ήχου)

Για να ελέγξει το αποτέλεσμα των ρυθμίσεων μπορεί να κάνει κλικ στο κουμπί “Play Test Sound” για να ακουστεί ένας δοκιμαστικός ήχος.

### 5.7.3 Ρυθμίσεις αυτόματης αποθήκευσης σημειώσεων (Editor Auto Save)

Αυτή η ρύθμιση καθορίζει το αν θα είναι μόνιμα ενεργή η αυτόματη αποθήκευση των αλλαγών των σημειώσεων η όχι. Η συγκεκριμένη ρύθμιση μπορεί να αντικατασταθεί προσωρινά εντός του editor των σημειώσεων, και η προκαθορισμένη κατάσταση της είναι να είναι ανενεργή.

### 5.7.4 Σημειώσεις καθηγητή (Teacher Notes)

Εδώ ο χρήστης έχει τη δυνατότητα να εμφανίσει ή να κρύψει τις σημειώσεις του καθηγητή που ενδέχεται να βρίσκονται μέσα στα έγγραφα ύλης καθώς και τις ασκήσεις, με την ενεργοποίηση ή την απενεργοποίηση της συγκεκριμένης ρύθμισης.

Η συγκεκριμένη ρύθμιση ξεκινά με το να είναι πάντα ενεργή καθώς οι σημειώσεις του καθηγητή είναι άκρος απαραίτητες εξαιτίας του επικουρικού τους χαρακτήρα αλλά κρίθηκε σκόπιμη η προσθήκη της δυνατότητας απόκρυψής τους, γιατί είναι σημαντικό να παρέχουμε στον χρήστη τη δυνατότητα να έχει αυτός τον έλεγχο, παρόλο που μπορεί να μην είναι σωστή η επιλογή της απενεργοποίησης.

### 5.7.5 Λειτουργία υποστήριξης εκτός σύνδεσης (Offline Support Mode)

Για την επίτευξη της παροχής των λειτουργιών της εφαρμογής σε περίπτωση που δεν είναι δυνατή η σύνδεση με τον εξυπηρετητή ή υπάρχει γενική αδυναμία σύνδεσης στο διαδίκτυο, έχει γίνει η προσθήκη της δυνατότητας λειτουργίας υποστήριξης εκτός σύνδεσης.

Αυτό που γίνεται συνοπτικά είναι ότι μόλις ενεργοποιηθεί αυτή η ρύθμιση τότε κάθε αρχείο που αιτείται και έπειτα δέχεται η εφαρμογή από τον εξυπηρετητή αποθηκεύεται εντός του περιηγητή ιστού σε ένα είδους cache.

Έπειτα, έχοντας ως δεδομένο το γεγονός ότι έχει γίνει η αποθήκευση όλων των άμεσα κρίσιμων αρχείων για την εκκίνηση της εφαρμογής (για τον λόγο αυτόν η εφαρμογή προτείνει στον χρήστη να κάνει μια ολική ανανέωση κάνοντας κλικ στο κουμπί “Restart Application”), θα μπορεί ο χρήστης να ανοίξει την εφαρμογή τουλάχιστον μέχρι και την αρχική οθόνη.

Ανάλογα με ποια αρχεία της εφαρμογής βρίσκονται στο cache, υπάρχουν και οι ανάλογες διαθέσιμες λειτουργίες (πχ αν έχει γίνει η τοποθέτηση ενός εγγράφου ύλης στο cache, τότε ο χρήστης θα μπορέσει να το διαβάσει εκτός σύνδεσης).

Εάν το/τα αρχεία που υλοποιούν μια λειτουργία δεν υπάρχουν μέσα στο cache, τότε η εφαρμογή ειδοποιεί τον χρήστη με τη χρήση ενός toast.

Για να έχει τα καλύτερα δυνατά αποτελέσματα με τη συγκεκριμένη λειτουργία, ο χρήστης ενθαρρύνεται να την ενεργοποιήσει από την αρχή, έτσι ώστε σταδιακά να γίνει η φόρτωση ολόκληρης της εφαρμογής.

Με αυτόν τον τρόπο μπορεί να έχει σχεδόν πλήρη ανεξάρτηση από τον εξυπηρετητή.

#### 5.7.5.1 Συμπληρωματικές λεπτομέρειες

Για την υλοποίηση αυτής της λειτουργίας γίνεται η χρήσης ενός service worker και η διαδικασία χρήσης του είναι η εξής.

Πρώτα πρέπει να γίνει η εγγραφή (registration) του service worker έτσι ώστε ο περιηγητής ιστού να γνωρίζει ότι πρέπει να τον χρησιμοποιεί.

Η εγγραφή του γίνεται στο αρχείο service-worker-initialization.js, στον φάκελο AuxiliaryScripts (Σχήμα 5.54). Γίνεται έλεγχος για το αν είναι ενεργή η λειτουργία, και εάν είναι σε θέση ο περιηγητής ιστού να χρησιμοποιήσει service workers.

Το αρχείο στο οποίο αναφέρεται το παραπάνω script είναι το sw.js (Σχήμα 5.55) (αυτό είναι στο όνομα που χρησιμοποιείται συνήθως για έναν service worker), που βρίσκεται στον φάκελο app, έτσι ώστε να μπορεί να κάνει cache όλα τα αρχεία μέσα σε αυτόν τον φάκελο (η τοποθέτηση του αρχείου έχει μεγάλη σημασία καθώς το αρχείο μπορεί να έχει πρόσβαση μόνο στα αρχεία του φακέλου στον οποίον βρίσκεται). Ένας service worker μπορεί να προγραμματιστεί με διάφορες τεχνικές ως προς τον τρόπο που αποφασίζει να τοποθετήσει αρχεία στο cache. Στη συγκεκριμένη εφαρμογή έχει γίνει η χρήση της τεχνικής network first cache after.

Δηλαδή η εφαρμογή πρώτα αιτείται ένα αρχείο από τον εξυπηρετητή, και εάν μπορέσει να το λάβει χωρίς σφάλμα και δεν το έχει μέσα στο cache, κάνει τις απαραίτητες ενέργειες και το προσθέτει.

Εάν η εφαρμογή κάνει μια αίτηση για ένα αρχείο που έχει ήδη στην κατοχή της και το εν λόγω αρχείο δεν έχει αλλάξει στον εξυπηρετητή, τότε αυτό δεν αναμεταδίδεται γλιτώνοντας δικτυακούς πόρους.

```
function initializeServiceWorker() {
  if (
    window.localStorage.getItem("ServiceWorkerState") !== null &&
    window.localStorage.getItem("ServiceWorkerState") === "true"
  ) {
    const registerServiceWorker = async () => {
      if ("serviceWorker" in navigator) {
        try {
          const registration = await navigator.serviceWorker.register("../sw.js");
          if (registration.installing) {
            console.log("Service worker installing");
          } else if (registration.waiting) {
            console.log("Service worker installed");
          } else if (registration.active) {
            console.log("Service worker active");
          }
        } catch (error) {
          console.error('Registration failed with ${error}');
        }
      }
    };
    registerServiceWorker();
  }
}

initializeServiceWorker();

export {initializeServiceWorker}
```

Σχήμα 5.54: Περιεχόμενα αρχείου service-worker-initialization.js

Τώρα αν για κάποιο λόγο δεν μπορεί να γίνει σύνδεση με τον εξυπηρετητή (λόγο προσωρινής παύσης λειτουργίας του ή λόγω πλήρους διακοπής του δικτύου) η εφαρμογή θα κάνει την αίτηση στο cache, και

Κεφάλαιο 5ο:

αν βρεθεί το αιτούμενο αρχείο επιστρέφεται στην οντότητα που το ζήτησε, σαν να μην υπάρχει κάποιο σφάλμα.

### 5.7.6 Ρυθμίσεις των Modals (Application Modals)

Τα modals χρησιμοποιούνται εκτενώς για τη διεκπεραίωση διάφορων κρίσιμων λειτουργιών της εφαρμογής, οπότε η δυνατότητα της αλλαγής του τρόπου αλληλεπίδρασης τους είναι αρκετά σημαντικός.

```
self.addEventListener("fetch", function (event) {
  let request = event.request;
  event.respondWith(
    fetch(request)
      .then(function (response) {
        let copy = response.clone();
        event.waitUntil(
          caches.open("app").then(async function (cache) {
            try {
              return await cache.put(request, copy);
            } catch (error) { }
          })
        );
        return response;
      })
      .catch(async function (_) {
        const response = await caches.match(request);
        return response;
      })
  );
});
```

Σχήμα 5.55: Περιεχόμενα αρχείου sw.js

Υπάρχουν τρεις επιλογές που μπορεί να διαλέξει ο χρήστης για να αλλάξει τη συμπεριφορά του backdrop (το φόντο) ενός modal :

- Enabled : είναι η προκαθορισμένη ρύθμιση. Το φόντο είναι ορατό και ο χρήστης μπορεί να κλείσει το modal κάνοντας κλικ στο φόντο.
- Disabled: το φόντο δε φαίνεται πια και ο χρήστης δεν μπορεί να κάνει κλικ στο φόντο για να κλείσει το modal.
- Static: το φόντο είναι ορατό αλλά ο χρήστης δεν μπορεί να κάνει κλικ στο φόντο για να κλείσει το modal.

Εκτός αυτών των ρυθμίσεων μπορεί να ενεργοποιήσει ή να απενεργοποιήσει τη δυνατότητα κλεισίματος του modal με τη χρήση του πλήκτρου ESC.

Για να ελέγξει τις αλλαγές που έγιναν μπορεί να πατήσει το κουμπί “Open test modal”, που θα προκαλέσει το άνοιγμα ενός γενικού modal.

### 5.7.7 Ρυθμίσεις των Toasts (Application Toasts)

Όπως και με το modal, η δυνατότητα εξατομίκευσης του τρόπου λειτουργίας των toast κρίθηκε αρκετά σημαντική.

Με τα toasts ο χρήστης μπορεί να αλλάξει το αν θα εμφανίζονται με ένα fade-in-animation η όχι, σε περίπτωση που θέλει να είναι άμεση η εμφάνιση.

Επίσης, μπορεί να αλλάξει το αν θα εξαφανίζεται μόνο του το toast μετά από ένα προκαθορισμένο χρονικό διάστημα που μπορεί να αλλάξει με τη χρήση της μπάρας, η οποία έχει ως ελάχιστο χρόνο εμφάνισης τα 100ms και μέγιστο χρόνο τα 15000ms (15 δευτερόλεπτα), ή εάν το επιθυμεί να

απενεργοποιήσει τη λειτουργία αυτόματης απόκρυψης (με αυτή την ενέργεια απενεργοποιείται η μπάρα αλλαγής του χρόνου).

Ο λόγος που μπορεί να θελήσει να αλλάξει κάποιος τον προκαθορισμένο χρόνο εμφάνισης (2500ms) ή την αυτόματη απόκρυψη (η προκαθορισμένη ρύθμιση είναι ενεργή) είναι γιατί αντιμετωπίζει κάποια δυσκολία στην ανάγνωση του κειμένου των toast.

Για να μπορέσει να ελέγξει το αποτέλεσμα των ρυθμίσεων μπορεί να κάνει κλικ στο κουμπί “Open test toast”, που θα προκαλέσει το άνοιγμα ενός γενικού toast.

### 5.7.8 Διαχείριση δεδομένων εφαρμογής

Στην εφαρμογή λαμβάνει μέρος η αποθήκευση διαφόρων ειδών δεδομένων (πχ σημειώσεις, σκορ των κουίζ, τιμές ρυθμίσεων κτλ). Για τη διαχείριση αυτών των δεδομένων, η εφαρμογή παρέχει τρεις βασικές μεθόδους, την εξαγωγή την εισαγωγή και τη διαγραφή.

Η εξαγωγή των δεδομένων μπορεί να γίνει κάνοντας κλικ στο κουμπί “Export”. Αυτό θα έχει ως αποτέλεσμα τη μεταφορά όλων των δεδομένων στο clipboard του χρήστη.

Έπειτα τα δεδομένα αυτά μπορούν να τοποθετηθούν σε ένα αρχείο για τη μακροχρόνια αποθήκευση τους ή την αποστολή τους με κάποιον άλλον.

Με την εισαγωγή των δεδομένων, που μπορεί να γίνει κάνοντας κλικ στο κουμπί “Import”, ο χρήστης μπορεί να προβεί στην επικόλληση των δεδομένων που αποκτά από τη λειτουργία της εξαγωγής μέσα στο εμφανιζόμενο modal. Μετά, με το πάτημα του κουμπιού “Import” του modal γίνεται η αντικατάσταση όλων των προϋπάρχων δεδομένων με τα νέα εισαγόμενα.

Με αυτόν τον τρόπο ένας χρήστης μπορεί να μεταφέρει όλα του τα δεδομένα από έναν υπολογιστή σε έναν άλλον ή να μοιραστεί την πρόοδο του με έναν άλλον χρήστη.

Τέλος, υπάρχει η δυνατότητα της ολικής διαγραφής όλων των δεδομένων της εφαρμογής κάνοντας κλικ στο κουμπί “Delete”.

Πρέπει ο χρήστης να είναι πολύ προσεκτικός με τη συγκεκριμένη ενέργεια (πριν την ολοκλήρωση της προβάλλεται ένα ενημερωτικό modal στο οποίο πρέπει να επιβεβαιώσει τις προθέσεις του), καθώς επρόκειτο για ολική επαναφορά της εφαρμογής στις εργοστασιακές τις ρυθμίσεις καθώς και ανανέωση της ίδιας της σελίδας.

### 5.7.9 Επαναφορά ρυθμίσεων (Reset Settings To Default)

Με τη συγκεκριμένη λειτουργία επαναφέρουμε όλες τις ρυθμίσεις στις προκαθορισμένες του τιμές. Η διαφορά με ολική διαγραφή είναι ότι εδώ επηρεάζονται μόνο οι ρυθμίσεις και δεν τροποποιούνται καθόλου τα αποθηκευμένα δεδομένα του χρήστη.

### 5.7.10 Συμπληρωματικές λεπτομέρειες

Το view των ρυθμίσεων υλοποιείται στο αρχείο settings.js, αλλά σε αντίθεση με τα υπόλοιπα ViewComponents δεν είναι υπεύθυνο μόνο για την απεικόνιση των ρυθμίσεων αλλά κάνει και την εργασία του μεσάζοντα ανάμεσα στα διάφορα συστήματα της εφαρμογής και τους SettingsManagers που χρησιμοποιούν.

## Κεφάλαιο 5ο:

Ουσιαστικά μέσα του έχει διάφορες συναρτήσεις (έχουν το πρόθεμα “update”) η οποίες λειτουργούν σαν ξεκινήτες των συναρτήσεων που βρίσκονται μέσα στους εκάστοτε SettingsManagers, ανάλογα με πια ρύθμιση αλλάζει.

Εκτός από την εκκίνηση αυτών των συναρτήσεων, την πρώτη φορά που γίνεται η φόρτωση του view διαβάζει ή αρχικοποιεί (ανάλογα με το ποία ενέργεια απαιτείται) όλες τις απαραίτητες τιμές των ρυθμίσεων από την localStorage και έπειτα συγχρονίζει τις επιλογές των συστατικών διεπαφής του με τις τιμές που διάβασε.

### 5.8 View Ειδοποιήσεων εφαρμογής (Notifications)

Ο καθηγητής έχει τη δυνατότητα να στέλνει σε όλους του συνδεδεμένους χρήστες ειδοποιήσεις (ο τρόπος αποστολής του θα εξηγηθεί στην υποενότητα 6.2.1.4).

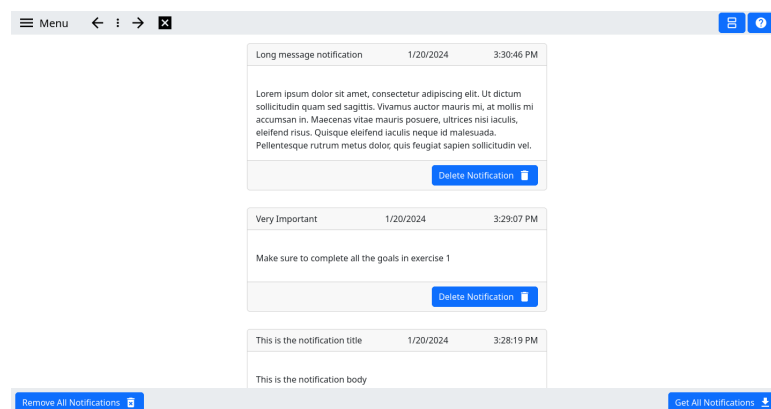
Αυτές οι ειδοποιήσεις έχουν ως κύριο σκοπό να ενημερώσουν τους χρήστες για τυχόν αλλαγές που μπορεί να έχουν γίνει σε κάποια κομμάτια της εφαρμογής (πχ η προσθήκη ενός νέου εγγράφου ή κουίζ, η αλλαγή στους στόχους μια άσκησης) αλλά και για την προτροπή τους στη μελέτη και την ενασχόληση με το μάθημα (πχ δίνοντας του εργασίες να κάνουν εντός της εφαρμογής, και τη μετέπειτα αποστολή των αποτελεσμάτων τους). Το view των ειδοποιήσεων φαίνεται στο Σχήμα 5.56.

Την πρώτη φορά που ο χρήστης ανοίγει αυτό το view τότε γίνεται η λήψη όλων των ειδοποιήσεων από τον εξυπηρετητή και η απεικόνιση τους με χρήση καρτών, η σειρά των οποίων είναι από την πιο πρόσφατη ημερολογιακά στην πιο παλιά.

Η κάθε μια ειδοποίηση αναγράφει πάνω της :

- Τον τίτλο στην πάνω αριστερή γωνία της κάρτας.
- Την ημερομηνία δημιουργίας της στη μέση της μπάρας.
- Την ώρα δημιουργίας της στην πάνω δεξιά γωνία της κάρτας.
- Το περιεχόμενο της στο κέντρο της κάρτας.
- Ένα κουμπί διαγραφής της ειδοποίησης στην κάτω δεξιά γωνία.

Κάνοντας κλικ σε οποιαδήποτε σημείο της ειδοποίησης (εκτός του κουμπιού διαγραφής), τότε θα ανοίξει ένα modal το οποίο παρουσιάζει την ειδοποίηση όπως θα φαινόταν όταν ελήφθη για πρώτη φορά.



Σχήμα 5.56: Προβολή view ειδοποιήσεων

Εκτός των ειδοποιήσεων στο view υπάρχουν και δύο άλλα κουμπιά. Το κουμπί στην κάτω αριστερή γωνία με όνομα “Remove All Notifications”, διαγράφει όλες τις τρέχουσες ειδοποιήσεις.

Το άλλο κουμπί, που βρίσκεται στην κάτω δεξιά γωνία με όνομα “Get All Notifications” κάνει το ακριβώς αντίθετο, δηλαδή ζητάει από τον εξυπηρετητή όλες τις ειδοποιήσεις που έχει αποθηκευμένες.

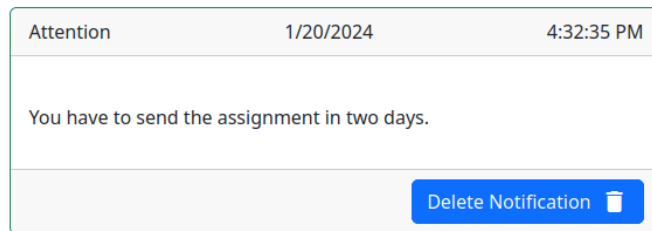
Όταν συμβεί αυτό τότε εμφανίζεται στο πάνω μέρος του view ένα μήνυμα που λέει ότι δεν υπάρχουν ειδοποιήσεις προς προβολή.

Μόλις σταλεί από τον καθηγητή μια νέα ειδοποίηση τότε αυτή θα εμφανιστεί στο view με μία πράσινη περίμετρο, όπως φαίνεται στο Σχήμα 5.57.

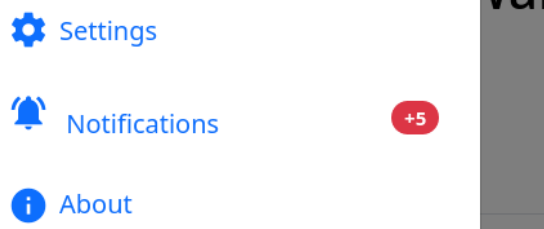
Εκτός αυτών των λειτουργιών, ο χρήστης έχει τη δυνατότητα να εμποδίζει τις ειδοποιήσεις από το να τον διακόπτουν με την εμφάνιση των modal μέσω τις ενεργοποίησης της ρύθμισης Do Not Disturb Mode.

Όταν αυτή η ρύθμιση είναι ενεργή τότε ο αριθμός των ειδοποιήσεων που δεν έχει δει ο χρήστης εμφανίζεται στο μενού πλοήγησης με τη μορφή που φαίνεται στο Σχήμα 5.58.

Μόλις ο χρήστης αποφασίσει να δει τις νέες ειδοποιήσεις, τότε ο αριθμός που εμφανίζεται στο μενού θα εξαφανιστεί.



Σχήμα 5.57: Εμφάνισή νέας ειδοποίησης



Σχήμα 5.58: Εμφάνιση του αριθμού των ειδοποιήσεων στο μενού

### 5.8.1 Συμπληρωματικές λεπτομέρειες

Το view υλοποιείται στο αρχείο notifications.js, και το σύστημα της διαχείρισης των ειδοποιήσεων στον φάκελο NotificationManager.

## Κεφάλαιο 5ο:

Στο αρχείο `notification-component.js` γίνεται η υλοποίηση της κάρτας που εμφανίζει τις ειδοποιήσεις στο view. Στο αρχείο `notification-manager.js` λαμβάνουν μέρος όλες οι λειτουργίες που έχουν σχέση με τη λήψη, διαγραφή και γενικότερα με τη διαχείριση των ειδοποιήσεων.

Κάθε μια ειδοποίηση περιέχει εσωτερικά ένα hash (δημιουργείται με τρόπο που επιχειρεί να διασφαλίσει τη μοναδικότητα), έτσι ώστε να μπορεί να γίνει η εύκολη εύρεση τους όταν αυτό απαιτείται. Στον εξυπηρετητή υπάρχει ένα διαθέσιμο URL το οποίο όταν δέχεται μια αίτηση από ένα χρήστη επιστρέφει όλες τις αποθηκευμένες ειδοποιήσεις σε μορφή JSON.

Όμως επειδή ο κάθε χρήστης ενδέχεται να έχει ένα μεταβλητό αριθμό ειδοποιήσεων αποθηκευμένων, είτε επειδή δεν προβαίνει στο άνοιγμα τους, τις διαγράφει ή έχει καιρό να ανοίξει την εφαρμογή με αποτέλεσμα να συσσωρευτούν, πρέπει να υπάρχει ένας τρόπος του να μπορεί να λάβει μόνο τις ειδοποιήσεις που δεν έχει ήδη, έτσι ώστε να είναι συγχρονισμένος με τον εξυπηρετητή.

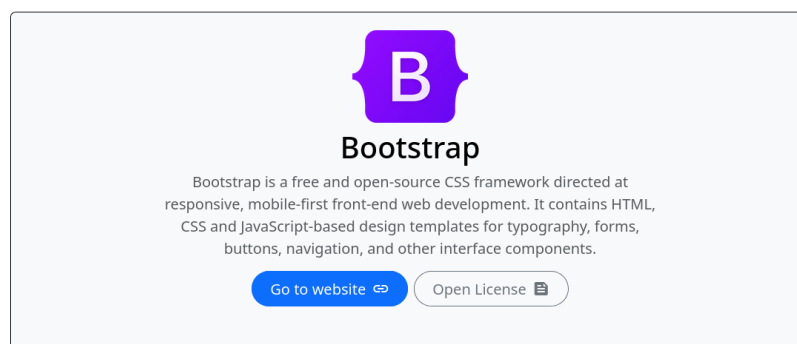
Κάθε φορά που ο χρήστης λαμβάνει τις ειδοποιήσεις από τον εξυπηρετητή, αποθηκεύει το hash της πιο πρόσφατης ειδοποίησης που έχει εκείνη τη στιγμή.

Έπειτα κάθε φορά που γίνεται ο επανέλεγχος για τυχόν νέες ειδοποιήσεις, μαζί με το αίτημα στον εξυπηρετητή στέλνεται και το αποθηκευμένο hash. Μόλις ο εξυπηρετητής λάβει και ένα hash με την αίτηση τότε επιστέφει πίσω μόνο όσες ειδοποιήσεις εμφανίζονται μετά από αυτή που υποδεικνύει το hash.

### 5.9 View Παρουσίασης τεχνολογιών (About)

Λόγου του μεγάλου αριθμού εργαλείων και βιβλιοθηκών που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής κρίθηκε σκόπιμη η παράθεση του εντός της ίδιας της εφαρμογής με τη μορφή που φαίνεται στο Σχήμα 5.60. Η τεχνολογίες εμφανίζονται είτε από αριστερά είτε από δεξιά με τη χρήση ενός `fade-in-animation` όταν ο χρήστης κάνει κύλιση του view προς τα κάτω. Πάνω σε κάθε κάρτα υπάρχει το λογότυπο της τεχνολογίας (αν υπάρχει), μια σύντομη περιγραφή της τεχνολογίας καθώς και δύο κουμπιά. Αυτό με το όνομα “Go to website” ανοίγει σε ξεχωριστό παράθυρο την κύρια ιστοσελίδα της τεχνολογίας, όπου μπορεί κανείς να βρει περαιτέρω πληροφορίες καθώς και συνδέσμους λήψης της.

Το δεύτερο κουμπί με όνομα “Open Licence”, ανοίγει σε ξεχωριστό παράθυρο την αρμόζουσα άδεια την οποία χρησιμοποιεί η εκάστοτε τεχνολογία (εάν δεν υπάρχει άδεια τότε δεν εμφανίζεται το κουμπί). Μια πλήρη κάρτα φαίνεται στο Σχήμα 5.59.



Σχήμα 5.59: Εμφάνιση κάρτας τεχνολογίας

### 5.9.1 Λογότυπο εφαρμογής

Το λογότυπο της εφαρμογής σχεδιάστηκε έτσι ώστε να είναι ευδιάκριτο και σε ανοιχτόχρωμο αλλά και σε σκουρόχρωμο θέμα, για αυτό τον λόγο η περίμετρος του είναι γαλάζια και το κύριο χρώμα του είναι το μαύρο.

Αυτό που απεικονίζει είναι τη γέννηση της γνώσης, που απεικονίζεται ως μια λάμπα. Έπειτα αυτή η γνώση ωριμάζει και γίνεται πιο ολοκληρωμένη, σφαιρική όπως φαίνεται στην αριστερή πλευρά. Με την ωρίμανση της αρχικής ιδέας αυτή δίνει καρπό σε μια ακόμα μεγαλύτερη της, όπως φαίνεται στη δεξιά πλευρά. Αλλά είναι αναπόφευκτα συνδεδεμένες μεταξύ τους. Η άσπρη γραμμή προστέθηκε για να εμπνεύσει συναισθήματα παρειδολίας.



Σχήμα 5.60: Λογότυπο εφαρμογής

### 5.9.2 Συμπληρωματικές λεπτομέρειες

Μέσα στον φάκελο ViewComponents υπάρχει ένα αρχείο με όνομα about.json.

Εκεί είναι όλες οι εγγραφές που χρησιμοποιεί το view (που υλοποιείται στο αρχείο about.js) για να κάνει την απεικόνιση των τεχνολογιών.

Μια πλήρη εγγραφή του αρχείου αυτού φαίνεται στο Σχήμα 5.61

```
{
  "img": "../Images/bootstrap_logo.png",
  "title": "Bootstrap",
  "description": "Bootstrap is a free and open-source CSS framework directed
  at responsive, mobile-first front-end web development. It contains HTML,
  CSS and JavaScript-based design templates for typography, forms, buttons,
  navigation, and other interface components.",
  "url": "https://getbootstrap.com/",
  "license": "https://opensource.org/licenses/mit/",
  "direction": "left"
},
```

Σχήμα 5.61: Εγγραφή του αρχείου about.json

Τα πεδία μια εγγραφής είναι τα εξής (είναι όλα τους string) :

- img : ορίζει το PATH στο οποίο μπορεί να βρεθεί το λογότυπο.
- title : ορίζει τον τίτλο της τεχνολογίας.

Κεφάλαιο 5ο:

- description : ορίζει την εγγραφή της τεχνολογίας.
- url : ορίζει τον σύνδεσμο που θα ανοίξει την αρχική σελίδα της τεχνολογίας.
- license : ορίζει τον σύνδεσμο που θα ανοίξει την ιστοσελίδα της εκάστοτε άδειας.
- direction : ορίζει την κατεύθυνση από την οποία μπορεί να εμφανιστεί η τεχνολογία (μπορεί να είναι left ή right).

Με τη χρήση αυτού του συστήματος μπορεί πολύ εύκολα να γίνει η προσθήκη των σχετικών πληροφοριών μιας καινούριας τεχνολογίας σε μελλοντικές επαυξήσεις της εφαρμογής.

### **5.10 Επίλογος**

Σε αυτό το κεφάλαιο έγινε μια πλήρη παράθεση όλων των λειτουργιών που βρίσκεται σε δυνατότητα να εκτελέσει η εφαρμογή (με την εμβάθυνση σε πτυχές που έχριζαν περαιτέρω ανάλυση) καθώς και τις απαραίτητες ενέργειες που πρέπει να κάνει κάποιος για να την προσαρμόσει στις δικές του ανάγκες (κυρίως αναφερόμενοι στα αρχεία που βρίσκονται στον φάκελο ContentFiles).

## Κεφάλαιο 6ο: Λεπτομέρειες Εξυπηρετητή και Εγκατάστασης

### 6.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται μια μικρή ανασκόπηση στη μεριά του εξυπηρετητή και σε κάποιες λίγο πιο εξειδικευμένες ενέργειες που εκτελεί σε σχέση με τις προκαθορισμένες (την απλή μεταφορά αρχείο ως αποτέλεσμα της λήψης μια αίτησης από τον πελάτη) καθώς και την παράθεση συνδέσμων που αναφέρονται στις διάφορες μεθόδους και πλατφόρμες που μπορεί κανείς να “σερβίρει” την εφαρμογή.

### 6.2 Ο εξυπηρετητής

Το αρχείο που “περιέχει” τον κώδικα του εξυπηρετητή είναι το `index.js`.

Βέβαια, υπάρχει μια ολόκληρη υποδομή κάτω από αυτό το αρχείο καθώς κάνουμε χρήση του Node.js που μας επιτρέπει να χρησιμοποιούμε τη JavaScript αλλά παρέχει και τον ίδιο τον εξυπηρετητή, καθώς και το Express.js, που αυτοματοποιεί πολλές από τις πιο ανιαρές λειτουργίες που πρέπει να συγγράψει κάποιος όταν θέλει να υλοποιήσει μια εφαρμογή ιστού και πρόκειται για ένα στρώμα αφαίρεσης που βρίσκεται πάνω από το Node.js.

Απλά αυτό το αρχείο είναι σημείο εκκίνησης του εξυπηρετητή και υλοποιεί τις λειτουργίες που δε θα εκτελούσε ένας απλός εξυπηρετητής μια στατικής ιστοσελίδας. Εκτός αυτού το Express.js παρέχει και πακέτα επέκτασης που του επιτρέπουν να τροποποιεί τις εισερχόμενες αλλά και τις εξερχόμενες αιτήσεις προς/από τον εξυπηρετητή.

#### 6.2.1 Δομή του αρχείου `index.js`

##### 6.2.1.1 Βασική αρχικοποίηση

Το πρώτο τμήμα το αρχείο φαίνεται στο Σχήμα 6.1, και περιέχει όλα τα απαραίτητα `import statements` για τη λειτουργία του εξυπηρετητή καθώς και όλες τις απαραίτητες αρχικοποιήσεις των μεταβλητών που απαιτούνται.

```

1  const express = require("express");
2  const compression = require("compression");
3  const bodyParser = require("body-parser");
4  const { Server } = require("socket.io");
5  const fs = require("fs");
6  const path = require("path");
7
8  const app = express();
9  const port = 3000;
10 const rootPath = path.join(__dirname, "app");
11 app.use(compression());
12 app.use("/", express.static(rootPath));
13 app.use(bodyParser.json());
14
15 const server = app.listen(port, () => {
16   console.log(`App listening on port ${port}`);
17   readNotifications();
18 });
19
20 const io = new Server(server);
21
22 let notifications = [];
```

Σχήμα 6.1: Πρώτο τμήμα αρχείου `index.js`

## Κεφάλαιο 6ο:

Πιο συγκεκριμένα από τις γραμμές 1-6 λαμβάνουν μέρος τα `import`. Στις γραμμές 8-10, γίνεται η εκκίνηση του `express.js`, η αρχικοποίηση του αριθμού `port` στην οποία θα ακούει ο εξυπηρετητής και ο ορισμός του φακέλου που περιέχει όλα τα αρχεία της εφαρμογής (στην προκειμένη περίπτωση είναι ο `app`).

Έπειτα στις γραμμές 11-13 γίνεται η χρήση κάποιων πακέτων επεκτάσεων του `express.js` που ονομάζονται `middlewares`. Αυτό που κάνουν είναι παρέχουν έξτρα λειτουργίες στο `express.js` αλλά και να προβαίνουν στην τροποποίηση των πακέτων.

Στην εφαρμογή γίνεται η χρήση του `compression middleware`, για της συμπίεση όλων των αποσταλμένων αρχείων από τον εξυπηρετητή έτσι ώστε να μειωθεί ο χρόνος λήψης του από τον πελάτη καθώς και να αυξηθεί το φαινομενικά διαθέσιμο `bandwidth`, παρέχοντας έτσι μια πιο ευχάριστη εμπειρία στον χρήστη.

Εκτός αυτού του `middleware` γίνεται η χρήση του `bodyParser` για την προσθήκη της ικανότητας ανάγνωσης και εγγραφής `JSON` εντός των αποσταλμένων και ληφθέντων αιτήσεων.

Στη γραμμή 15 γίνεται η πραγματική εκκίνηση του εξυπηρετητή και στη 20 γραμμή αρχικοποιείται το `socket.io`, για να διαχειριστεί την αμφίδρομη επικοινωνία με τους χρήστες.

### 6.2.1.2 Έλεγχος αλλαγών αρχείων

Η εφαρμογή μπορεί να αντιδράσει σε τυχόν αλλαγές που συμβαίνουν σε διάφορα αρχεία.

```
28 function writeNotifications() {
29   fs.writeFileSync("notification_log.json", JSON.stringify(notifications));
30 }
31
32 const fileWatcherDefinitions = [
33   {
34     url: "./app/ContentFiles/PdfContentFiles/",
35     fileName: "pdf_definitions.json",
36     intent: "pdfDefinitionsChange",
37   },
38   {
39     url: "./app/ContentFiles/ExerciseContentFiles",
40     fileName: "exercise_definitions.json",
41     intent: "exerciseDefinitionsChange",
42   },
43   {
44     url: "./app/ContentFiles/QuizContentFiles",
45     fileName: "quiz_definitions.json",
46     intent: "quizDefinitionsChange",
47   },
48 ];
49
50
51 function setFileWatcher(url, fileName, intent) {
52   fs.watchFile(`${url}/${fileName}`, function (_, _) {
53     io.emit(intent);
54     console.log(`Intent - ${intent} - Sent`);
55   });
56 }
57
58 fileWatcherDefinitions.forEach((definition) => {
59   setFileWatcher(definition.url, definition.fileName, definition.intent);
60 });
```

Σχήμα 6.2: Εγγραφές αρχείων προς παρακολούθηση

Ο τρόπος που το καταφέρνει αυτό είναι έχοντας μέσα στον εξυπηρετητή μια λίστα εγγραφών για το ποια αρχεία πρέπει να παρακολουθούνται και ποια ενέργεια πρέπει να λαμβάνει μέρος όταν εντοπιστεί μια αλλαγή.

Στη γραμμή 32 στο Σχήμα 6.2 ορίζονται αυτές οι εγγραφές ακολουθώντας την ακόλουθη δομή (όλα τα πεδία είναι `string`):

- `url` : ορίζει το που βρίσκεται το εν λόγω αρχείο μέσα στους φακέλους της εφαρμογής.
- `fileName` : ορίζει το όνομα του αρχείου για το οποίο ενδιαφερόμαστε.

- `intent` : ορίζει μια ενέργεια εκτελείται στον πελάτη όταν δεχτεί το μήνυμα από τον εξυπηρετητή (αυτό το `intent` το επεξεργάζεται η μεριά του πελάτη στο αρχείο `socket-definitions.js`, εκεί είναι ορισμένη μια ενέργεια με ακριβώς το ίδιο όνομα που έχει το `intent`).

Εκτός αυτών των εγγραφών υπάρχει και μια συνάρτηση με όνομα `setFileWatcher` (ορίζεται στη γραμμή 51 στο Σχήμα 6.2) η οποία έχοντας ως παραμέτρους τα προαναφερθέντα πεδία μιας εγγραφής ορίζει μια οντότητα που έχει σκοπό να παρακολουθεί το εν λόγω αρχείο για αλλαγές.

Μόλις λάβει μέρος μια αλλαγή, τότε με τη χρήση του `socket.io` στέλνει στον πελάτη το ανάλογο `intent` (στη γραμμή 53 αυτό γίνεται με το κάλεσμα του `io.emit(intent)`).

### 6.2.1.3 Διαχείριση των διαδρομών

Για να ανοίξουμε μια διαδικτυακή εφαρμογή πρέπει να επισκεφτούμε ένα συγκεκριμένο URL.

Μια τέτοια εφαρμογή μπορεί να ορίζει διάφορα τέτοια URL ή αλλιώς διαδρομές, και ανάλογα με τη διαδρομή μπορεί να εκτελούνται συγκεκριμένες ενέργειες ή να γίνεται η ανακατεύθυνση σε μία άλλη διαδρομή (ή και `endpoint`).

Αυτές οι διαδρομές ορίζονται στην εφαρμογή με τον τρόπο που παρουσιάζεται στο Σχήμα 6.3.

```

100
101 app.post("/notify", async (req, res) => {
102   let userName = "test";
103   let password = "123";
104   password = await digestMessage(password);
105   if (userName === req.body.userName && password === req.body.password) {
106     io.emit("teacherNotification", req.body);
107     notifications.push(req.body);
108     writeNotifications();
109     res.json(req.body);
110   }
111 });
112
113 app.get("/", (_, res) => {
114   res.sendFile("/index.html", { root: rootPath });
115 });
116
117 app.get("/*", (_, res) => res.redirect("/"));
118
```

Σχήμα 6.3: Ορισμός κάποιων διαδρομών στο αρχείο `index.js`

Για παράδειγμα, στη γραμμή 113 δίνουμε τις οδηγίες στον εξυπηρετητή όταν ένας χρήστης πλοηγήσει στην εφαρμογή να του επιστέφεται το αρχείο `index.html` (το σημείο εκκίνησης της εφαρμογής για τον πελάτη).

Η τελευταία εγγραφή στη γραμμή 117 δίνει την οδηγία της ανακατεύθυνσης του χρήστη στην αρχική σελίδα εάν προσπαθήσει να πλοηγηθεί σε μια διαδρομή που δεν έχει οριστεί πιο νωρίς στο αρχείο (η συγκεκριμένη εγγραφή πρέπει να είναι η τελευταία γιατί αλλιώς δε λειτουργεί η ανακατεύθυνση).

Οι διαδρομές αυτές έχουν τη μορφοποίηση [ονομα-εξυπηρετητή:πόρτα/διαδρομή].

Στην προκειμένη περίπτωση θα ήταν `localhost:3000/` γιατί τρέχουμε την εφαρμογή τοπικά.

### 6.2.1.4 Διαχείριση των ειδοποιήσεων

Η διαχείριση των ειδοποιήσεων υλοποιείται με τη χρήση τριών αρχείων.

## Κεφάλαιο 6ο:

Το πρώτο είναι το `notifications_log.json` στο οποίο αποθηκεύονται όλες οι ειδοποιήσεις που δημιουργούνται από τον καθηγητή (το αρχείο αυτό επίσης παρακολουθείται για τυχόν αλλαγές).

Κάθε φορά που ξεκινάει ο εξυπηρετητής αυτό που κάνει είναι να διαβάσει αυτό το αρχείο και να αποθηκεύει όλες τις εγγραφές στη μνήμη του, έτσι ώστε να μη χρειάζεται να κάνει ανάγνωση του αρχείου κάθε φορά που ένας χρήστης αιτείται τις σημειώσεις, προκαλώντας αχρείαστη καθυστέρηση.

Οι ειδοποιήσεις δημιουργούνται με τη χρήση μιας “υποεφαρμογής” που ουσιαστικά πρόκειται για μια άλλη διαδρομή στον εξυπηρετητή.

Αυτό το endpoint είναι το `/teacher` στο οποίο μόλις γίνει η πλοήγηση εμφανίζεται η οθόνη που φαίνεται στο Σχήμα 6.4. Η συγκεκριμένη οθόνη υλοποιείται στο αρχείο `teacher.html` που βρίσκεται μέσα στον φάκελο `app`.

Από εδώ μπορεί ο καθηγητής, εφόσον εισάγει το σωστό User Name και Password, να συντάξει μια ειδοποίηση. Τα πεδία αυτά ορίζονται μέσα στο αρχείο `index.js` (στο Σχήμα 6.3 οι γραμμές 102 και 103).

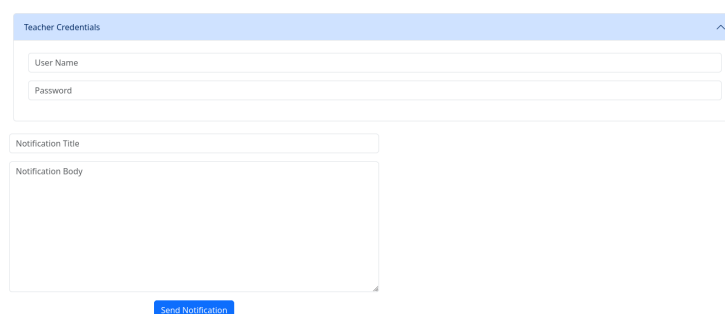
Ο λόγο ύπαρξης τους είναι για να μην μπορέσει κάποιος που δεν είναι ο καθηγητής να στείλει ειδοποιήσεις (σε κανονική χρήση της εφαρμογής πρέπει να γίνει η αλλαγή των προκαθορισμένων τιμών).

Το User Name και Password αλλοιώνονται με τη χρήση του αλγόριθμου `sha-256` έτσι ώστε να μη στέλνονται σε μορφή απλού κειμένου στον εξυπηρετητή.

Μια σημείωση μπορεί να έχει τίτλο και πρέπει να έχει περιεχόμενο (μπορεί να είναι κείμενο ή κώδικας HTML που αποδίδεται αργότερα στον πελάτη), στην περίπτωση που δεν υπάρχει αυτό ή εάν δεν έχει γίνει η συμπλήρωση του User Name και Password τότε ο καθηγητής ενημερώνεται με ένα `alert`.

Μόλις ο καθηγητής κάνει κλικ στο κουμπί “Send Notification” τότε άλλα τα πεδία που συμπλήρωσε στη σελίδα στέλνονται με μορφή JSON στον εξυπηρετητή μέσω ενός endpoint (το `/notify`).

Εκεί ο εξυπηρετητής κάνει την αυθεντικοποίηση του καθηγητή και εάν είναι επιτυχής τότε στέλνει τα περιεχόμενα της ειδοποίησης (τον τίτλο και το κείμενο) με τη χρήση `socket` στους συνδεδεμένους χρήστες, προσθέτει μια εγγραφή στο αρχείο `notifications_log.json` και ανανεώνει την εσωτερική του μνήμη.



Σχήμα 6.4: Εμφάνιση αρχικής οθόνης `/teacher`

## 6.3 Τρόποι εγκατάστασης

### 6.3.1 Τοπική εγκατάσταση

Ο λόγος για να τρέχει κανείς την εφαρμογή τοπικά είναι για τον έλεγχο της συμπεριφοράς διάφορων προσθηκών και αλλαγών που ενδέχεται να χρειάζονται αποσφαλμάτωση πριν την έκθεσή τους στους χρήστες. Για να γίνει αυτό πρέπει πρώτα να γίνει η εγκατάσταση του Node.js στον υπολογιστή.

Ανάλογα με το λειτουργικό σύστημα που χρησιμοποιείται αλλάζει και η διαδικασία εγκατάστασης. Πληροφορίες για αυτή τη διαδικασία μπορούν να βρεθούν [εδώ](#).

Μετά την εγκατάσταση πρέπει να εκτελεστεί η εντολή `[node index.js]` από ένα τερματικό μέσα στον φάκελο που περιέχει το εν λόγω αρχείο.

Έπειτα με τη χρήση ενός περιηγητή ιστού μπορεί ο χρήστης να πλοηγηθεί στον URL [\[http://localhost:3000/\]](http://localhost:3000/) για να βρεθεί στην αρχική οθόνη (Σχήμα 5.1).

### 6.3.2 Προώθηση της τοπικής εγκατάστασης

Όταν ο αριθμός των χρηστών είναι σχετικά μικρός θα μπορούσε να γίνει η προώθηση του εξυπηρετητή που δημιουργείται με την εκτέλεση των βημάτων που περιγράφονται στην προηγούμενη υπολότητα.

Υπάρχουν πολλές εταιρίες που παρέχουν αυτή τη λειτουργία, είτε δωρεάν είτε με κάποιο είδος συνδρομής. Μια λίστα με τέτοιες εταιρίες μπορεί να βρεθεί [εδώ](#).

Ανάλογα με διαλεγμένη εταιρία τα βήματα που απαιτούνται ενδέχεται να διαφέρουν μεταξύ τους και πρέπει να γίνει κατανοητό ότι αυτή η μέθοδος απευθύνεται σε μικρό αριθμό χρηστών και κυρίως για δοκιμαστικούς λόγους.

### 6.3.3 Χρήση ιδιωτικού εξυπηρετητή

Σε περίπτωση που υπάρχει είδη διαθέσιμος υπολογιστής που μπορεί να εκτελέσει τα καθήκοντα ενός εξυπηρετητή, μπορεί σχετικά εύκολα να προσαρμοστεί για να εκτελέσει την εφαρμογή.

Υπάρχουν πολύ τρόποι που μπορεί να γίνει αυτό, ανάλογα με πιο λειτουργικό σύστημα και πια προϋπάρχουσα εφαρμογή εκτελεί τις λειτουργίες του εξυπηρετητή.

Για αυτό χρειάζεται περαιτέρω έρευνα για την κατάλληλη μέθοδο προσέγγισης.

[Εδώ](#) βρίσκονται οδηγίες για της προσαρμογή του Apache στο λειτουργικό σύστημα Ubuntu.

[Εδώ](#) βρίσκονται οδηγίες για την προσαρμογή του NGINX στο λειτουργικό σύστημα Ubuntu.

### 6.3.4 Πλατφόρμες ανάρτησης εφαρμογών Node.js

Υπάρχουν πάρα πολλές πλατφόρμες που διαχειρίζονται εφαρμογές node.js. Η κάθε μια προσφέρει διάφορες υπηρεσίες που κάνουν τη διαχείριση μιας τέτοιας εφαρμογής πολύ πιο εύκολης σε σχέση με τις προαναφερόμενες μεθόδους.

Για κάθε μία τα βήματα για το ανέβασμα μιας τέτοιας εφαρμογής θα είναι διαφορετικά, οπότε πρέπει να γίνει περαιτέρω έρευνα για την κατάλληλη μέθοδο προσέγγισης.

Κεφάλαιο 6ο:

Κάποιες από αυτές είναι συνδρομητικές ενώ κάποιες άλλες παρέχουν δωρεάν κάποιες υπηρεσίες, αλλά αυτές είναι συνήθως πολύ περιορισμένες ως προς του πόρους που παρέχουν. Μερικές από αυτές τις πλατφόρμες μπορούν να βρεθούν [εδώ](#).

#### **6.4 Επίλογος**

Στο κεφάλαιο αυτό έλαβε μέρος η παρουσίαση κάποιων λεπτομερειών υλοποίησης από τη μεριά του εξυπηρετητή καθώς και μια πληθώρα διαφορετικών μεθόδων από τις οποίες μπορεί να επιλέξει κάποιος για το πως καλύτερα θα διανέμει την προσαρμοσμένη εφαρμογή στους χρήστες.

## Κεφάλαιο 7ο: Συμπεράσματα και Προτάσεις Βελτίωσης

Η δημιουργία ενός πλαισίου εκπαιδευτικής εφαρμογής έχει διάφορες πτυχές που πρέπει να εξεταστούν μεθοδικά για να επιτευχθεί ένα ικανοποιητικό αποτέλεσμα. Πρέπει να επιτευχθεί μια ισορροπία ανάμεσα στις παρεχόμενες λειτουργίες και στην πολυπλοκότητα που αυτές προσδίδουν στο πλαίσιο.

Ένα τέτοιο πλαίσιο οφείλει να παρέχει εύκολη τροποποίηση της λειτουργίας των διάφορων τμημάτων του με μεθόδους που είναι απλές στην κατανόηση και προσφέρουν μεγάλη ευελιξία.

Επιπλέον, πρέπει να προσαρμόζεται εύκολα σε διάφορες συνθήκες εγκατάστασης καθώς και σε παραλλαγές του περιεχομένου που καλείται να διαχειριστεί.

Όλα αυτά μπορούν να επιτευχθούν με τη χρήση των κατάλληλων εργαλείων και τεχνολογιών σε συνδυασμό με τη σχεδίαση ενός πλαισίου αρχιτεκτονικής που προάγει την τμηματοποίηση του κώδικα και την εύκολη μελλοντική επεκτασιμότητα.

Η παρούσα εργασία επιχειρεί να παρουσιάσει ένα πρωτότυπο του πλαισίου, χρησιμοποιώντας εδραιωμένα και δοκιμασμένα στον χρόνο εργαλεία, καθώς και χρησιμοποιώντας κατάλληλες αρχιτεκτονικές μεθόδους για τον σχεδιασμό του πλαισίου. Ωστόσο, αναγνωρίζοντας την εγγενή δυνατότητα βελτίωσης, καθώς και τη συνεχή αναζήτηση καινοτομίας στον χώρο του λογισμικού, οι παρακάτω ενότητες προτείνουν δυνατότητες για περαιτέρω βελτιστοποίηση και επέκταση αυτού του πλαισίου.

### 7.1 Υποστήριξη πολλαπλών γλωσσών

Στην παρούσα φάση, η εφαρμογή υποστηρίζει μόνο την αγγλική γλώσσα, στα μενού και τα υπόλοιπα σημεία που εμφανίζουν κείμενο. Παρόλο που τα κουίζ παρέχουν τη δυνατότητα εναλλαγής μεταξύ ελληνικών και αγγλικών (όπου αυτό είναι δυνατό), θα ήταν ωφέλιμο να υποστηρίζεται η εναλλαγή της γλώσσα σε όλη την εφαρμογή.

Δυστυχώς, ο σχεδιασμός της δεν ξεκίνησε με τη συγκεκριμένη λειτουργικότητα ως βασικό κανόνα, οπότε η προσθήκη της απαιτεί μια σχετικά σημαντική αναγραφή μεγάλων τμημάτων της εφαρμογής, καθώς και την παροχή κατάλληλων μεταφράσεων για όλα τα τμήματα που παρουσιάζουν κάποια μορφή κειμένου στον χρήστη. Το ιδανικό θα ήταν να χρησιμοποιηθούν τεχνολογίες [i18n](#), οι οποίες αποτελούν το ντε φάκτο πρότυπο για τη διεθνοποίηση εφαρμογών, καθιστώντας έτσι την εφαρμογή πιο προσιτή σε ένα ευρύτερο κοινό.

### 7.2 Προσθήκη περισσότερων τύπων ερωτήσεων

Στον τρέχοντα σχεδιασμό της εφαρμογής παρέχονται 4 τύποι ερωτήσεων, οι οποίοι μπορούν να συνδυαστούν μεταξύ τους για να παρέχουν τη δυνατότητα δημιουργήθηκαν ενός αρκετά μεγάλου αριθμού κουίζ, αλλά είναι προφανές ότι όσο αυξάνεται ο αριθμός των διαθέσιμων τύπων ερωτήσεων τόσο αυξάνεται και η ποικιλία των κουίζ. Θα μπορούσε να γίνει η προσθήκη ερωτήσεων με εικόνες, βίντεο ή ήχο, καθιστώντας τις πιο διαδραστικές και ενδιαφέρουσες.

## Κεφάλαιο 7ο:

Φυσικά, θα πρέπει επίσης να ληφθεί υπόψη η παροχή εναλλακτικών ερωτήσεων για άτομα τα οποία ενδέχεται να αντιμετωπίζουν διάφορες δυσκολίες οι οποίες είναι δυνατόν να καταστήσουν την επεξεργασία τους δύσκολη (πχ προβλήματα ακοής).

### 7.3 Δυνατότητα τυποποίησης των στόχων

Οι στόχοι των ασκήσεων συνδέονται άμεσα με όλες τις μεταβλητές που χρησιμοποιούνται στην ίδια την υλοποίηση της άσκησης. Αν και αυτό είναι αναγκαίο για τη λειτουργία τους, θα ήταν σχετικά πιο εύχρηστο να υπήρχαν τρόποι συγγραφής στόχων βασισμένοι σε χαρακτηριστικά που μοιράζονται οι περισσότερες ασκήσεις (π.χ. κουμπιά ρυθμίσεων, συστατικά διεπαφής που υλοποιούν την οπτικοποίηση κάποιας οντότητας).

Με αυτόν τον τρόπο, λόγω του υψηλού βαθμού επαναχρησιμοποίησης στοιχείων από άσκηση σε άσκηση, θα μπορούσε να υπάρξει ένα είδος script, το οποίο θα καθορίζει τους στόχους με γενικές παραμέτρους χωρίς να απαιτείται επέμβαση στον κώδικα, χωρίς φυσικά να απαγορεύεται η επέμβαση όπου είναι απαραίτητη για την υλοποίηση στόχων αυξημένης πολυπλοκότητας.

### 7.4 Υποστήριξη μεγαλύτερου φάσματος συσκευών

Η εφαρμογή έχει σχεδιαστεί για χρήση σε σταθερούς υπολογιστές και laptop. Αυτό προϋποθέτει μια σχετικά μεγάλη οθόνη για την παρουσίαση της, πράγμα που δεν είναι πάντα απαραίτητο να ισχύει τη σήμερον εποχή, όπου η χρήση των κινητών συσκευών (smart phone και tablet) είναι μεγαλύτερη από ποτέ. Αξίζει να σημειωθεί ότι η εφαρμογή μπορεί να χρησιμοποιηθεί στις προαναφερθείσες συσκευές χωρίς κάποιο ιδιαίτερο πρόβλημα, αλλά είναι άκρως απαραίτητη η προσαρμογή της για τη μεγιστοποίηση της ευχρηστίας σε αυτές τις συσκευές, καθώς ενέχουν ιδιαίτερες προκλήσεις.

### 7.5 Προσθήκη περαιτέρω μεθόδων απόδοσης ύλης

Μέχρι στιγμής η παράδοση της ύλης γίνεται μέσω αρχείων PDF. Η συγκεκριμένη μέθοδος παρέχει αρκετά πλεονεκτήματα όπως η εύκολη μεταφορά αρχείων λόγω του μεγέθους τους, το γεγονός ότι είναι πανταχού παρούσες καθώς υποστηρίζονται σε κάθε λειτουργικό σύστημα, η δυνατότητα εμπλουτισμού τους με υπερ-συνδέσμους σε χρήσιμους εξωτερικούς ιστότοπους κ.α.

Σίγουρα, όμως δεν αποτελούν τη μοναδική λύση. Για παράδειγμα, θα μπορούσε να συμπεριληφθεί βίντεο ή ήχος για την αύξηση της διαδραστικότητας της ύλης, καθώς και άλλες μέθοδοι παράδοσης κειμένου, όπως σελίδες HTML και αρχεία TXT. Αυτό επιτρέπει τη συσσώρευση χρήσιμων πληροφοριών, οι οποίες σχετίζονται άμεσα με το μάθημα σε διάφορες μορφές πιθανής διανομής χωρίς την ανάγκη επεξεργασίας.

### 7.6 Προσθήκη μεθόδων οργάνωσης σημειώσεων

Ο τρόπος παρουσίασης των σημειώσεων βασίζεται στη χρονολογική τους σειρά, όπου οι πιο πρόσφατες βρίσκονται δεξιά και οι παλιότερες στα αριστερά, χωρίς να λαμβάνονται υπόψιν τυχόν διαγραφές σημειώσεων.

Αυτό το σύστημα είναι εύχρηστο, αλλά στερείται επιλογών εξατομίκευσης. Θα μπορούσε να υπάρχει η επιλογή ταξινόμησης με βάση το μέγεθος των σημειώσεων, ή τις πιο πρόσφατες αλλαγές.

Υπάρχουν πολλοί άλλοι τρόποι που θα μπορούσε να γίνει η ταξινόμηση καθώς και η απόκρυψη σημειώσεων με χρήση διάφορων φίλτρων. Τέτοιου είδους ρυθμίσεις επιτρέπουν την εξατομίκευση της εμπειρίας των χρηστών και τη βελτίωση της χρηστικότητας.

### 7.7 Μείωση εξάρτησης από τη βιβλιοθήκη Bootstrap

Τα εργαλεία που παρέχει η Bootstrap για την επιτάχυνση και τη διευκόλυνση της ανάπτυξης διαδικτυακών εφαρμογών ήταν ιδιαίτερα χρήσιμα για τη διεκπεραίωση της συγκεκριμένης εφαρμογής, αλλά η υπερβολική χρήση της καθιστά δυσκολότερη τη μετάβαση σε άλλες τεχνολογίες στο μέλλον.

Αν και τα περισσότερα λειτουργικά κομμάτια της εφαρμογής είναι γραμμένα σε JavaScript, πράγμα που κάνει ιδιαίτερα εύκολη τη μεταφορά τους σε άλλα συστήματα, η εμφάνιση της εφαρμογής (μενού, modal, toast κτλ) εξαρτάται από την Bootstrap.

Μειώνοντας αυτή τη συσχέτιση, μπορούμε μελλοντικά να αντικαταστήσουμε την παρούσα τεχνολογία με μια που προσφέρει καλύτερη λειτουργικότητα, ασφάλεια και διάφορες άλλες λειτουργίες που μπορεί να είναι θεμιτές στο μέλλον, καθιστώντας τον κώδικα πιο φορητό και αρθρωτό.

### 7.8 Συμμετοχή χρηστών στη δημιουργία κουίζ

Στην εφαρμογή όλα τα κουίζ δημιουργούνται, τροποποιούνται και διαγράφονται από τον καθηγητή του μαθήματος. Αν και αυτό είναι το σύνηθες, θα ήταν ενδιαφέρον και χρήσιμο να προστεθεί η δυνατότητα δημιουργίας νέων κουίζ από τους χρήστες, προτείνοντας ιδέες για πιθανές ερωτήσεις που θα μπορούσαν να προστεθούν στα κουίζ ή ακόμη και δημοσιεύοντας αυτές του τις ιδέες μέσα στην εφαρμογή.

Ο καθηγητής θα έχει τη δυνατότητα να εγκρίνει ή να απορρίψει τις ερωτήσεις, ή να επικοινωνήσει με τον χρήστη για τυχόν αλλαγές που θα χρειαζόταν να γίνουν για να ενσωματωθούν οι προτάσεις του μέσα στην εφαρμογή.

Περαιτέρω, χρήσιμη θα ήταν και η δυνατότητα δημιουργίας εξατομικευμένων κουίζ από τους χρήστες με χρήση των ήδη υπαρχόντων κουίζ. Με βάση τα κουίζ που είναι ξεκλειδωμένα θα μπορούσε ο κάθε χρήστης να επιλέγει ποιες ερωτήσεις θα ήθελε να συμπεριλάβει στο καινούριο κουίζ καθώς και να τροποποιήσει όλες τις διαθέσιμες ρυθμίσεις. Έτσι θα μπορούσε για παράδειγμα να κάνει εξάσκηση σε ένα κουίζ που δυσκολεύεται με λίγο πιο εύκολες ρυθμίσεις (πχ αυξάνοντας τον διαθέσιμο χρόνο, αν υπάρχει χρονικό όριο ή να ενεργοποιήσει βοήθεια που δεν είναι διαθέσιμη.)

### 7.9 Προσθήκη παιχνιδιών σχετικών με το αντικείμενο μελέτης

Τα συγκεκριμένα παιχνίδια θα διαφέρουν από τις απλές ασκήσεις που παρέχονται από την εφαρμογή. Θα έχουν και αυτά παραμετροποιήσιμες ρυθμίσεις και θα επιτρέπουν πχ στους χρήστες να αξιολογήσουν τις ικανότητες του πάνω στην επίλυση ενός προβλήματος έναντι ενός εξελικτικού αλγόριθμου. Έτσι θα μπορούσαν να προστεθούν στοιχεία, όπως η συλλογή πόντων ή άλλοι τρόποι προσδιορισμού του νικητή, τα οποία θα αυξήσουν την ενεργή συμμετοχή των χρηστών και θα τους παρακινήσουν να πειραματιστούν με τις ρυθμίσεις και να εμβαθύνουν στις γνώσεις του πάνω στο πρόβλημα.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] “The importance of educational technology in teaching.” Accessed: Jan. 16, 2024. [Online]. Available: <https://cyberleninka.ru/article/n/the-importance-of-educational-technology-in-teaching/viewer>
- [2] N. Selwyn, *Education in a Digital World: Global Perspectives on Technology and Education*. Routledge, 2012.
- [3] *Popular Mechanics*. Hearst Magazines, 1954.
- [4] “Education App Revenue and Usage Statistics (2024),” Business of Apps. Accessed: Jan. 28, 2024. [Online]. Available: <https://www.businessofapps.com/data/education-app-market/>
- [5] J. Hamari, “Gamification,” in *The Blackwell Encyclopedia of Sociology*, John Wiley & Sons, Ltd, 2019, pp. 1–3. doi: 10.1002/9781405165518.wbeos1321.
- [6] “Why are Educational Apps gaining popularity these days?,” Web And Mobile App Development Company. Accessed: Jan. 28, 2024. [Online]. Available: <https://www.digital-dividend.com/en/why-are-educational-apps-gaining-popularity-these-days/>
- [7] “Important of Educational App in 2021.” Accessed: Jan. 28, 2024. [Online]. Available: <https://www.linkedin.com/pulse/important-educational-app-2021-wowinfotech>
- [8] “Top language learning apps by downloads 2023,” Statista. Accessed: Jan. 28, 2024. [Online]. Available: <https://www.statista.com/statistics/1239522/top-language-learning-apps-downloads/>
- [9] “Learn a language for free,” Duolingo. Accessed: Jan. 28, 2024. [Online]. Available: <https://www.duolingo.com/>
- [10] “Duolingo\_Efficacy\_Study.pdf.” Accessed: Jan. 28, 2024. [Online]. Available: [https://www.languagezen.com/pt/about/english/Duolingo\\_Efficacy\\_Study.pdf](https://www.languagezen.com/pt/about/english/Duolingo_Efficacy_Study.pdf)
- [11] O. E. Hetzroni and I. Banin, “The Effect of Educational Software, Video Modelling and Group Discussion on Social-Skill Acquisition Among Students with Mild Intellectual Disabilities,” *Journal of Applied Research in Intellectual Disabilities*, vol. 30, no. 4, pp. 757–773, 2017, doi: 10.1111/jar.12271.
- [12] “Genetic-Algorithm-Guide-[Tom-Mathew].pdf.” Accessed: Jan. 28, 2024. [Online]. Available: [http://datajobstest.com/data-science-repo/Genetic-Algorithm-Guide-\[Tom-Mathew\].pdf](http://datajobstest.com/data-science-repo/Genetic-Algorithm-Guide-[Tom-Mathew].pdf)
- [13] A. Venables and G. Tan, “A ‘Hands on’ Strategy for Teaching Genetic Algorithms to Undergraduates,” *Journal of Information Technology Education: Research*, vol. 6, no. 1, pp. 249–256, Jan. 2007.
- [14] Z. Michalewicz and M. Michalewicz, “Teaching evolutionary algorithms,” in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Jul. 1999, pp. 1702–1706 Vol. 3. doi: 10.1109/CEC.1999.785479.
- [15] S. Salcedo-Sanz, J. A. Portilla-Figueras, E. G. Ortiz-Garcia, A. M. Perez-Bellido, and X. Yao, “Teaching Advanced Features of Evolutionary Algorithms Using Japanese Puzzles,” *IEEE Transactions on Education*, vol. 50, no. 2, pp. 151–156, May 2007, doi: 10.1109/TE.2007.893170.
- [16] J. Barrachina, P. Garrido, M. Fogue, J. A. Sanguesa, and F. J. Martinez, “Understanding Evolutionary Algorithms through Interactive Graphical Applications,” vol. 10, no. 10, 2016.
- [17] “JavaScript - MDN Web Docs Glossary: Definitions of Web-related terms | MDN.” Accessed: Dec. 28, 2023. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/JavaScript>
- [18] “Document Object Model (DOM) - Web APIs | MDN.” Accessed: Dec. 29, 2023. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model/Introduction](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)
- [19] “Fetch API - Web APIs | MDN.” Accessed: Dec. 22, 2023. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API)
- [20] “How many pages of text will one megabyte hold?” Accessed: Dec. 29, 2023. [Online]. Available: [https://pc.net/helpcenter/how\\_much\\_text\\_in\\_one\\_megabyte](https://pc.net/helpcenter/how_much_text_in_one_megabyte)

- [21] “Clipboard API - Web APIs | MDN.” Accessed: Dec. 29, 2023. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Clipboard\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Clipboard_API)
- [22] “Canvas API - Web APIs | MDN.” Accessed: Dec. 29, 2023. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API)
- [23] “Web Workers API - Web APIs | MDN.” Accessed: Dec. 29, 2023. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Workers\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API)
- [24] “Service Worker API - Web APIs | MDN.” Accessed: Dec. 29, 2023. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Service\\_Worker\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API)
- [25] “Web Components - Web APIs | MDN.” Accessed: Dec. 29, 2023. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_components](https://developer.mozilla.org/en-US/docs/Web/API/Web_components)
- [26] “What is open source? | Opensource.com.” Accessed: Dec. 22, 2023. [Online]. Available: <https://opensource.com/resources/what-open-source>
- [27] “Responsive Web Design (RWD) and User Experience,” Nielsen Norman Group. Accessed: Dec. 22, 2023. [Online]. Available: <https://www.nngroup.com/articles/responsive-web-design-definition/>
- [28] “What is user interface (UI)? Definition from SearchAppArchitecture,” App Architecture. Accessed: Dec. 22, 2023. [Online]. Available: <https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI>
- [29] “Material Design,” Material Design. Accessed: Dec. 22, 2023. [Online]. Available: <https://m3.material.io/>
- [30] “HTML Standard.” Accessed: Dec. 22, 2023. [Online]. Available: <https://html.spec.whatwg.org/multipage/>
- [31] “JSON.” Accessed: Dec. 22, 2023. [Online]. Available: <https://www.json.org/json-el.html>
- [32] “plotly/plotly.js.” Plotly, Dec. 21, 2023. Accessed: Dec. 22, 2023. [Online]. Available: <https://github.com/plotly/plotly.js>
- [33] “PDF.js.” Mozilla, Dec. 26, 2023. Accessed: Dec. 26, 2023. [Online]. Available: <https://github.com/mozilla/pdf.js>
- [34] “Express - Node.js web application framework.” Accessed: Dec. 27, 2023. [Online]. Available: <https://expressjs.com/>
- [35] “Introduction | Socket.IO.” Accessed: Dec. 27, 2023. [Online]. Available: <https://socket.io/docs/v4/>
- [36] “SVG Repo - Free SVG Vectors and Icons,” SVG Repo. Accessed: Dec. 23, 2023. [Online]. Available: <https://www.svgrepo.com/>
- [37] “Introduction - SVG 1.0 - 20010904.” Accessed: Dec. 23, 2023. [Online]. Available: <https://www.w3.org/TR/SVG10/intro.html>
- [38] “Download FREE Sound Effects at ZapSplat,” ZapSplat - Download free sound effects. Accessed: Dec. 23, 2023. [Online]. Available: <https://www.zapsplat.com/>
- [39] “The Impact of Ambient Sound on Emotions - ProQuest.” Accessed: Dec. 23, 2023. [Online]. Available: <https://www.proquest.com/openview/224ab93555452e81df28b08e04fe0787/1?pq-origsite=gscholar&cbl=18750>
- [40] F. Lardinois, “Microsoft Launches Visual Studio Code, A Free Cross-Platform Code Editor For OS X, Linux And Windows,” TechCrunch. Accessed: Dec. 23, 2023. [Online]. Available: <https://techcrunch.com/2015/04/29/microsoft-shocks-the-world-with-visual-studio-code-a-free-code-editor-for-os-x-linux-and-windows/>
- [41] “How to Run Code in VS Code,” Alphr. Accessed: Dec. 23, 2023. [Online]. Available: <https://www.alphr.com/vs-code-run-code/>
- [42] “About | Inkscape.” Accessed: Dec. 23, 2023. [Online]. Available: <https://inkscape.org/about/>
- [43] “JavaScript modules - JavaScript | MDN.” Accessed: Dec. 31, 2023. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules>
- [44] “Left-Side Vertical Navigation on Desktop: Scalable, Responsive, and Easy to Scan,” Nielsen Norman Group. Accessed: Jan. 14, 2024. [Online]. Available: <https://www.nngroup.com/articles/vertical-nav/>