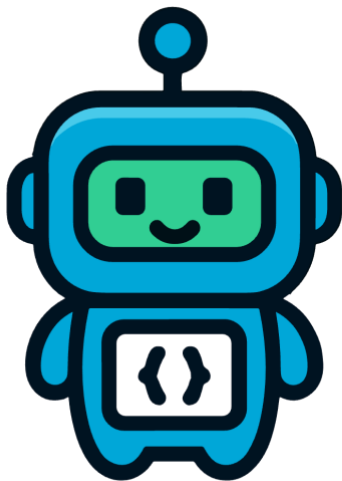




ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
«Διαδικτυακό παιχνίδι εκμάθησης προγραμματισμού  
για την γλώσσα C»



CodeIT

Του φοιτητή  
Ανεσιάδη Θεοχάρη  
Αρ. Μητρώου: 185147

Επιβλέπων  
Γουλιάνας Κωνσταντίνος  
Βαθμίδα: Καθηγητής

Θεσσαλονίκη 2025

Τίτλος Δ.Ε. Διαδικτυακό παιχνίδι εκμάθησης προγραμματισμού για την γλώσσα C

Κωδικός Δ.Ε. 25162

Ονοματεπώνυμο φοιτητή Ανεσιάδης Θεοχάρης

Ονοματεπώνυμο εισηγητή Γουλιάνας Κωνσταντίνος

Ημερομηνία ανάληψης Δ.Ε. 10/3/2025

Ημερομηνία περάτωσης Δ.Ε. 11/9/2025

*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 Ανεσιάδη Θεοχάρη 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 αυτήν την διπλωματική εργασία στην οικογένειά μου.»*



## Πρόλογος

Ο προγραμματισμός αποτελεί μία από τις σημαντικότερες δεξιότητες της ψηφιακής εποχής. Δεν είναι μόνο εργαλείο για τη δημιουργία λογισμικού, αλλά συμβάλλει στην καλλιέργεια λογικής σκέψης, στην ανάπτυξη της δημιουργικότητας και στην επίλυση προβλημάτων. Παρά τη σημασία του, η πρώτη επαφή με τον προγραμματισμό συχνά αποδεικνύεται δύσκολη, ειδικά για όσους δεν διαθέτουν προηγούμενη εμπειρία. Η πολυπλοκότητα των συντακτικών κανόνων, τα τεχνικά εμπόδια και η έλλειψη παρακίνησης αποτελούν συχνά αποθαρρυντικούς παράγοντες.

Η παρούσα εργασία εστιάζει στη δημιουργία μιας εκπαιδευτικής εφαρμογής που διευκολύνει την εκμάθηση της γλώσσας C. Μέσα από έναν συνδυασμό θεωρίας, πρακτικής και στοιχείων παιχνιδιοποίησης, η εφαρμογή προσφέρει μια προσιτή και ευχάριστη εμπειρία μάθησης. Ο χρήστης έχει τη δυνατότητα να εξασκείται σε πραγματικό χρόνο μέσω ενσωματωμένου compiler, ενώ το φιλικό περιβάλλον και η σταδιακή παρουσίαση της ύλης μειώνουν τα εμπόδια κατανόησης και ενισχύουν την αυτοπεποίθηση του μαθητή.

Η εφαρμογή απευθύνεται κυρίως σε αρχάριους, ανεξαρτήτως ηλικίας και υποβάθρου, που θέλουν να κάνουν τα πρώτα τους βήματα στον προγραμματισμό. Με αυτόν τον τρόπο, η εκμάθηση της C γίνεται πιο άμεση και προσιτή, συνδυάζοντας την εκπαιδευτική αξία με μια συμμετοχική και διαδραστική εμπειρία που μπορεί να λειτουργήσει ως γέφυρα για περαιτέρω ενασχόληση με την πληροφορική.

## Περίληψη

Η συγκεκριμένη πτυχιακή εργασία αναλύει την ανάπτυξη του CodeIT, μιας διαδικτυακής πλατφόρμας για τη διδασκαλία της γλώσσας προγραμματισμού C. Σκοπός της είναι να εξασφαλίσει μια ευχάριστη και αποδοτική εμπειρία μάθησης για τους νέους χρήστες. Ο προγραμματισμός είναι μία από τις πιο κρίσιμες δεξιότητες στη σημερινή ψηφιακή εποχή, ωστόσο συχνά έχει μια δύσκολη καμπύλη εκμάθησης που μπορεί να αποθαρρύνει τους αρχάριους μαθητές. Η εφαρμογή χρησιμοποιεί σύγχρονες τεχνολογίες, όπως η React, για να προσφέρει ένα φιλόξενο και αποδοτικό περιβάλλον εκμάθησης. Το CodeIT χρησιμοποιεί τη μέθοδο της παιχνιδοποίησης, παρέχοντας πόντους εμπειρίας, επίπεδα, συνεχείς επιδόσεις και οπτικά στοιχεία που δείχνουν την πρόοδο του χρήστη. Με αυτόν τον τρόπο, διατηρεί το ενδιαφέρον των χρηστών και ενισχύει το κίνητρό τους για σταθερή συμμετοχή. Η εκπαιδευτική διαδικασία συνδυάζει την παρουσίαση θεωρητικών εννοιών με μια απλή και κατανοητή γλώσσα. Περιλαμβάνει διαδραστικά κουίζ που επιτρέπουν τον άμεσο έλεγχο της κατανόησης, καθώς επίσης έναν ενσωματωμένο online compiler. Αυτός ο compiler δίνει τη δυνατότητα εκτέλεσης κώδικα C απευθείας στην πλατφόρμα, χωρίς να χρειάζεται η χρήση εξωτερικών εργαλείων ή εγκαταστάσεων. Το CodeIT απευθύνεται σε μαθητές, φοιτητές και οποιονδήποτε ενδιαφέρεται να αποκτήσει γνώσεις στον προγραμματισμό μέσω πρακτικών και διαδραστικών μεθόδων. Συνδυάζοντας θεωρία, πρακτική άσκηση και στοιχεία παιχνιδοποίησης, στοχεύει να διευκολύνει την είσοδο στη γλώσσα C και να παρέχει μια θετική και ενθαρρυντική πρώτη επαφή με τον κόσμο του προγραμματισμού.

# «Online programming learning game for the C language»

«Anesiadis Theocharis»

## **Abstract**

This thesis presents the case for CodeIT, an online tool for teaching the C programming language that is meant to be a fun yet meaningful learning experience for new users. The importance of programming as a skill set in the digital age cannot be understated, yet it can be a difficult skill to learn and often discourages beginners due to its steep learning curve. The application uses modern technologies, like React, to create an inviting and non-threatening environment when learning. CodeIT incorporates a gamification option to offer experience points, levels, streaks, and visual means of tracking progress. This increases user retention and solidifies their motivation to keep coming back. The learning is based on theory, presented simply and easily understood, and combined with interactive quizzes to verify understanding of the subject immediately after theory, along with an online compiler. The compiler allows users to write and execute C code in the app without having to use another tool or installation. CodeIT is for students, college-level learners, or even anyone who would like to learn programming, in such a practical and interactive manner. The aim is to combine theory with practical and interactive, enjoyable gamified experiences to lower the barrier to enter the C language and create a positive and motivational first interaction with the programming world.

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους εκείνους που με στήριξαν κατά τη διάρκεια της ακαδημαϊκής μου θητείας είτε ψυχολογικά, είτε ηθικά, ακόμα και οικονομικά. Πρώτα από όλους, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή, κ. Γουλιάννα Κωνσταντίνο, για την εμπιστοσύνη, την καθοδήγηση και την εμπειρία που μου προσέφερε τόσο απλόχερα. Θα ήθελα επίσης να ευχαριστήσω την οικογένειά μου, την κοπέλα και τους φίλους μου, που με συνόδευσαν σε αυτό το ταξίδι. Τέλος, εκφράζω την ευγνωμοσύνη μου στους συμφοιτητές και καθηγητές μου, για όλες τις απόψεις. Σας ευχαριστώ όλους θερμά.

# Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract .....	vii
Ευχαριστίες .....	viii
Περιεχόμενα .....	ix
Κατάλογος Σχημάτων .....	xii
Συντομογραφίες .....	xiii
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Παιχνιδοποιημένη Μάθηση.....	2
1.2 Η Σημασία της Παρακίνησης στη Διαδικασία Μάθησης.....	3
Κεφάλαιο 2ο: Παρόμοιες Εφαρμογές .....	5
2.1 Duolingo .....	5
2.2 Mimo .....	8
2.3 SoloLearn.....	10
2.4 Knackbout.....	11
2.5 LeetCode.....	12
Κεφάλαιο 3ο: CodeIT.....	14
3.1 Σκοπός και Κανόνες .....	14
3.2 Σχεδιασμός .....	14
3.3 Χαρακτηριστικά και διαφορές με παρόμοιες Εφαρμογές .....	16
Κεφάλαιο 4ο: Τεχνολογία React .....	19
4.1 Βασικές αρχές λειτουργίας.....	19
4.1.1 Component-Based Αρχιτεκτονική .....	19
4.1.2 Props και State.....	20
4.1.3 Εικονικό DOM και διαδικασία συμφιλίωσης .....	20
4.1.4 Άγκιστρα (Hooks) .....	21
4.1.5 Δηλωτική Προσέγγιση .....	21
4.1.6 Πλοήγηση και αρχιτεκτονική εφαρμογών μίας σελίδας.....	21
4.1.7 Αντιδραστικότητα και άμεση ενημέρωση διεπαφής.....	21
4.2 Πλεονεκτήματα και αδυναμίες της React.....	22
4.2.1 Πλεονεκτήματα της React.....	22
4.2.2 Αδυναμίες της React .....	22
Κεφάλαιο 5ο: Υλικό Ασκήσεων.....	24
5.1 Κριτήρια δημιουργίας ασκήσεων .....	24

5.2	Είδη ασκήσεων στην πλατφόρμα .....	24
5.3	Σύνδεση θεωρίας και πράξης.....	26
5.4	Παιχνιδοποίηση και αξιολόγηση .....	26
Κεφάλαιο 6ο:	Συμπληρωματικές Τεχνολογίες και Εργαλεία Ανάπτυξης.....	28
6.1	Judge0 API .....	28
6.2	Βιβλιοθήκες και Εργαλεία React.....	28
Κεφάλαιο 7ο:	Supabase και βάσεις δεδομένων .....	30
7.1	Τι είναι το Supabase .....	30
7.2	Χρήση του Supabase στο CodeIT .....	30
7.3	Δομή της Βάσης Δεδομένων .....	31
7.4	Πλεονεκτήματα Χρήσης του Supabase .....	32
Κεφάλαιο 8ο:	Κώδικας.....	33
8.1	Αρχιτεκτονική.....	33
8.1.1	Πλεονεκτήματα μιας σωστά δομημένης αρχιτεκτονικής κώδικα .....	34
8.1.2	ES Modules και Import/Export .....	34
8.2	Σελίδες εφαρμογής .....	34
8.2.1	Κύριες σελίδες.....	35
8.2.2	Σελίδες λογαριασμού και διαχείρισης.....	42
8.3	Κοινά Στοιχεία Διεπαφής .....	45
8.3.1	Header .....	45
8.3.2	Footer .....	45
8.3.3	Card .....	46
8.3.4	Editor.....	47
8.4	Επικοινωνία με Backend και Διαχείριση Δεδομένων .....	48
8.4.1	Επικοινωνία με Supabase.....	48
8.4.2	Ενσωμάτωση Judge0 API .....	49
8.4.3	Σύνδεση Frontend και Backend στη ροή της μάθησης .....	50
8.5	Πολυγλωσσική Υποστήριξη.....	50
8.5.1	Ενσωμάτωση react-i18next .....	50
8.5.2	Δυναμική αλλαγή γλώσσας.....	51
8.5.3	Πολυγλωσσικό περιεχόμενο μαθημάτων .....	51
8.5.4	Σύνδεση με Components .....	52
8.6	Υλοποίηση UI/UX.....	52
8.7	Υλοποίηση Στοιχείων Gamification .....	53
Κεφάλαιο 9ο:	Συμπέρασμα .....	55
9.1	Μελλοντικές προσθήκες.....	55
9.2	Συνολική αποτίμηση.....	56

Βιβλιογραφία..... 58

## Κατάλογος Σχημάτων

Σχήμα 1: Η παιχνιδοποιημένη μάθηση και τα χαρακτηριστικά της .....	3
Σχήμα 2: Επιλογές πρώτης εισόδου στο Duolingo με τη μασκότ Duo.....	6
Σχήμα 3: Σελίδα Leaderboard στο Duolingo .....	7
Σχήμα 4: Στάδιο παιχνιδιού στο Duolingo.....	8
Σχήμα 5: Επιλογή learning path στο Mimo .....	8
Σχήμα 6: Στάδιο παιχνιδιού στο Mimo .....	10
Σχήμα 7: Παράδειγμα άσκησης στο SoloLearn .....	11
Σχήμα 8: Περιβάλλον εργασίας στο LeetCode με παράδειγμα προβλήματος. ....	13
Σχήμα 9: Δομή μαθημάτων .....	15
Σχήμα 10: Χρώματα που χρησιμοποιήθηκαν στην εφαρμογή σε μορφή HEX.....	16
Σχήμα 11: Λογότυπο React και κύρια χαρακτηριστικά.....	19
Σχήμα 12: props και states .....	20
Σχήμα 13: Παραδείγματα ασκήσεων πολλαπλής επιλογής και σωστού/λάθους .....	25
Σχήμα 14: Παραδείγματα ασκήσεων συμπλήρωσης κενού και εκτέλεσης κώδικα.....	25
Σχήμα 15: Πως λειτουργεί το Judge0 στο CodeIT .....	29
Σχήμα 16: Περιβάλλον Supabase.....	30
Σχήμα 17: Παράδειγμα δομής στην βάση δεδομένων του Supabase.....	31
Σχήμα 18: Αρχική σελίδα.....	36
Σχήμα 19: Σελίδα Lessons .....	37
Σχήμα 20: Εμφάνιση βοήθειας.....	38
Σχήμα 21: Σελίδα LessonScreen .....	39
Σχήμα 22: Σελίδα Dashboard .....	41
Σχήμα 23: Σελίδα Practice .....	42
Σχήμα 24: Σελίδα Signup (αριστερά) και σελίδα Login (δεξιά).....	43
Σχήμα 25: Σελίδα Admin Dashboard.....	44
Σχήμα 26: Header ως ανώνυμος χρήστης (αριστερά) και ως συνδεδεμένος χρήστης (δεξιά). ....	45
Σχήμα 27: Διάφορες παραλλαγές των Card components.....	46
Σχήμα 28: Λανθασμένη έξοδος στον Editor .....	47
Σχήμα 29: Παράδειγμα κώδικα που δείχνει την ανάκτηση δεδομένων μαθημάτων από το Supabase.....	49
Σχήμα 30: Αλλαγή γλώσσας μέσω εικονιδίων στο Footer .....	51
Σχήμα 31: Στιγμιότυπο οθόνης στην ολοκλήρωση του μαθήματος.....	53

## Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΠΙΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
SPA	Single Page Application
DOM	Document Object Model
ΒΔ	Βάση Δεδομένων
CSS	Cascading Style Sheets
JSX	JavaScript XML
JSON	JavaScript Object Notation
API	Application Programming Interface
SDK	Software Development Kit
UI	User Interface
UX	User Experience
IDE	Integrated Development Environment
RLS	Row-Level Security
OAuth	Open Authorization



## Κεφάλαιο 1ο: Εισαγωγή

Ο προγραμματισμός είναι μία από τις πιο σημαντικές δεξιότητες που μπορεί να διαθέτει κάποιος στην εποχή που διανύουμε. Η ικανότητα του προγραμματισμού έχει πολλές χρήσεις και μάλιστα, σε πολλούς τομείς, όπως η δημιουργία λογισμικού και εφαρμογών, που είναι ευρέως γνωστές, μέχρι και την τεχνητή νοημοσύνη, την ανάλυση δεδομένων ακόμα και την διάγνωση ασθενειών. Η εκμάθηση προγραμματισμού δεν αφορά μόνο την απόκτηση τεχνικών δεξιοτήτων, αποτελεί βασικό παράγοντα για την ανάπτυξη λογικής σκέψης, την ικανότητα επίλυσης προβλημάτων και της δημιουργικότητας. Αυτά είναι μόνο μερικά από τα πολλά στοιχεία που κάνουν τον προγραμματισμό μια δεξιότητα που γίνεται όλο και πιο περιζήτητη τόσο στην αγορά εργασίας, όσο και στον τομέα της εκπαίδευσης.

Μεταξύ των πολλών γλωσσών προγραμματισμού που έχουν δημιουργηθεί, η γλώσσα C έχει μια ιδιαίτερη θέση. Αναπτύχθηκε τη δεκαετία του 1970 και έκτοτε παραμένει θεμελιώδης για τον τομέα της πληροφορικής. Χρησιμοποιείται ευρέως στην ανάπτυξη λειτουργικών συστημάτων, σε εφαρμογές χαμηλού επιπέδου, σε ενσωματωμένα συστήματα και σε λογισμικό που απαιτεί υψηλή απόδοση. Η γλώσσα C διακρίνεται για την ικανότητά της να παρέχει άμεση πρόσβαση στο υλικό του υπολογιστή, δίνοντας τη δυνατότητα στον προγραμματιστή να χειρίζεται τη μνήμη και τους πόρους του συστήματος με μεγάλη ακρίβεια. Αυτή η δυνατότητα το καθιστά ένα εξαιρετικό εργαλείο για να κατανοήσουμε σε βάθος πώς λειτουργούν τα υπολογιστικά συστήματα.

Η εκμάθηση της γλώσσας C παρέχει στους μαθητές μια ισχυρή βάση, η οποία τους επιτρέπει να επεκτείνουν τις γνώσεις τους σε άλλες προγραμματιστικές γλώσσες, όπως η C++, η Java ή η Python. Οι έννοιες που διδάσκονται μέσω της γλώσσας C, όπως οι μεταβλητές, οι δομές ελέγχου, η διαχείριση μνήμης και οι δείκτες, είναι θεμελιώδεις για την κατανόηση του τρόπου λειτουργίας του κώδικα σε πιο χαμηλό επίπεδο. Πολλές σύγχρονες γλώσσες δεν παρέχουν την ίδια εις βάθος προσέγγιση σε αυτά τα θέματα.

Αν και η γλώσσα προγραμματισμού C έχει πολλά πλεονεκτήματα και είναι χρήσιμη, η εκμάθησή της μπορεί να παρουσιάσει προκλήσεις, κυρίως για τους αρχάριους. Η σύνταξή της, έστω κι αν είναι απλή και ευέλικτη, μπορεί να φαίνεται αυστηρή και απαιτητική σε άτομα χωρίς προηγούμενη εμπειρία. Η κατανόηση εννοιών όπως η διαχείριση της μνήμης, οι δείκτες και οι διαφορετικοί τύποι δεδομένων μπορεί να είναι δύσκολη για αρκετούς μαθητές. Επιπλέον, το γεγονός ότι δεν υπάρχουν διαθέσιμες ποιοτικές βιβλιοθήκες όπως σε άλλες πιο πρόσφατες γλώσσες σημαίνει ότι οι μαθητές πρέπει να αναπτύσσουν πολλές λειτουργίες από την αρχή. Αυτό απαιτεί χρόνο και αφοσίωση.

Η αρχική επαφή με τον προγραμματισμό συνήθως φέρνει πολλές προκλήσεις. Οι νέοι μαθητές συναντούν ένα άγνωστο λεξιλόγιο, που περιλαμβάνει όρους όπως η μεταγλώττιση, τα σφάλματα σύνταξης, οι μεταβλητές, οι δείκτες και οι βρόχοι επανάληψης. Η κατανόηση αυτών των εννοιών απαιτεί μια σταδιακή εξοικείωση με τη θεωρία και την πράξη. Η απουσία άμεσων ανατροφοδοτήσεων και η δυσκολία στην οπτικοποίηση της λειτουργίας του κώδικα μπορεί να αποθαρρύνει τους αρχάριους, μειώνοντας το κίνητρο για περαιτέρω ενασχόληση.

Η διαδικασία εκμάθησης του προγραμματισμού χρειάζεται καλά οργανωμένο υλικό που θα παρουσιάζει τις έννοιες σταδιακά, συνδυάζοντας τη θεωρία με την πρακτική εφαρμογή. Οι πιο αποδοτικές μέθοδοι είναι εκείνες που προσφέρουν ενεργή συμμετοχή των μαθητών, μέσω ασκήσεων, παραδειγμάτων και πειραματισμού με τον κώδικα. Η πρακτική εξάσκηση ενισχύει την κατανόηση και επιτρέπει στους μαθητές να αναγνωρίζουν και να διορθώνουν τα λάθη τους, αποκτώντας έτσι περισσότερη αυτοπεποίθηση στη χρήση της γλώσσας.

Η διαδικασία του προγραμματισμού, και ειδικότερα η εκμάθηση της γλώσσας C, είναι σίγουρα απαιτητική αλλά και πολύ χρήσιμη [1]. Παρόλο που υπάρχουν προκλήσεις, η κατανόηση των βασικών αρχών που διδάσκει η C παρέχει στον μαθητή τα απαραίτητα εργαλεία για να πορευτεί με επιτυχία στο πεδίο της πληροφορικής. Η οργανωμένη και συστηματική εκμάθηση της μπορεί να αποτελέσει τον

## Κεφάλαιο 1ο:

βασικό πυλώνα για την ενασχόληση με πιο εξειδικευμένα πεδία, όπως είναι η ανάπτυξη λογισμικού, η μηχανική λογισμικού και η βελτιστοποίηση επιδόσεων.

### 1.1 Παιγνιδοποιημένη Μάθηση

Η παιγνιδοποιημένη εκπαίδευση, ή αλλιώς gamified learning, είναι μια σύγχρονη διδακτική μέθοδος που ενσωματώνει στοιχεία και μηχανισμούς από το κόσμο των παιχνιδιών. Στόχος της είναι να αυξήσει το κίνητρο και τη συμμετοχή των μαθητών στη διαδικασία μάθησης. Η βασική σκέψη προέρχεται από την παρατήρηση ότι οι άνθρωποι, ανεξαρτήτως της ηλικίας τους, αντιδρούν θετικά σε δραστηριότητες που περιλαμβάνουν προκλήσεις, άμεση ανατροφοδότηση και αίσθηση επιτυχίας [2] [3]. Ενσωματώνοντας αυτά τα στοιχεία σε εκπαιδευτικά πλαίσια, η διαδικασία μάθησης μπορεί να εξελιχθεί από μια στατική και συνήθως απαιτητική εμπειρία σε κάτι πιο ευχάριστο, ζωντανό και ελκυστικό.

Στην πραγματικότητα, η παιγνιδοποίηση δεν σημαίνει ότι το μάθημα γίνεται παιχνίδι. Αντίθετα, αναγνωρίζει και ενσωματώνει στοιχεία που συναντούμε σε παιχνίδια, με σκοπό να εμπλουτίσει την εμπειρία των μαθητών. Αυτά τα χαρακτηριστικά περιλαμβάνουν τη συγκέντρωση πόντων ως ανταμοιβή για την ολοκλήρωση διαφόρων δραστηριοτήτων, την επίτευξη βαθμίδων που δείχνουν πρόοδο, την απόκτηση μεταλλίων ή εμβλημάτων προς αναγνώριση των επιτυχιών και τις προκλήσεις που ενθαρρύνουν τον μαθητή να ξεπεράσει δυσκολίες. Η παρουσία οπτικών και ηχητικών ερεθισμάτων που συνοδεύουν αυτές τις δραστηριότητες ενισχύει ακόμα περισσότερο την αίσθηση της ολοκλήρωσης και της ικανοποίησης.

Η χρήση της παιγνιδοποίησης στην εκπαίδευση έχει αποκτήσει μεγάλη αποδοχή τα τελευταία είκοσι χρόνια, κυρίως λόγω της προόδου στις ψηφιακές τεχνολογίες, οι οποίες διευκολύνουν την εφαρμογή της [3] [4]. Στα εκπαιδευτικά περιβάλλοντα, οι μηχανισμοί παιχνιδιού μπορούν να αξιοποιηθούν για να ενθαρρύνουν τους μαθητές να επιμείνουν στην προσπάθειά τους, να παρακολουθούν την πρόδό τους και να συμμετέχουν πιο ενεργά με το διδακτικό υλικό. Η αίσθηση της προόδου, όταν εκφράζεται με ξεκάθαρο και ελκυστικό τρόπο, ενισχύει την ικανοποίηση και προάγει τη διάρκεια στην προσπάθεια, ακόμα και σε δύσκολες ή απαιτητικές καταστάσεις.

Η εκπαιδευτική αξία της παιγνιδοποίησης δεν περιορίζεται μόνον στη διαρκή ενθάρρυνση. Με την άμεση παροχή ανατροφοδότησης, οι μαθητές μπορούν να αντιληφθούν αμέσως αν η απάντησή τους είναι σωστή ή αν πρέπει να επανεξετάσουν τη μέθοδό τους. Η αδιάλειπτη αυτή συνεργασία μεταξύ προσπάθειας και αποτελέσματος διαμορφώνει έναν κύκλο μάθησης που προάγει τη σταδιακή εξέλιξη. Επίσης, η ύπαρξη σαφώς καθορισμένων στόχων, είτε μικρών είτε μεγάλων, παρέχει στον μαθητή την αίσθηση ότι προχωράει σε μια συγκεκριμένη κατεύθυνση και ότι κάθε βήμα έχει σημασία.

Η παιγνιδοποίηση μπορεί επίσης να βοηθήσει στην ανάπτυξη δεξιοτήτων που επεκτείνονται πέρα από το συγκεκριμένο επιστημονικό πεδίο [4]. Η αποτελεσματική διαχείριση του χρόνου, η στρατηγική σκέψη, η συνεργασία με συμμαθητές σε ομαδικές δραστηριότητες και η επίμονη προσπάθεια μπροστά σε προκλήσεις είναι μόνο ορισμένα από τα οφέλη που μπορεί να προκύψουν. Ιδιαίτερα σε τομείς που θεωρούνται δύσκολοι, όπως τα μαθηματικά, οι φυσικές επιστήμες ή ο προγραμματισμός, η χρήση παιχνιδιών μπορεί να λειτουργήσει ως μέσο που μειώνει την πίεση και καλλιεργεί μια πιο θετική προσέγγιση στη διαδικασία της μάθησης.

Ένα κρίσιμο στοιχείο της παιγνιδοποίησης είναι η προσαρμογή της εμπειρίας. Με τη διαφοροποίηση του περιεχομένου και των προκλήσεων σύμφωνα με το επίπεδο και τις ανάγκες του μαθητή, η διαδικασία αποκτά πιο προσωπικό χαρακτήρα και συνδέεται καλύτερα με τους στόχους του. Με αυτόν τον τρόπο, αποφεύγεται η εντύπωση ότι το περιεχόμενο είναι είτε πολύ απλό είτε εξαιρετικά δύσκολο, γεγονός που μπορεί να προκαλέσει απώλεια ενδιαφέροντος.

Η ανάπτυξη κινήτρων μέσω της παιγνιδοποίησης συνδέεται και με την ψυχολογική θεωρία της αυτοκαθοριζόμενης μάθησης, σύμφωνα με την οποία οι άνθρωποι κινητοποιούνται όταν ικανοποιούνται βασικές ανάγκες, όπως η αυτονομία, η ικανότητα και η κοινωνική σύνδεση [5] [6]. Οι μηχανισμοί παιχνιδιού μπορούν να καλύψουν αυτές τις ανάγκες, δίνοντας στον μαθητή την αίσθηση

ελέγχου της πορείας του, αναγνωρίζοντας και ανταμείβοντας τις δεξιότητές του και δημιουργώντας ευκαιρίες για αλληλεπίδραση με άλλους.

Στον τομέα της εκπαίδευσης, η παιχνιδοποίηση δεν είναι αποκλειστικά ψηφιακή. Μπορεί να χρησιμοποιηθεί και σε πραγματικούς χώρους μάθησης μέσω διαγωνισμών, επιτραπέζιων παιχνιδιών ή δραστηριοτήτων ρόλων. Ωστόσο, οι ψηφιακές τεχνολογίες έχουν επεκτείνει σημαντικά τις δυνατότητες εφαρμογής τους. Αυτό καθιστά εφικτή την ταυτόχρονη παρακολούθηση της προόδου πολλών μαθητών, τη γρήγορη παροχή ανατροφοδότησης και την εύκολη προσαρμογή του περιεχομένου.



Σχήμα 1: Η παιχνιδοποιημένη μάθηση και τα χαρακτηριστικά της

Συνολικά, η παιχνιδοποιημένη μάθηση μπορεί να αλλάξει ριζικά τον τρόπο που οι μαθητές αλληλεπιδρούν με το εκπαιδευτικό υλικό, κάνοντάς την πιο ελκυστική και αποδοτική. Μέσω της πρόκλησης, των ανταμοιβών και της αίσθησης προόδου, οι μαθητές ενθαρρύνονται να συμμετέχουν ενεργά και να διατηρούν τη δέσμευσή τους σε μακροχρόνια βάση. Έτσι αναπτύσσουν θετικές μαθησιακές συνήθειες που θα τους ακολουθούν καθ' όλη την εκπαιδευτική τους πορεία [4].

## 1.2 Η Σημασία της Παρακίνησης στη Διαδικασία Μάθησης

Η παρακίνηση είναι ένας από τους πιο σημαντικούς παράγοντες που επηρεάζουν την αποτελεσματικότητα της διαδικασίας μάθησης. Όποιο κι αν είναι το γνωστικό αντικείμενο, η παρουσία κινήτρων παίζει καθοριστικό ρόλο στην αφοσίωση, την επιμονή και τη συμμετοχή των μαθητών. Η μάθηση δεν είναι απλώς μια παθητική διαδικασία. Αντίθετα, προϋποθέτει συνεχή προσπάθεια, αφοσίωση και διάθεση για πειραματισμούς. Χωρίς εσωτερικά ή εξωτερικά κίνητρα, ακόμα και το καλύτερο οργανωμένο εκπαιδευτικό υλικό μπορεί να μην καταφέρει να προσελκύσει ή να διατηρήσει την προσοχή του μαθητή [5] [6].

Η παρακίνηση για τη μάθηση μπορεί να κατηγοριοποιηθεί σε δύο κύριους τύπους: την εσωτερική και την εξωτερική. Η εσωτερική παρακίνηση προέρχεται από την επιθυμία του ατόμου να αποκτήσει γνώσεις και να κατανοήσει ένα θέμα, είτε λόγω προσωπικού ενδιαφέροντος είτε εξαιτίας της ικανοποίησης που προσφέρει η διαδικασία της μάθησης. Αντίθετα, η εξωτερική παρακίνηση στηρίζεται σε παράγοντες που προέρχονται από το περιβάλλον, όπως είναι οι ανταμοιβές, οι βαθμολογίες, τα πιστοποιητικά ή η κοινωνική αναγνώριση. Αμφότερες οι μορφές κινήτρου μπορούν να έχουν θετικά αποτελέσματα, αλλά η εσωτερική παρακίνηση θεωρείται πιο βιώσιμη σε μακροπρόθεσμο επίπεδο. Αυτό συμβαίνει επειδή συνδέεται με μια πιο βαθιά κατανόηση και ισχυρή δέσμευση στο θέμα.

## Κεφάλαιο 1ο:

Μελέτες στον τομέα της εκπαίδευσης αποκαλύπτουν ότι οι μαθητές που έχουν υψηλό επίπεδο κινήτρου δείχνουν μεγαλύτερη ικανότητα αυτορρύθμισης [6]. Αυτό σημαίνει ότι μπορούν να διαχειρίζονται τον χρόνο και την ενέργειά τους με πιο αποτελεσματικό τρόπο. Ταυτόχρονα, αυτοί είναι πιο ανθεκτικοί στις προκλήσεις και έχουν λιγότερες πιθανότητες να τα παρατήσουν σε μια δύσκολη διαδικασία. Η παρακίνηση έχει σημαντική επίδραση στο βάθος που προσεγγίζουν οι μαθητές τη μάθηση: αυτοί που έχουν πραγματικό ενδιαφέρον για ένα θέμα συχνά αναζητούν περισσότερες πληροφορίες, θέτουν ερωτήσεις και συνδέουν τις νέες γνώσεις με προηγούμενες εμπειρίες τους.

Στο σημερινό εκπαιδευτικό πλαίσιο, η πρόκληση είναι να αναπτύξουμε στρατηγικές που θα αυξήσουν την παρακίνηση, ιδιαίτερα όταν το θέμα θεωρείται πολύπλοκο ή απαιτητικό, κάτι που συμβαίνει συχνά με τον προγραμματισμό. Η εφαρμογή μεθόδων που προάγουν την ενεργή συμμετοχή, παρέχουν άμεση ανατροφοδότηση και επιτρέπουν σταδιακή πρόοδο μπορεί να παίξει καθοριστικό ρόλο στην αύξηση του ενδιαφέροντος των μαθητών. Ταυτόχρονα, η θέσπιση ξεκάθαρων στόχων, που μπορούν να επιτευχθούν με μικρές αλλά συνεχείς κινήσεις, δημιουργεί μια αίσθηση επιτυχίας και ενισχύει την αποφασιστικότητα για συνέχιση της προσπάθειας.

Το κίνητρο έχει ισχυρή σχέση με την ψυχολογική ικανοποίηση που αποκομίζει ο μαθητής από τη διαδικασία της μάθησης. Όταν η εμπειρία είναι ευχάριστη, οι μαθητές τείνουν να συνεχίζουν τις προσπάθειές τους και να δημιουργούν μια μόνιμη σύνδεση με το αντικείμενο που μαθαίνουν. Γι' αυτόν τον λόγο, η κατανόηση και η αξιοποίηση της παρακίνησης δεν είναι απλώς ένα βοηθητικό εργαλείο, αλλά ένας θεμέλιος λίθος για την επιτυχία της εκπαιδευτικής διαδικασίας στο σύνολό της.

## Κεφάλαιο 2ο: Παρόμοιες Εφαρμογές

Η δημιουργία εφαρμογών που συνδυάζουν την εκπαίδευση με στοιχεία παιχνιδιών έχει σημειώσει σημαντική αύξηση τα τελευταία χρόνια, καθώς οι χρήστες επιδιώκουν πιο αλληλεπιδραστικούς και ελκυστικούς τρόπους για να μάθουν. Η παιχνιδοποίηση διαθέτει μηχανισμούς που ενισχύουν την κίνητρο, αυξάνουν τη δέσμευση και βελτιώνουν την εμπειρία των χρηστών, κάνοντάς την εκπαιδευτική διαδικασία πιο αποδοτική και ευχάριστη. Οι πλατφόρμες που χρησιμοποιούν αυτές τις μεθόδους έχουν καταφέρει να προσελκύσουν εκατομμύρια μαθητές σε όλο τον κόσμο, δείχνοντας ότι η εκπαίδευση μπορεί να γίνει μια ευχάριστη και συχνά εθιστική δραστηριότητα.

Η πρωτογενής πηγή έμπνευσης για αυτήν την εργασία ήταν το Duolingo, μία από τις πιο αναγνωρίσιμες πλατφόρμες για τη διδασκαλία ξένων γλωσσών. Αυτή η πλατφόρμα έχει καταφέρει να σχηματίσει μια παγκόσμια κοινότητα μαθητών μέσω μιας προσεγγίσεως που συνδυάζει φιλικότητα και στοιχεία παιχνιδιού. Ταυτόχρονα, πλατφόρμες όπως το Mimo έχουν υιοθετήσει παρόμοιες μεθόδους στην εκπαίδευση προγραμματισμού, χρησιμοποιώντας την αποτελεσματικότητα των μικρών βημάτων, της άμεσης ανάδρασης και της συνεχούς επιβράβευσης.

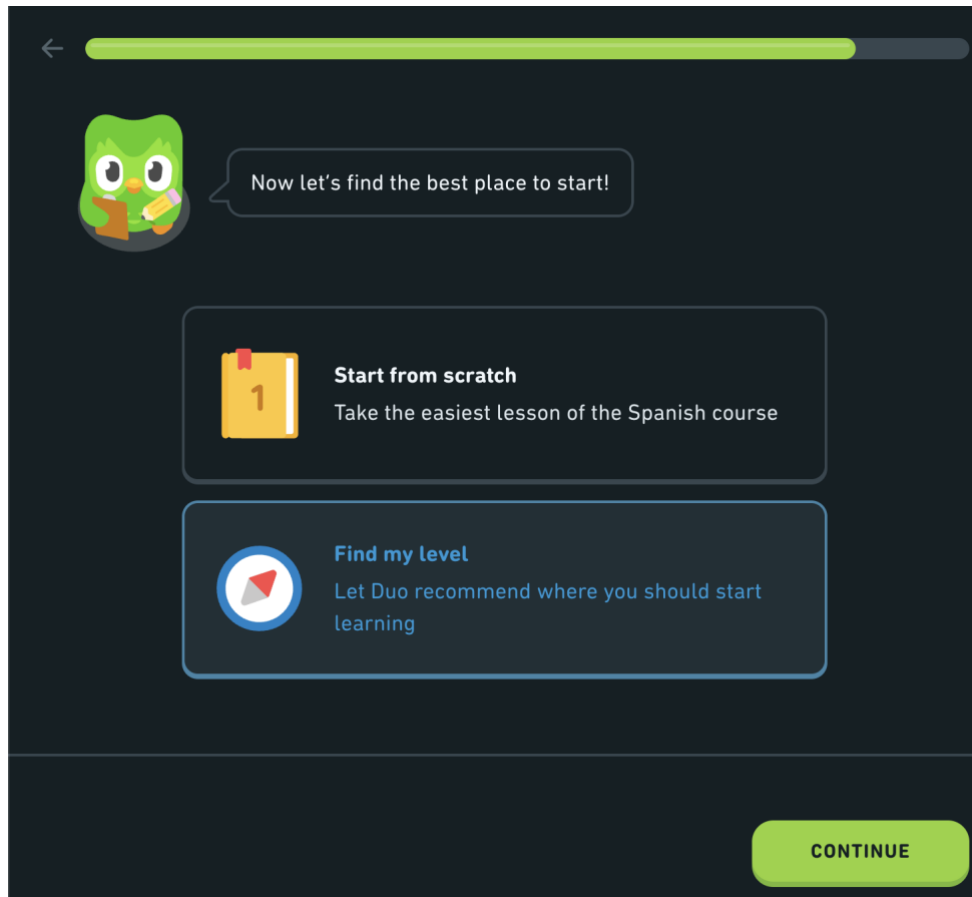
Σε αυτό το κεφάλαιο θα εξεταστούν αυτές οι τρεις εφαρμογές, με σκοπό να κατανοηθούν οι μέθοδοι που χρησιμοποιούν και η εμπειρία που παρέχουν στους χρήστες.

### 2.1 Duolingo

Το Duolingo ιδρύθηκε το 2011 από τους Luis von Ahn και Severin Hacker, με κύριο σκοπό να κάνει την εκμάθηση ξένων γλωσσών προσιτή σε όλους, ανεξάρτητα από τα οικονομικά ή εκπαιδευτικά τους προσόντα. Από την αρχή της παρουσίας της, η πλατφόρμα αυτή ξεχώρισε από τις κλασικές τεχνικές διδασκαλίας, ενσωματώνοντας πλούσια στοιχεία παιχνιδοποίησης και προσφέροντας μια εμπειρία που συνδυάζει τη μάθηση με τη διασκέδαση.

Τα μαθήματα στο Duolingo διαρθρώνονται σε θεματικές ενότητες που περιλαμβάνουν ποικιλία επιπέδων δυσκολίας και δεξιοτήτων. Αυτές οι ενότητες εστιάζουν σε τομείς όπως το λεξιλόγιο, η γραμματική, η κατανόηση κειμένου και η προφορική εξάσκηση. Κάθε τμήμα αποτελείται από σύντομα, αυτοτελή μαθήματα που μπορούν να ολοκληρωθούν μέσα σε λίγα λεπτά. Αυτή η μέθοδος διευκολύνει τη συμμετοχή και προάγει την επαναληπτική μάθηση.

Κατά την πρώτη είσοδο, ο χρήστης καλείται να επιλέξει τη γλώσσα που θέλει να μάθει, καθώς και τον καθημερινό χρόνο που μπορεί να αφιερώνει καθημερινά. Μέσω ενός σύντομου τεστ τοποθέτησης (placement test), η εφαρμογή εκτιμά το επίπεδό του και προσαρμόζει τα μαθήματα ανάλογα. Στη συνέχεια, η πορεία μάθησης εξατομικεύεται, καθώς οι ασκήσεις που εμφανίζονται βασίζονται στις επιδόσεις του χρήστη. Εάν γίνουν λάθη, παρόμοιες ασκήσεις επανεμφανίζονται συχνότερα για καλύτερη εμπέδωση.

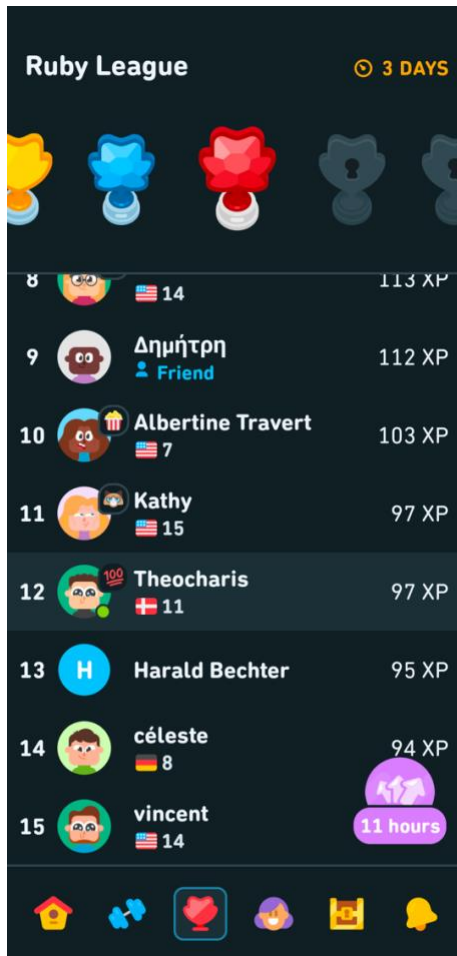


Σχήμα 2: Επιλογές πρώτης εισόδου στο Duolingo με τη μασκότ Duo

Αξίζει επίσης να αναφερθεί η τεχνολογική διάσταση του Duolingo. Η πλατφόρμα αξιοποιεί μεθόδους τεχνητής νοημοσύνης και μηχανικής μάθησης, ώστε να προσαρμόζει δυναμικά την πορεία του κάθε μαθητή. Μέσω της ανάλυσης των απαντήσεων, η εφαρμογή εντοπίζει τις αδυναμίες του χρήστη και επαναλαμβάνει αντίστοιχες ασκήσεις με μεγαλύτερη συχνότητα, εφαρμόζοντας έτσι εξατομικευμένη μάθηση. Επιπλέον, τα δεδομένα που συλλέγονται από εκατομμύρια χρήστες παγκοσμίως επιτρέπουν στη Duolingo να βελτιώνει συνεχώς την ακρίβεια των ασκήσεων και την αποτελεσματικότητα του εκπαιδευτικού της υλικού.

Η αξιοποίηση στοιχείων παιχνιδιού είναι βασικό στοιχείο της εμπειρίας του χρήστη. Οι μαθητές συγκεντρώνουν πόντους εμπειρίας (XP) για κάθε μάθημα που ολοκληρώνουν, γεγονός που τους δίνει τη δυνατότητα να παρακολουθούν την πρόοδό τους και να επιτυγχάνουν νέα επίπεδα. Το σύστημα με τις συνεχόμενες ημέρες εξάσκησης προωθεί τη σταθερότητα, ενώ οι πίνακες κατάταξης προσθέτουν μια διάσταση υγιούς ανταγωνισμού. Επιπλέον, η πλατφόρμα χορηγεί εμβλήματα για συγκεκριμένα επιτεύγματα και χρησιμοποιεί την εικονική της μασκότ, τον «Duo», μια πράσινη κουκουβάγια, ως συνεχή υπενθύμιση και κίνητρο.

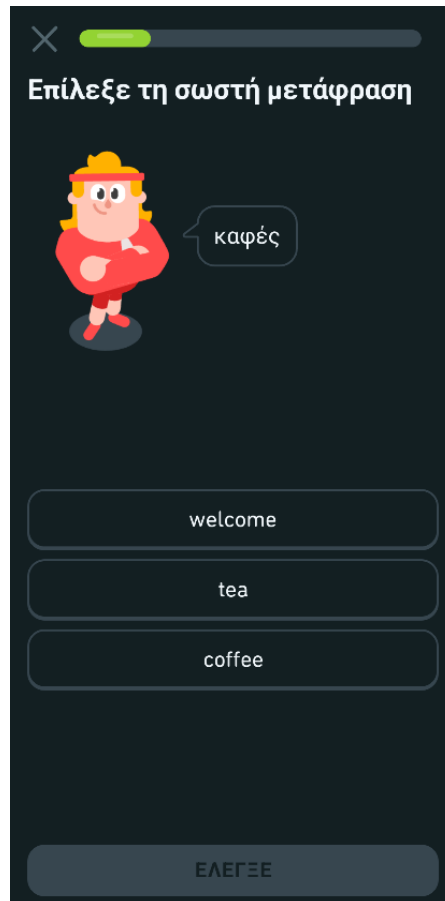
Πέρα από την ατομική πρόοδο, το Duolingo ενισχύει και την κοινωνική διάσταση της μάθησης. Κάθε χρήστης διαθέτει προφίλ όπου αποτυπώνεται το επίπεδο, οι πόντοι εμπειρίας και τα επιτεύγματά του. Η εφαρμογή επιτρέπει τη σύνδεση με φίλους, με δυνατότητα παρακολούθησης της πορείας τους, διατήρησης κοινών streaks και αποστολής μικρών “δώρων” XP ως κίνητρο. Παράλληλα, τα συστήματα κατάταξης (leaderboards) εισάγουν μια διάσταση ανταγωνισμού, καθώς οι χρήστες συγκρίνονται με φίλους ή με την παγκόσμια κοινότητα. Με αυτόν τον τρόπο, η διαδικασία της μάθησης αποκτά κοινωνικό χαρακτήρα και ενισχύεται η συνέπεια μέσω του φιλικού ανταγωνισμού.



Σχήμα 3: Σελίδα Leaderboard στο Duolingo

Η δημιουργία του περιβάλλοντος χρήστη διακρίνεται από την απλότητα, τα έντονα χρώματα και καθαρά εικονίδια, προσφέροντας μια ελκυστική και φιλική εμπειρία στους χρήστες. Η πλατφόρμα προσφέρει άμεση ανατροφοδότηση, εντοπίζοντας τα λάθη και προτείνοντας διορθώσεις σε πραγματικό χρόνο. Αυτό συμβάλλει στην ενεργή συμμετοχή των χρηστών και επιταχύνει τη διαδικασία εκμάθησης. Η επιτυχία του Duolingo στηρίζεται στην ενοποίηση μιας προσεγγίσιμης εκπαιδευτικής μεθόδου με μηχανισμούς που ενθαρρύνουν τους χρήστες να παραμένουν αφοσιωμένοι στο μάθημα για μεγάλο χρονικό διάστημα. Ως αποτέλεσμα, έχει γίνει μία από τις πιο γνωστές και συνηθισμένες εκπαιδευτικές εφαρμογές στον κόσμο, ασκώντας σημαντική επιρροή στον σχεδιασμό πολλών άλλων ψηφιακών πλατφορμών εκμάθησης.

Η μασκότ, καθώς και τα συστήματα streaks, XP και επιπέδων, προάγουν τη συνέπεια και ενισχύουν τη συναισθηματική σύνδεση με τον χρήστη [7] [8]. Παράλληλα, το Duolingo ακολουθεί το μοντέλο freemium, προσφέροντας δωρεάν βασική πρόσβαση για όλους, ενώ η συνδρομητική έκδοση (Super Duolingo) παρέχει πρόσθετα οφέλη, όπως απουσία διαφημίσεων, δυνατότητα offline χρήσης και ενισχυμένα στατιστικά προόδου. Αυτό το μοντέλο διευρύνει την προσβασιμότητα, διατηρώντας την εφαρμογή ανοιχτή σε ευρύ κοινό, αλλά και βιώσιμη σε μακροχρόνιο επίπεδο.

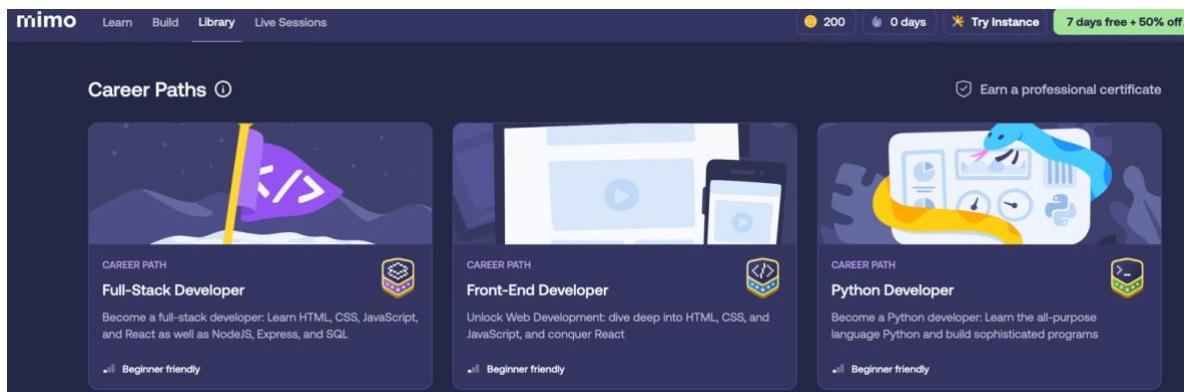


Σχήμα 4: Στάδιο παιχνιδιού στο Duolingo

## 2.2 Mimo

Το Mimo αποτελεί μια εφαρμογή που βοηθά στην εκμάθηση προγραμματισμού, εστιάζοντας σε γλώσσες όπως η Python, η JavaScript και το C++ [9]. Επίσης, διδάσκει θεμελιώδεις έννοιες σχετικά με την ανάπτυξη ιστοσελίδων και εφαρμογών. Σκοπός της δημιουργίας του είναι να διευκολύνει τους αρχάριους στον προγραμματισμό, προσφέροντας σύντομα και κατανοητά μαθήματα που μπορούν να ολοκληρωθούν σε λίγα μόνο λεπτά.

Όταν ο χρήστης ανοίγει για πρώτη φορά την εφαρμογή, του ζητείται να διαλέξει τη διαδρομή μάθησης που τον ενδιαφέρει, όπως Python, ανάπτυξη ιστοσελίδων ή επιστήμη δεδομένων.



Σχήμα 5: Επιλογή learning path στο Mimo

Η μέθοδος του εστιάζει στην πρακτική εξάσκηση με τη βοήθεια μίνι ασκήσεων και quiz, τα οποία προσφέρουν άμεσες απαντήσεις. Τα στοιχεία παιχνιδοποίησης, όπως οι πόντοι εμπειρίας (XP), τα συστήματα σερί και τα επιτεύγματα, προάγουν τη συνέπεια και την αδιάκοπη πρόοδο.

Το περιβάλλον του MIMO είναι ευανάγνωστο και πρακτικό, δίνοντας έμφαση στην καθοδήγηση βήμα προς βήμα. Η εφαρμογή διαθέτει επίσης «μονοπάτια μάθησης», τα οποία οργανώνουν το περιεχόμενο σύμφωνα με τους στόχους του χρήστη, διευκολύνοντας έτσι την κατανόηση και προσαρμόζοντας τη διαδικασία εκμάθησης στις συγκεκριμένες ανάγκες του. Μερικά από τα αξιοσημείωτα χαρακτηριστικά του παιχνιδιού είναι τα εξής:

- **Σύντομα και κατανοητά μαθήματα:** Κάθε μάθημα έχει σχεδιαστεί έτσι ώστε να μπορεί να ολοκληρωθεί σε λίγα λεπτά, δίνοντας τη δυνατότητα στον χρήστη να μάθει ακόμη και αν διαθέτει περιορισμένο χρόνο.
- **Διαδραστικά κουίζ και ασκήσεις:** Ο χρήστης συμμετέχει ενεργά με ερωτήσεις πολλαπλής επιλογής, συμπλήρωση κώδικα και σύντομες προγραμματιστικές προκλήσεις.
- **Συστήματα XP, streaks και επιτεύγματα:** Η πρόοδος επιβραβεύεται με πόντους εμπειρίας, συνεχείς ημέρες δραστηριότητας και εικονικά «παράσημα» για την εκπλήρωση στόχων.
- **Προσωποποιημένα μονοπάτια μάθησης:** Η εφαρμογή προσαρμόζει το υλικό σύμφωνα με το επίπεδο και τους στόχους του χρήστη. χ. εκμάθηση για ανάπτυξη ιστοσελίδων, κινητές εφαρμογές ή επιστήμη δεδομένων.
- **Άμεσες ανατροφοδοτήσεις:** Κάθε απάντηση ή κώδικας αξιολογούνται άμεσα, με διορθώσεις και επεξηγήσεις ώστε ο χρήστης να κατανοεί τα λάθη του.
- **Μικρές εργασίες:** Σε προχωρημένα στάδια, οι χρήστες καλούνται να αναπτύξουν μικρές εφαρμογές ή σενάρια, εφαρμόζοντας τις γνώσεις τους στην πράξη.



Σχήμα 6: Στάδιο παιχνιδιού στο Mimo

Το Mimo χρησιμοποιεί παρόμοιες στρατηγικές κινήτρων στον τομέα της εκμάθησης προγραμματισμού. Μέσω σύντομων μαθημάτων, πρακτικών ασκήσεων και μικρών έργων, παρέχει μια σταδιακά εξελισσόμενη διαδρομή ανάπτυξης δεξιοτήτων που προσαρμόζεται στις ανάγκες του χρήστη. Τα συστήματα XP, οι σειρές επιτυχιών και τα επιτεύγματα λειτουργούν ως κίνητρα, ενώ η άμεση ανατροφοδότηση διευκολύνει την ταχεία κατανόηση και διορθώνει λάθη.

Το Duolingo και το Mimo χρησιμοποιούν την παιχνιδοποίηση ως κύριο μέσο για να βελτιώσουν τη μαθησιακή διαδικασία, αλλά κάθε εφαρμογή την εφαρμόζει σε διαφορετικό τομέα.

Και οι δύο πλατφόρμες αποδεικνύουν ότι η συνδυασμένη αξιοποίηση μικρών μαθησιακών τμημάτων, άμεσης ενορχηστρωμένης ανατροφοδότησης και μηχανισμών επιβράβευσης μπορεί να καταστήσει τη διαδικασία μάθησης ισχυρή και αποτελεσματική, ανεξαρτήτως του αντικειμένου που διδάσκεται.

### 2.3 SoloLearn

Το SoloLearn είναι μια πλατφόρμα εκμάθησης προγραμματισμού που δημιουργήθηκε το 2014, με στόχο να προσφέρει μια προσιτή και ευχάριστη εμπειρία μάθησης για χρήστες κάθε επιπέδου. Διατίθεται τόσο σε κινητές συσκευές όσο και μέσω web, δίνοντας ιδιαίτερη έμφαση στην εκμάθηση εν κινήσει. Υποστηρίζει πολλές γλώσσες προγραμματισμού, όπως C, C++, Python, Java, JavaScript, HTML, CSS και SQL, καλύπτοντας ένα ευρύ φάσμα τεχνολογιών. [10]

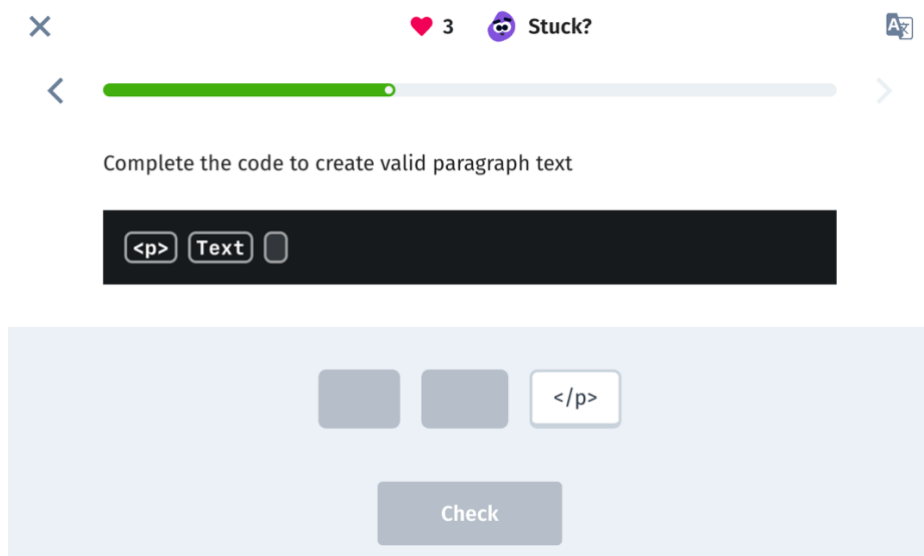
Τα μαθήματα είναι δομημένα σε σύντομες ενότητες που περιλαμβάνουν θεωρία, πρακτικές ασκήσεις και ερωτήσεις αξιολόγησης. Ο χρήστης μπορεί να γράψει και να εκτελέσει κώδικα απευθείας μέσα στην εφαρμογή μέσω ενός ενσωματωμένου code editor. Στα πιο προχωρημένα στάδια, παρέχεται η δυνατότητα δημιουργίας μικρών projects που βοηθούν στην εφαρμογή των γνώσεων στην πράξη. Η

πορεία μάθησης προσαρμόζεται σταδιακά ανάλογα με τις επιδόσεις του χρήστη, ώστε να ανταποκρίνεται στο επίπεδό του.

Όπως και άλλες επιτυχημένες πλατφόρμες, το SoloLearn αξιοποιεί στοιχεία παιχνιδοποίησης, όπως πόντους εμπειρίας (XP), streaks, εμβλήματα για επιτεύγματα και πίνακες κατάταξης, ώστε να παρακινεί τους χρήστες και να τους ενθαρρύνει να παραμένουν ενεργοί. Αυτοί οι μηχανισμοί επιβράβευσης καθιστούν τη μάθηση πιο ελκυστική και δημιουργούν ένα περιβάλλον συνεχούς προόδου.

Ένα από τα στοιχεία που διαφοροποιεί το SoloLearn είναι η έντονη κοινοτική διάσταση. Οι χρήστες μπορούν να ανεβάσουν δικό τους κώδικα, να υποβάλλουν ερωτήσεις και να λαμβάνουν απαντήσεις από την κοινότητα, αλλά και να αλληλεπιδρούν μεταξύ τους με likes, σχόλια και followers. Επιπλέον, οργανώνονται προγραμματιστικές προκλήσεις (Code Challenges), οι οποίες ενισχύουν τη συνεργασία αλλά και τον υγιή ανταγωνισμό. Με αυτόν τον τρόπο, η μάθηση συνδυάζεται με την αλληλεπίδραση σε μια ενεργή κοινότητα, ενισχύοντας τη δέσμευση και το ενδιαφέρον των χρηστών.

Το περιβάλλον χρήσης του SoloLearn είναι καθαρό και εύχρηστο, με σχεδιασμό που επιτρέπει τη γρήγορη πλοήγηση και την άμεση εκτέλεση κώδικα. Σε σχέση με το Mimo, το οποίο εστιάζει περισσότερο σε σύντομα και αυστηρά δομημένα μαθήματα, το SoloLearn δίνει έμφαση στην κοινωνική αλληλεπίδραση και στην ανταλλαγή γνώσεων μεταξύ των μελών της κοινότητας, καθιστώντας τη διαδικασία της μάθησης πιο συνεργατική και διαδραστική.



Σχήμα 7: Παράδειγμα άσκησης στο SoloLearn

Στις ασκήσεις του SoloLearn, οι ερωτήσεις παρουσιάζονται ως μικρά, στοχευμένα tasks πάνω σε σύντομα αποσπάσματα κώδικα. Ο χρήστης καλείται είτε να επιλέξει τη σωστή απάντηση είτε να συμπληρώσει το snippet, συχνά με drag-and-drop “tokens” (π.χ. ετικέτες HTML όπως στο παράδειγμα) ή με πληκτρολόγηση, ώστε να προκύψει έγκυρος κώδικας. Με το Check γίνεται άμεση αξιολόγηση: σε λάθος απόπειρα επισημαίνονται τα σημεία που χρειάζονται διόρθωση και εμφανίζονται σύντομες εξηγήσεις/υπαινιγμοί, ενώ στη σωστή λύση απονέμονται XP και ενημερώνεται η πρόοδος. Τα λανθασμένα θέματα επανέρχονται αργότερα για επανάληψη, ώστε να εδραιωθεί η γνώση, και η επιτυχής ολοκλήρωση ξεκλειδώνει την επόμενη ενότητα: σε ορισμένους τύπους άσκησης παρέχεται και εκτέλεση κώδικα μέσα από ενσωματωμένο editor.

## 2.4 Knackbout

Το Knackbout είναι μια διαδικτυακή πλατφόρμα που επικεντρώνεται στην εκμάθηση προγραμματισμού μέσω κουίζ, προσφέροντας μια πιο παιχνιδιάρικη και διαδραστική εμπειρία για τους χρήστες. [11]

## Κεφάλαιο 2ο:

Ειδικεύεται στη γλώσσα προγραμματισμού C, παρέχοντας ερωτήσεις που είναι οργανωμένες σε τρία επίπεδα δυσκολίας: αρχάριο, μεσαίο και προχωρημένο. Με αυτόν τον τρόπο, κάθε χρήστης μπορεί να ξεκινήσει από το επίπεδο που ταιριάζει καλύτερα στις γνώσεις του και να εξελιχθεί σταδιακά.

Κάθε ερώτηση συνοδεύεται από χρονικό περιορισμό, ο οποίος ενθαρρύνει τον χρήστη να σκέφτεται γρήγορα και να αναπτύσσει δεξιότητες λήψης αποφάσεων υπό πίεση, προσομοιώνοντας έτσι πραγματικές συνθήκες όπου η ταχύτητα και η ακρίβεια είναι σημαντικές. Οι σωστές απαντήσεις ανταμείβονται με πόντους, οι οποίοι αποτελούν τη βασική μονάδα προόδου στην πλατφόρμα. Παράλληλα, οι επιδόσεις καταγράφονται μέσω ενός συστήματος badges (διακριτικά επιτεύγματος), καθώς και με τη συλλογή ενός εικονικού νομίσματος που ονομάζεται "Knackbout cash". Αυτό το νόμισμα μπορεί να χρησιμοποιηθεί για την ξεκλείδωμα νέων επιπέδων ή τη συμμετοχή σε ειδικές προκλήσεις, προσφέροντας έτσι ένα επιπλέον κίνητρο για συνεχή δραστηριότητα.

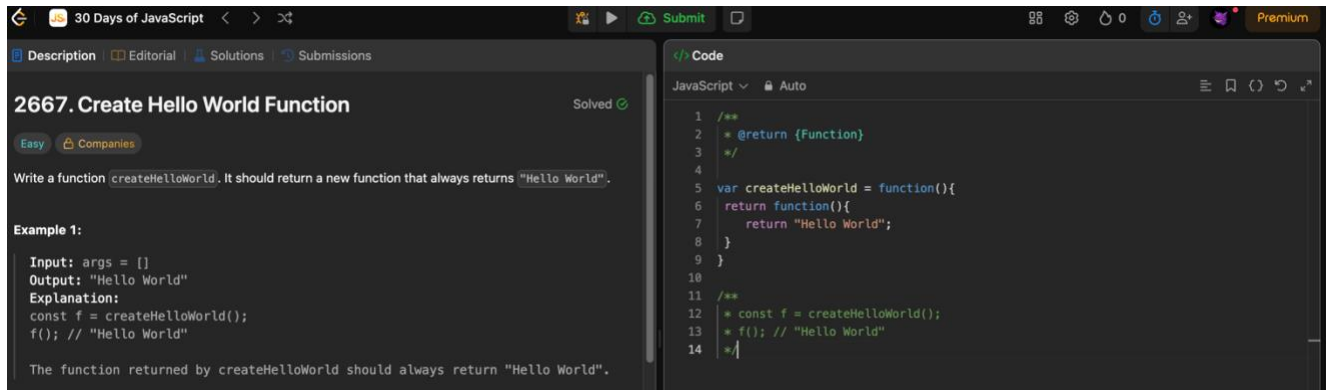
Παρά την απλότητά του, το Knackbout αποτελεί χαρακτηριστικό παράδειγμα μιας εφαρμογής που καταφέρνει να μετατρέψει τη μάθηση σε διασκεδαστική εμπειρία. Μέσα από την απλή διεπαφή του, την ξεκάθαρη δομή των ερωτήσεων και τους μηχανισμούς ανταμοιβής, δείχνει πώς ακόμα και ένα βασικό εκπαιδευτικό εργαλείο μπορεί να αξιοποιήσει την παιχνιδοποίηση για να ενθαρρύνει την αφοσίωση και να διευκολύνει την εκμάθηση της γλώσσας C.

### 2.5 LeetCode

Το LeetCode είναι μια από τις πιο γνωστές πλατφόρμες για την εξάσκηση στον προγραμματισμό και τη βελτίωση των δεξιοτήτων επίλυσης προβλημάτων. [12] Αν και δεν επικεντρώνεται αποκλειστικά στη γλώσσα C, διαθέτει μια εκτενή βιβλιοθήκη ασκήσεων που καλύπτουν βασικές και προχωρημένες έννοιες, όπως δομές δεδομένων, αλγόριθμους και τεχνικές βελτιστοποίησης. Η πλατφόρμα χρησιμοποιείται ευρέως από προγραμματιστές που προετοιμάζονται για τεχνικές συνεντεύξεις, καθώς προσφέρει ρεαλιστικά προβλήματα που αντικατοπτρίζουν τις απαιτήσεις μεγάλων εταιρειών τεχνολογίας, όπως Google, Microsoft, Amazon και Meta. Παράλληλα, πολλές εταιρείες αξιοποιούν το LeetCode και ως εργαλείο διεξαγωγής live συνεντεύξεων, επιτρέποντας την αξιολόγηση των δεξιοτήτων προγραμματισμού των υποψηφίων σε πραγματικό χρόνο, χωρίς την ανάγκη εξωτερικών εργαλείων ή IDE.

Παρά τον πιο επαγγελματικό χαρακτήρα της, η πλατφόρμα ενσωματώνει στοιχεία παιχνιδοποίησης που παρακινούν τους χρήστες. Οι ασκήσεις είναι οργανωμένες σε επίπεδα δυσκολίας (*easy*, *medium*, *hard*), ενώ οι συμμετέχοντες κερδίζουν πόντους και badges με την ολοκλήρωση προκλήσεων. Διαθέτει επίσης leaderboards που καταγράφουν την παγκόσμια κατάταξη των χρηστών, καθώς και διαγωνισμούς (contests) σε πραγματικό χρόνο, όπου οι προγραμματιστές καλούνται να λύσουν προβλήματα κάτω από χρονική πίεση, ανταγωνιζόμενοι άλλους συμμετέχοντες.

Αυτά τα χαρακτηριστικά συνδυάζουν τη μάθηση με τον ανταγωνισμό, δημιουργώντας ένα περιβάλλον που ενθαρρύνει τη συνεχή βελτίωση και εξάσκηση, ενώ ταυτόχρονα προετοιμάζει τους χρήστες για πραγματικές συνθήκες συνεντεύξεων και επαγγελματικές προκλήσεις.



Σχήμα 8: Περιβάλλον εργασίας στο LeetCode με παράδειγμα προβλήματος.

Στο LeetCode, οι ασκήσεις παρουσιάζονται ως προγραμματιστικά προβλήματα που συνοδεύονται από περιγραφή, παραδείγματα εισόδου και εξόδου, καθώς και επεξήγηση των απαιτήσεων. Ο χρήστης γράφει τον κώδικά του σε έναν ενσωματωμένο online editor, επιλέγοντας τη γλώσσα προγραμματισμού που επιθυμεί, όπως C, C++, Python ή Java. Πατώντας Run, το σύστημα ελέγχει τον κώδικα σε προκαθορισμένα test cases, παρέχοντας άμεση ανατροφοδότηση για σωστές ή λανθασμένες απαντήσεις. Αν ο κώδικας περάσει όλα τα tests, μπορεί να γίνει υποβολή (Submit), ώστε να συγκριθεί η λύση με εκείνες άλλων χρηστών και να καταχωρηθεί η βαθμολογία.

Αυτό το διαδραστικό περιβάλλον επιτρέπει στους χρήστες όχι μόνο να εξασκούνται στην πράξη, αλλά και να παρακολουθούν την πρόδό τους μέσα από στατιστικά στοιχεία και πίνακες κατάταξης, ενώ η δομή των ασκήσεων προσομοιώνει πραγματικές τεχνικές συνεντεύξεις.

## Κεφάλαιο 3ο: CodeIT

Η πλατφόρμα CodeIT σχεδιάστηκε για να παρέχει μια σύγχρονη, διαδραστική και παιχνιδιάρικη εμπειρία στην εκμάθηση της γλώσσας προγραμματισμού C. Απαντά στη ζήτηση για εκπαιδευτικά εργαλεία που συνδυάζουν θεωρία με πρακτική εφαρμογή, δίνοντας την ευκαιρία στους μαθητές να εξασκήσουν τις γνώσεις τους σε πραγματικές συνθήκες προγραμματισμού στην ίδια την πλατφόρμα της εφαρμογής.

Η φιλοσοφία ανάπτυξης του CodeIT στηρίζεται στην πεποίθηση ότι η διαδικασία εκμάθησης προγραμματισμού πρέπει να είναι μια ενεργητική και αλληλεπιδραστική εμπειρία, που θα κρατά ζωντανό το ενδιαφέρον των μαθητών. Για αυτόν τον λόγο, περιλαμβάνει μηχανισμούς παιχνιδοποίησης όπως πόντους εμπειρίας (XP), επίπεδα και streaks, με σκοπό να διασφαλίσει τη συνεχή παρακίνηση και την αύξηση της δέσμευσης των χρηστών.

### 3.1 Σκοπός και Κανόνες

Ο κύριος στόχος του CodeIT είναι να δημιουργήσει ένα περιβάλλον εκμάθησης της γλώσσας προγραμματισμού C που να είναι τόσο ολοκληρωμένο όσο και προσβάσιμο. Αυτό το περιβάλλον συνδυάζει τις θεωρητικές αρχές με την πρακτική εξάσκηση μέσα σε ένα ενιαίο σύστημα. Ο στόχος είναι να δημιουργηθεί ένα εκπαιδευτικό εργαλείο που θα διδάσκει όχι μόνο τους κανόνες και τις τεχνικές της σύνταξης, αλλά και θα ενισχύει δεξιότητες όπως η επίλυση προβλημάτων, η λογική σκέψη και η ικανότητα προσαρμογής σε νέα προγραμματιστικά περιβάλλοντα.

Για να επιτευχθεί αυτός ο στόχος, η πλατφόρμα εφαρμόζει συγκεκριμένους κανόνες λειτουργίας που διασφαλίζουν τη συνέπεια και την αποτελεσματικότητα της μάθησης:

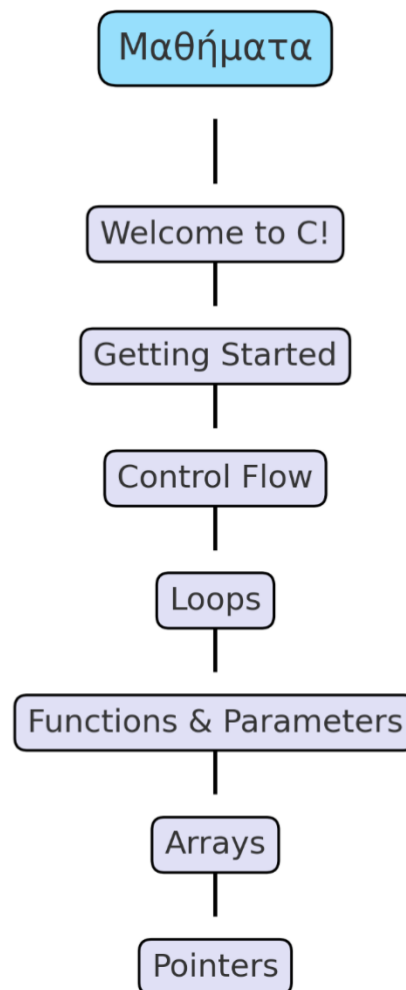
- Δομημένη πρόοδος: Το περιεχόμενο χωρίζεται σε θεματικά κομμάτια, καλύπτοντας σταδιακά από τις βασικές έννοιες μέχρι πιο περίπλοκα θέματα.
- Συνδυάζοντας θεωρία και πράξη: Κάθε μάθημα περιέχει θεωρητικό περιεχόμενο, quiz για την κατανόηση του υλικού, καθώς και ασκήσεις με πραγματικό κώδικα χρησιμοποιώντας τον ενσωματωμένο online compiler όποτε χρειάζεται.
- Παιχνιδοποίηση: Οι χρήστες αποκτούν πόντους εμπειρίας, ανεβαίνουν επίπεδα, διατηρούν την πρόοδο τους και λαμβάνουν ανταμοιβές, ενισχύοντας έτσι την επιθυμία τους να συνεχίσουν.
- Άμεση ανατροφοδότηση: Η πλατφόρμα προσφέρει αυτόματα σχόλια για κάθε απάντηση ή εκτέλεση κώδικα, διευκολύνοντας την άμεση διόρθωση σφαλμάτων.
- Αυτονομία μαθητή: Οι χρήστες μπορούν να προχωρούν με τον δικό τους ρυθμό και έχουν τη δυνατότητα να επαναλαμβάνουν μαθήματα ή βήματα όποτε το επιθυμούν.

Μέσω αυτών των αρχών, το CodeIT στοχεύει να προσφέρει μια μαθησιακή εμπειρία που είναι ευχάριστη αλλά και αποτελεσματική, παρέχοντας σταθερές προκλήσεις για συνεχή εξέλιξη.

### 3.2 Σχεδιασμός

Ο σχεδιασμός του CodeIT βασίστηκε στην πεποίθηση ότι η εκμάθηση προγραμματισμού μπορεί και πρέπει να αποτελεί μια ευχάριστη διαδικασία, ακόμα και για εκείνους που δεν έχουν καμία προηγούμενη εμπειρία με τον κώδικα. Για αυτόν τον λόγο, η γενική προσέγγιση εστιάζει στη σαφήνεια, την απλότητα και την άμεση πρόσβαση στα περιεχόμενα. Από τη στιγμή που ο χρήστης εισέρχεται στην πλατφόρμα, αντικρίζει ένα φιλικό και ευανάγνωστο περιβάλλον, όπου η πλοήγηση είναι ξεκάθαρη και η διάταξη του περιεχομένου είναι λογικά οργανωμένη. Η διεπαφή έχει σχεδιαστεί για να ελαχιστοποιεί τον γνωστικό φόρτο, επιτρέποντας στους μαθητές να επικεντρώνονται στη διαδικασία της μάθησης αντί να προσπαθούν να κατανοήσουν πώς λειτουργεί η εφαρμογή.

Η οργάνωση του περιεχομένου είναι ένα βασικό στοιχείο του σχεδιασμού. Το υλικό οργανώνεται σε θεματικές ενότητες που ακολουθούν μια λογική πρόοδο. Ξεκινά από τις βασικές έννοιες, όπως η σύνταξη και οι μεταβλητές, και προχωρά σε πιο προχωρημένα θέματα, όπως είναι οι δείκτες και η διαχείριση μνήμης. Κάθε τμήμα περιέχει υποτμήματα, στα οποία ο χρήστης μπορεί να κατανοήσει μια συγκεκριμένη έννοια μέσω θεωρητικών κειμένων, παραδειγμάτων κώδικα και διαδραστικών ασκήσεων. Η προσθήκη quiz με πολλές επιλογές, ασκήσεων για συμπλήρωση κενών και ασκήσεις προγραμματισμού ενισχύει τη δραστηριοποίηση των μαθητών, εμποδίζοντας την παθητική παρακολούθηση.



Σχήμα 9: Δομή μαθημάτων

Η διαδικασία εκπαίδευσης έχει σχεδιαστεί ώστε να συνδυάζει τη θεωρία με την πράξη. Οι χρήστες έχουν τη δυνατότητα να ασκηθούν απευθείας στον ενσωματωμένο online compiler, χωρίς να απαιτείται η εγκατάσταση εξωτερικών εργαλείων ή η πραγματοποίηση περίπλοκων ρυθμίσεων. Αυτό δεν καταργεί απλώς τεχνικά εμπόδια, αλλά διευκολύνει επίσης την άμεση σύνδεση της θεωρίας με την πράξη. Η σταδιακή αύξηση της δυσκολίας εξασφαλίζει ότι το νέο υλικό στηρίζεται σε έννοιες που ο μαθητής έχει ήδη μάθει. Παράλληλα, οι επαναλήψεις σε κρίσιμα σημεία λειτουργούν ως «γέφυρες» για την ενίσχυση της γνώσης πριν από την παρουσίαση νέων θεμάτων.

Τα στοιχεία παιχνιδιοποίησης παίζουν καθοριστικό ρόλο στον σχεδιασμό. Το σύστημα πόντων εμπειρίας, τα streaks, τα επίπεδα και οι επιβραβεύσεις έχουν σχεδιαστεί με τέτοιο τρόπο ώστε να παροτρύνουν τους χρήστες χωρίς να τους αποσπούν από τον κύριο στόχο μάθησης. Για παράδειγμα, η απόκτηση XP συμβαίνει αμέσως μετά την ολοκλήρωση μιας δραστηριότητας, ενισχύοντας έτσι την αίσθηση της επιτυχίας και της προόδου.

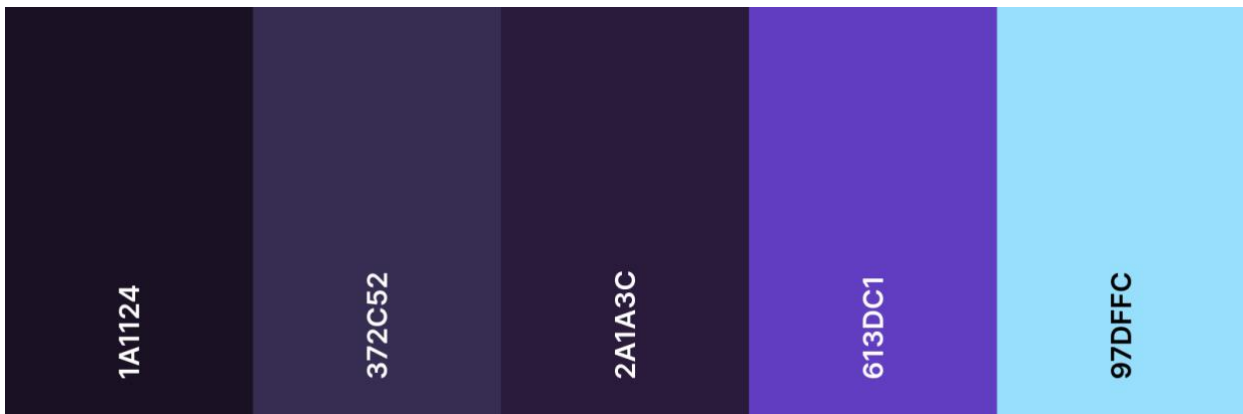
## Κεφάλαιο 3ο:

Τα streaks λειτουργούν ως ένας μετρητής συνεχόμενων ημερών δραστηριότητας. Κάθε φορά που ο χρήστης ολοκληρώνει τουλάχιστον ένα μάθημα μέσα στην ημέρα, το streak αυξάνεται κατά μία μονάδα. Αυτή η συνεχής καταγραφή προόδου προσδίδει μια αίσθηση επιτεύγματος και βοηθά τον χρήστη να δημιουργήσει μια σταθερή μαθησιακή συνήθεια.

Οι οπτικές ενδείξεις προόδου, όπως οι μπάρες ολοκλήρωσης μαθημάτων, παρέχουν μια σαφή εικόνα της πορείας του χρήστη, ενώ οι ειδοποιήσεις για streaks υπενθυμίζουν την αξία της συνέπειας και διατηρούν το ενδιαφέρον σε βάθος χρόνου.

Η πλατφόρμα είναι φτιαγμένη για να επιτρέπει την πρόσβαση χωρίς να απαιτείται η δημιουργία λογαριασμού, δίνοντας έτσι τη δυνατότητα στον χρήστη να ξεκινήσει αμέσως την εκμάθησή του χωρίς καμία διαδικασία εγγραφής. Ωστόσο, εάν ο χρήστης αποφασίσει να μην συνδεθεί, η πρόόδός του δεν θα αποθηκευτεί, με αποτέλεσμα να μην έχει τη δυνατότητα να παρακολουθεί τα στατιστικά ή τα επιτεύγματά του με την πάροδο του χρόνου. Η δυνατότητα σύνδεσης επιτρέπει την πλήρη παρακολούθηση της προόδου, τη χρήση στοιχείων παιχνιδοποίησης και μια προσωποποιημένη εμπειρία εκμάθησης.

Η επιλογή χρωμάτων βασίζεται τόσο στην αισθητική όσο και στην ψυχολογία του χρήστη. Ο συνδυασμός σκούρων φόντων με φωτεινές, έντονες λεπτομέρειες δημιουργεί μια ισχυρή αντίθεση που καθιστά το περιεχόμενο πιο ευανάγνωστο και ταυτόχρονα προσδίδει μια σύγχρονη και επαγγελματική αίσθηση. Οι έντονες αποχρώσεις χρησιμοποιούνται σε πλήκτρα, δείκτες προόδου και εικονίδια για να εντυπωσιάσουν τους χρήστες στα κρίσιμα σημεία της διεπαφής και να καθοδηγήσουν το βλέμμα τους.



Σχήμα 10: Χρώματα που χρησιμοποιήθηκαν στην εφαρμογή σε μορφή HEX

Η ευελιξία και η προσβασιμότητα αποτελούν επίσης κεντρικές παραμέτρους του σχεδιασμού. Η πλατφόρμα είναι πλήρως responsive, εξασφαλίζοντας ότι η εμπειρία παραμένει ομαλή και αποδοτική σε κάθε συσκευή, από υπολογιστές μέχρι κινητές συσκευές. Η δυνατότητα να ξεκινήσει κανείς τη μάθηση χωρίς εγκατάσταση διευκολύνει κάθε χρήστη, μειώνοντας σημαντικά τα εμπόδια στην πρόσβαση. Επιπλέον, η εφαρμογή καθαρής τυπογραφίας και κατάλληλων χρωματικών αντιθέσεων ενισχύει την αναγνωσιμότητα και κάνει την πλατφόρμα πιο προσβάσιμη σε άτομα με ήπιες οπτικές προκλήσεις.

Συνοψίζοντας, ο σχεδιασμός του CodeIT δεν εστιάζεται μόνο στην εμφάνιση και τη λειτουργία, αλλά επίσης εξετάζει την ψυχολογία της εκπαίδευσης. Μέσω της ισορροπίας ανάμεσα στις προκλήσεις και τις επιβραβεύσεις, καθώς και με την άμεση παροχή ανατροφοδότησης και την οπτική απεικόνιση της προόδου, διαμορφώνεται ένα περιβάλλον που ενθαρρύνει τους χρήστες να παραμένουν δραστήριοι και να επιστρέφουν στην πλατφόρμα για συνεχή ανάπτυξη.

### 3.3 Χαρακτηριστικά και διαφορές με παρόμοιες Εφαρμογές

Στον τομέα των διαδικτυακών εκπαιδευτικών πλατφορμών, εφαρμογές όπως το Duolingo και το Mimo έχουν γίνει δημοφιλείς λόγω της δημιουργικής τους προσέγγισης στην παιχνιδοποιημένη μάθηση. Αν

και αυτές οι πλατφόρμες είναι εξαιρετικά αποτελεσματικές σε πολλές θεματικές, η προσέγγισή τους συχνά είναι γενική και δεν εμβαθύνει, καθώς αγγίζουν ταυτόχρονα διάφορα γνωστικά πεδία. Αντίθετα, το CodeIT εστιάζει αποκλειστικά στην εκμάθηση της γλώσσας C, επιτρέποντας την ανάπτυξη ενός περιβάλλοντος που είναι απολύτως προσαρμοσμένο στις ανάγκες αυτού του στόχου.

Σε αυτό το πλαίσιο, η πλατφόρμα παρουσιάζει ουσιαστικές και πρακτικές διαφοροποιήσεις που επηρεάζουν άμεσα την εμπειρία και την αποτελεσματικότητα της μάθησης. Οι βασικές διαφορές του CodeIT σε σύγκριση με άλλες παρόμοιες εφαρμογές περιλαμβάνουν:

- **Εξατομίκευση στη γλώσσα C:** Πολλές γνωστές πλατφόρμες προσφέρουν εκπαίδευση σε πολλές γλώσσες προγραμματισμού ή και σε πεδία που δεν σχετίζονται άμεσα με την πληροφορική. Αυτό συνήθως έχει ως αποτέλεσμα μια περιορισμένη αναλυτικότητα στο περιεχόμενο για κάθε συγκεκριμένο θέμα. Το CodeIT αφιερώνει όλη τη δομή και το περιεχόμενο του μαθήματος αποκλειστικά στη C, επιτρέποντας την ανάπτυξη θεματικών ενοτήτων με πλήρη ανάλυση και περισσότερα παραδείγματα κώδικα, χωρίς να θυσιάζεται η λεπτομέρεια για χάρη της ποικιλίας.
- **Συνδυασμός θεωρίας, ερωτήσεων και εφαρμογής κώδικα σε έναν ενιαίο χώρο:** Σε πολλές άλλες πλατφόρμες, η θεωρητική γνώση και η πρακτική πλευρά συνήθως διαχωρίζονται σε διαφορετικά περιβάλλοντα ή ακόμα και σε ανεξάρτητες εφαρμογές. Το CodeIT προσφέρει τη δυνατότητα στους χρήστες να διαβάσουν θεωρητικές πληροφορίες, να απαντήσουν σε ερωτήσεις κατανόησης και να γράψουν ή να εκτελέσουν κώδικα, όλα σε μία σελίδα. Με αυτόν τον τρόπο περιορίζεται η ανάγκη για αλλαγές εργαλείων, επιταχύνεται η διαδικασία εκμάθησης και διατηρείται το ενδιαφέρον του μαθητή.
- **Ενσωματωμένος διαδικτυακός μεταγλωττιστής:** Το CodeIT διαθέτει έναν πλήρως ενσωματωμένο μεταγλωττιστή στο μαθησιακό περιβάλλον, επιτρέποντας την εκτέλεση κώδικα C σε πραγματικό χρόνο. Αντίθετα, σε άλλες πλατφόρμες οι χρήστες συχνά πρέπει να εγκαταστήσουν εξωτερικά προγράμματα ή να αξιοποιήσουν τρίτα εργαλεία. Με το CodeIT, η διαδικασία είναι γρήγορη και δεν χρειάζεται καμία τεχνική προετοιμασία.
- **Δυνατότητα χρήσης χωρίς λογαριασμό:** Ο χρήστης μπορεί να ξεκινήσει την εκμάθηση αμέσως, χωρίς να δημιουργήσει λογαριασμό, κάτι που μειώνει το αρχικό «φράγμα εισόδου». Ωστόσο, για την αποθήκευση της προόδου και την πλήρη αξιοποίηση των λειτουργιών, η δημιουργία λογαριασμού παραμένει διαθέσιμη ως επιλογή.
- **100% δωρεάν χρήση:** Σε αντίθεση με πολλές άλλες εφαρμογές που ακολουθούν μοντέλο freemium, το CodeIT προσφέρει όλα τα περιεχόμενα και τις βασικές του δυνατότητες χωρίς καμία χρέωση. Ο χρήστης δεν θα βρεθεί ποτέ μπροστά σε «κλειδωμένα» μαθήματα ή λειτουργίες που απαιτούν πληρωμή.
- **Απεριόριστη χρήση hints:** Σε πολλές πλατφόρμες, η δυνατότητα χρήσης βοηθητικών υποδείξεων είναι περιορισμένη και συχνά σχετίζεται με μικροχρεώσεις ή ένα σύστημα «ζώων» που εξαντλούνται. Το CodeIT επιτρέπει την απεριόριστη πρόσβαση στα hints, ενθαρρύνοντας την αυτοδιόρθωση και τη μάθηση χωρίς να δημιουργεί οικονομικά εμπόδια ή άγχος στον χρήστη.
- **Υποστήριξη της ελληνικής γλώσσας:** Το CodeIT είναι η μοναδική πλατφόρμα του είδους της που προσφέρει πλήρη υποστήριξη στην ελληνική γλώσσα. Αυτό καθιστά το περιεχόμενο προσιτό σε μαθητές που δεν είναι εξοικειωμένοι με τα αγγλικά, δίνοντας τη δυνατότητα εκμάθησης της C στη μητρική τους γλώσσα, χωρίς γλωσσικά εμπόδια.

Η ύπαρξη αυτών των διαφορών δεν παρέχει μόνο τεχνικά πλεονεκτήματα, αλλά επηρεάζει σημαντικά τη σχέση του μαθητή με το υλικό που διδάσκεται. Σε ένα περιβάλλον που έχει αποκλειστική εστίαση στη μάθηση, χωρίς οικονομικούς περιορισμούς και σχεδιασμένο για να υποστηρίξει την πρακτική εξάσκηση, ο χρήστης μπορεί να αφιερώσει τον χρόνο του μόνο στη διαδικασία της εκμάθησης. Ταυτόχρονα, η έλλειψη πίεσης από «ζώες» ή προθεσμίες, σε συνδυασμό με την πρόσβαση στη μητρική

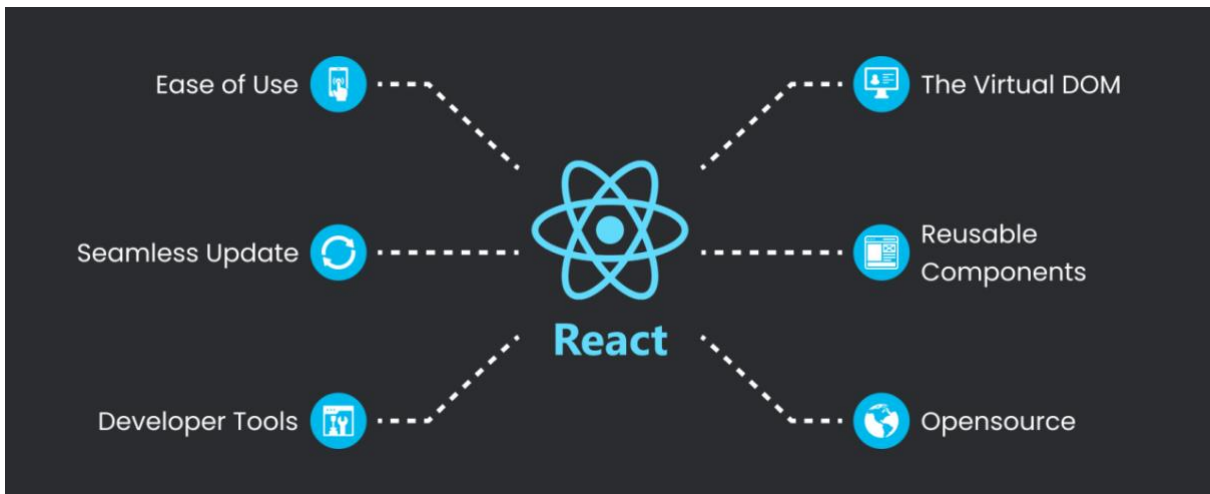
### Κεφάλαιο 3ο:

γλώσσα, ενισχύει την αυτοπεποίθηση και το κίνητρο για προχώρημα. Έτσι, το CodeIT δεν είναι μόνο ένα ακόμα εργαλείο εκμάθησης, αλλά λειτουργεί ως ένας ψηφιακός χώρος όπου η μάθηση της γλώσσας προγραμματισμού C γίνεται πιο ανθρώπινη, προσιτή και αποτελεσματική.

## Κεφάλαιο 4ο: Τεχνολογία React

Η React αποτελεί μια σύγχρονη βιβλιοθήκη JavaScript που χρησιμοποιείται για τη δημιουργία διεπαφών χρήστη, κυρίως σε εφαρμογές μιας μόνο σελίδας (Single Page Applications - SPA) [13]. Δημιουργήθηκε από το Facebook και στηρίζεται στην επαναχρησιμοποίηση στοιχείων (components) καθώς και στην αποδοτική ενημέρωση του περιβάλλοντος μέσω του Virtual DOM.

Η δηλωτική της προσέγγιση επιτρέπει στον προγραμματιστή να περιγράφει πώς πρέπει να εμφανίζεται η διεπαφή για μια συγκεκριμένη κατάσταση, αφήνοντας στη βιβλιοθήκη τη διαχείριση των απαραίτητων αλλαγών. Η επιλογή της React για το CodeIT οφείλεται στη μεγάλη της ευελιξία, την εξαιρετική απόδοση και τη ισχυρή υποστήριξη που παρέχει η κοινότητα. Αυτά τα χαρακτηριστικά την καθιστούν ιδανική για την ανάπτυξη δυναμικών, διαδραστικών και επεκτάσιμων εκπαιδευτικών εφαρμογών.



Σχήμα 11: Λογότυπο React και κύρια χαρακτηριστικά

### 4.1 Βασικές αρχές λειτουργίας

Η λειτουργία της React βασίζεται σε έναν συνδυασμό τεχνολογικών αρχών και προγραμματιστικών προτύπων που αποσκοπούν στη δημιουργία δυναμικών, επεκτάσιμων και αποδοτικών διεπαφών χρήστη. Η κατανόηση αυτών των αρχών είναι κρίσιμη για την πλήρη αξιοποίηση των δυνατοτήτων της βιβλιοθήκης και αποτέλεσε θεμέλιο για την ανάπτυξη της πλατφόρμας CodeIT.

#### 4.1.1 Component-Based Αρχιτεκτονική

Η βασική ιδέα της React είναι ότι η διεπαφή χρήστη χωρίζεται σε components — μικρές, αυτόνομες και επαναχρησιμοποιήσιμες μονάδες κώδικα. Κάθε component περιλαμβάνει:

- Τη λογική (JavaScript/JSX)
- Την εμφάνιση (HTML-like markup μέσω JSX).
- Την τοπική του κατάσταση (state).

Αυτό το μοντέλο ακολουθεί την αρχή "Διαίρει και βασίλευε", διευκολύνοντας τη συντήρηση και την ανάπτυξη σύνθετων εφαρμογών. Τα components μπορούν να είναι:

- Λειτουργικά (Functional Components) – απλές συναρτήσεις που επιστρέφουν JSX.

## Κεφάλαιο 4ο:

- Κλασικά (Class Components) – βασίζονται σε ES6 κλάσεις (λιγότερο διαδεδομένα πλέον λόγω hooks).

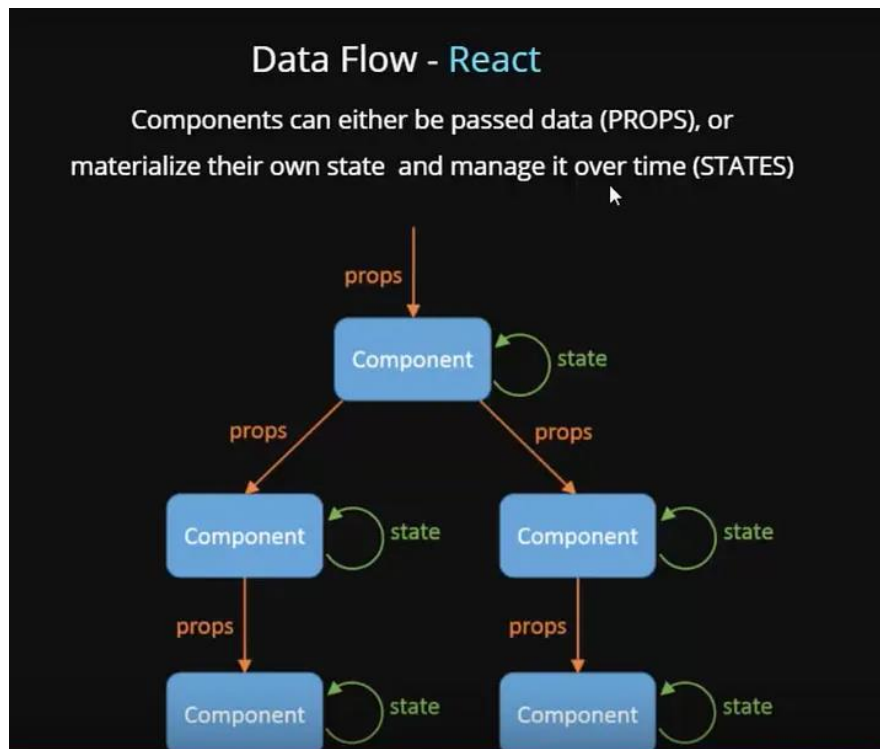
Η React ενθαρρύνει τη δημιουργία ιεραρχιών components, όπου τα γονικά στοιχεία συντονίζουν την κατάσταση και μεταβιβάζουν δεδομένα στα παιδικά μέσω props.

### 4.1.2 Props και State

Η ροή δεδομένων σε μια εφαρμογή React ακολουθεί την αρχή του μονόδρομου data flow:

- Props (properties): Παράμετροι που δίνονται σε ένα component από τον γονέα του. Είναι αμετάβλητες από την πλευρά του παιδιού.
- State : Εσωτερικά δεδομένα που διαχειρίζεται το ίδιο το component. Μπορούν να αλλάξουν με την κλήση της κατάλληλης συνάρτησης ενημέρωσης (π.χ. setState ή useState).

Αυτή η διάκριση επιτρέπει την προβλεψιμότητα στη συμπεριφορά της εφαρμογής. Οι αλλαγές στο state οδηγούν αυτόματα σε επανασχεδίαση (re-render) του component



Σχήμα 12: props και states

### 4.1.3 Εικονικό DOM και διαδικασία συμφιλίωσης

Η άμεση τροποποίηση του πραγματικού DOM ενός διαδικτυακού εγγράφου είναι μια διαδικασία που μπορεί να είναι αργή και απαιτητική σε πόρους, ειδικά όταν η εφαρμογή περιλαμβάνει συχνές αλλαγές στην εμφάνιση. Για την αντιμετώπιση αυτού του προβλήματος, η React χρησιμοποιεί μια εικονική αναπαράσταση του DOM, γνωστή ως Εικονικό DOM [13].

Το Εικονικό DOM είναι ένα ελαφρύ αντίγραφο της δομής της σελίδας που αποθηκεύεται στη μνήμη. Όταν αλλάζουν τα δεδομένα της εφαρμογής, η React δημιουργεί μια νέα εκδοχή του Εικονικού DOM και την συγκρίνει με την προηγούμενη μέσω μιας διαδικασίας που ονομάζεται σύγκριση διαφορών

(diffing). Από αυτή τη σύγκριση εντοπίζονται μόνο οι αλλαγές που χρειάζονται, και στη συνέχεια εφαρμόζονται αποκλειστικά αυτές οι αλλαγές στο πραγματικό DOM.

Αυτή η μέθοδος, γνωστή και ως διαδικασία συμφιλίωσης (reconciliation), επιτρέπει την ελαχιστοποίηση των περιττών επεμβάσεων στο DOM και οδηγεί σε καλύτερη απόδοση και ταχύτερη απόκριση της διεπαφής.

#### 4.1.4 Άγκιστρα (Hooks)

Με την εξέλιξη της React, εισήχθη η δυνατότητα χρήσης αγκίστρων, η οποία επιτρέπει σε συναρτήσεις-component (λειτουργικά components) να έχουν δυνατότητες που παλαιότερα ήταν διαθέσιμες μόνο σε component κλάσεων.

Τα άγκιστρα είναι ειδικές συναρτήσεις που επιτρέπουν:

- Τη δημιουργία και διαχείριση εσωτερικής κατάστασης ενός component (π.χ. useState)
- Την εκτέλεση ενεργειών σε συγκεκριμένα στάδια του κύκλου ζωής του component (π.χ. useEffect)
- Την ανταλλαγή δεδομένων μεταξύ components χωρίς αλυσίδα ιδιοτήτων (useContext)
- Τη βελτιστοποίηση της απόδοσης με αποθήκευση υπολογισμένων τιμών ή συναρτήσεων (useMemo, useCallback)
- Τη διαχείριση πιο σύνθετης κατάστασης μέσω συναρτήσεων μείωσης (useReducer)

Η χρήση αγκίστρων επιτρέπει πιο καθαρό και κατανοητό κώδικα, με λιγότερες επαναλήψεις και μεγαλύτερη ευελιξία στην ανάπτυξη.

#### 4.1.5 Δηλωτική Προσέγγιση

Η React βασίζεται σε μια δηλωτική λογική, όπου ο προγραμματιστής περιγράφει το τελικό αποτέλεσμα που θέλει να δει στην οθόνη, και η βιβλιοθήκη αναλαμβάνει να το υλοποιήσει με τον πιο αποδοτικό τρόπο.

Σε αντίθεση με την προστακτική λογική, όπου ο προγραμματιστής δίνει βήμα-βήμα οδηγίες για την αλλαγή της διεπαφής, η δηλωτική προσέγγιση μειώνει την πολυπλοκότητα του κώδικα, περιορίζει τα σφάλματα και κάνει την εφαρμογή πιο προβλέψιμη και εύκολη στη συντήρηση.

#### 4.1.6 Πλοήγηση και αρχιτεκτονική εφαρμογών μίας σελίδας

Η React υποστηρίζει τη δημιουργία εφαρμογών μίας σελίδας (Single Page Applications), όπου η πλοήγηση ανάμεσα σε διαφορετικές ενότητες δεν απαιτεί επαναφόρτωση της σελίδας. Αυτό επιτυγχάνεται μέσω ειδικών βιβλιοθηκών πλοήγησης, οι οποίες διαχειρίζονται τις διαδρομές και την εμφάνιση του κατάλληλου περιεχομένου ανάλογα με τη διεύθυνση που έχει ζητήσει ο χρήστης.

Με αυτήν την αρχιτεκτονική, η εμπειρία χρήσης γίνεται πιο ομαλή, καθώς οι μεταβάσεις είναι άμεσες και η εφαρμογή μπορεί να διατηρεί την κατάστασή της χωρίς απώλειες δεδομένων κατά την πλοήγηση.

#### 4.1.7 Αντιδραστικότητα και άμεση ενημέρωση διεπαφής

Ένα από τα βασικά χαρακτηριστικά της React είναι η ικανότητά της να ενημερώνει άμεσα τη διεπαφή χρήστη όταν μεταβάλλονται τα δεδομένα. Η αντιδραστικότητα αυτή προκύπτει από τη στενή σύνδεση μεταξύ της κατάστασης της εφαρμογής και της απεικόνισής της στην οθόνη.

## Κεφάλαιο 4ο:

Κάθε αλλαγή στην κατάσταση ή στις ιδιότητες ενός component οδηγεί σε αυτόματη ανανέωση του τμήματος της διεπαφής που επηρεάζεται. Με αυτόν τον τρόπο εξασφαλίζεται ότι ο χρήστης βλέπει πάντοτε την πιο πρόσφατη και ακριβή αναπαράσταση των δεδομένων, γεγονός που είναι ιδιαίτερα σημαντικό σε εφαρμογές που βασίζονται στην αλληλεπίδραση και την άμεση ανατροφοδότηση.

### 4.2 Πλεονεκτήματα και αδυναμίες της React

Η React έχει καθιερωθεί ως μία από τις πιο δημοφιλείς τεχνολογίες για την ανάπτυξη διεπαφών χρήστη, κυρίως λόγω της αρχιτεκτονικής της, της απόδοσης και της ευελιξίας που προσφέρει. Ωστόσο, η χρήση της συνοδεύεται και από ορισμένους περιορισμούς, οι οποίοι πρέπει να αξιολογούνται κατά τον σχεδιασμό ενός έργου.

#### 4.2.1 Πλεονεκτήματα της React

- Αρχιτεκτονική βασισμένη σε components: Η React επιτρέπει τη διάσπαση της διεπαφής χρήστη σε μικρά, αυτόνομα και επαναχρησιμοποιήσιμα components. Αυτό οδηγεί σε καλύτερη οργάνωση του κώδικα, διευκολύνει τη συντήρηση και επιτρέπει την ανάπτυξη μεγάλων έργων από ομάδες που εργάζονται ταυτόχρονα σε διαφορετικά τμήματα της εφαρμογής.
- Υψηλή απόδοση μέσω Εικονικού DOM: Η χρήση εικονικής αναπαράστασης του DOM και η διαδικασία συμφιλίωσης μειώνουν δραστικά τις περιττές αλλαγές στο πραγματικό DOM. Ως αποτέλεσμα, οι ενημερώσεις της διεπαφής είναι ταχύτερες, προσφέροντας ομαλή εμπειρία χρήσης ακόμη και σε εφαρμογές με συχνές μεταβολές δεδομένων.
- Δηλωτική προσέγγιση: Ο προγραμματιστής περιγράφει την τελική μορφή της διεπαφής με βάση την τρέχουσα κατάσταση των δεδομένων, χωρίς να χρειάζεται να καθορίσει αναλυτικά τα βήματα τροποποίησης. Αυτό μειώνει την πολυπλοκότητα, περιορίζει τον κίνδυνο σφαλμάτων και καθιστά τον κώδικα πιο προβλέψιμο.
- Ευελιξία και προσαρμοστικότητα: Η React μπορεί να χρησιμοποιηθεί αυτόνομα ή σε συνδυασμό με άλλες βιβλιοθήκες και πλαίσια. Υποστηρίζει την ενσωμάτωση πρόσθετων εργαλείων για διαχείριση κατάστασης, δρομολόγηση, επικοινωνία με διακομιστές και προσαρμοζόμενη στις απαιτήσεις του εκάστοτε έργου.
- Μεγάλη κοινότητα και πλούσιο οικοσύστημα: Η εκτενής κοινότητα ανάπτυξης παρέχει βιβλιοθήκες, plugins, οδηγούς και παραδείγματα. Η ύπαρξη αυτών των πόρων μειώνει τον χρόνο επίλυσης προβλημάτων και επιταχύνει την ανάπτυξη.
- Συμβατότητα με διαφορετικά περιβάλλοντα ανάπτυξης: Η React μπορεί να χρησιμοποιηθεί όχι μόνο για την ανάπτυξη διαδικτυακών εφαρμογών, αλλά και για τη δημιουργία εφαρμογών για κινητές συσκευές μέσω του React Native, καθώς και για την ανάπτυξη desktop εφαρμογών με κατάλληλα frameworks. Αυτή η δυνατότητα επαναχρησιμοποίησης γνώσεων και κώδικα σε πολλαπλές πλατφόρμες μειώνει τον χρόνο και το κόστος ανάπτυξης.
- Δυνατότητα σταδιακής υιοθέτησης: Η React μπορεί να ενσωματωθεί σε υπάρχουσες εφαρμογές τμηματικά, χωρίς να απαιτείται πλήρης ανακατασκευή του έργου. Αυτό σημαίνει ότι μια ομάδα μπορεί να την υιοθετήσει σε μικρά τμήματα της εφαρμογής και σταδιακά να επεκτείνει τη χρήση της, μειώνοντας το ρίσκο και διευκολύνοντας τη μετάβαση.

#### 4.2.2 Αδυναμίες της React

- Συνεχής και ταχεία εξέλιξη: Η React και οι σχετικές βιβλιοθήκες της εξελίσσονται με μεγάλη ταχύτητα, γεγονός που μπορεί να δημιουργεί δυσκολίες παρακολούθησης των αλλαγών. Οι προγραμματιστές συχνά χρειάζεται να ενημερώνουν τις γνώσεις τους και να προσαρμόζουν τον κώδικα σε νέες εκδόσεις.

- Περιορισμοί στη βελτιστοποίηση για μηχανές αναζήτησης: Ως τεχνολογία που δημιουργεί δυναμικό περιεχόμενο στον φυλλομετρητή, η React ενδέχεται να μην προσφέρει βέλτιστη ευρετηρίαση από μηχανές αναζήτησης. Για να ξεπεραστεί αυτός ο περιορισμός, συχνά απαιτούνται τεχνικές όπως pre-rendering στον διακομιστή.
- Εξάρτηση από εξωτερικές βιβλιοθήκες: Παρόλο που η React είναι ισχυρή, πολλές βασικές λειτουργίες, όπως η πλήρης διαχείριση κατάστασης ή η δρομολόγηση, απαιτούν την ενσωμάτωση επιπλέον βιβλιοθηκών. Αυτό μπορεί να αυξήσει την πολυπλοκότητα της ανάπτυξης και να απαιτεί πρόσθετη συντήρηση.
- Περιορισμένος αριθμός βιβλιοθηκών τρίτων κατασκευαστών: Η Flutter όντας σχετικά νέα δεν μπορεί να συγκριθεί με τις εγγενείς γλώσσες προγραμματισμού. Οι προγραμματιστές εξακολουθούν να χρειάζεται να ξοδεύουν περισσότερο χρόνο για την κατασκευή όσο το δυνατόν περισσότερων βιβλιοθηκών.

Συνοψίζοντας, η React αποτελεί μια από τις πιο ολοκληρωμένες και ευέλικτες τεχνολογίες για την ανάπτυξη σύγχρονων εφαρμογών στο διαδίκτυο. Ο συνδυασμός υψηλής απόδοσης, επαναχρησιμοποίησης κώδικα και δηλωτικής προσέγγισης επιτρέπει την ταχεία και αποτελεσματική δημιουργία πολύπλοκων διεπαφών χρήστη. Η συμβατότητα με πολλαπλές πλατφόρμες και η δυνατότητα σταδιακής υιοθέτησης μειώνουν το ρίσκο ανάπτυξης, ενώ η μεγάλη κοινότητα και το πλούσιο οικοσύστημα εργαλείων παρέχουν σταθερή υποστήριξη και καινοτόμες λύσεις.

Παρά τα πλεονεκτήματα, η επιτυχής αξιοποίηση της React απαιτεί επαρκή τεχνική κατάρτιση, συνεχή παρακολούθηση των εξελίξεων και προσεκτική διαχείριση της πολυπλοκότητας που μπορεί να προκύψει από τη χρήση πολλών εξωτερικών βιβλιοθηκών. Με τον κατάλληλο σχεδιασμό και την εφαρμογή βέλτιστων πρακτικών, τα μειονεκτήματα μπορούν να περιοριστούν, επιτρέποντας την πλήρη εκμετάλλευση των δυνατοτήτων της για την ανάπτυξη επεκτάσιμων, διαδραστικών και φιλικών προς τον χρήστη εφαρμογών.

## Κεφάλαιο 5ο: Υλικό Ασκήσεων

Οι ερωτήσεις και οι ασκήσεις αποτελούν θεμελιώδες στοιχείο της μαθησιακής διαδικασίας, καθώς δίνουν τη δυνατότητα στον εκπαιδευόμενο να εφαρμόσει άμεσα τις γνώσεις που απέκτησε. Η ενεργητική συμμετοχή, μέσω της επίλυσης πρακτικών προβλημάτων, ενισχύει την κατανόηση των εννοιών και συμβάλλει στην καλύτερη αφομοίωση του διδακτικού περιεχομένου.

Η πλατφόρμα CodeIT βασίζεται στην αρχή ότι η θεωρητική παρουσίαση πρέπει να ακολουθείται από στοχευμένες ασκήσεις, οι οποίες ελέγχουν την κατανόηση και επιτρέπουν στον χρήστη να προχωρά στο επόμενο βήμα μόνο όταν έχει επιτύχει επαρκές επίπεδο γνώσης. Με αυτόν τον τρόπο, οι ασκήσεις δεν λειτουργούν απλώς ως μέσο αξιολόγησης, αλλά ως ενεργό μέρος της μαθησιακής εμπειρίας.

Η συστηματική ενσωμάτωση των ασκήσεων στις ενότητες του μαθήματος συμβάλλει στη σταδιακή ανάπτυξη δεξιοτήτων, ενώ η άμεση ανατροφοδότηση βοηθά τον εκπαιδευόμενο να αναγνωρίζει και να διορθώνει τα λάθη του. Η προσέγγιση αυτή συνάδει με σύγχρονες παιδαγωγικές μεθόδους που δίνουν έμφαση στη βιωματική μάθηση και την εξατομικευμένη πρόοδο.

### 5.1 Κριτήρια δημιουργίας ασκήσεων

Ο σχεδιασμός των ασκήσεων στην πλατφόρμα CodeIT βασίζεται σε συγκεκριμένα παιδαγωγικά και τεχνικά κριτήρια, ώστε να εξασφαλίζεται η αποτελεσματικότητα της μαθησιακής εμπειρίας και η διατήρηση του ενδιαφέροντος του χρήστη.

- **Εκπαιδευτικοί στόχοι:** Κάθε άσκηση συνδέεται με συγκεκριμένους μαθησιακούς στόχους που προκύπτουν από την ενότητα θεωρίας που προηγείται. Ο στόχος μπορεί να είναι η κατανόηση μιας νέας έννοιας, η εξάσκηση σε έναν κανόνα σύνταξης ή η επίλυση ενός πρακτικού προβλήματος.
- **Κλιμάκωση δυσκολίας:** Οι ασκήσεις διαβαθμίζονται από απλές σε πιο σύνθετες, επιτρέποντας στον χρήστη να εξοικειώνεται σταδιακά με την ύλη. Οι πρώτες ερωτήσεις εστιάζουν σε βασικές έννοιες, ενώ οι επόμενες απαιτούν συνδυασμό γνώσεων και μεγαλύτερη κριτική σκέψη.
- **Σαφήνεια διατύπωσης:** Η γλώσσα που χρησιμοποιείται είναι απλή και κατανοητή, αποφεύγοντας περιττούς τεχνικούς όρους που μπορεί να μπερδέψουν έναν αρχάριο. Οι οδηγίες είναι συγκεκριμένες, ώστε ο εκπαιδευόμενος να γνωρίζει ακριβώς τι ζητείται.
- **Άμεση σύνδεση με το θεωρητικό περιεχόμενο:** Οι ασκήσεις σχεδιάζονται ώστε να ακολουθούν άμεσα την παρουσίαση της θεωρίας, εξασφαλίζοντας ότι ο χρήστης μπορεί να εφαρμόσει τις γνώσεις του όσο αυτές είναι ακόμη «φρέσκες».
- **Δυνατότητα πολλαπλών προσπαθειών:** Η πλατφόρμα ενθαρρύνει την εξάσκηση χωρίς τον φόβο της αποτυχίας, επιτρέποντας στον χρήστη να επαναλάβει την άσκηση έως ότου την ολοκληρώσει σωστά. Με αυτόν τον τρόπο ενισχύεται η μαθησιακή αυτοπεποίθηση.
- **Ανατροφοδότηση και επεξηγήσεις:** Μετά από κάθε προσπάθεια, παρέχεται άμεση ανατροφοδότηση για το αν η απάντηση είναι σωστή ή λανθασμένη, καθώς και επεξηγήσεις που βοηθούν στην κατανόηση του λάθους. Αυτό ενισχύει την αυτοδιόρθωση και τη σταδιακή βελτίωση.

### 5.2 Είδη ασκήσεων στην πλατφόρμα

Η πλατφόρμα CodeIT υποστηρίζει διάφορους τύπους ασκήσεων, ώστε να καλύπτει διαφορετικές μαθησιακές ανάγκες και να διατηρεί το ενδιαφέρον του χρήστη. Κάθε τύπος άσκησης έχει σχεδιαστεί

με σκοπό να ενισχύει συγκεκριμένες δεξιότητες και να αξιοποιεί κατάλληλα τις δυνατότητες του περιβάλλοντος.

1. Ερωτήσεις πολλαπλής επιλογής: Οι χρήστες καλούνται να επιλέξουν τη σωστή απάντηση ανάμεσα σε δύο ή περισσότερες επιλογές. Ο τύπος αυτός είναι ιδανικός για έλεγχο θεωρητικών γνώσεων και βασικών εννοιών. Επιτρέπει επίσης την ένταξη παραπλανητικών επιλογών (distractors) που ενισχύουν την κριτική σκέψη.
2. Ερωτήσεις σωστού/λάθους: Πρόκειται για απλές και γρήγορες ασκήσεις, όπου ο χρήστης καλείται να κρίνει αν μια πρόταση είναι ορθή ή λανθασμένη. Χρησιμοποιούνται συχνά για επανάληψη βασικών εννοιών ή για γρήγορη αξιολόγηση κατανόησης.
3. Ερωτήσεις συμπλήρωσης κενού: Ο χρήστης πρέπει να συμπληρώσει την απουσιάζουσα λέξη, σύμβολο ή τμήμα κώδικα σε μια πρόταση ή παράδειγμα. Ο τύπος αυτός ενισχύει την ακρίβεια στη σύνταξη και βοηθά στην απομνημόνευση κρίσιμων στοιχείων.
4. Ασκήσεις εκτέλεσης κώδικα: Με τη χρήση του ενσωματωμένου μεταγλωττιστή, ο χρήστης γράφει και εκτελεί πραγματικό κώδικα σε γλώσσα C, λαμβάνοντας άμεση ανατροφοδότηση. Αυτές οι ασκήσεις προσφέρουν πρακτική εξάσκηση και επιτρέπουν τη σύνδεση θεωρίας και εφαρμογής.

Η C υπάρχει εδώ και πολύ καιρό! Για τι πιστεύεις ότι χρησιμοποιείται;

Χρειάζεσαι βοήθεια;

Να γράφεις ιστορίες

Να δημιουργείς προγράμματα υπολογιστή

Να φτιάχνεις καφέ

Να ζωγραφίζεις εικόνες

Η C είναι μια γλώσσα που χρησιμοποιούμε για να δίνουμε εντολές στους υπολογιστές.

Χρειάζεσαι βοήθεια;

Σωστό

Λάθος

Υποβολή

Σχήμα 13: Παραδείγματα ασκήσεων πολλαπλής επιλογής και σωστού/λάθους

**Ερώτηση 3 από 10**

Στον κώδικα `void greet() { printf("Hi"); }`, το όνομα της συνάρτησης είναι \_\_\_\_\_.

Χρειάζεσαι βοήθεια;

greet()

Υποβολή

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n"); // Change this
    line
    return 0;
}
```

Εκτέλεση Κώδικα Υποβολή

Σχήμα 14: Παραδείγματα ασκήσεων συμπλήρωσης κενού και εκτέλεσης κώδικα

## Κεφάλαιο 5ο:

Κάθε κατηγορία άσκησης συνοδεύεται από μηχανισμό αυτόματης αξιολόγησης και επεξηγηματικά σχόλια, ώστε ο εκπαιδευόμενος να κατανοεί άμεσα το αποτέλεσμα της προσπάθειάς του.

### 5.3 Σύνδεση θεωρίας και πράξης

Η αποτελεσματική μάθηση προϋποθέτει τη στενή σύνδεση της θεωρητικής γνώσης με την πρακτική εφαρμογή της. Στην πλατφόρμα CodeIT, οι ασκήσεις έχουν σχεδιαστεί έτσι ώστε να ενσωματώνονται άμεσα μετά την παρουσίαση του αντίστοιχου θεωρητικού περιεχομένου. Με αυτόν τον τρόπο, ο χρήστης έχει την ευκαιρία να εφαρμόσει τις νέες έννοιες όσο αυτές είναι ακόμη πρόσφατες στη μνήμη του.

Η διαδοχή θεωρίας και άσκησης ακολουθεί μια λογική πορεία που μιμείται τον φυσικό τρόπο μάθησης:

1. Παρουσίαση της έννοιας ή του κανόνα με απλά παραδείγματα και κατανοητή γλώσσα. Παροχή άσκησης που απαιτεί την εφαρμογή του.
2. Παροχή άσκησης που απαιτεί την άμεση εφαρμογή της νέας γνώσης, ώστε ο εκπαιδευόμενος να δοκιμάσει τις δυνάμεις του.
3. Άμεση ανατροφοδότηση για το αν η απάντηση είναι σωστή ή λανθασμένη, με ενθαρρυντικό ύφος και επεξηγήσεις όπου χρειάζεται.

Αυτή η μεθοδικότητα εξασφαλίζει ότι η θεωρία δεν μένει σε αφηρημένο επίπεδο, αλλά μετατρέπεται σε πρακτική δεξιότητα. Η συνεχής εναλλαγή ανάμεσα σε κατανόηση και δράση ενισχύει την ενεργητική συμμετοχή του χρήστη και καλλιεργεί μια πιο σταθερή και ανθεκτική μνήμη. Παράλληλα, η διαδικασία της επανάληψης και η αίσθηση επιτυχίας που συνοδεύει την επίλυση κάθε άσκησης λειτουργούν ως μηχανισμός παρακίνησης, ενισχύοντας την αυτοπεποίθηση και το ενδιαφέρον του εκπαιδευόμενου να συνεχίσει.

Επιπλέον, η πρακτική εφαρμογή μέσα από σύντομες, στοχευμένες δραστηριότητες βοηθά τον μαθητευόμενο να αναγνωρίζει λάθη, να τα διορθώνει και να κατανοεί καλύτερα τις λεπτές αποχρώσεις της γλώσσας C. Με τον τρόπο αυτό, η μάθηση δεν αποτελεί μια παθητική διαδικασία παρακολούθησης, αλλά μια ενεργητική και διαδραστική εμπειρία που δημιουργεί στέρεες βάσεις για την περαιτέρω εξέλιξη.

### 5.4 Παιχνιδοποίηση και αξιολόγηση

Η εφαρμογή αξιοποιεί στοιχεία παιχνιδοποίησης προκειμένου να ενισχύσει την παρακίνηση των χρηστών και να διατηρήσει το ενδιαφέρον τους για συνεχή εξάσκηση. Η ένταξη των ασκήσεων σε ένα σύστημα ανταμοιβών και προόδου μετατρέπει τη μαθησιακή διαδικασία σε μια πιο ευχάριστη και ελκυστική εμπειρία, προσδίδοντάς της χαρακτηριστικά παιχνιδιού που κάνουν τον χρήστη να αισθάνεται ότι συμμετέχει σε μια εξερεύνηση με στόχους και επιτεύγματα.

Κύρια στοιχεία του συστήματος παιχνιδοποίησης:

1. Πόντοι εμπειρίας (XP): Κάθε ολοκληρωμένη άσκηση αποδίδει πόντους εμπειρίας, οι οποίοι λειτουργούν ως μέτρο ανταμοιβής για την προσπάθεια του χρήστη. Η συγκέντρωση XP δεν είναι απλώς ένας αριθμός, αλλά μια ορατή ένδειξη προόδου που ενισχύει το αίσθημα προσωπικής επιτυχίας.
2. Επίπεδα προόδου: Η αύξηση επιπέδου λειτουργεί ως απτή απόδειξη της εξέλιξης του χρήστη. Κάθε νέο επίπεδο μπορεί να συνοδεύεται από εικονικές ανταμοιβές, badges ή πρόσβαση σε νέο περιεχόμενο, δημιουργώντας μια αίσθηση συνέχειας και επιπλέον κίνητρα για να συνεχιστεί η προσπάθεια.
3. Σειρά συνεχόμενης συμμετοχής (streaks): Η καθημερινή εξάσκηση επιβραβεύεται με streaks, που ενισχύουν τη συνέπεια και ενθαρρύνουν τον χρήστη να υιοθετήσει μια σταθερή μαθησιακή ρουτίνα. Η απώλεια ενός streak μπορεί να λειτουργήσει ως κίνητρο για επιπλέον δέσμευση, καθώς ο χρήστης επιδιώκει να διατηρήσει το σερί του.

4. Άμεση ανατροφοδότηση: Κάθε απάντηση συνοδεύεται από σαφή ένδειξη επιτυχίας ή αποτυχίας, καθώς και από επεξηγήσεις όπου απαιτείται. Αυτό το στοιχείο αυξάνει τη διαδραστικότητα, μειώνει την αβεβαιότητα και επιτρέπει την ταχεία κατανόηση και διόρθωση λαθών.
5. Στατιστικά και παρακολούθηση προόδου: Ο χρήστης έχει πρόσβαση σε αναλυτικά στοιχεία σχετικά με την πορεία του, όπως τα μαθήματα που έχει ολοκληρώσει, τα ΧΡ που έχει συγκεντρώσει και τις περιοχές όπου χρειάζεται περισσότερη εξάσκηση. Έτσι, αποκτά μια συνολική εικόνα της εξέλιξής του και μπορεί να θέτει ρεαλιστικούς προσωπικούς στόχους.

Η χρήση αυτών των στοιχείων δεν αποσκοπεί μόνο στη διασκέδαση· στοχεύει κυρίως στην ενίσχυση της εσωτερικής παρακίνησης, μέσα από την ικανοποίηση που προσφέρει η προσωπική πρόοδος, αλλά και της εξωτερικής παρακίνησης, μέσω ορατών ανταμοιβών και επιτευγμάτων. Με αυτόν τον τρόπο, καλλιεργείται μεγαλύτερη αφοσίωση, αυξάνεται η διάρκεια ενασχόλησης με την εφαρμογή και βελτιώνονται τα μαθησιακά αποτελέσματα σε βάθος χρόνου.

## Κεφάλαιο 6ο: Συμπληρωματικές Τεχνολογίες και Εργαλεία Ανάπτυξης

Στο κεφάλαιο αυτό παρουσιάζονται οι επιπλέον τεχνολογίες και εργαλεία που υποστηρίζουν τη λειτουργία και την εμπειρία χρήσης της εφαρμογής CodeIT. Αν και δεν αποτελούν τον κεντρικό πυρήνα ανάπτυξης, διαδραματίζουν σημαντικό ρόλο στη βελτίωση της αλληλεπίδρασης του χρήστη, στην ομαλή λειτουργία του συστήματος και στη συνολική αισθητική του περιβάλλοντος. Εξετάζονται ο διαδικτυακός μεταγλωττιστής Judge0, οι βασικές βιβλιοθήκες React που ενσωματώθηκαν, καθώς και τα εργαλεία σχεδίασης και εφέ που συμβάλλουν στην ελκυστικότητα και τη χρηστικότητα της πλατφόρμας.

### 6.1 Judge0 API

Το Judge0 αποτελεί έναν διαδικτυακό μεταγλωττιστή και εκτελεστή κώδικα, ο οποίος υποστηρίζει πληθώρα γλωσσών προγραμματισμού, μεταξύ των οποίων και η C [14]. Η υπηρεσία βασίζεται σε ένα API που δέχεται κώδικα, τον εκτελεί σε απομακρυσμένο περιβάλλον και επιστρέφει στον χρήστη τα αποτελέσματα της εκτέλεσης (τυπική έξοδο, σφάλματα, χρόνο εκτέλεσης). Με αυτόν τον τρόπο, η διαδικασία ανάπτυξης και δοκιμής προγραμμάτων μεταφέρεται εξ ολοκλήρου στο διαδίκτυο, χωρίς να απαιτείται τοπική εγκατάσταση μεταγλωττιστών ή πρόσθετου λογισμικού.

Στην εφαρμογή CodeIT, το Judge0 αξιοποιείται για την αξιολόγηση των ασκήσεων προγραμματισμού. Οι μαθητές μπορούν να γράφουν τον κώδικά τους απευθείας από τον φυλλομετρητή τους, να τον εκτελούν σε πραγματικό χρόνο και να λαμβάνουν άμεσα ανατροφοδότηση. Αυτό καθιστά την εμπειρία μάθησης πιο διαδραστική και προσιτή, καθώς εξαλείφει τεχνικά εμπόδια που συχνά αποθαρρύνουν αρχάριους προγραμματιστές.

Η επιλογή του Judge0 έγινε για μια σειρά από λόγους:

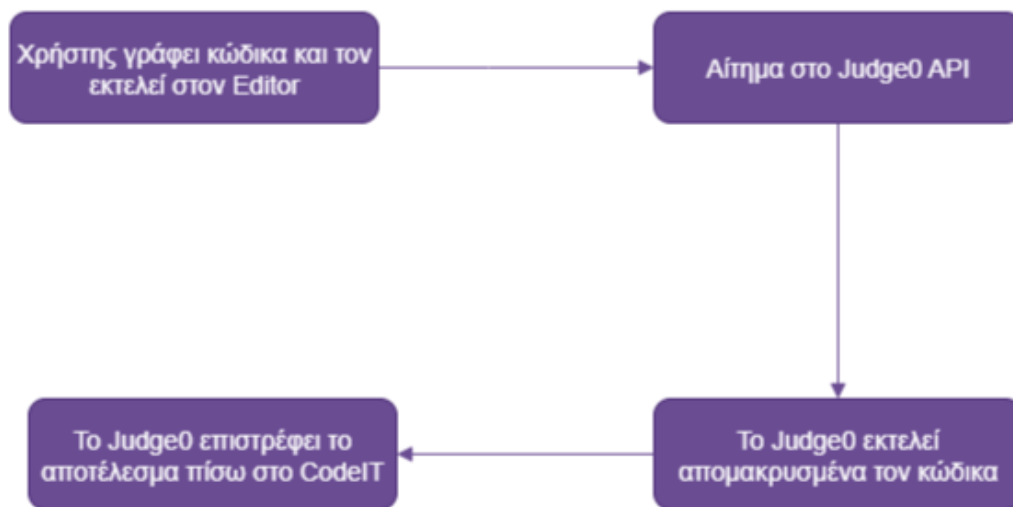
- Ευκολία ενσωμάτωσης: Το API διαθέτει σαφή τεκμηρίωση και απλές μεθόδους κλήσης, επιτρέποντας την ταχεία σύνδεσή του με εφαρμογές ιστού.
- Υποστήριξη πολλών γλωσσών: Παρόλο που η πλατφόρμα CodeIT επικεντρώνεται στη C, η ύπαρξη υποστήριξης για δεκάδες άλλες γλώσσες ανοίγει δυνατότητες μελλοντικής επέκτασης.
- Αξιοπιστία και ταχύτητα: Οι κλήσεις επιστρέφουν αποτελέσματα μέσα σε ελάχιστο χρόνο, προσφέροντας ομαλή εμπειρία χρήσης.
- Διαθεσιμότητα δωρεάν έκδοσης: Η ύπαρξη δημόσιου και δωρεάν endpoint καθιστά το Judge0 προσιτό για εκπαιδευτικά έργα.
- Ενεργή κοινότητα υποστήριξης: Η συνεχής ανάπτυξη και τακτική συντήρηση εξασφαλίζουν σταθερότητα και επικαιροποίηση.

Μέσω του Judge0, η πλατφόρμα CodeIT καταφέρνει να γεφυρώσει τη θεωρία με την πράξη: κάθε θεωρητικό μάθημα μπορεί να συνοδεύεται από ένα πραγματικό κομμάτι κώδικα που εκτελείται και αξιολογείται άμεσα. Έτσι, η μάθηση παύει να είναι παθητική και μετατρέπεται σε βιωματική διαδικασία, όπου οι μαθητές πειραματίζονται, κάνουν λάθη, τα διορθώνουν και τελικά εδραιώνουν τις γνώσεις τους πιο αποτελεσματικά.

### 6.2 Βιβλιοθήκες και Εργαλεία React

Για την ανάπτυξη της εφαρμογής CodeIT χρησιμοποιήθηκαν ορισμένες εξειδικευμένες βιβλιοθήκες του οικοσυστήματος React, οι οποίες πρόσθεσαν κρίσιμες λειτουργίες και συνέβαλαν στη βελτίωση της συνολικής εμπειρίας χρήσης.

- `react-simple-code-editor`: Χρησιμοποιείται για την εισαγωγή και επεξεργασία κώδικα μέσα από το περιβάλλον της εφαρμογής [15]. Με αυτή τη βιβλιοθήκη, οι χρήστες έχουν στη διάθεσή τους ένα ελαφρύ αλλά λειτουργικό `code editor`, ο οποίος υποστηρίζει βασικές δυνατότητες επεξεργασίας. Η επιλογή της βασίστηκε στην απλότητα ενσωμάτωσης και στη δυνατότητα προσαρμογής στις ανάγκες της πλατφόρμας.
- `Prism.js`: Σε συνδυασμό με τον `code editor`, το `Prism.js` αναλαμβάνει την επισήμανση σύνταξης (`syntax highlighting`) [16]. Η λειτουργία αυτή βελτιώνει σημαντικά την αναγνωσιμότητα του κώδικα, καθώς διαφορετικά στοιχεία (λέξεις-κλειδιά, συναρτήσεις, αριθμοί) επισημαίνονται με χρώματα. Έτσι, η κατανόηση των παραδειγμάτων και των λύσεων από τον μαθητή γίνεται πιο εύκολη και ευχάριστη.
- `react-confetti`: Προσφέρει τα εφέ εορτασμού που εμφανίζονται κατά την ολοκλήρωση ενός μαθήματος [17]. Η λειτουργία αυτή, αν και αισθητικά απλή, ενισχύει το αίσθημα επιβράβευσης και δημιουργεί θετική εμπειρία χρήσης, συμβάλλοντας στα στοιχεία παιχνιδιοποίησης που χαρακτηρίζουν την πλατφόρμα.
- `react-i18next`: Αποτελεί τον βασικό μηχανισμό πολυγλωσσικής υποστήριξης της εφαρμογής. Μέσω αυτής της βιβλιοθήκης, το περιβάλλον μπορεί να προσαρμόζεται δυναμικά στη γλώσσα που επιλέγει ο χρήστης (π.χ. Αγγλικά ή Ελληνικά), χωρίς να απαιτείται επανεκκίνηση ή ξεχωριστή έκδοση της εφαρμογής. Αυτό διευρύνει το κοινό στο οποίο απευθύνεται το `CodeIT` και βελτιώνει την προσβασιμότητα [18].



Σχήμα 15: Πως λειτουργεί το `Judge0` στο `CodeIT`

Η χρήση αυτών των εργαλείων ενίσχυσε τη διαδραστικότητα, βελτίωσε την αισθητική εμπειρία και υποστήριξε την προσβασιμότητα της πλατφόρμας. Παράλληλα, κατέστη δυνατή η διατήρηση μιας απλής, καθαρής και κατανοητής διεπαφής, ώστε ακόμη και οι αρχάριοι χρήστες να μπορούν να πλοηγηθούν και να εστιάσουν στη μάθηση χωρίς περιττές δυσκολίες.

## Κεφάλαιο 7ο: Supabase και βάσεις δεδομένων

Η εφαρμογή CodeIT βασίζεται σε μια αξιόπιστη και ευέλικτη υποδομή βάσης δεδομένων για την αποθήκευση και διαχείριση των πληροφοριών που σχετίζονται με τους χρήστες, τα μαθήματα και την πρόοδο τους. Για τον σκοπό αυτό επιλέχθηκε η πλατφόρμα Supabase, η οποία προσφέρει ολοκληρωμένες υπηρεσίες backend, όπως διαχείριση βάσης δεδομένων, ταυτοποίηση χρηστών και real-time λειτουργίες, μέσα από ένα εύχρηστο και σύγχρονο περιβάλλον.

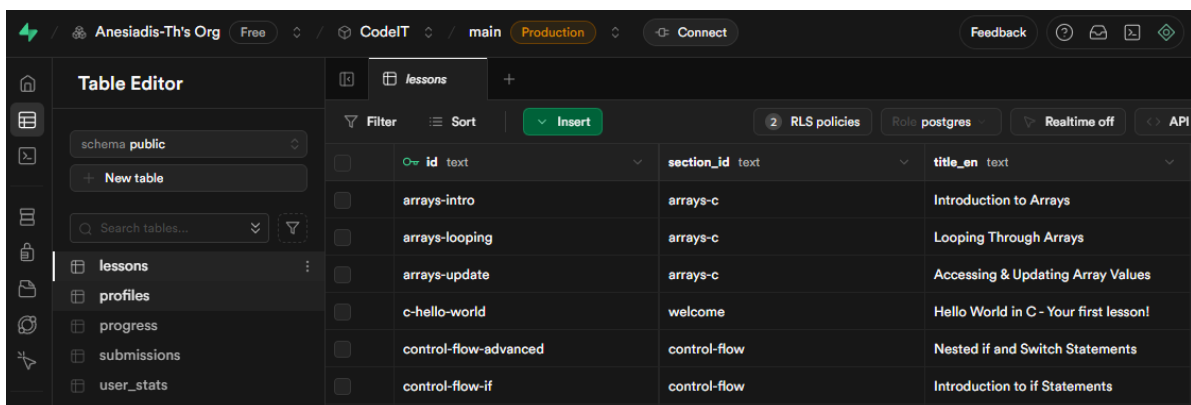
Το Supabase επιτρέπει την ομαλή επικοινωνία μεταξύ της React εφαρμογής και της βάσης δεδομένων, διασφαλίζοντας ταυτόχρονα την ασφάλεια των δεδομένων μέσω μηχανισμών όπως το Row Level Security. Στις επόμενες ενότητες παρουσιάζεται ο ρόλος του Supabase στην ανάπτυξη του CodeIT, η δομή των βασικών πινάκων και τα πλεονεκτήματα που προσφέρει αυτή η επιλογή σε σχέση με εναλλακτικές λύσεις.

### 7.1 Τι είναι το Supabase

Το Supabase είναι μια πλατφόρμα ανοιχτού κώδικα που παρέχει υπηρεσίες backend ως υπηρεσία (Backend-as-a-Service) [19]. Χτίζεται πάνω στη βάση δεδομένων PostgreSQL και προσφέρει μια σειρά από ενσωματωμένες λειτουργίες, όπως διαχείριση αυθεντικοποίησης χρηστών, αποθήκευση δεδομένων, real-time ενημερώσεις και αποθήκευση αρχείων.

Ουσιαστικά, το Supabase λειτουργεί ως ένας έτοιμος διακομιστής που συνδέεται εύκολα με εφαρμογές web και mobile, εξαλείφοντας την ανάγκη για την ανάπτυξη και συντήρηση μιας πλήρους υποδομής backend από το μηδέν. Μέσα από το ενσωματωμένο API που δημιουργείται αυτόματα για κάθε βάση δεδομένων, οι προγραμματιστές μπορούν να πραγματοποιούν άμεσα ερωτήματα και ενημερώσεις δεδομένων, χωρίς περίπλοκες ρυθμίσεις.

Η επιλογή του Supabase για την ανάπτυξη του CodeIT βασίστηκε στην ευκολία ενσωμάτωσης, στη συμβατότητα με React και στην ύπαρξη προηγμένων λειτουργιών ασφαλείας, οι οποίες είναι απαραίτητες σε μια εφαρμογή που διαχειρίζεται προσωπικά δεδομένα χρηστών και πληροφορίες προόδου.



Σχήμα 16: Περιβάλλον Supabase

### 7.2 Χρήση του Supabase στο CodeIT

Στην εφαρμογή CodeIT, το Supabase αποτελεί τον βασικό μηχανισμό διαχείρισης δεδομένων και ελέγχου πρόσβασης. Μέσω της σύνδεσης με τη βάση δεδομένων, αποθηκεύονται όλα τα απαραίτητα στοιχεία που σχετίζονται με τη λειτουργία της πλατφόρμας, όπως τα μαθήματα, η πρόοδος των χρηστών, οι βαθμολογίες και οι υποβολές κώδικα.

Ένας από τους σημαντικότερους ρόλους του Supabase είναι η αυθεντικοποίηση χρηστών. Η πλατφόρμα παρέχει πολλαπλές μεθόδους σύνδεσης, όπως email και κωδικό πρόσβασης, καθώς και OAuth μέσω Google και GitHub. Έτσι, οι χρήστες μπορούν να επιλέξουν τον τρόπο που τους εξυπηρετεί καλύτερα, ενώ η εφαρμογή διασφαλίζει την προστασία των προσωπικών τους δεδομένων.

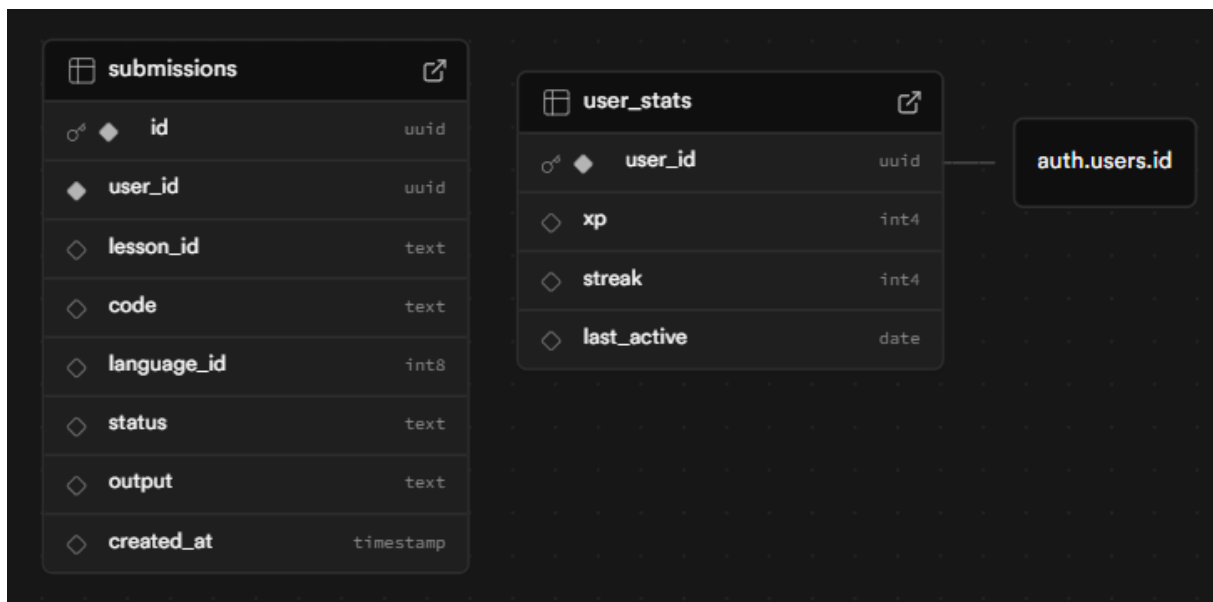
Επιπλέον, το Supabase αξιοποιείται για την παρακολούθηση της προόδου κάθε χρήστη. Κάθε φορά που ολοκληρώνεται ένα μάθημα, καταγράφεται αυτόματα η πρόοδος στη βάση, μαζί με την αντίστοιχη απόδοση σε εμπειρία (XP) και streak. Με αυτόν τον τρόπο, οι πληροφορίες είναι πάντα ενημερωμένες και συγχρονισμένες, ανεξάρτητα από τη συσκευή που χρησιμοποιεί ο χρήστης.

### 7.3 Δομή της Βάσης Δεδομένων

Η βάση δεδομένων του CodeIT, που υλοποιείται μέσω του Supabase, είναι σχεδιασμένη ώστε να καλύπτει όλες τις ανάγκες αποθήκευσης και ανάκτησης πληροφοριών της πλατφόρμας. Η δομή της οργανώνεται σε πίνακες με σαφείς ρόλους και σχέσεις μεταξύ τους, διασφαλίζοντας την ομαλή λειτουργία και την επεκτασιμότητα της εφαρμογής.

Οι βασικοί πίνακες περιλαμβάνουν:

- **lessons:** Περιέχει τις πληροφορίες κάθε μαθήματος, όπως τίτλο, περιεχόμενο, εισαγωγικό κείμενο και βήματα (steps). Υπάρχει υποστήριξη για πολλαπλές γλώσσες, με ξεχωριστά πεδία για κάθε μετάφραση.
- **progress:** Καταγράφει την πρόοδο κάθε χρήστη, συνδέοντας το αναγνωριστικό του μαθήματος με την κατάσταση ολοκλήρωσης.
- **user\_stats:** Αποθηκεύει στατιστικά στοιχεία, όπως επίπεδο, συνολικά XP και streak.
- **submissions:** Περιέχει τις υποβολές κώδικα των χρηστών, μαζί με το αποτέλεσμά τους (σωστό ή λανθασμένο) και την έξοδο του προγράμματος.



Σχήμα 17: Παράδειγμα δομής στην βάση δεδομένων του Supabase

Η οργάνωση αυτή επιτρέπει την εύκολη ανάκτηση δεδομένων, όπως π.χ. το να εμφανιστεί στον χρήστη ποια μαθήματα έχει ολοκληρώσει, ποια είναι σε εξέλιξη και ποια εκκρεμούν. Παράλληλα, η χρήση μηχανισμών ασφαλείας όπως το Row Level Security εξασφαλίζει ότι κάθε χρήστης έχει πρόσβαση μόνο στα δικά του δεδομένα.

## 7.4 Πλεονεκτήματα Χρήσης του Supabase

Η επιλογή του Supabase για την ανάπτυξη του CodeIT προσφέρει σημαντικά οφέλη, τόσο σε επίπεδο λειτουργικότητας όσο και σε επίπεδο ανάπτυξης. Ένα από τα βασικά πλεονεκτήματα είναι η ευκολία ενσωμάτωσης με εφαρμογές React, καθώς το Supabase παρέχει έτοιμες βιβλιοθήκες και τεκμηρίωση που επιτρέπουν γρήγορη και ασφαλή σύνδεση με τη βάση δεδομένων.

Η ύπαρξη αυτόματα δημιουργημένου API για κάθε πίνακα μειώνει τον χρόνο ανάπτυξης και απλοποιεί την αλληλεπίδραση με τα δεδομένα. Παράλληλα, η πλατφόρμα προσφέρει real-time δυνατότητες, επιτρέποντας την άμεση ενημέρωση του περιβάλλοντος του χρήστη όταν αλλάζουν δεδομένα στη βάση.

Σε επίπεδο ασφάλειας, το Row Level Security εξασφαλίζει ότι κάθε χρήστης μπορεί να δει και να επεξεργαστεί μόνο τα δικά του δεδομένα, ενισχύοντας την προστασία προσωπικών πληροφοριών. Επιπλέον, η υποστήριξη πολλαπλών μεθόδων αυθεντικοποίησης (email/password, Google, GitHub) προσφέρει ευελιξία και καλύτερη εμπειρία σύνδεσης.

Ένα ακόμη σημαντικό πλεονέκτημα είναι ότι το Supabase διαθέτει δωρεάν πλάνο, το οποίο καλύπτει πλήρως τις ανάγκες της εφαρμογής κατά την ανάπτυξη και τις δοκιμές, χωρίς επιπλέον κόστος. Αυτό το χαρακτηριστικό το καθιστά ιδιαίτερα ελκυστική επιλογή για ακαδημαϊκά έργα και μικρές ομάδες ανάπτυξης.

Συνολικά, το Supabase συνδυάζει την ευκολία χρήσης με προηγμένες δυνατότητες, καθιστώντας το μια ιδανική λύση για εκπαιδευτικές εφαρμογές που απαιτούν ασφάλεια, ταχύτητα και επεκτασιμότητα.

## Κεφάλαιο 8ο: Κώδικας

Αυτό το κεφάλαιο εστιάζει στον κώδικα της εφαρμογής CodeIT και παρέχει λεπτομερή ανάλυση της δομής του. Σκοπός είναι να καταγραφεί με οργανωμένο τρόπο η αρχιτεκτονική που επιλέχθηκε, τα οφέλη μιας καλά δομημένης βάσης κώδικα και η ανάλυση των διαφορετικών σελίδων και στοιχείων της διεπαφής. Η προσέγγιση αυτή αποδεικνύει όχι μόνο την τεχνική εκτέλεση, αλλά και τη σχεδιαστική φιλοσοφία που υποστηρίζει την επεκτασιμότητα, την ευκολία χρήσης και τη συντήρηση της πλατφόρμας.

### 8.1 Αρχιτεκτονική

Η εφαρμογή CodeIT ακολουθεί μια αρθρωτή και επεκτάσιμη αρχιτεκτονική, οργανωμένη με βάση τις αρχές του component-based design της React. Ο κώδικας χωρίζεται σε φακέλους με σαφή ρόλο, γεγονός που διευκολύνει την πλοήγηση, την κατανόηση και τη συντήρηση από οποιονδήποτε προγραμματιστή εργαστεί στο έργο. Οι βασικοί φάκελοι είναι οι εξής:

- **components:** Περιλαμβάνει επαναχρησιμοποιούμενα στοιχεία παρουσίασης και διάταξης που συμβάλλουν στη σύνθεση των σελίδων:
  - Header.jsx: Πλοήγηση, εμφάνιση στοιχείων χρήστη (επίπεδο, XP, streak) και επιλογές λογαριασμού.
  - Footer.jsx: Σύνδεσμοι GitHub/LinkedIn και εναλλαγή γλώσσας (EN/GR) με αποθήκευση προτίμησης στο localStorage.
  - Card.jsx: ουδέτερο container για τίτλο/εικονίδιο/περιεχόμενο, αξιοποιείται στην Home και αλλού για συνεπή παρουσίαση πληροφορίας.
- **pages:** Στο αυτόν το φάκελο, βρίσκονται όλες οι σελίδες της εφαρμογής:
  - Home: η αρχική σελίδα με παρουσίαση της πλατφόρμας.
  - Lessons: λίστα μαθημάτων οργανωμένη σε ενότητες.
  - LessonScreen: η σελίδα όπου ο χρήστης απαντάει σε quiz, γράφει κώδικα και κερδίζει XP.
  - Dashboard: τα στατιστικά του χρήστη (επίπεδο, XP, streaks).
  - Practice: χώρος για ελεύθερη εκτέλεση κώδικα.
  - Signup και Login: εγγραφή και σύνδεση χρηστών, με υποστήριξη Google και GitHub.
  - AdminDashboard και AdminLessonEditor: σελίδες για διαχειριστές που δημιουργούν και επεξεργάζονται μαθήματα.
- **lib** Φάκελος με αρχεία που διαχειρίζονται την επικοινωνία με τις υπηρεσίες που χρησιμοποιεί η εφαρμογή:
  - supabaseClient: συνδέει την εφαρμογή με το Supabase (βάση δεδομένων και χρήστες).
  - lessonService: φέρνει τα μαθήματα από τη βάση.
  - progressService: αποθηκεύει και ελέγχει την πρόοδο κάθε χρήστη.
  - statsService: χειρίζεται πόντους XP και επίπεδα.

- `judge0Service`: στέλνει κώδικα στο Judge0 API και επιστρέφει το αποτέλεσμα.
- **styles** Περιέχει τα CSS αρχεία της εφαρμογής.:
  - `globals.module.css`: γενικά στυλ που χρησιμοποιούνται σε πολλές σελίδες.
  - `header.module.css`, `footer.module.css`: στυλ ειδικά για Header και Footer.
  - `progressService`: αποθηκεύει και ελέγχει την πρόοδο κάθε χρήστη.
  - `editor.module.css`: στυλ για τον επεξεργαστή κώδικα.

### 8.1.1 Πλεονεκτήματα μιας σωστά δομημένης αρχιτεκτονικής κώδικα

Η σωστή οργάνωση του κώδικα δεν κάνει απλώς την εφαρμογή πιο «τακτοποιημένη». Έχει συγκεκριμένα πλεονεκτήματα που επηρεάζουν άμεσα τη συντήρηση, την ανάπτυξη νέων λειτουργιών και τη γενικότερη ποιότητα του λογισμικού.

- Ευκολία κατανόησης: Όταν ο κώδικας είναι οργανωμένος σε φακέλους με σαφή ρόλο (`components`, `pages`, `lib`, `styles`), ο προγραμματιστής καταλαβαίνει γρήγορα πού βρίσκεται κάθε λειτουργία. Αυτό μειώνει τον χρόνο εκμάθησης για νέους συνεργάτες στο έργο.
- Επαναχρησιμοποίηση: Τα `components` (π.χ. `Header`, `Footer`, `Card`) μπορούν να χρησιμοποιηθούν σε πολλές σελίδες χωρίς να γράφεται ξανά ο ίδιος κώδικας. Αυτό εξασφαλίζει μια συνεπή εμφάνιση και επιτυγχάνει οικονομία χρόνου.
- Συντηρησιμότητα: Όταν χρειαστεί μια αλλαγή, ο προγραμματιστής ξέρει ακριβώς σε ποιο αρχείο να επέμβει. Για παράδειγμα, αν αλλάξει ο τρόπος εμφάνισης του `Footer`, αρκεί να ενημερωθεί το αντίστοιχο `component`, χωρίς να επηρεαστούν άλλα μέρη της εφαρμογής.
- Επεκτασιμότητα: Η αρχιτεκτονική επιτρέπει να προστεθούν εύκολα νέες σελίδες ή νέες λειτουργίες χωρίς να επηρεαστεί ο υπάρχων κώδικας. Για παράδειγμα, μπορεί να προστεθεί ένα νέο μάθημα ή μια νέα σελίδα διαχείρισης χωρίς να αλλάξουν οι βασικές δομές.
- Ευκολία ελέγχου: Επειδή ο κώδικας είναι χωρισμένος σε μικρά, αυτόνομα κομμάτια, είναι πιο εύκολο να ελεγχθεί. Για παράδειγμα, μπορεί να ελεγχθεί μόνο η λειτουργία του `judge0Service`, χωρίς να χρειαστεί να φορτωθεί ολόκληρη η σελίδα `LessonScreen`.

### 8.1.2 ES Modules και Import/Export

Η εφαρμογή αξιοποιεί ES Modules (ECMAScript Modules) για την οργάνωση του κώδικα. Κάθε `component` ή βιβλιοθήκη ορίζεται σε ξεχωριστό αρχείο και γίνεται διαθέσιμο μέσω `export`. Στη συνέχεια, όπου χρειάζεται, γίνεται `import`. Παράδειγμα:

- Το `Footer.jsx` κάνει `export default Footer`
- Στη σελίδα `Home.jsx` γίνεται `import Footer from "../components/Footer"`;

Με αυτόν τον τρόπο επιτυγχάνεται σαφής εξάρτηση ανάμεσα στα μέρη του κώδικα και αποφεύγονται περιττές επαναλήψεις.

## 8.2 Σελίδες εφαρμογής

Οι σελίδες της εφαρμογής CodeIT οργανώνονται σε τρεις βασικές κατηγορίες:

- Κύριες σελίδες που αφορούν τη μαθησιακή εμπειρία του χρήστη.

- Σελίδες λογαριασμού και διαχείρισης.
- Κοινά στοιχεία διεπαφής (components) που χρησιμοποιούνται παντού.

## 8.2.1 Κύριες σελίδες

Οι κύριες σελίδες αποτελούν την «καρδιά» της εφαρμογής CodeIT, αφού εδώ διαδραματίζεται η μαθησιακή εμπειρία. Κάθε μία από αυτές έχει ξεχωριστό ρόλο στη ροή του χρήστη: από την πρώτη γνωριμία με την πλατφόρμα, μέχρι την αλληλεπίδραση με τα μαθήματα, την παρακολούθηση της προόδου και την ελεύθερη εξάσκηση.

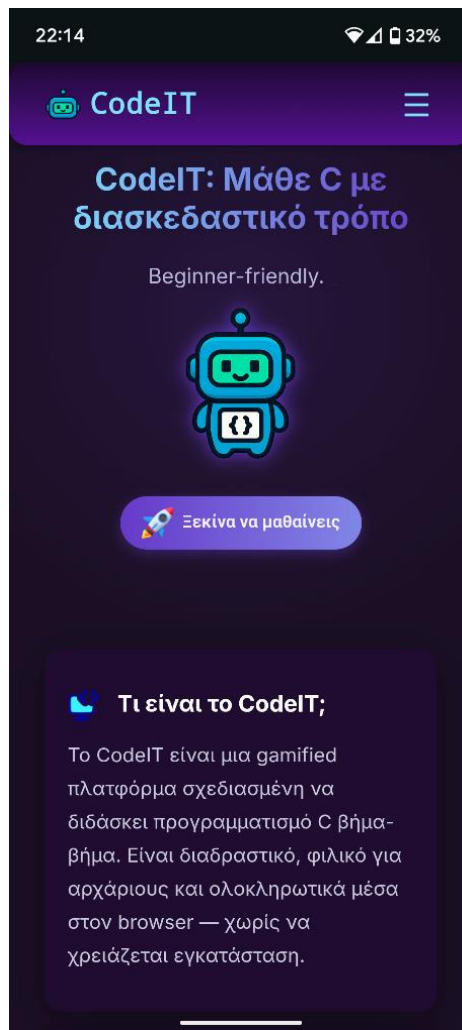
### 8.2.1.1 Home

Η αρχική σελίδα της εφαρμογής, Home, αποτελεί το πρώτο σημείο επαφής του χρήστη με την πλατφόρμα και είναι σχεδιασμένη ώστε να μεταδίδει άμεσα τον χαρακτήρα της: φιλική, παιχνιδιάρικη και προσανατολισμένη στη μάθηση. Στο επάνω μέρος της σελίδας δεσπόζει το Hero section, με έναν κεντρικό τίτλο σε έντονο χρωματισμό που αναδεικνύει την ταυτότητα της εφαρμογής. Δίπλα του εμφανίζεται ένα δυναμικό tagline, υλοποιημένο με animation τύπου “typewriter”, το οποίο εναλλάσσει λέξεις-κλειδιά όπως *Gamified*, *Beginner-friendly* και *100% in-browser*. Με αυτόν τον τρόπο, ο επισκέπτης αντιλαμβάνεται άμεσα τα βασικά χαρακτηριστικά της εμπειρίας που πρόκειται να ζήσει. Η μασκότ της εφαρμογής συμπληρώνει οπτικά το μήνυμα, εμφανιζόμενη με animation που δημιουργεί την αίσθηση αιώρησης και ενισχύει την παιχνιδιάρικη ατμόσφαιρα. Ένα εμφανές κουμπί Call-to-Action καλεί τον χρήστη να ξεκινήσει τη μαθησιακή του πορεία μεταφέροντάς τον στη σελίδα Lessons.

Ακολουθούν τα ενημερωτικά τμήματα της Home, τα οποία υλοποιούνται με τη βοήθεια του επαναχρησιμοποιούμενου component Card. Μέσα από αυτά παρουσιάζονται συνοπτικά το τι είναι η εφαρμογή, ποια είναι τα βασικά χαρακτηριστικά της και με ποιον τρόπο ο χρήστης ανταμείβεται για την πρόοδό του. Η χρήση καρτών με εικονίδια και οργανωμένο κείμενο συμβάλλει στη σαφήνεια και στη συνέπεια του οπτικού σχεδιασμού, ενώ ταυτόχρονα αναδεικνύει τα πλεονεκτήματα της αρχιτεκτονικής βασισμένης σε components.

Σε επίπεδο λειτουργίας, η Home αποτελεί μια καθαρά στατική σελίδα, χωρίς άμεσες κλήσεις προς το backend. Ωστόσο, ο σχεδιασμός της αξιοποιεί πλήρως την επαναχρησιμοποίηση components, κάτι που καθιστά ευέλικτη την εξέλιξη και τη συντήρησή της. Η σελίδα έχει υλοποιηθεί με responsive σχεδιασμό, ώστε να εμφανίζεται σωστά σε διαφορετικούς τύπους συσκευών, από κινητά και tablets έως οθόνες υπολογιστή. Η διάταξη προσαρμόζεται αυτόματα μέσω CSS modules και media queries, ενώ τα animations έχουν σχεδιαστεί ώστε να είναι ελαφριά και να μην επηρεάζουν αρνητικά την απόδοση σε μικρότερες συσκευές.

Συνολικά, η Home δεν λειτουργεί μόνο ως εισαγωγή στην εφαρμογή, αλλά ως ένα καλοσχεδιασμένο «κατώφλι» που δημιουργεί θετική πρώτη εντύπωση, καθοδηγεί τον χρήστη στο επόμενο βήμα και μεταδίδει με σαφήνεια το ύφος και τον σκοπό του CodeIT.



Σχήμα 18: Αρχική σελίδα

### 8.2.1.2 Lessons

Η σελίδα Lessons συγκεντρώνει όλα τα διαθέσιμα μαθήματα και τα παρουσιάζει σε οργανωμένη μορφή, αποτελώντας το κεντρικό σημείο πλοήγησης στη μαθησιακή εμπειρία. Σε επίπεδο υλοποίησης, η σελίδα έχει αναπτυχθεί ως React component (Lessons.jsx) και αξιοποιεί hooks όπως useState και useEffect για τη διαχείριση δεδομένων και την επικοινωνία με το backend.

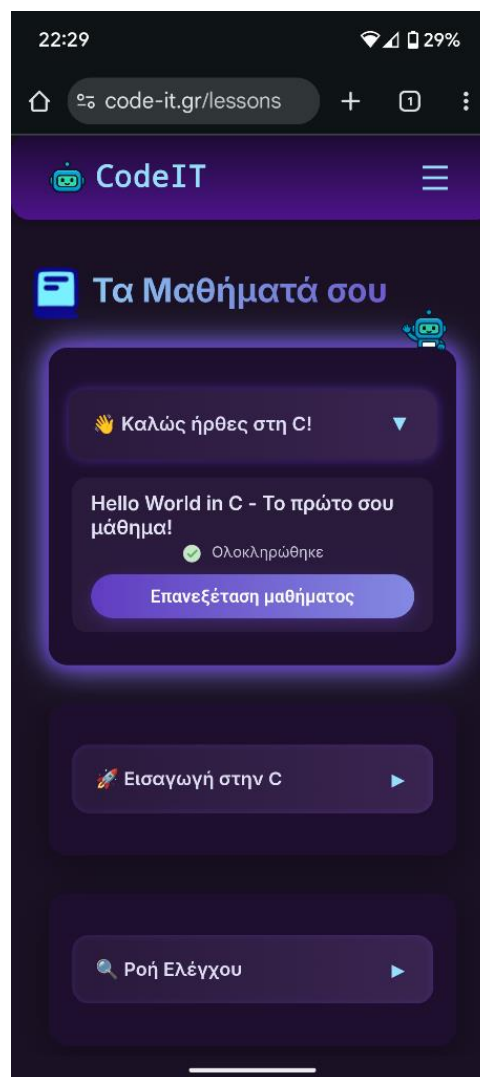
Η δομή των μαθημάτων βασίζεται στη λογική του accordion. Το ποια ενότητα είναι ανοιχτή ή κλειστή καθορίζεται μέσω ενός state, π.χ. `const [openSection, setOpenSection] = useState(null)`, το οποίο ενημερώνεται κάθε φορά που ο χρήστης κάνει κλικ σε μια θεματική ενότητα. Η απόδοση των μαθημάτων γίνεται με τη χρήση της μεθόδου `map()`, που διατρέχει το αντικείμενο μαθημάτων και δημιουργεί δυναμικά το αντίστοιχο JSX.

Τα δεδομένα των μαθημάτων αντλούνται από τη βάση δεδομένων μέσω Supabase. Κατά τη φόρτωση της σελίδας, εκτελείται ένα `useEffect` που καλεί τη συνάρτηση `fetchAllLessons()`. Αυτή επικοινωνεί με τον πίνακα `lessons` και επιστρέφει μια λίστα με όλα τα μαθήματα, ταξινομημένα με βάση το `section_id`. Στη συνέχεια, για κάθε μάθημα γίνεται έλεγχος της τρέχουσας γλώσσας του χρήστη, ώστε να εμφανίζεται ο κατάλληλος τίτλος. Αυτό επιτυγχάνεται με απλή δυναμική πρόσβαση στα πεδία της βάσης: `lesson["title_" + lang]`.

Ιδιαίτερη σημασία έχει η ενσωμάτωση της προόδου του χρήστη. Για τον σκοπό αυτό, εκτελείται ένα δεύτερο query στο Supabase, το οποίο ανακτά εγγραφές από τον πίνακα progress. Οι εγγραφές αυτές περιλαμβάνουν το lesson\_id και μια ένδειξη ολοκλήρωσης. Στη συνέχεια δημιουργείται ένας τοπικός χάρτης (progressMap), όπου κάθε μάθημα συσχετίζεται με την κατάστασή του (π.χ. ολοκληρωμένο, σε εξέλιξη, μη ξεκινήμένο). Αυτός ο μηχανισμός επιτρέπει την άμεση απόδοση οπτικών ενδείξεων δίπλα σε κάθε τίτλο μαθήματος.

Η εμπειρία διαφοροποιείται ανάμεσα σε ανώνυμους και συνδεδεμένους χρήστες. Για τους πρώτους, η σελίδα εμφανίζει τα διαθέσιμα μαθήματα χωρίς όμως να αποθηκεύει μόνιμα πρόοδο. Για τους δεύτερους, κάθε ολοκλήρωση μαθήματος ενημερώνει αυτόματα τον πίνακα progress, κάτι που επιτρέπει συνεχή παρακολούθηση και επαναφορά της πορείας ανεξάρτητα από τη συσκευή πρόσβασης. Παράλληλα, η Lessons αξιοποιεί το react-router-dom, ώστε κάθε επιλογή μαθήματος να οδηγεί με ασφάλεια στη σελίδα διδασκαλίας (LessonScreen). Το routing βασίζεται στο lessonId που περνάει ως παράμετρος στο URL, με χρήση του useNavigate για τη μετάβαση.

Συνολικά, η Lessons δεν είναι μια απλή λίστα, αλλά ένας δυναμικός συνδυασμός React state management, Supabase queries και πολυγλωσσικής υποστήριξης, που αποδίδει στον χρήστη έναν ξεκάθαρο χάρτη μάθησης. Ο σχεδιασμός της εξασφαλίζει ότι ο χρήστης γνωρίζει πάντα πού βρίσκεται, ενώ ο κώδικας που τη στηρίζει είναι οργανωμένος, επεκτάσιμος και προσαρμόσιμος σε μελλοντικές αλλαγές.



Σχήμα 19: Σελίδα Lessons

### 8.2.1.3 Lessons Screen

Η σελίδα LessonScreen αποτελεί τον πυρήνα της μαθησιακής εμπειρίας, καθώς εδώ διεξάγεται η πραγματική διδασκαλία. Σε επίπεδο υλοποίησης, πρόκειται για ένα React component (LessonScreen.jsx) που συνδυάζει παρουσίαση θεωρίας, αλληλεπίδραση μέσω κουίζ και άσκηση με εκτέλεση κώδικα. Ο σχεδιασμός της είναι δυναμικός και βασίζεται σε συνεχή αλληλεπίδραση με το backend, ώστε η πρόοδος του χρήστη να ενημερώνεται άμεσα.

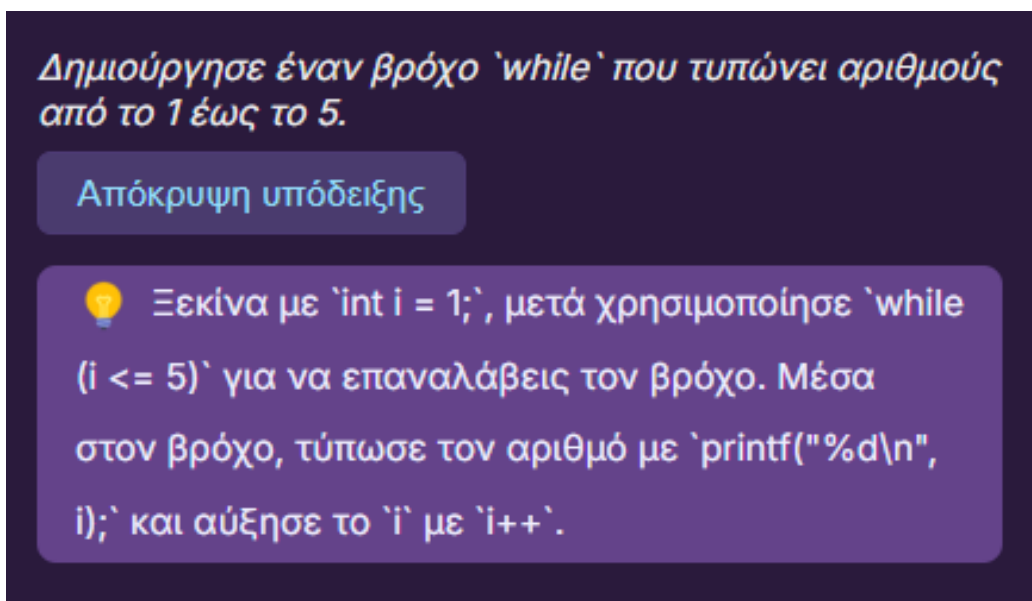
Η φόρτωση του περιεχομένου πραγματοποιείται με τη βοήθεια του useParams, το οποίο αντλεί το lessonId από το URL. Μέσα από ένα useEffect, καλείται η συνάρτηση fetchLessonById() η οποία επικοινωνεί με τον πίνακα lessons στο Supabase και επιστρέφει το αντίστοιχο αντικείμενο. Η πληροφορία αποθηκεύεται σε τοπικό state (const [lesson, setLesson] = useState(null)), μαζί με το σύνολο των βημάτων (steps) που θα ακολουθήσει ο χρήστης.

Η ροή μάθησης υλοποιείται με έναν μηχανισμό δείκτη ερωτήσεων (questionIndex). Με κάθε αλληλεπίδραση, όπως επιλογή απάντησης ή εκτέλεση κώδικα, ο δείκτης αυξάνεται και φορτώνεται το επόμενο βήμα. Η πρόοδος αποτυπώνεται οπτικά με progress bar, η οποία υπολογίζεται με βάση το ποσοστό ολοκλήρωσης των steps.

Η LessonScreen υποστηρίζει διαφορετικούς τύπους βημάτων: τα multiple-choice κουίζ αποδίδονται με λίστες επιλογών, τα fill-in-the-blank απαιτούν εισαγωγή κειμένου με έλεγχο αντιστοίχισης, ενώ τα code tasks ενσωματώνουν πλήρη editor για συγγραφή και εκτέλεση C κώδικα. Ο editor βασίζεται στο react-simple-code-editor και χρησιμοποιεί το PrismJS για syntax highlighting, ώστε ο κώδικας να εμφανίζεται με χρωματική μορφοποίηση. Η εκτέλεση πραγματοποιείται μέσω του Judge

API, με αποστολή request που περιέχει τον κώδικα του χρήστη και λήψη της εξόδου ή των σφαλμάτων σε ασύγχρονη ροή (async/await).

Ένα ακόμη στοιχείο είναι η λειτουργία “Need Help?”, η οποία εμφανίζει συμβουλή με animation unfolding, προσφέροντας υποστήριξη στον χρήστη όταν δυσκολεύεται. Με αυτόν τον τρόπο, το μάθημα δεν μένει στην απλή εξέταση γνώσης αλλά ενσωματώνει παιδαγωγικά εργαλεία καθοδήγησης.



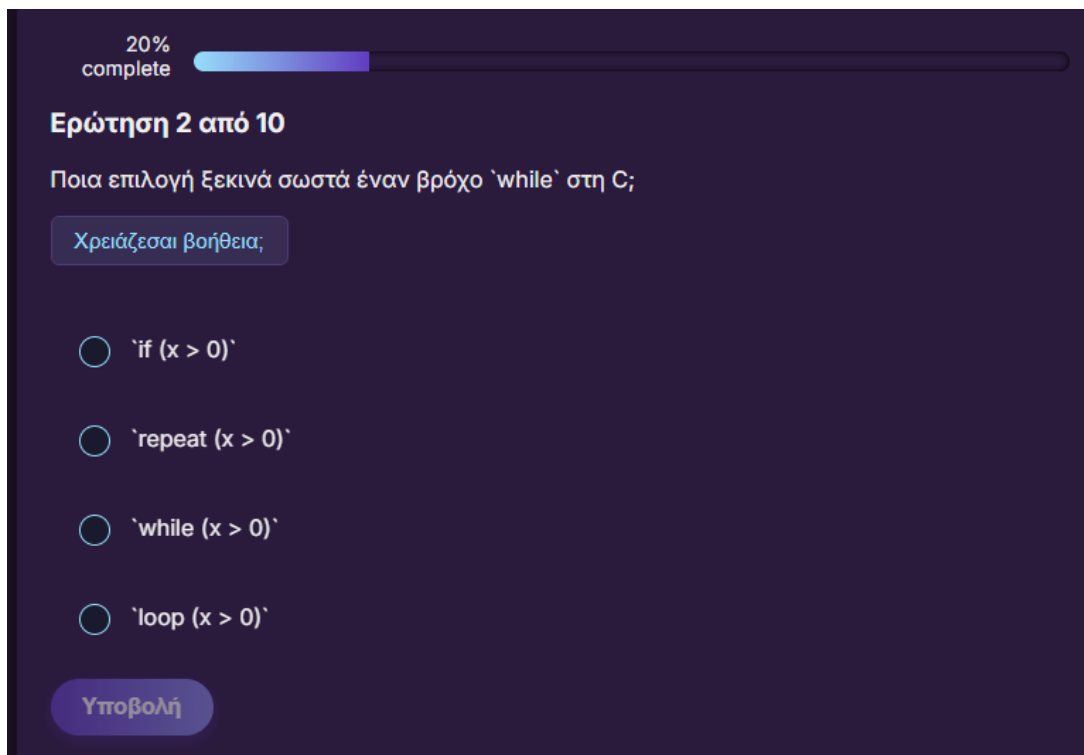
Σχήμα 20: Εμφάνιση βοήθειας

Η σελίδα διαφοροποιεί την εμπειρία ανάλογα με το αν ο χρήστης είναι επισκέπτης ή έχει λογαριασμό::

- **Συνδεδεμένος χρήστης:** Όταν ολοκληρωθεί ένα μάθημα, η εφαρμογή ενημερώνει το προφίλ του χρήστη στη βάση, καταγράφοντας την πρόοδο και απονέμοντας τους πόντους εμπειρίας (XP). Τα δεδομένα αυτά είναι αποθηκευμένα μόνιμα, επιτρέποντας στον χρήστη να συνεχίσει από εκεί που σταμάτησε, να παρακολουθεί το επίπεδό του και να διατηρεί streaks.
- **Ανώνυμος χρήστης:** Στην περίπτωση που ο χρήστης δεν έχει λογαριασμό, η εφαρμογή του επιτρέπει κανονικά να ολοκληρώσει το μάθημα και να δει τα αποτελέσματα επί τόπου. Ωστόσο, η πρόοδος και τα XP δεν αποθηκεύονται στη βάση. Αν επιστρέψει αργότερα, ξεκινάει από την αρχή. Με τον τρόπο αυτό, οι επισκέπτες μπορούν να δοκιμάσουν την εμπειρία χωρίς δεσμεύσεις, αλλά ενθαρρύνονται να δημιουργήσουν λογαριασμό για να επωφεληθούν από το σύστημα ανταμοιβών και παρακολούθησης προόδου.

Με αυτόν τον τρόπο, η LessonScreen συνδυάζει διαδραστική μάθηση με προσωποποιημένη εμπειρία, προσαρμόζοντας τη λειτουργικότητά της ανάλογα με το αν ο χρήστης είναι επισκέπτης ή μέλος.

Η ολοκλήρωση ενός μαθήματος συνοδεύεται από οπτικά εφέ. Το react-confetti δημιουργεί εορταστική ατμόσφαιρα, ενώ εμφανίζεται μήνυμα συγχαρητηρίων και η μασκότ σε εορταστική στάση. Έτσι, το μάθημα κλείνει με θετικό τρόπο, ενισχύοντας την παρακίνηση και το αίσθημα επίτευξης.



Σχήμα 21: Σελίδα LessonScreen

Η LessonScreen είναι το πιο σύνθετο κομμάτι του CodeIT, καθώς συνδυάζει δυναμική φόρτωση δεδομένων, state management, πολυμορφία ερωτήσεων και ενσωμάτωση εξωτερικών API σε μία ενιαία εμπειρία. Ο κώδικας της είναι δομημένος με τρόπο που επιτρέπει την εύκολη υποστήριξη νέων τύπων ερωτήσεων στο μέλλον, αναδεικνύοντας την αξία της αρθρωτής αρχιτεκτονικής που ακολουθείται σε όλη την εφαρμογή.

#### 8.2.1.4 Dashboard

Η σελίδα Dashboard λειτουργεί ως το προσωπικό κέντρο ελέγχου κάθε χρήστη, συγκεντρώνοντας πληροφορίες για την πορεία και τα επιτεύγματά του. Σε επίπεδο κώδικα, υλοποιείται ως React

component (Dashboard.jsx) και βασίζεται σε δεδομένα που αντλούνται δυναμικά από το Supabase, τα οποία στη συνέχεια αποθηκεύονται σε state μεταβλητές με τη χρήση των hooks useState και useEffect.

Κατά τη φόρτωση της σελίδας, η εφαρμογή εκτελεί μια σειρά από κλήσεις στο backend. Αρχικά γίνεται αναγνώριση του χρήστη μέσω `supabase.auth.getUser()`. Στη συνέχεια, αν ο χρήστης είναι συνδεδεμένος, καλείται η συνάρτηση `getUserStats()` για να ανακτηθούν τα στατιστικά στοιχεία του (επίπεδο, XP, streak). Παράλληλα, μέσω της `fetchAllLessons()` αντλούνται όλα τα μαθήματα και μέσω της `fetchUserProgress()` δημιουργείται ένας χάρτης (progressMap) που συσχετίζει κάθε μάθημα με την κατάσταση ολοκλήρωσής του. Αυτά τα δεδομένα επιτρέπουν την ακριβή απεικόνιση της προόδου στον χρήστη.

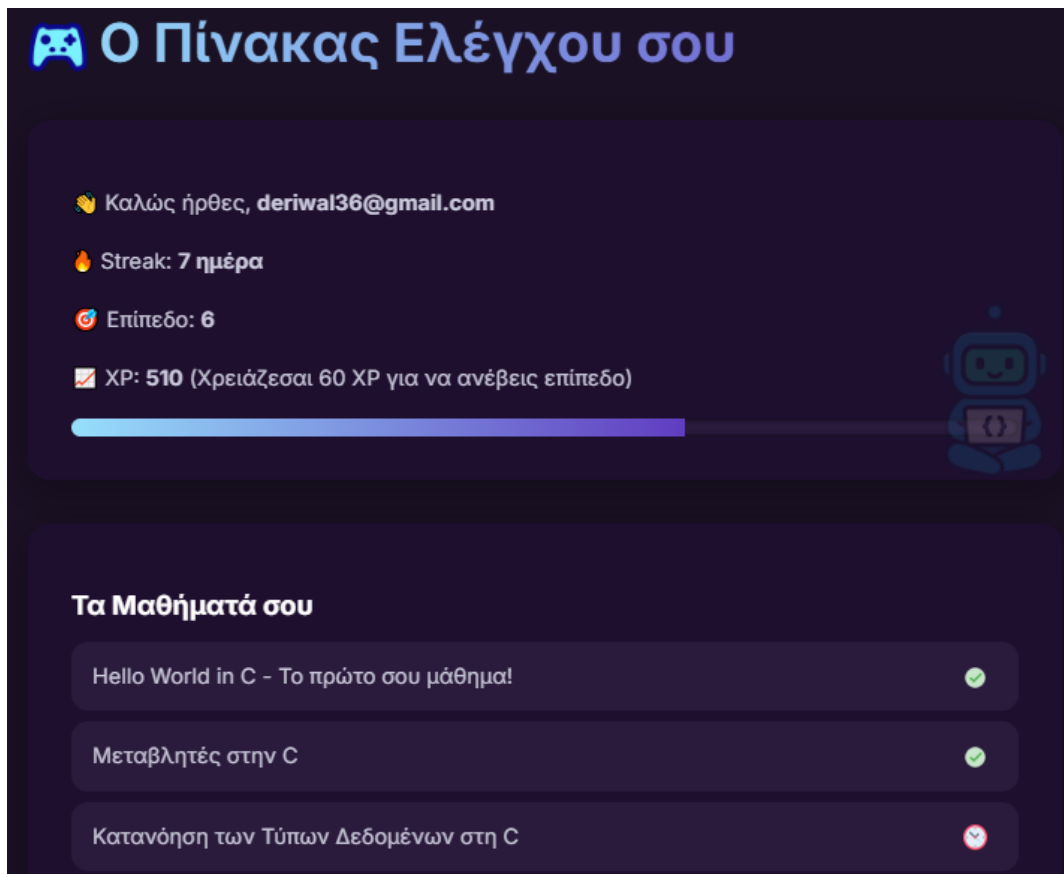
Η πληροφορία παρουσιάζεται στον χρήστη με τη μορφή προόδου και ανταμοιβών, ενισχύοντας το στοιχείο της παιχνιδοποίησης. Ειδικότερα:

- Πρόοδος μαθημάτων: εμφανίζεται ποια μαθήματα έχουν ολοκληρωθεί και ποια εκκρεμούν. Η πληροφορία προέρχεται από τον πίνακα `progress`.
- Σύστημα XP και Levels: κάθε ολοκλήρωση μαθήματος προσθέτει XP στον πίνακα `stats`. Η μπάρα προόδου προς το επόμενο level ενημερώνεται δυναμικά, με τον απαιτούμενο αριθμό XP να αυξάνεται προοδευτικά ανά επίπεδο.
- Streaks: εμφανίζεται με ένδειξη πόσες συνεχόμενες ημέρες ο χρήστης δραστηριοποιείται στην πλατφόρμα. Η λογική βασίζεται σε timestamps που καταγράφονται στο Supabase.

Η σελίδα διαφοροποιεί την εμπειρία ανάλογα με το αν ο χρήστης είναι επισκέπτης ή συνδεδεμένος:

- **Ανώνυμος χρήστης:** βλέπει μια απλουστευμένη έκδοση με προσωρινά δεδομένα, αλλά χωρίς καταγραφή μακροχρόνιας προόδου.
- **Συνδεδεμένος χρήστης:** έχει πλήρη πρόσβαση στα στατιστικά του, με αποθήκευση και συγχρονισμό δεδομένων σε πραγματικό χρόνο. Αυτός ο μηχανισμός ενισχύει την παρακίνηση, καθώς η πρόοδος αποκτά μονιμότητα και ο χρήστης αισθάνεται ότι οι προσπάθειές του αποδίδουν.

Από άποψη διεπαφής, η Dashboard παρουσιάζει την πληροφορία με απλό και καθαρό τρόπο. Χρησιμοποιεί progress bars, αριθμητικούς μετρητές και εικονίδια που δίνουν αίσθηση παιχνιδιού, χωρίς να αποσπών τον χρήστη από τον κύριο στόχο. Η αισθητική παραμένει συμβατή με το υπόλοιπο περιβάλλον, αξιοποιώντας κοινά components και στυλ από τα CSS modules.



Σχήμα 22: Σελίδα Dashboard

Σε τεχνικό επίπεδο, η Dashboard αποτελεί παράδειγμα συνδυασμού πολλαπλών πηγών δεδομένων (μαθήματα, πρόοδος, στατιστικά) και παρουσίασής τους σε ενιαία μορφή. Ο κώδικας είναι οργανωμένος με τέτοιο τρόπο ώστε να μπορεί εύκολα να επεκταθεί, για παράδειγμα με μελλοντικές λειτουργίες όπως πίνακες κατάταξης (leaderboards) ή badges. Με τον τρόπο αυτό, η Dashboard δεν είναι μόνο εργαλείο παρακολούθησης, αλλά και μηχανισμός ενίσχυσης της παρακίνησης, ενσωματώνοντας πλήρως τη λογική της παιχνιδοποίησης.

### 8.2.1.5 Practice

Η σελίδα Practice προσφέρει στον χρήστη έναν χώρο ελεύθερης εξάσκησης, όπου μπορεί να γράψει και να εκτελέσει δικό του κώδικα σε C. Αποτελεί ένα περιβάλλον δοκιμών το οποίο επιτρέπει την πειραματική προσέγγιση και τη δημιουργική εξερεύνηση, χωρίς τον περιορισμό των βημάτων που ορίζονται στα μαθήματα. Παράλληλα, λειτουργεί ως γέφυρα ανάμεσα στη θεωρητική γνώση και την πρακτική εφαρμογή, ενισχύοντας την ικανότητα του χρήστη να μεταφέρει τις έννοιες που διδάχθηκε σε πραγματικά παραδείγματα. Αξιοποιεί τον κοινό Editor της εφαρμογής, ο οποίος προσφέρει τη δυνατότητα εισαγωγής και εκτέλεσης κώδικα απευθείας μέσα στο περιβάλλον του CodeIT. Η επικοινωνία με το Judge0 API επιτρέπει την άμεση εκτέλεση του κώδικα και την εμφάνιση των αποτελεσμάτων στον χρήστη σε πραγματικό χρόνο.

Η σχεδίαση της σελίδας είναι εσκεμμένα λιτή, με κεντρικό στοιχείο τον Editor και το κουμπί εκτέλεσης. Έτσι ο χρήστης μπορεί να επικεντρωθεί πλήρως στον πειραματισμό, είτε πρόκειται για δοκιμή μιας μικρής εντολής είτε για την ανάπτυξη ενός ολοκληρωμένου προγράμματος.

## Εξάσκηση κώδικα C

```
#include <stdio.h>
int main() {
    printf("Hello, C!");
    return 0;
}
```

Εκτέλεση Κώδικα

✓ Έξοδος: Hello, C!

Σχήμα 23: Σελίδα Practice

Με τον τρόπο αυτό, η Practice λειτουργεί ως γέφυρα ανάμεσα στη θεωρητική γνώση και την πρακτική εφαρμογή, παρέχοντας ένα περιβάλλον που ενθαρρύνει την ενεργή μάθηση και τη δημιουργική διερεύνηση της γλώσσας C.

### 8.2.2 Σελίδες λογαριασμού και διαχείρισης

Εκτός από τις κύριες σελίδες μάθησης, η εφαρμογή περιλαμβάνει σελίδες που σχετίζονται με τη διαχείριση λογαριασμού και περιεχομένου. Μέσω αυτών, ο χρήστης μπορεί να εγγραφεί, να συνδεθεί και να αποθηκεύσει την πρόοδό του, ενώ ο διαχειριστής έχει στη διάθεσή του εργαλεία για την οργάνωση και επεξεργασία μαθημάτων.

#### 8.2.2.1 Signup & Login

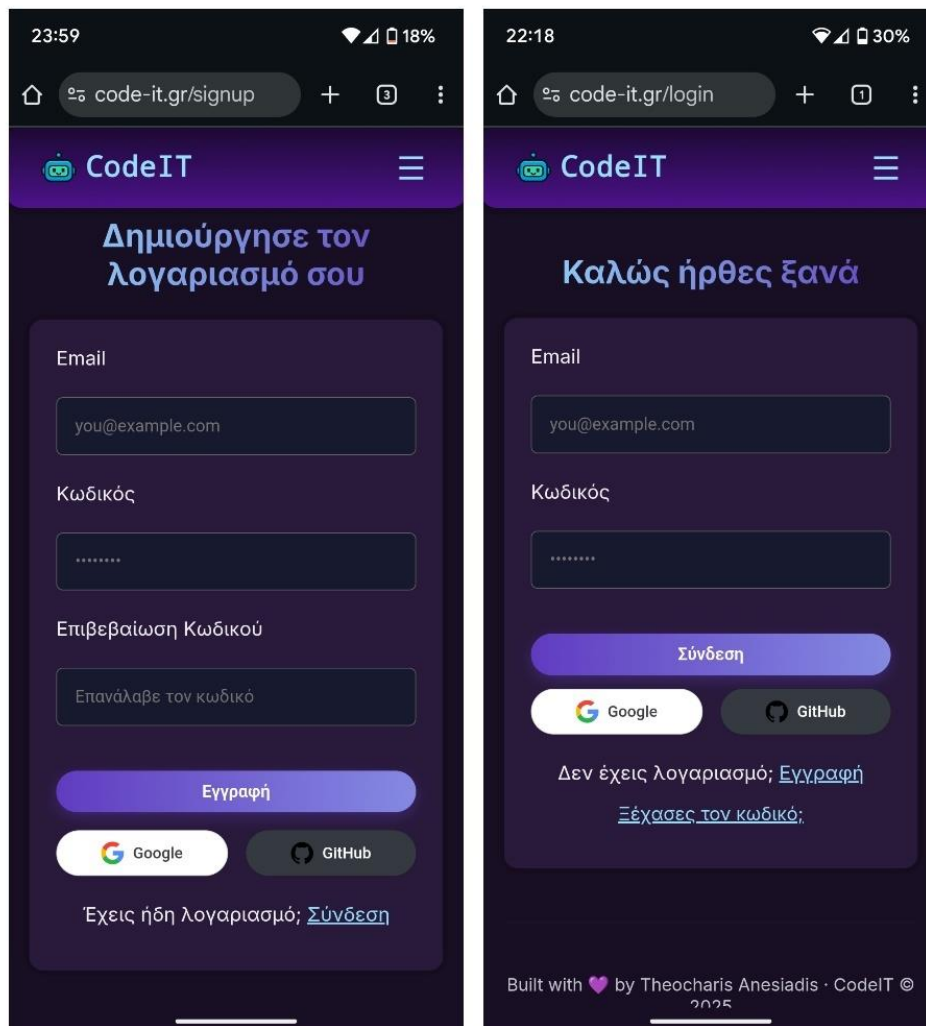
Οι σελίδες Signup & Login αποτελούν το σημείο αυθεντικοποίησης των χρηστών, επιτρέποντάς τους να δημιουργήσουν λογαριασμό ή να συνδεθούν σε υπάρχοντα. Οι δύο λειτουργίες συνδυάζονται με κοινή αρχιτεκτονική κώδικα και ενοποιημένο στυλ για συνέπεια στην εμπειρία χρήσης.

Η ταυτοποίηση βασίζεται στο Supabase Auth, το οποίο παρέχει έτοιμες μεθόδους εγγραφής και σύνδεσης. Ο χρήστης μπορεί να επιλέξει είτε την κλασική αυθεντικοποίηση με email/κωδικό πρόσβασης είτε τη χρήση OAuth μέσω παρόχων όπως Google και GitHub. Η ροή ξεκινά με κλήση της συνάρτησης `supabase.auth.signUp()` για την εγγραφή ή `supabase.auth.signInWithPassword()` για τη σύνδεση. Οι μέθοδοι αυτοί επιστρέφουν αντικείμενα με πληροφορίες χρήστη και tokens τα οποία αποθηκεύονται αυτόματα στο session, εξασφαλίζοντας ασφαλή πρόσβαση σε προστατευμένες λειτουργίες.

Η σελίδα ενσωματώνει επίσης δυνατότητα επαναφοράς κωδικού. Αν ο χρήστης ξεχάσει τον κωδικό του, μπορεί να ζητήσει reset μέσω Supabase, το οποίο αποστέλλει αυτόματα email με οδηγίες. Για

ταχύτερη εμπειρία, υλοποιήθηκε και inline reset, όπου το αίτημα στέλνεται απευθείας από τη φόρμα σύνδεσης.

Από άποψη UX, δόθηκε έμφαση στη σαφήνεια και απλότητα. Η φόρμα περιλαμβάνει ελάχιστα πεδία (email και κωδικό), ενώ τα μηνύματα επιτυχίας ή λάθους εμφανίζονται με καθαρά χρωματικά στοιχεία και εικονίδια, ώστε ο χρήστης να γνωρίζει άμεσα την κατάσταση.



Σχήμα 24: Σελίδα Signup (αριστερά) και σελίδα Login (δεξιά)

### 8.2.2.2 Admin Dashboard

Η σελίδα Admin Dashboard αποτελεί το κεντρικό εργαλείο διαχείρισης του CodeIT, προσφέροντας στους διαχειριστές τη δυνατότητα να ελέγχουν και να επεξεργάζονται το περιεχόμενο της πλατφόρμας. Σε αντίθεση με τις υπόλοιπες σελίδες που απευθύνονται στους μαθητές, το Admin Dashboard είναι προσβάσιμο μόνο σε χρήστες με αυξημένα δικαιώματα, εξασφαλίζοντας ότι η δημιουργία και η τροποποίηση μαθημάτων γίνεται με ασφαλή και ελεγχόμενο τρόπο.

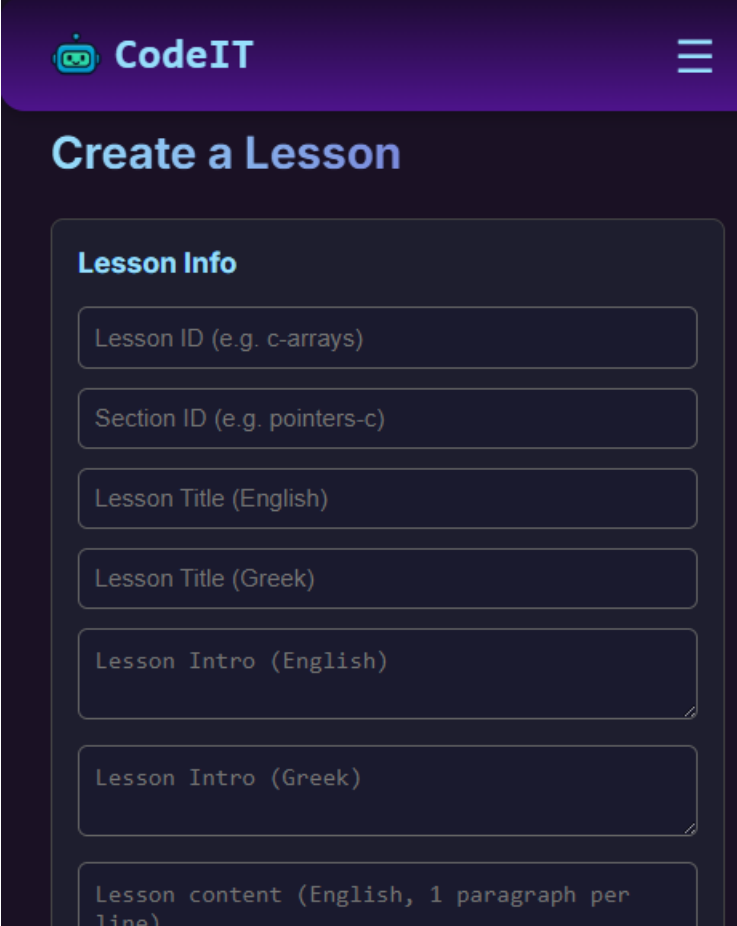
Η σελίδα διασυνδέεται άμεσα με το Supabase για ανάγνωση και τροποποίηση δεδομένων. Κατά την είσοδο, ελέγχεται το προφίλ του χρήστη μέσω του μηχανισμού Row Level Security (RLS). Αν ο χρήστης δεν διαθέτει τον ρόλο διαχειριστή, η πρόσβαση απορρίπτεται και ο χρήστης ανακατευθύνεται.

Το Admin Dashboard συγκεντρώνει σε μία κεντρική οθόνη βασικές ενέργειες διαχείρισης:

- Προβολή λίστας μαθημάτων: γίνεται κλήση στο Supabase για ανάκτηση όλων των εγγραφών από τον πίνακα lessons, με δυνατότητα ταξινόμησης και αναζήτησης.
- Δημιουργία νέου μαθήματος: μέσω φόρμας που οδηγεί στον AdminLessonEditor, οι διαχειριστές μπορούν να προσθέσουν τίτλους, περιεχόμενο, βήματα (steps) και κώδικα.
- Επεξεργασία ή διαγραφή μαθημάτων: κάθε μάθημα συνοδεύεται από επιλογές για γρήγορη ενημέρωση ή οριστική διαγραφή από τη βάση.

Η σελίδα αξιοποιεί React hooks για state management και επικοινωνεί με το Supabase μέσω ασύγχρονων συναρτήσεων. Ενδεικτικά, η φόρτωση όλων των μαθημάτων πραγματοποιείται μέσα από ένα useEffect που καλεί `supabase.from("lessons").select("*")`, ενώ οι ενέργειες δημιουργίας/ενημέρωσης γίνονται με insert, update και delete queries.

Συνολικά, το Admin Dashboard ενσαρκώνει τη φιλοσοφία ότι μια εκπαιδευτική πλατφόρμα δεν χρειάζεται μόνο καλοσχεδιασμένη εμπειρία χρήστη, αλλά και ισχυρά εργαλεία διαχείρισης περιεχομένου, τα οποία διασφαλίζουν την ποιότητα, την οργάνωση και την εξέλιξη του μαθησιακού υλικού.



The image shows a mobile application interface for 'CodeIT'. At the top, there is a purple header with the 'CodeIT' logo and a hamburger menu icon. Below the header, the main title is 'Create a Lesson'. The form is titled 'Lesson Info' and contains several input fields: 'Lesson ID (e.g. c-arrays)', 'Section ID (e.g. pointers-c)', 'Lesson Title (English)', 'Lesson Title (Greek)', 'Lesson Intro (English)', 'Lesson Intro (Greek)', and 'Lesson content (English, 1 paragraph per line)'. Each field has a light blue border and a light blue background. The form is set against a dark purple background.

Σχήμα 25: Σελίδα Admin Dashboard

## 8.3 Κοινά Στοιχεία Διεπαφής

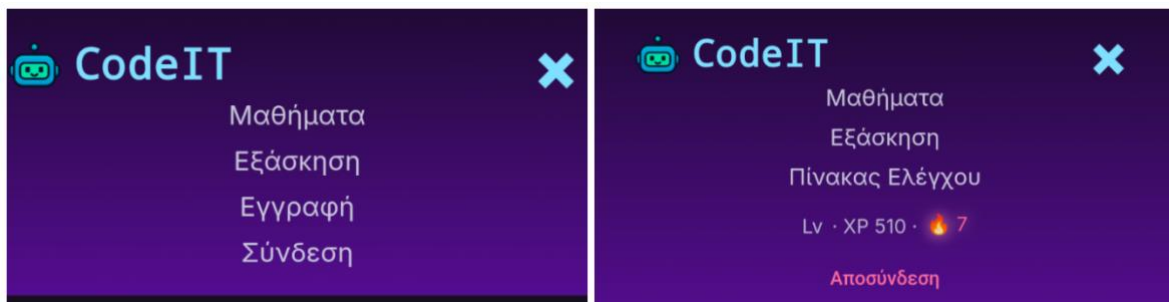
Εκτός από τις βασικές σελίδες της εφαρμογής, το CodeIT περιλαμβάνει και μια σειρά από κοινά στοιχεία διεπαφής (UI components), τα οποία επαναχρησιμοποιούνται σε διαφορετικά σημεία. Αυτά τα στοιχεία δεν αποτελούν αυτόνομες σελίδες, αλλά δομικά κομμάτια που διασφαλίζουν συνέπεια στον σχεδιασμό και απλοποιούν την ανάπτυξη. Η χρήση τους επιτρέπει την ομοιομορφία στην εμπειρία χρήστη, μειώνει τον πλεονασμό κώδικα και διευκολύνει τη μελλοντική συντήρηση.

### 8.3.1 Header

Το Header αποτελεί το βασικό σημείο πλοήγησης της εφαρμογής. Υλοποιείται ως component (Header.jsx) και περιλαμβάνει το λογότυπο, τα βασικά links πλοήγησης (Lessons, Practice, Dashboard), καθώς και επιλογές αυθεντικοποίησης (Login/Logout). Σε επίπεδο κώδικα, το Header είναι “sticky”, παραμένοντας στην κορυφή της σελίδας κατά την κύλιση, ώστε να παρέχει σταθερή πρόσβαση στο μενού.

Η αρχιτεκτονική του βασίζεται σε conditional rendering:

- Αν ο χρήστης είναι ανώνυμος, εμφανίζονται οι επιλογές Sign Up και Login.
- Αν ο χρήστης είναι συνδεδεμένος, εμφανίζεται η επιλογή Logout καθώς και πρόσβαση στο Dashboard.



Σχήμα 26: Header ως ανώνυμος χρήστης (αριστερά) και ως συνδεδεμένος χρήστης (δεξιά)

### 8.3.2 Footer

Το Footer αποτελεί το στοιχείο που κλείνει κάθε σελίδα της εφαρμογής και παρέχει βασικές πληροφορίες και συνδέσμους. Υλοποιείται ως component (Footer.jsx) και διατηρεί απλή, λιτή δομή, ώστε να μην αποσπά τον χρήστη από το κύριο περιεχόμενο.

Στη βασική του μορφή περιλαμβάνει ένα σταθερό μήνυμα πνευματικών δικαιωμάτων, με αναφορά στο όνομα του δημιουργού και στο έτος, ενώ ακριβώς από κάτω ενσωματώνονται σύνδεσμοι προς GitHub και LinkedIn. Οι σύνδεσμοι αυτοί εμφανίζονται με τη μορφή εικονιδίων και ανοίγουν σε νέο παράθυρο, ώστε ο χρήστης να μην χάνει την τρέχουσα συνεδρία στην εφαρμογή. Ο σχεδιασμός αφήνει περιθώριο για μελλοντική επέκταση, όπως προσθήκη social media icons ή σελίδων υποστήριξης.

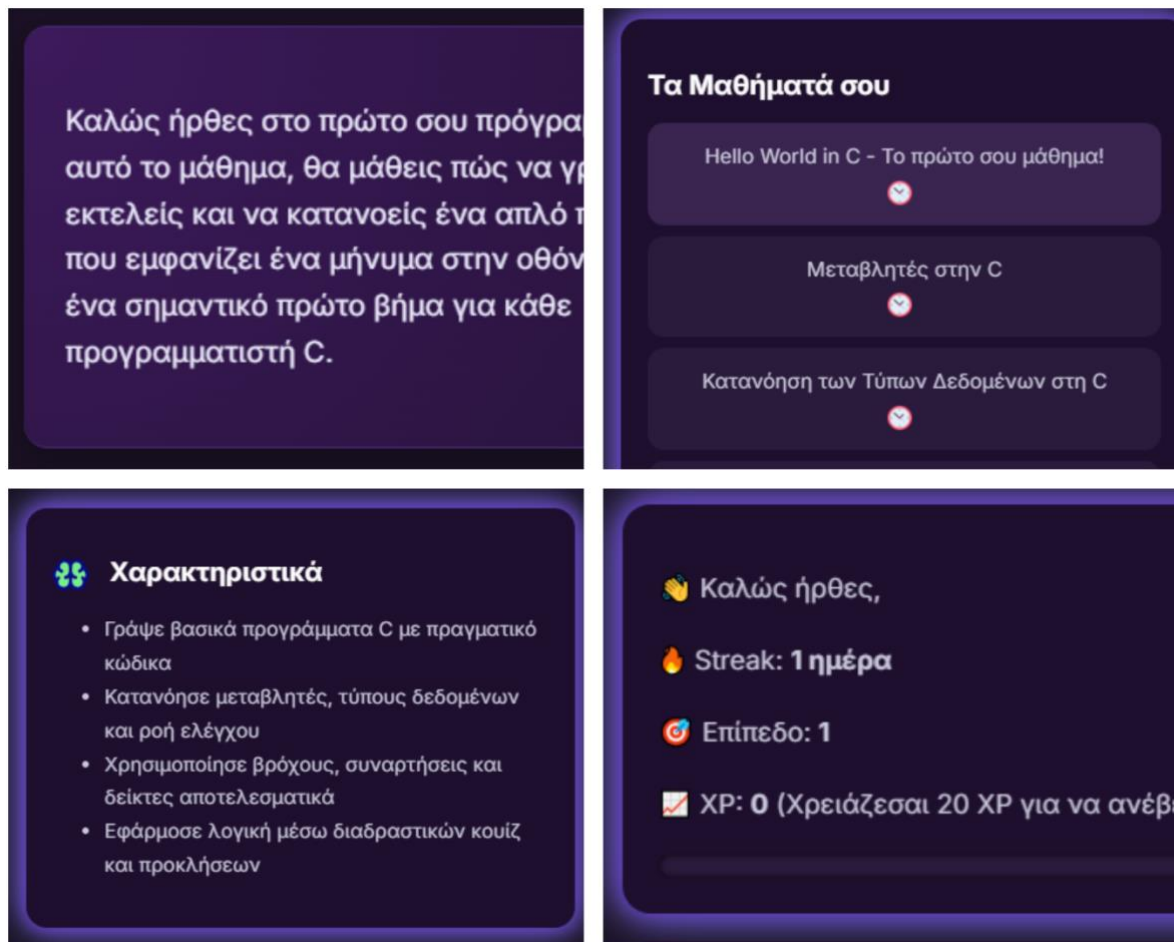
Παρότι πρόκειται για ένα απλό component, η σημασία του είναι διπλή: ολοκληρώνει οπτικά κάθε σελίδα προσδίδοντας αίσθηση πληρότητας, ενώ ταυτόχρονα προσφέρει σταθερά σημεία αναφοράς για τον χρήστη.

### 8.3.3 Card

Το Card είναι ένα επαναχρησιμοποιούμενο component, αναφέρεται σε ένα σύνολο από επαναχρησιμοποιήσιμα components που έχουν κοινή αισθητική και χρησιμοποιούνται για να παρουσιάσουν περιεχόμενο με οργανωμένο και ευανάγνωστο τρόπο. Αποτελούν βασικά δομικά στοιχεία για σελίδες όπως το Home, το Lessons, αλλά και την LessonScreen.

Οι κάρτες χρησιμοποιούνται σε διαφορετικά πλαίσια με μικρές παραλλαγές: οι InfoCard στο Home προβάλλουν τα χαρακτηριστικά της εφαρμογής, οι quizCard στη LessonScreen περιλαμβάνουν ερωτήσεις και επιλογές απαντήσεων, ενώ οι editorCard ενσωματώνουν τον κώδικα και τα αποτελέσματα εκτέλεσης. Παρά τις διαφορές στη λειτουργικότητα, όλες βασίζονται στην ίδια σχεδιαστική φιλοσοφία: καθαρή παρουσίαση, οριοθετημένο πλαίσιο και έμφαση στο περιεχόμενο.

Η χρήση των Card components προσφέρει σημαντικά πλεονεκτήματα στην ανάπτυξη: μειώνει την επανάληψη κώδικα, διασφαλίζει ομοιομορφία σε όλη την εφαρμογή και διευκολύνει τη συντήρηση. Αν χρειαστεί αλλαγή στο design, αυτή μπορεί να γίνει σε ένα κεντρικό σημείο και να εφαρμοστεί αυτόματα σε όλα τα μέρη που χρησιμοποιούν Card.

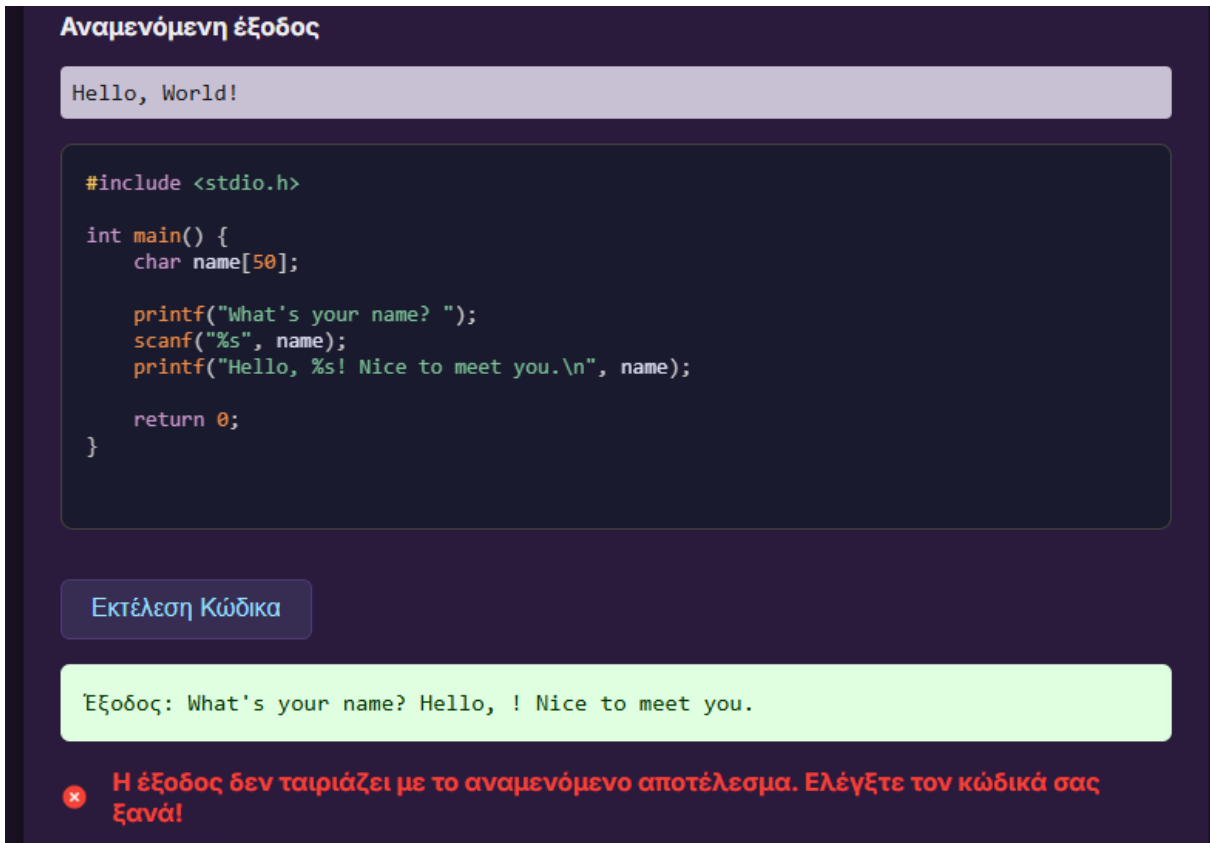


Σχήμα 27: Διάφορες παραλλαγές των Card components

Παρότι πρόκειται για σχετικά απλά components, η σημασία τους είναι καθοριστική. Χάρη σε αυτά, η εφαρμογή αποκτά συνέπεια και επαγγελματική αισθητική, ενώ ταυτόχρονα παραμένει εύκολα επεκτάσιμη. Ο διαχωρισμός μεταξύ InfoCard, quizCard και editorCard δείχνει πώς η ίδια βασική ιδέα μπορεί να προσαρμοστεί σε διαφορετικές ανάγκες, χωρίς να απαιτείται εκ νέου ανάπτυξη από το μηδέν.

### 8.3.4 Editor

Ο Editor αποτελεί ένα από τα πιο σημαντικά κοινά στοιχεία διεπαφής στο CodeIT, καθώς δίνει τη δυνατότητα στον χρήστη να γράψει, να εκτελέσει και να ελέγξει κώδικα C μέσα από το περιβάλλον της εφαρμογής, χωρίς να απαιτείται εξωτερική εγκατάσταση εργαλείων. Υλοποιείται ως React component και βασίζεται στη βιβλιοθήκη react-simple-code-editor, σε συνδυασμό με το PrismJS για την επισήμανση σύνταξης (syntax highlighting) [15] [16].



Σχήμα 28: Λανθασμένη έξοδος στον Editor

Φορτώνει με προεπιλογή είτε κενό χώρο (στη σελίδα Practice) είτε έτοιμο κώδικα/σκαλωσιά (starter code) όταν χρησιμοποιείται σε μάθημα. Η γραφή κώδικα γίνεται σε περιβάλλον με σκούρο θέμα, ευανάγνωστη γραμματοσειρά και σαφή διάκριση των συντακτικών στοιχείων (λέξεις-κλειδιά, σχόλια, μεταβλητές). Το syntax highlighting υλοποιείται με τη βιβλιοθήκη PrismJS και το theme prism-tomorrow.css.

Ο Editor περιλαμβάνει επίσης κουμπί Run, το οποίο ενεργοποιεί την εκτέλεση του κώδικα μέσω του Judge0 API [14]. Το component καλεί τη συνάρτηση runCCode(), η οποία στέλνει τον κώδικα στον απομακρυσμένο compiler και επιστρέφει είτε το αποτέλεσμα είτε μήνυμα σφάλματος. Το output εμφανίζεται άμεσα κάτω από τον editor, ώστε ο χρήστης να έχει πλήρη εικόνα της λειτουργίας του προγράμματός του.

Η σχεδίαση δίνει έμφαση στην ανατροφοδότηση: όταν το αποτέλεσμα αντιστοιχεί στο αναμενόμενο, εμφανίζεται πράσινο μήνυμα με ένδειξη επιτυχίας· σε διαφορετική περίπτωση εμφανίζεται κόκκινο μήνυμα με ένδειξη σφάλματος. Η υλοποίηση αυτή στηρίζεται σε state management με React hooks (useState) και conditional rendering, εξασφαλίζοντας ότι κάθε αλλαγή στο input αντικατοπτρίζεται άμεσα στο UI.

Από άποψη εμπειρίας χρήστη, ο Editor είναι πλήρως responsive, ώστε να λειτουργεί εξίσου καλά σε μικρές και μεγάλες οθόνες. Επιπλέον, το styling έχει απομονωθεί σε ξεχωριστό CSS module, διευκολύνοντας μελλοντικές βελτιώσεις όπως line numbering, αυτόματη εσοχή (auto-indent) ή ακόμη και υποστήριξη για πολλαπλές γλώσσες προγραμματισμού.

Σε τεχνικό επίπεδο, ο Editor αναδεικνύει την αξία των επαναχρησιμοποιήσιμων components: πρόκειται για το ίδιο στοιχείο που εμφανίζεται τόσο στη σελίδα Practice, όσο και στη LessonScreen, με μικρές μόνο παραμετροποιήσεις. Με τον τρόπο αυτό, η εφαρμογή αποφεύγει τον πλεονασμό κώδικα και παραμένει εύκολα επεκτάσιμη.

## 8.4 Επικοινωνία με Backend και Διαχείριση Δεδομένων

Η αρχιτεκτονική του κώδικα στην εφαρμογή CodeIT δίνει ιδιαίτερη έμφαση στην ομαλή και αξιόπιστη επικοινωνία με το backend. Στο σημείο αυτό δεν εξετάζουμε τη δομή της βάσης δεδομένων – η οποία παρουσιάστηκε αναλυτικά στο Κεφάλαιο 7 – αλλά τον τρόπο με τον οποίο τα React components αλληλεπιδρούν με τα εξωτερικά συστήματα. Η εφαρμογή βασίζεται κυρίως σε δύο υπηρεσίες: το Supabase, που παρέχει αυθεντικοποίηση και αποθήκευση δεδομένων, και το Judge0 API, που αναλαμβάνει τη μεταγλώττιση και εκτέλεση κώδικα C σε πραγματικό χρόνο.

Η επικοινωνία με το backend υλοποιείται μέσω ασύγχρονων κλήσεων (async/await), οι οποίες ενσωματώνονται στη ροή των React components μέσω hooks όπως useEffect και useState. Με αυτόν τον τρόπο, τα δεδομένα χρήστη (π.χ. πρόοδος, στατιστικά) και τα δεδομένα μαθημάτων (π.χ. περιεχόμενο, quiz, tasks) αντλούνται δυναμικά από το Supabase, ενώ η εκτέλεση κώδικα προωθείται στο Judge0 API και τα αποτελέσματα επιστρέφουν άμεσα στον χρήστη.

Για να διατηρείται καθαρός ο κώδικας και να αποφεύγεται η άμεση «ανάμειξη» UI και λογικής, η εφαρμογή ακολουθεί αρχή διαχωρισμού ευθυνών (separation of concerns). Έτσι, η επικοινωνία με το backend είναι συγκεντρωμένη σε αρχεία υπηρεσιών (lessonService.js, progressService.js, statsService.js), τα οποία εκτελούν τα απαραίτητα queries και επιστρέφουν τα δεδομένα στα components. Με αυτόν τον τρόπο, η εφαρμογή παραμένει επεκτάσιμη, πιο ευανάγνωστη και εύκολη στη συντήρηση.

### 8.4.1 Επικοινωνία με Supabase

Το Supabase αποτελεί το βασικό backend της εφαρμογής CodeIT, καλύπτοντας δύο κρίσιμους άξονες: την αυθεντικοποίηση των χρηστών και την αποθήκευση δεδομένων. Η επικοινωνία υλοποιείται μέσω της client βιβλιοθήκης που παρέχει το Supabase, γεγονός που δίνει τη δυνατότητα στα React components να αλληλεπιδρούν απευθείας με τη βάση δεδομένων, χωρίς την ανάγκη για επιπλέον server ενδιάμεσης επεξεργασίας [19].

Στο κομμάτι της αυθεντικοποίησης, η εφαρμογή υποστηρίζει πολλαπλά σενάρια: εγγραφή και σύνδεση με email και κωδικό, είσοδο μέσω OAuth παρόχων (Google και GitHub), αλλά και ανώνυμες συνεδρίες για χρήστες που θέλουν να δοκιμάσουν το περιβάλλον χωρίς να δημιουργήσουν λογαριασμό. Η ευελιξία αυτή εξασφαλίζει ευκολία πρόσβασης και χαμηλό εμπόδιο εισόδου για νέους χρήστες.

Αναφορικά με τα δεδομένα μαθημάτων και προόδου, το Supabase διαχειρίζεται τόσο την ανάκτηση όσο και την ενημέρωσή τους. Τα στοιχεία των μαθημάτων (τίτλοι, περιεχόμενο, ερωτήσεις) αποθηκεύονται στη βάση και ανακτώνται δυναμικά ανάλογα με τη γλώσσα που έχει επιλέξει ο χρήστης. Παράλληλα, η πρόοδος του κάθε μαθητή (π.χ. ποια μαθήματα έχει ολοκληρώσει, ποια βρίσκονται σε εξέλιξη) ενημερώνεται σε πραγματικό χρόνο, ώστε το Dashboard να εμφανίζει πάντα τα πιο πρόσφατα στατιστικά.

```

import { supabase } from "../supabaseClient";

async function fetchLessons() {
  const { data, error } = await supabase
    .from("lessons")
    .select("id, title_en, title_gr");

  if (error) {
    console.error("Error fetching lessons:", error);
    return [];
  }

  return data;
}

```

Σχήμα 29: Παράδειγμα κώδικα που δείχνει την ανάκτηση δεδομένων μαθημάτων από το Supabase

Η εφαρμογή ακολουθεί αρχή διαχωρισμού ευθυνών: οι κλήσεις προς το Supabase συγκεντρώνονται σε ξεχωριστά αρχεία υπηρεσιών, τα οποία αναλαμβάνουν όλη τη λογική επικοινωνίας με το backend. Με αυτόν τον τρόπο, τα React components επικεντρώνονται αποκλειστικά στην παρουσίαση, ενώ η αλληλεπίδραση με τη βάση παραμένει απομονωμένη και ευκολότερη στη συντήρηση.

Τέλος, η δυναμική ενσωμάτωση των δεδομένων στο περιβάλλον χρήστη πραγματοποιείται μέσω React hooks. Για παράδειγμα, κατά την πλοήγηση στο Dashboard, αρχικά εμφανίζεται ένας οπτικός δείκτης φόρτωσης, ενώ στη συνέχεια παρουσιάζονται οι πληροφορίες προόδου και στατιστικών που έχουν αντληθεί από το Supabase. Η προσέγγιση αυτή συνδυάζει καλή εμπειρία χρήστη με ασφάλεια και ακεραιότητα δεδομένων.

## 8.4.2 Ενσωμάτωση Judge0 API

Η δεύτερη βασική υπηρεσία που υποστηρίζει το backend της εφαρμογής είναι το Judge0 API, το οποίο αναλαμβάνει την εκτέλεση κώδικα C σε πραγματικό χρόνο. Το Judge0 λειτουργεί ως online compiler: λαμβάνει τον κώδικα που πληκτρολογεί ο χρήστης μέσα στον ενσωματωμένο editor, τον μεταγλωττίζει σε απομακρυσμένο περιβάλλον και επιστρέφει το αποτέλεσμα της εκτέλεσης, είτε αυτό είναι η επιθυμητή έξοδος είτε μηνύματα σφάλματος.

Η επικοινωνία με το Judge0 API γίνεται μέσω HTTP αιτημάτων, τα οποία στέλνουν το πρόγραμμα προς εκτέλεση και περιμένουν την απάντηση. Το μοντέλο αυτό επιτρέπει την άμεση αξιολόγηση της ορθότητας του κώδικα, καθώς η εφαρμογή συγκρίνει την έξοδο που επιστρέφεται με την αναμενόμενη απάντηση κάθε άσκησης. Αν οι δύο τιμές ταιριάζουν, ο χρήστης προχωρά στο επόμενο βήμα· σε διαφορετική περίπτωση, ενημερώνεται με σαφή μηνύματα σφάλματος και μπορεί να διορθώσει τον κώδικά του.

Η ενσωμάτωση του Judge0 ενισχύει καθοριστικά τη μαθησιακή εμπειρία, καθώς προσφέρει στον χρήστη άμεση ανατροφοδότηση. Δεν απαιτείται τοπική εγκατάσταση compiler, ενώ η εκτέλεση γίνεται με ασφαλή και ελεγχόμενο τρόπο στο cloud. Έτσι, ακόμη και αρχάριοι χρήστες μπορούν να πειραματιστούν με κώδικα C χωρίς τεχνικές δυσκολίες στη ρύθμιση του περιβάλλοντος.

Σε επίπεδο υλοποίησης, η διαδικασία είναι πλήρως ενοποιημένη με το περιβάλλον React: η αποστολή του κώδικα, η λήψη της απόκρισης και η εμφάνιση των αποτελεσμάτων γίνονται μέσα από τα components, με χρήση React hooks για την ενημέρωση της διεπαφής. Ο χρήστης βλέπει σε πραγματικό χρόνο τόσο την επιτυχή έξοδο όσο και τυχόν μηνύματα σφαλμάτων μεταγλώττισης ή εκτέλεσης.

Η χρήση του Judge0 API αποτελεί θεμελιώδη καινοτομία για το CodeIT, αφού επιτρέπει την πρακτική εξάσκηση παράλληλα με τη θεωρητική εκμάθηση. Με τον τρόπο αυτό, η εφαρμογή ξεπερνά τα παραδοσιακά quiz και προσφέρει μια ολοκληρωμένη εμπειρία μάθησης, όπου η θεωρία συνδέεται άμεσα με την πράξη.

### 8.4.3 Σύνδεση Frontend και Backend στη ροή της μάθησης

Η εκπαιδευτική εμπειρία στο CodeIT βασίζεται στην αρμονική συνεργασία frontend και backend. Το περιβάλλον React αναλαμβάνει την παρουσίαση του περιεχομένου και τη διαδραστική πλοήγηση, ενώ οι υπηρεσίες Supabase και Judge0 παρέχουν τα απαραίτητα δεδομένα και τη δυνατότητα εκτέλεσης κώδικα.

Κατά την εκκίνηση ενός μαθήματος, το frontend αντλεί από το Supabase το θεωρητικό υλικό, τις ερωτήσεις και τυχόν tasks. Καθώς ο χρήστης απαντά ή εκτελεί κώδικα, οι ενέργειες αυτές διαβιβάζονται στο backend: οι απαντήσεις αποθηκεύονται στη βάση δεδομένων, η πρόοδος καταγράφεται και τα στατιστικά ενημερώνονται. Με τον τρόπο αυτό, το Dashboard παραμένει πάντα ενημερωμένο με το επίπεδο, τα XP και τη συνολική εικόνα του μαθητή.

Όταν ο χρήστης καλείται να υλοποιήσει μια άσκηση κώδικα, το frontend αποστέλλει το πρόγραμμα στο Judge0 API, το οποίο το μεταγλωττίζει και επιστρέφει το αποτέλεσμα. Στη συνέχεια, η εφαρμογή συγκρίνει αυτό το αποτέλεσμα με την αναμενόμενη έξοδο, ώστε να αποφασίσει αν η απάντηση είναι σωστή. Η διαδικασία αυτή ολοκληρώνεται σε δευτερόλεπτα, προσφέροντας στον μαθητή άμεση ανατροφοδότηση.

Η αλληλεπίδραση των τριών επιπέδων – UI, Supabase, Judge0 – είναι πλήρως ενοποιημένη. Οι χρήστες δεν αντιλαμβάνονται την τεχνική πολυπλοκότητα, αλλά απολαμβάνουν μια συνεχή και ρέουσα εμπειρία: παρακολουθούν το περιεχόμενο, απαντούν σε ερωτήσεις, γράφουν κώδικα και βλέπουν άμεσα τα αποτελέσματα, ενώ παράλληλα η πρόοδος τους αποθηκεύεται με ασφάλεια.

Η συγκεκριμένη αρχιτεκτονική καθιστά την εφαρμογή stateful στο επίπεδο του χρήστη (με συνεχή ενημέρωση της κατάστασης) και ταυτόχρονα stateless στο επίπεδο των υπηρεσιών (καθώς κάθε αίτημα προς το Supabase ή το Judge0 είναι αυτόνομο). Ο συνδυασμός αυτός εξασφαλίζει απόδοση, επεκτασιμότητα και ανθεκτικότητα, προσφέροντας μια ρεαλιστική εμπειρία μάθησης προγραμματισμού που συνδέει θεωρία και πράξη.

## 8.5 Πολυγλωσσική Υποστήριξη

Η υποστήριξη πολλαπλών γλωσσών αποτελεί βασικό στοιχείο της εφαρμογής CodeIT, καθώς στοχεύει σε διεθνές κοινό αλλά και σε ελληνόφωνους χρήστες. Η υλοποίηση δεν περιορίζεται σε μια απλή αλλαγή ετικετών στο UI, αλλά καλύπτει τόσο το περιβάλλον της εφαρμογής όσο και το περιεχόμενο των μαθημάτων.

### 8.5.1 Ενσωμάτωση react-i18next

Για τη διαχείριση μεταφράσεων χρησιμοποιήθηκε η βιβλιοθήκη react-i18next, που είναι ειδικά προσαρμοσμένη για React εφαρμογές. Η βιβλιοθήκη προσφέρει hooks όπως το useTranslation(), τα οποία επιτρέπουν στα components να «τραβούν» το σωστό string ανάλογα με την τρέχουσα γλώσσα.

Κάθε στοιχείο κειμένου ορίζεται σε ξεχωριστό translation αρχείο (π.χ. en.json, gr.json). Εκεί περιέχονται όλα τα UI strings οργανωμένα σε κατηγορίες, όπως header, home, dashboard, lesson. Έτσι, το component δεν εμφανίζει hard-coded κείμενα αλλά κλειδιά μετάφρασης, π.χ.:

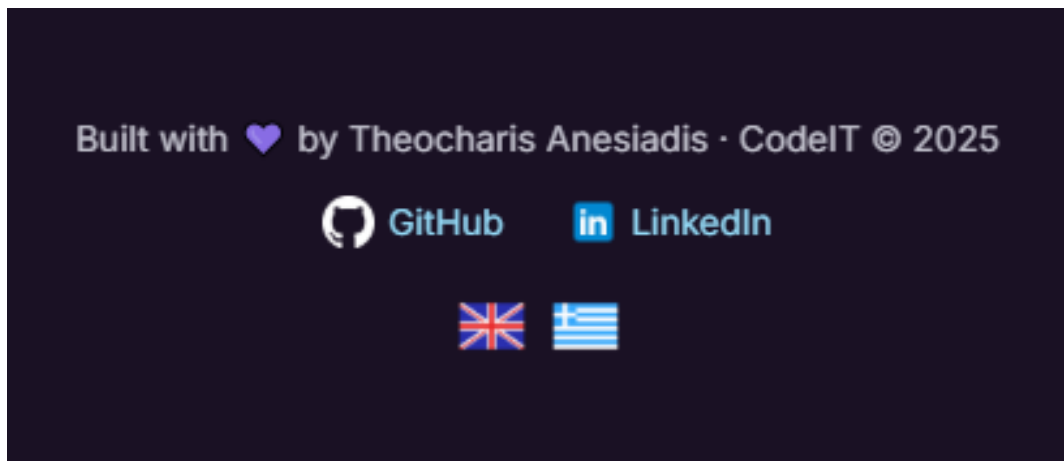
- t("header.lessons") → εμφανίζει “Lessons” ή “Μαθήματα”
- t("lesson.completed") → εμφανίζει “Lesson completed!” ή “Μάθημα ολοκληρώθηκε!”

Με αυτόν τον τρόπο, ο κώδικας παραμένει καθαρός και ανεξάρτητος από τη γλώσσα.

## 8.5.2 Δυναμική αλλαγή γλώσσας

Η αλλαγή γλώσσας γίνεται μέσω διακόπτη (language switcher) στο Footer, με δύο κουμπιά (σημαία Αγγλίας και Ελλάδας). Όταν ο χρήστης επιλέξει γλώσσα, καλείται η συνάρτηση `i18n.changeLanguage lng`, η οποία ενημερώνει κεντρικά τη γλώσσα της εφαρμογής.

Η επιλογή αποθηκεύεται στο `localStorage`, ώστε η εφαρμογή να θυμάται την προτίμηση του χρήστη ακόμη και μετά από επανεκκίνηση ή νέα σύνδεση. Αυτό εξασφαλίζει συνέπεια και καλύτερη εμπειρία χρήσης.



Σχήμα 30: Αλλαγή γλώσσας μέσω εικονιδίων στο Footer

## 8.5.3 Πολυγλωσσικό περιεχόμενο μαθημάτων

Το περιεχόμενο των μαθημάτων (τίτλοι, κείμενα, βήματα) δεν «ψήνεται» στα translation files· αντ' αυτού φυλάσσεται στη ΒΔ σε πολλαπλές γλωσσικές στήλες. Τα components αναζητούν την κατάλληλη στήλη με βάση το `i18n.language` (π.χ. `title_en` ή `title_gr`) και συνθέτουν δυναμικά το τελικό αντικείμενο που αποδίδεται στο UI.

- Στη σελίδα Lessons, το ερώτημα προς τα δεδομένα και ο χάρτης ονομάτων φροντίζουν ώστε η λίστα να εμφανίζει τον σωστό τίτλο για κάθε μάθημα ανά γλώσσα.
- Στο Dashboard, ο ίδιος μηχανισμός εφαρμόζεται όταν «φορτώνονται» όλα τα lessons για να παρουσιαστεί πρόδος/επόμενο μάθημα με σωστούς τίτλους.

Η επιλογή αυτής της προσέγγισης (data-driven i18n για περιεχόμενο) επιτρέπει τη διαχείριση/επιμέλεια μαθημάτων απευθείας από τη ΒΔ και διατηρεί τα translation files μικρά και «καθαρά» για στοιχεία UI.

## 8.5.4 Σύνδεση με Components

Η διεπαφή της εφαρμογής (Header, Footer, Dashboard, LessonScreen) συνδέεται απευθείας με το `i18next`. Για παράδειγμα:

- Στο Header, οι τίτλοι των συνδέσμων (Lessons, Practice, Dashboard, Login/Logout) εμφανίζονται δυναμικά μέσω μεταφράσεων.
- Στο Dashboard, όλα τα στατιστικά (XP, Level, Streak) περιγράφονται στη γλώσσα που έχει επιλεγεί.
- Στο LessonScreen, οι ερωτήσεις, τα hints και τα μηνύματα σωστής/λάθους απάντησης εμφανίζονται αμέσως στην κατάλληλη γλώσσα.

Έτσι, η αλλαγή γλώσσας δεν επηρεάζει μόνο το UI αλλά και το ίδιο το εκπαιδευτικό περιεχόμενο, προσφέροντας μια πραγματικά δίγλωσση εμπειρία.

## 8.6 Υλοποίηση UI/UX

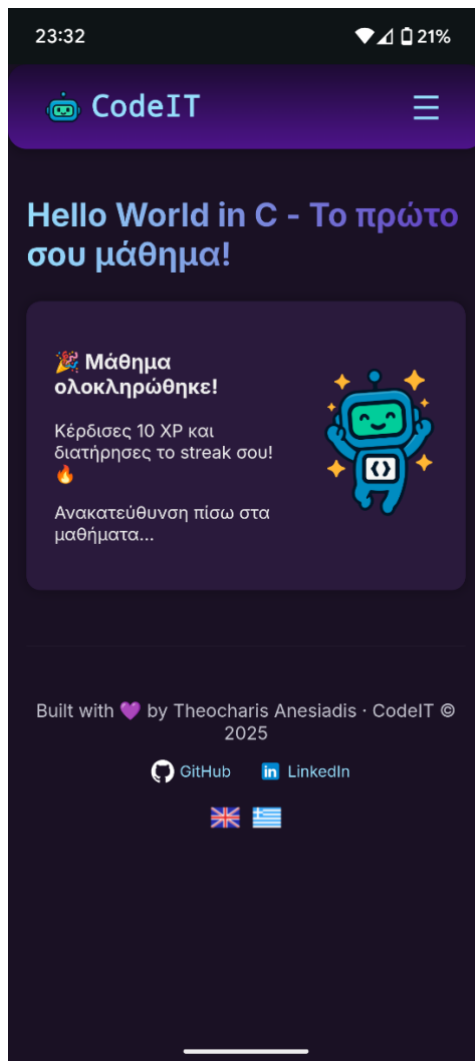
Η υλοποίηση του γραφικού περιβάλλοντος στο CodeIT βασίζεται σε CSS Modules, με το αρχείο `globals.module.css` να αποτελεί το κεντρικό σημείο ορισμού των στυλ. Η χρήση CSS Modules εξασφαλίζει απομόνωση ονομάτων κλάσεων, αποτρέποντας πιθανές συγκρούσεις και διατηρώντας τον κώδικα πιο καθαρό και οργανωμένο.

Ένα σημαντικό χαρακτηριστικό της υλοποίησης είναι η εκτεταμένη χρήση `animations`, τα οποία δίνουν ζωντάνια στη διεπαφή και καθιστούν την εμπειρία πιο ευχάριστη για τον χρήστη. Παραδείγματα περιλαμβάνουν:

- `fadeSlideIn` και `fadeUp`: χρησιμοποιούνται για την ομαλή εμφάνιση τίτλων, υπότιτλων και καρτών.
- `shake`: εφαρμόζεται σε λάθος απαντήσεις στα quiz, παρέχοντας άμεση οπτική ανατροφοδότηση.
- `pulseSuccess`: δίνει εφέ παλμού σε σωστές απαντήσεις, ενισχύοντας την αίσθηση επιτυχίας.

Η ανταπόκριση σε διαφορετικές συσκευές (responsiveness) υλοποιείται με `media queries`, που προσαρμόζουν μεγέθη τίτλων, `padding`, αλλά και τη διάταξη καρτών για tablet και κινητές συσκευές. Έτσι διασφαλίζεται ότι η εμπειρία παραμένει ομαλή ανεξάρτητα από την ανάλυση οθόνης.

Ιδιαίτερη σημασία έχει η χρήση εικονιδίων και μασκότ. Οι εικόνες της μασκότ Cody (π.χ. `cody_hello`, `codyCelebrate`) ενσωματώνονται σε διάφορες σελίδες (`Lessons.jsx`, `LessonScreen.jsx`) όχι μόνο ως διακοσμητικά στοιχεία αλλά και ως μέσα αλληλεπίδρασης: εμφανίζονται σε σημεία-κλειδιά, όπως κατά την ολοκλήρωση ενός μαθήματος, για να υπογραμμίσουν τη σημαντικότητα του επιτεύγματος.



Σχήμα 31: Στιγμιότυπο οθόνης στην ολοκλήρωση του μαθήματος

Τέλος, η υλοποίηση των components διεπαφής, όπως το Header, το Footer και οι Card, είναι στενά δεμένη με το στυλ του `globals.module.css`, ώστε να διατηρείται ενιαία αισθητική σε όλη την εφαρμογή. Η επαναχρησιμοποίηση κοινών κλάσεων (`buttonPrimary`, `cardStatic`, `progressBar`) μειώνει την πολυπλοκότητα και ενισχύει τη συνέπεια του UI.

## 8.7 Υλοποίηση Στοιχείων Gamification

Η υλοποίηση της παιχνιδοποίησης στο CodeIT βασίζεται στη συνεργασία πολλών αρχείων και services που αλληλεπιδρούν με το backend του Supabase.

Στο αρχείο `Dashboard.jsx` γίνεται η ανάκτηση των βασικών στατιστικών μέσω της συνάρτησης `getUserStats`. Τα δεδομένα προέρχονται από τον πίνακα της βάσης που αποθηκεύει το συνολικό XP, το streak και τα ολοκληρωμένα μαθήματα. Στη συνέχεια, στο frontend υπολογίζονται δυναμικά το επίπεδο του χρήστη και το ποσοστό προόδου, τα οποία αποτυπώνονται σε progress bar με χρήση κλάσεων από το `globals.module.css`.

Στο `LessonScreen.jsx` ενσωματώνονται οι κλήσεις προς τα services `completeLesson` και `awardXP`. Με κάθε επιτυχή ολοκλήρωση βήματος, καλείται το service που ενημερώνει τον πίνακα progress και προσθέτει XP στον χρήστη. Ο συγχρονισμός με το backend γίνεται σε πραγματικό χρόνο μέσω Supabase και το UI ανανεώνεται αυτόματα χάρη στη χρήση state hooks της React.

Η ανατροφοδότηση στο χρήστη υλοποιείται μέσω κατάστασης (state) και CSS animations. Για παράδειγμα, η μεταβλητή `isCorrect` ενεργοποιεί κλάσεις όπως `pulseSuccess` ή `shake` από το `globals.module.css`. Έτσι, κάθε απάντηση αποτυπώνεται άμεσα με πράσινα ή κόκκινα μηνύματα, οπτικά εφέ και συνοδευτικά εικονίδια.

Η ολοκλήρωση ενός μαθήματος ενεργοποιεί δύο διαδικασίες:

- Ενημερώνεται ο πίνακας `progress` στη βάση ώστε να καταγραφεί η πρόοδος.
- Στην πλευρά του frontend, εμφανίζεται το animation `confetti` και η μασκότ να πανηγυρίζει, ενισχύοντας το αίσθημα επιβράβευσης.

Η υλοποίηση των streaks βασίζεται σε ελέγχους ημερομηνιών στο backend: κάθε φορά που ο χρήστης ολοκληρώνει ένα μάθημα, καταγράφεται η ημερομηνία και το streak αυξάνεται αν υπάρχει συνεχόμενη δραστηριότητα. Το αποτέλεσμα προβάλλεται στο Dashboard με το εικονίδιο φωτιάς.

Συνολικά, τα στοιχεία παιχνιδοποίησης υλοποιούνται με συνδυασμό React components, CSS modules και Supabase services. Η React διαχειρίζεται την άμεση ανατροφοδότηση, το Supabase αποθηκεύει και επαληθεύει την πρόοδο και η διεπαφή αποδίδει τα αποτελέσματα με animations και εικονίδια. Έτσι η τεχνική υλοποίηση υποστηρίζει απευθείας τον παιδαγωγικό στόχο της συνεχούς παρακίνησης.

## Κεφάλαιο 9ο: Συμπέρασμα

Εν κατακλείδι, η παρούσα πτυχιακή εργασία είχε ως αντικείμενο τη δημιουργία του CodeIT, μιας καινοτόμου διαδικτυακής πλατφόρμας εκμάθησης της γλώσσας C, η οποία συνδυάζει θεωρητική γνώση, πρακτική εξάσκηση και στοιχεία παιχνιδοποίησης. Στόχος ήταν να αποδειχθεί ότι ο προγραμματισμός μπορεί να διδαχθεί με τρόπο διαδραστικό, ευχάριστο και προσβάσιμο, ακόμη και σε άτομα χωρίς προηγούμενη εμπειρία.

Μέσα από την ανάπτυξη της εφαρμογής αναδείχθηκε η αξία του συνδυασμού παιδαγωγικής και τεχνολογίας. Η δομή των μαθημάτων, με εισαγωγικό κείμενο, παραδείγματα, quiz και ασκήσεις κώδικα, προσφέρει μια ολοκληρωμένη μαθησιακή εμπειρία, όπου η θεωρία συνδέεται άμεσα με την πράξη. Ο ενσωματωμένος compiler επιτρέπει στους χρήστες να γράφουν και να εκτελούν κώδικα σε πραγματικό χρόνο, αφαιρώντας τεχνικά εμπόδια και καθιστώντας την εκμάθηση άμεση και βιωματική.

Από τεχνικής πλευράς, η πλατφόρμα αξιοποίησε σύγχρονα εργαλεία όπως το React για τη δημιουργία μιας ευέλικτης και δυναμικής διεπαφής, το Supabase για ασφαλή αποθήκευση δεδομένων και διαχείριση χρηστών, καθώς και το Judge0 API για την εκτέλεση κώδικα. Η ενσωμάτωση πολυγλωσσικής υποστήριξης μέσω του react-i18next πρόσθεσε ένα επιπλέον επίπεδο προσιτότητας, επιτρέποντας σε περισσότερους χρήστες να αξιοποιήσουν την εφαρμογή στη γλώσσα της επιλογής τους.

Η παιχνιδοποίηση αποτέλεσε καθοριστικό στοιχείο της υλοποίησης. Συστήματα XP, επιπέδων, streaks και οπτικών επιβραβεύσεων δημιουργούν κίνητρα για συνεχή χρήση, ενισχύουν τη δέσμευση και κάνουν τη μάθηση πιο ελκυστική. Η σύνδεση του backend με το frontend εξασφαλίζει ότι η πρόοδος αποθηκεύεται με ακρίβεια, ενώ τα animations και τα οπτικά εφέ προσφέρουν μια εμπειρία που μοιάζει περισσότερο με παιχνίδι παρά με παραδοσιακό εκπαιδευτικό εργαλείο.

Το CodeIT δείχνει ότι η μάθηση προγραμματισμού δεν χρειάζεται να περιορίζεται σε θεωρητικά εγχειρίδια ή σε δύσχρηστα περιβάλλοντα ανάπτυξης. Αντίθετα, μπορεί να γίνει μια προσβάσιμη, διασκεδαστική και ταυτόχρονα αποτελεσματική διαδικασία. Η δυνατότητα χρήσης χωρίς εγκατάσταση, η καθαρή και φιλική διεπαφή, καθώς και η άμεση παροχή ανατροφοδότησης καθιστούν την πλατφόρμα προσιτή σε ευρύ φάσμα χρηστών, από μαθητές και φοιτητές έως ενήλικες που θέλουν να κάνουν τα πρώτα τους βήματα στον προγραμματισμό.

Συνολικά, το έργο αυτό αποδεικνύει τις δυνατότητες της σύγχρονης τεχνολογίας να στηρίζει τη μαθησιακή διαδικασία, αξιοποιώντας πρακτικές που συνδυάζουν την εκπαίδευση με την ψυχαγωγία. Το CodeIT δεν είναι μόνο μια εφαρμογή για την εκμάθηση της C, αλλά ένα παράδειγμα για το πώς η τεχνολογία μπορεί να μετατρέψει την εκπαίδευση σε μια πιο αλληλεπιδραστική και παρακινητική εμπειρία.

### 9.1 Μελλοντικές προσθήκες

Παρόλο που το CodeIT αποτελεί ήδη μια λειτουργική και ολοκληρωμένη πλατφόρμα για την εκμάθηση της C, υπάρχουν πολλοί δρόμοι για περαιτέρω εξέλιξη που μπορούν να εμπλουτίσουν σημαντικά την εμπειρία του χρήστη και να επεκτείνουν τις δυνατότητες του συστήματος.

Μία από τις πιο προφανείς κατευθύνσεις είναι η επέκταση των μαθημάτων και η εισαγωγή νέων ασκήσεων. Η γλώσσα C διαθέτει πλούσιο οικοσύστημα εννοιών και τεχνικών που δεν έχουν ακόμη καλυφθεί πλήρως. Η δημιουργία νέων εννοιών για πιο προχωρημένα θέματα, όπως η διαχείριση αρχείων, οι δομές δεδομένων ή ο αντικειμενοστραφής προγραμματισμός μέσω C++, θα μπορούσε να δώσει στους χρήστες τη δυνατότητα να συνεχίσουν την πορεία τους χωρίς να χρειαστεί να αναζητήσουν άλλο εργαλείο.

Παράλληλα, η προσθήκη επιπλέον γλωσσών προγραμματισμού θα ενίσχυε τον ορίζοντα του CodeIT. Η υποστήριξη για γλώσσες όπως η Python, η Java ή η JavaScript θα καθιστούσε την πλατφόρμα πιο ευέλικτη, επιτρέποντας σε αρχάριους αλλά και πιο προχωρημένους χρήστες να επιλέξουν τη γλώσσα που τους ενδιαφέρει περισσότερο, πάντα μέσα στο ίδιο γνώριμο και φιλικό περιβάλλον.

Ένα ακόμη βήμα θα μπορούσε να είναι η εισαγωγή διαφορετικών σταδίων δυσκολίας, όπως αρχάριος, εξοικειωμένος ή προχωρημένος. Με τον τρόπο αυτό, το CodeIT θα προσαρμοζόταν καλύτερα στις ανάγκες του κάθε χρήστη, προσφέροντας μια εξατομικευμένη μαθησιακή εμπειρία. Ένας αρχάριος θα μπορούσε να κινηθεί με αργούς και απλούς ρυθμούς, ενώ ένας πιο έμπειρος θα είχε τη δυνατότητα να περάσει γρήγορα σε πιο απαιτητικά θέματα.

Η κοινωνική διάσταση είναι ένας ακόμη τομέας με μεγάλη δυναμική. Η ενίσχυση με leaderboards, συνεργατικές προκλήσεις ή η δυνατότητα ανταγωνισμού μεταξύ χρηστών θα μετέτρεπε τη μάθηση σε μια πιο διαδραστική και συλλογική εμπειρία. Η αίσθηση ανταγωνισμού και συνεργασίας μπορεί να αυξήσει σημαντικά τα κίνητρα, κάνοντας τους χρήστες να επιστρέφουν συχνότερα στην πλατφόρμα.

Επιπλέον, αξίζει να εξεταστεί η ανάπτυξη μιας mobile εφαρμογής για Android και iOS, που θα προσέφερε στον χρήστη την ίδια εμπειρία μάθησης από οποιαδήποτε συσκευή, με δυνατότητα offline πρόσβασης. Με αυτόν τον τρόπο, η πλατφόρμα θα γινόταν ακόμη πιο προσιτή, καλύπτοντας τις ανάγκες μιας γενιάς που βασίζεται έντονα στη χρήση κινητών συσκευών.

Μία ακόμη προοπτική είναι η βελτίωση του Admin panel, ώστε να δίνει περισσότερες δυνατότητες στους διαχειριστές για τη δημιουργία και την παρακολούθηση μαθημάτων. Η ενσωμάτωση στατιστικών αναφορών, γραφημάτων προόδου και αυτοματοποιημένων εργαλείων αξιολόγησης θα καθιστούσε τη διαχείριση πιο αποτελεσματική και θα βοηθούσε στην καλύτερη οργάνωση του περιεχομένου.

Τέλος, η αξιοποίηση τεχνητής νοημοσύνης θα μπορούσε να προσθέσει μια νέα διάσταση στο CodeIT. Η χρήση έξυπνων αλγορίθμων για την ανάλυση των απαντήσεων, η παροχή εξατομικευμένης ανατροφοδότησης ή ακόμη και η αυτόματη πρόταση ασκήσεων ανάλογα με τις ανάγκες του χρήστη, θα αναβάθμιζε αισθητά τη μαθησιακή εμπειρία.

Συνολικά, οι μελλοντικές προσθήκες δείχνουν ότι το CodeIT έχει τη δυνατότητα να ξελιχθεί από μια εφαρμογή εκμάθησης της C σε μια πολυδιάστατη εκπαιδευτική πλατφόρμα, που ενώνει τη μάθηση με την ψυχαγωγία και καλύπτει τις ανάγκες τόσο αρχάριων όσο και προχωρημένων χρηστών.

## 9.2 Συνολική αποτίμηση

Η ανάπτυξη του CodeIT απέδειξε ότι η εκμάθηση προγραμματισμού μπορεί να συνδυάσει τη σοβαρότητα της γνώσης με τη ζωντάνια του παιχνιδιού. Μέσα από τον συνδυασμό θεωρίας, πρακτικής και παιχνιδιοποίησης, δημιουργήθηκε ένα περιβάλλον που παρακινεί τον χρήστη να συμμετέχει ενεργά και να επανέρχεται συστηματικά στη διαδικασία μάθησης. Η χρήση σύγχρονων τεχνολογιών, όπως το React, το Supabase και το Judge0 API, διασφάλισε την τεχνική αρτιότητα και την αξιόπιστη λειτουργία της πλατφόρμας, ενώ η πολυγλωσσική υποστήριξη άνοιξε τον δρόμο για την αξιοποίησή της από ένα ευρύτερο κοινό.

Σε παιδαγωγικό επίπεδο, το έργο κατέδειξε ότι η θεωρία γίνεται ουσιαστική όταν συνδέεται άμεσα με την πράξη, και ότι η παρακίνηση ενισχύεται σημαντικά όταν συνοδεύεται από στοιχεία παιχνιδιού. Σε τεχνικό επίπεδο, η εργασία ανέδειξε πώς σύγχρονα εργαλεία ανοιχτού κώδικα μπορούν να αξιοποιηθούν αποτελεσματικά για την υλοποίηση ενός ολοκληρωμένου, διαδραστικού και επεκτάσιμου συστήματος.

Παράλληλα, η διαδικασία ανάπτυξης του CodeIT αποτέλεσε για τον δημιουργό μια πολύτιμη εμπειρία μάθησης. Η ενασχόληση με τεχνολογίες που απαιτούν τόσο καλή κατανόηση του frontend όσο και σύνδεση με το backend, η υλοποίηση πολυγλωσσικής υποστήριξης, η οργάνωση της βάσης δεδομένων και η βελτιστοποίηση της διεπαφής αποτέλεσαν προκλήσεις που συνέβαλαν στην απόκτηση βαθύτερων τεχνικών δεξιοτήτων. Επιπλέον, η εστίαση στην παιδαγωγική διάσταση του έργου ανέδειξε την αξία της απλότητας και της σαφήνειας στη διδασκαλία του προγραμματισμού, καθώς και τη σημασία της παρακίνησης στη διατήρηση της προσοχής του μαθητή.

Το CodeIT απέδειξε ότι η συνδυαστική δύναμη της παιγνιοποίησης, της αλληλεπίδρασης και των σύγχρονων τεχνολογιών μπορεί να καταστήσει την εκμάθηση του προγραμματισμού πιο προσιτή και ελκυστική για όλους.

## Βιβλιογραφία

- [1] B. W. Kernighan και D. M. Ritchie, *The C Programming Language*, 2 επιμ., 1988.
- [2] S. Deterding, D. Dixon, R. Khaled και L. Nacke, *From Game Design Elements to Gamefulness: Defining ‘Gamification’*, 2011.
- [3] J. Hamari, J. Koivisto και H. Sarsa, «Does Gamification Work?—A Literature Review of Empirical Studies on Gamification,» 2014.
- [4] M. Sailer και L. Homner, «The Gamification of Learning: A Meta-Analysis,» 2020.
- [5] E. L. Deci και R. M. Ryan, «The "What" and "Why" of Goal Pursuits: Human Needs and the Self-Determination of Behavior,» 2000.
- [6] R. M. Ryan και E. L. Deci, «Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being,» 2000.
- [7] X. Jiang και B. Pajak, «Reading and Listening Outcomes of Learners in the Duolingo English Course for Spanish Speakers,» Duolingo, 2022.
- [8] X. Jiang, «The Effectiveness of Duolingo English Courses in Developing Reading and Listening Proficiency,» Duolingo, 2024.
- [9] «Mimo — Learn to Code,» 2025. [Ηλεκτρονικό]. Available: <https://mimo.org/>.
- [10] «Sololearn: Learn to Code,» [Ηλεκτρονικό]. Available: <https://www.sololearn.com>.
- [11] «Knackbout | Game based learning platform,» [Ηλεκτρονικό]. Available: <https://knackbout.com/>.
- [12] «LeetCode - The World's Leading Online Programming Learning Platform,» [Ηλεκτρονικό]. Available: <https://leetcode.com/>.
- [13] «React — The library for web and native user interfaces,» 2025. [Ηλεκτρονικό]. Available: <https://react.dev>. [Πρόσβαση 19 8 2025].
- [14] «Judge0 CE — API Documentation,» 2025. [Ηλεκτρονικό]. Available: <https://ce.judge0.com/>.
- [15] «react-simple-code-editor — GitHub,» 2025. [Ηλεκτρονικό]. Available: <https://github.com/react-simple-code-editor/react-simple-code-editor>. [Πρόσβαση 19 8 2025].
- [16] «Prism.js — Lightweight, extensible syntax highlighter,» 2025. [Ηλεκτρονικό]. Available: <https://prismjs.com/>. [Πρόσβαση 19 8 2025].
- [17] «react-confetti — npm package,» 2025. [Ηλεκτρονικό]. Available: <https://www.npmjs.com/package/react-confetti>. [Πρόσβαση 19 8 2025].
- [18] «react-il8next Documentation,» 2025. [Ηλεκτρονικό]. Available: <https://react.il8next.com/>. [Πρόσβαση 19 8 2025].
- [19] «Supabase Docs,» 2025. [Ηλεκτρονικό]. Available: <https://supabase.com/docs>. [Πρόσβαση 19 8 2025].