



ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη web εφαρμογής για τη συλλογή, την παρουσίαση και τη διανομή δεδομένων που αφορούν την εισαγωγή νέων φοιτητών στην τριτοβάθμια εκπαίδευση μέσω των πανελλαδικών εξετάσεων»

**vaseis-app**

*Δείτε και συγκρίνετε εύκολα την εξέλιξη των βάσεων εισαγωγής των τμημάτων ΑΕΙ μέσω των Πανελλαδικών Εξετάσεων!*

**vaseis-app**  
**API**

*Διαδικτυακό API για την ανάκτηση δεδομένων σχετικών με τις Πανελλαδικές Εξετάσεις για την ανάπτυξη εφαρμογών.*

**Του φοιτητή**  
**Βασίλειου Ιωαννίδη**  
**Αρ. Μητρώου: 154453**

**Επιβλέπων**  
**Στέφανος Ουγιάρογλου**  
**Προσωπικό Ε.ΔΙ.Π.**

13 Ιουνίου 2021

Ανάπτυξη web εφαρμογής για τη συλλογή, την παρουσίαση και τη διανομή δεδομένων που αφορούν την εισαγωγή νέων φοιτητών στην τριτοβάθμια εκπαίδευση μέσω των πανελλαδικών εξετάσεων

20199

Φοιτητής: Βασίλειος Ιωαννίδης

Εισηγητής: Στέφανος Ουγιάρογλου

Ημερομηνία ανάληψης: 12-10-2020

Ημερομηνία περάτωσης: 07-06-2021

*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 Βασίλειου Ιωαννίδη που την εκπόνησε. 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, δε 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 πτυχιακής 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.



## Περίληψη

Το Υπουργείο Παιδείας δημοσιεύει κάθε χρόνο με την ανακοίνωση των βάσεων εισαγωγής των Πανελλαδικών Εξετάσεων, ένα πλήθος αρχείων που καλύπτουν όλες τις κατηγορίες υποψηφίων. Τα αρχεία αυτά, περιλαμβάνουν τις βάσεις εισαγωγής του κάθε τμήματος, καθώς και τα στατιστικά προτιμήσεων των υποψηφίων. Αυτό καθιστά δύσκολο για τους υποψήφιους μαθητές την αναζήτηση του ιστορικού βάσεων εισαγωγής, διότι θα πρέπει να ελέγξουν πολλά διαφορετικά αρχεία για κάθε έτος, έτσι ώστε να μπορούν να βγάλουν ένα συμπέρασμα για την πορεία των βάσεων εισαγωγής. Στόχος αυτής της πτυχιακής εργασίας ήταν να λύσει το πρόβλημα αυτό. Αυτό έγινε με την οργάνωση αυτών των δεδομένων, τη δημιουργία ενός ανοιχτού διαδικτυακού API και την ανάπτυξη μίας διαδικτυακής εφαρμογής. Το API προσφέρει τη δυνατότητα στον καθένα να προσπελάσει τις πληροφορίες με σκοπό να αναπτύξει μία δικιά του εφαρμογή. Η εφαρμογή δίνει τη δυνατότητα στους χρήστες να δουν εύκολα το ιστορικό των βάσεων εισαγωγής και των στατιστικών προτιμήσεων των τμημάτων της επιλογής τους.

«Development of a web application for the collection, presentation and distribution of data relative to the insertion of new students to higher education via the panhellenic exams»

«Vasileios Ioannidis»

## **Abstract**

The Ministry of Education publishes each year alongside the announcement of the entrance bases of the Panhellenic Exams, a set of files which covers all the candidate's categories. These files, include the entrance bases of each department, as well as the candidates' department choice statistics. By this, the search of the entrance base history becomes harder for the students. The objective of this Degree Thesis was the solution to this problem. This happened through the organisation of the released data, the creation of an open web API and the development of a web application. The API offers its users the ability to retrieve the data, in order to develop their own application. The web application offers an easy way for its users to compare the entrance bases and the choice statistics of the departments the choose.

# Περιεχόμενα

Περίληψη . . . . .	iii
Abstract . . . . .	iv
Περιεχόμενα . . . . .	v
Κατάλογος Σχημάτων . . . . .	vii
Κατάλογος Πινάκων . . . . .	viii
Συντομογραφίες . . . . .	ix
<b>1 Εισαγωγή . . . . .</b>	<b>1</b>
1.1 Εισαγωγή . . . . .	1
1.2 Σύστημα Εξετάσεων . . . . .	1
1.3 Κίνητρο . . . . .	2
1.4 Συνεισφορά . . . . .	2
1.5 Οργάνωση της εργασίας . . . . .	3
<b>2 Τεχνολογίες . . . . .</b>	<b>5</b>
2.1 Εισαγωγή . . . . .	5
2.2 HTML και CSS . . . . .	5
2.2.1 HTML . . . . .	5
2.2.2 CSS . . . . .	6
2.3 PHP: Hypertext Preprocessor . . . . .	7
2.4 SQL και MariaDB . . . . .	8
2.4.1 SQL . . . . .	8
2.4.2 MariaDB . . . . .	9
2.5 JavaScript . . . . .	9
2.5.1 JSON . . . . .	10
2.5.2 Chart.js . . . . .	11
2.5.3 Prism . . . . .	13
2.6 RESTful API . . . . .	14
2.7 Επίλογος . . . . .	14
<b>3 Backend και Διαδικτυακό API . . . . .</b>	<b>15</b>
3.1 Εισαγωγή . . . . .	15
3.2 Σχεδιασμός Βάσης Δεδομένων . . . . .	15
3.3 Ιστότοπος Μεταφόρτωσης Αρχείων Δεδομένων . . . . .	18
3.4 Διαδικτυακό API . . . . .	22
3.5 Παραδείγματα Κλήσεων του API . . . . .	24
3.5.1 Αναζήτηση Πανεπιστημίων με χρήση μοναδικού κωδικού . . . . .	25
3.5.2 Αναζήτηση βάσεων . . . . .	25
3.5.3 Αναζήτηση βάσεων εισαγωγής με χρήση φίλτρων . . . . .	26
3.5.4 Αναζήτηση βάσεων εισαγωγής με χρήση κριτηρίων . . . . .	28
3.5.5 Αναζήτηση Βάσεων με ερωτήματα της έκδοσης 1.0 . . . . .	29
3.5.6 Αναζήτηση Στατιστικών . . . . .	30
3.5.7 Αναζήτηση Στατιστικών με ερωτήματα της έκδοσης 1.0 . . . . .	32
3.6 Επίλογος . . . . .	33
<b>4 Διαδικτυακή Εφαρμογή . . . . .</b>	<b>35</b>
4.1 Εισαγωγή . . . . .	35
4.2 Ιστορίες Χρήστη . . . . .	35
4.3 Ανάλυση Κώδικα . . . . .	36
4.3.1 Αρχική Σελίδα . . . . .	36

4.3.2	Σελίδα Δεδομένων . . . . .	40
<b>5</b>	<b>Παρουσίαση Εφαρμογής</b>	<b>47</b>
5.0.1	Αρχική Σελίδα . . . . .	47
5.0.2	Σελίδα Δεδομένων . . . . .	48
<b>6</b>	<b>Κλίμακα Ευχρηστίας Συστήματος</b>	<b>55</b>
6.1	Πιστεύω ότι θα χρησιμοποιούσα την εφαρμογή συχνά . . . . .	56
6.2	Πιστεύω ότι η εφαρμογή είναι αχρείαστα περίπλοκη . . . . .	56
6.3	Πιστεύω ότι η εφαρμογή ήταν εύκολη στη χρήση . . . . .	57
6.4	Πιστεύω ότι θα χρειαζόμουν βοήθεια για να χρησιμοποιήσω την εφαρμογή . . . . .	57
6.5	Πιστεύω ότι οι διάφορες λειτουργίες της εφαρμογής ήταν καλά ενσωματωμένες . . . . .	58
6.6	Πιστεύω ότι η σχεδίαση της εφαρμογής ήταν μη συνεπής . . . . .	59
6.7	Πιστεύω ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν εύκολα την εφαρμογή . . . . .	59
6.8	Πιστεύω ότι η εφαρμογή ήταν δύσχρηστη . . . . .	59
6.9	Αισθάνθηκα σίγουρος κατά τη χρήση της εφαρμογής . . . . .	59
6.10	Έπρεπε να μάθω πολλά πράγματα για να χρησιμοποιήσω την εφαρμογή . . . . .	60
6.11	Το σημαντικότερο χαρακτηριστικό της εφαρμογής που λείπει . . . . .	61
6.12	Επίλογος . . . . .	61
<b>7</b>	<b>Συμπεράσματα και προτάσεις βελτίωσης</b>	<b>62</b>
7.1	Συμπεράσματα . . . . .	62
7.2	Μελλοντικές Επεκτάσεις . . . . .	63
	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b>	<b>64</b>

## Κατάλογος Σχημάτων

2.1	Παράδειγμα κώδικα HTML	5
2.2	Παράδειγμα κώδικα CSS	7
2.3	Παράδειγμα κώδικα PHP	8
2.4	Παράδειγμα κώδικα SQL	9
2.5	Παράδειγμα κώδικα JavaScript	10
2.6	Παράδειγμα δεδομένων σε μορφή JSON	11
2.7	Παράδειγμα γραφήματος που παράχθηκε από την Chart.js	11
2.8	Παράδειγμα κώδικα για τη δημιουργία γραφήματος με τη χρήση της Chart.js	12
2.9	Παράδειγμα παρουσίασης κώδικα με τη χρήση της Prism	13
2.10	Παράδειγμα κώδικα για την υλοποίηση επισήμανσης σύνταξης με τη χρήση της Prism	13
3.1	Η δομή του πίνακα university	15
3.2	Η δομή του πίνακα dept	16
3.3	Η δομή του πίνακα examtype	16
3.4	Η δομή του πίνακα specialcat	16
3.5	Η δομή του πίνακα base	17
3.6	Η δομή του πίνακα statistics	17
3.7	Το διάγραμμα οντοτήτων - συσχετίσεων της βάσης δεδομένων	18
3.8	Ο ιστότοπος μεταφόρτωσης των αρχείων δεδομένων	19
3.9	Ο αλγόριθμος επεξεργασίας δεδομένων	20
3.10	Η αλληλουχία εκτέλεσης των ερωτημάτων SQL για το αρχείο δεδομένων των βάσεων εισαγωγής	22
3.11	Μέρος του εγχειριδίου οδηγιών του διαδικτυακού API.	23
3.12	Αλγόριθμος μορφοποίησης δεδομένων.	24
3.13	Μέρος του εγχειριδίου οδηγιών του διαδικτυακού API.	25
3.14	Η δομή του συνδέσμου για την αναζήτηση πανεπιστημίων.	25
3.15	Παράδειγμα αναζήτησης Πανεπιστημίου.	25
3.16	Παράδειγμα αναζήτησης βάσεων εισαγωγής με κριτήρια το έτος και το τμήμα.	26
3.17	Τμήμα των αποτελεσμάτων από την εκτέλεση της κλήσης του API.	26
3.18	Στιγμιότυπο από το εγχειρίδιο χρήσης του API, που εξηγούνται τα φίλτρα.	28
3.19	Παράδειγμα συνδέσμου με χρήση φίλτρων.	28
3.20	Η δομή αναζήτησης βάσεων εισαγωγής με χρήση κριτηρίων (μέγιστη βάση εισαγωγής, όρος αναζήτησης, έτος).	29
3.21	Παράδειγμα απόκρισης του API μετά από αναζήτηση βάσεων εισαγωγής με χρήση κριτηρίων.	29
3.22	Η δομή του συνδέσμου αναζήτησης βάσεων της έκδοσης 1.0	30
3.23	Παράδειγμα απόκρισης του API μετά από αναζήτηση βάσεων εισαγωγής με χρήση ερωτημάτων της έκδοσης 1.0	30
3.24	Η δομή του συνδέσμου αναζήτησης στατιστικών	31
3.25	Παράδειγμα απόκρισης του API σε αναζήτηση στατιστικών	31
3.26	Η δομή της όψης “statistics_v1”	32
3.27	Παράδειγμα απόκρισης του API σε αναζήτηση στατιστικών με χρήση ερωτημάτων της έκδοσης 1.0	33
4.1	Στιγμιότυπο από τη σελίδα διαχείρισης των ιστοριών χρήστη του Trello	36
4.2	Η συνάρτηση fillList	37
4.3	Μέρος των βοηθητικών συναρτήσεων για τη διαχείριση του API	37
4.4	Η συνάρτηση filterDepts	39
4.5	Μέρος του αλγορίθμου διαχείρισης της υποβολής φόρμας	40
4.6	Ο βρόγχος επανάληψης που λαμβάνει τα αρχικά δεδομένα από το API	41

4.7	Ο αλγόριθμος που αρχικοποιεί τις λεπτομερείς πληροφορίες της κατηγορίας 10% για το πρώτο τμήμα	41
4.8	Ο αλγόριθμος που εκτελείται μόλις ενεργοποιηθεί ο ακροατής γεγονότων για τη φόρτωση της σελίδας	42
4.9	Μέρος της συνάρτησης workData	43
4.10	Ο αλγόριθμος της συνάρτησης removeDept	44
4.11	Ο αλγόριθμος της συνάρτησης changeType	45
4.12	Ο αλγόριθμος της συνάρτησης buildSelect	46
5.1	Το πρώτο μέρος της σελίδας επιλογής	47
5.2	Το παράθυρο επιλογής τμημάτων της αρχικής σελίδας	48
5.3	Το δεύτερο μέρος της σελίδας επιλογής	49
5.4	Το πάνω μέρος της σελίδας δεδομένων	50
5.5	Το πάνω μέρος της σελίδας δεδομένων μετά από αλληλεπίδραση του χρήστη	50
5.6	Το τμήμα πληροφοριών βάσεων εισαγωγής της σελίδας δεδομένων	51
5.7	Το τμήμα πληροφοριών των στατιστικών προτιμήσεων της σελίδας δεδομένων	51
5.8	Το κέντρο διαχείρισης επιλεγμένων τμημάτων	52
5.9	Η λειτουργία αναζήτησης για την προσθήκη νέων τμημάτων	53
5.10	Η αρχική σελίδα της διαδικτυακής εφαρμογής σε συσκευή με μικρή οθόνη	53
5.11	Η σελίδα δεδομένων της διαδικτυακής εφαρμογής σε συσκευή με μικρή οθόνη	54
6.1	Οι απαντήσεις των χρηστών στη φράση “Πιστεύω ότι θα χρησιμοποιούσα την εφαρμογή συχνά”	56
6.2	Οι απαντήσεις των χρηστών στη φράση “Πιστεύω ότι η εφαρμογή είναι αχρείαστα περίπλοκη”	57
6.3	Οι απαντήσεις των χρηστών στη φράση “Πιστεύω ότι η εφαρμογή ήταν εύκολη στη χρήση”	57
6.4	Οι απαντήσεις των χρηστών στη φράση “Πιστεύω ότι θα χρειαζόμουν βοήθεια για να χρησιμοποιήσω την εφαρμογή”	58
6.5	Οι απαντήσεις των χρηστών στην πρόταση “Πιστεύω ότι οι διάφορες λειτουργίες της εφαρμογής ήταν καλά ενσωματωμένες”	58
6.6	Οι απαντήσεις των χρηστών στην πρόταση “Πιστεύω ότι η σχεδίαση της εφαρμογής ήταν μη συνεπής”	59
6.7	Οι απαντήσεις των χρηστών στην πρόταση “Πιστεύω ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν εύκολα την εφαρμογή”	60
6.8	Οι απαντήσεις των χρηστών στην πρόταση “Πιστεύω ότι η εφαρμογή ήταν δύσχρηστη”	60
6.9	Οι απαντήσεις των χρηστών στην πρόταση “Αισθάνθηκα σίγουρος κατά τη χρήση της εφαρμογής”	60
6.10	Οι απαντήσεις των χρηστών στην πρόταση “Έπρεπε να μάθω πολλά πράγματα για να χρησιμοποιήσω την εφαρμογή”	61
6.11	Οι απαντήσεις των χρηστών στην πρόταση “Το σημαντικότερο χαρακτηριστικό της εφαρμογής που λείπει”	61

## Κατάλογος Πινάκων

## Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
ΥΠΑΙΘ	Υπουργείο Παιδείας και Θρησκευμάτων
API	Application Programming Interface

## Κεφάλαιο 1ο: Εισαγωγή

### 1.1 Εισαγωγή

Στο κεφάλαιο αυτό, θα αναπτυχθούν πληροφορίες σχετικά με τις Πανελλαδικές Εξετάσεις και τον τρόπο που το Υπουργείο Παιδείας και Θρησκευμάτων (ΥΠΑΙΘ) δημοσιεύει τις βάσεις εισαγωγής και τα στατιστικά κάθε έτος. Ακόμα, θα αναλυθεί τι έδωσε το κίνητρο για την εκπόνηση αυτής της εργασίας και η συνεισφορά της. Τέλος, θα γίνει αναφορά στην οργάνωση της.

### 1.2 Σύστημα Εξετάσεων

Οι υποψήφιοι για την εισαγωγή στην τριτοβάθμια εκπαίδευση, πρέπει να συμμετάσχουν στις πανελλαδικές εξετάσεις που διενεργεί κάθε χρόνο το Υπουργείο Παιδείας και Θρησκευμάτων. Αυτοί εξετάζονται σε 4 μαθήματα. Αξίζει να σημειωθεί ότι παλαιότερα και ανάλογα τη νομοθεσία ο αριθμός αυτός ήταν διαφορετικός. Αυτά τα μαθήματα ορίζονται από την ομάδα προσανατολισμού που θα επιλέξουν οι μαθητές, με εξαίρεση το μάθημα της Νεοελληνικής Γλώσσας και Λογοτεχνίας που είναι κοινό σε όλες τις ομάδες. Επιπρόσθετα, ως προϋπόθεση για την εισαγωγή σε κάποια πανεπιστημιακά τμήματα απαιτείται η εξέταση σε κάποια επιπλέον μαθήματα (ξένες γλώσσες, σχέδιο κ.α.).

Τα μαθήματα της κάθε ομάδας έχουν διαφορετικό συντελεστή βαρύτητας, οπότε και η επίδοση των υποψηφίων στο καθένα επηρεάζει είτε περισσότερο είτε λιγότερο την τελική βαθμολογία, η οποία υπολογίζεται με τον πολλαπλασιασμό των βαθμών που πέτυχε ο υποψήφιος με τους αντίστοιχο συντελεστή του κάθε μαθήματος.

Με την ανακοίνωση και δημοσίευση των επιδόσεων των υποψηφίων από το ΥΠΑΙΘ, οι υποψήφιοι συμπληρώνουν το μηχανογραφικό δελτίο στο οποίο ορίζουν με σειρά προτίμησης τα τμήματα των τριτοβάθμιων ιδρυμάτων που επιθυμούν να εισαχθούν. Τέλος, με βάση τις προτιμήσεις και τις βαθμολογίες των υποψηφίων, το ΥΠΑΙΘ ανακοινώνει τις βάσεις εισαγωγής για όλες τις κατηγορίες για το έτος εξέτασης, καθώς και στατιστικά, όπως τα στατιστικά επιδόσεων και προτιμήσεων των υποψηφίων.

Το σύστημα των πανελλαδικών εξετάσεων είναι σύνθετο, διότι εκτός από τον διαχωρισμό των Γενικών και τα Επαγγελματικών Λυκείων, υπάρχει και ο διαχωρισμός στις κατηγορίες 90% και 10%. Ο τύπος 90% αναφέρεται στις βάσεις εισαγωγής και στατιστικά, τα οποία καθορίζονται από τα αποτελέσματα των εξετάσεων τους έτους που διενεργούνται οι εξετάσεις. Ο τύπος 10% αναφέρεται σε αυτά, που καθορίζονται από τις αιτήσεις των υποψηφίων των δύο προηγούμενων ετών, οι οποίοι επιθυμούν να εισαχθούν σε κάποιο ίδρυμα με βάση τα αποτελέσματα της χρονιάς που εξετάστηκαν και ανταγωνίζονται αυτούς που έκαναν την ανάλογη αίτηση.

Παράλληλα, ισχύουν οι ειδικές κατηγορίες οι οποίες έχουν παραλλαγές στο πέρασμα του χρό-

νου. Μερικά παραδείγματα αυτών είναι οι βάσεις εισαγωγής για τους υποψήφιους που ανήκουν σε τρίτεκνη ή πολύτεκνη οικογένεια, για τους μουσουλμάνους και για τους Έλληνες του εξωτερικού. Ακόμη, για συγκεκριμένες σχολές όπως οι Αστυνομικές υπάρχουν διαφορετικές κατηγορίες όπως αυτές για τους πολίτες και για τους αστυνομικούς.

### 1.3 Κίνητρο

Ο σύνθετος τρόπος δημοσιοποίησης των αποτελεσμάτων των Πανελλαδικών Εξετάσεων μας οδήγησε στην εκπόνηση αυτής της εργασίας. Το Υπουργείο Παιδείας δημοσιεύει κάθε χρόνο στο χώρο των ανακοινώσεων του μία λίστα από υπολογιστικά φύλλα, που το καθένα αντιστοιχεί στους διαφορετικούς τύπους και στις διαφορετικές κατηγορίες.

Στα υπολογιστικά φύλλα σχετικά με τις βάσεις εισαγωγής περιλαμβάνονται ο κωδικός και ο τίτλος της κάθε σχολής, το πανεπιστήμιο στο οποίο ανήκει, το επιστημονικό πεδίο, τις βάσεις εισαγωγής του πρώτου και του τελευταίου και το είδος της θέσης που αντιστοιχούν αυτές οι βάσεις. Στα αρχεία που αφορούν τα στατιστικά προτιμήσεων επίσης περιλαμβάνονται τα στοιχεία των τμημάτων και επιπρόσθετα περιλαμβάνουν τον αριθμό των υποψηφίων που επέλεξαν τη σχολή από πρώτη ως έκτη επιλογή για τους επιτυχόντες και από πρώτη ως τρίτη για το σύνολο των υποψηφίων. Παρατηρήθηκε ότι υπάρχουν διαφορές και προσθαφαιρέσεις πεδίων σε αυτά τα αρχεία σε κάποια από τα έτη που έχουν περάσει.

Αρχικά, ως πρώτο πρόβλημα εντοπίζεται η μη ανάρτηση αυτών των αρχείων σε έναν συγκεκριμένο και εύκολα προσπελάσιμο χώρο, στον οποίο να βρίσκονται συγκεντρωτικά και οργανωμένα όλα τα αρχεία για όλες τις χρονιές. Αυτό καθιστά δύσκολη για τον υποψήφιο ή για τον όποιο ενδιαφερόμενο τη συλλογή αυτών των αρχείων. Επιπλέον, τα ίδια τα αρχεία και η δομή τους καθιστά δύσκολη την αναζήτηση και την ανάλυση των δεδομένων καθώς ο ενδιαφερόμενος θα πρέπει μελετήσει πολλά διαφορετικά αρχεία για τις κατηγορίες που τον ενδιαφέρει, έτσι ώστε να βγάλει κάποιο συμπέρασμα.

Στόχος της εργασίας είναι να επιτύχει ευκολότερη αναζήτηση και ανάλυση αυτών των δεδομένων, η παρουσίαση αυτών, αλλά και να προσφέρει έναν εύκολο τρόπο να αξιοποιηθούν αυτά τα δεδομένα από προγραμματιστές, με τις μεθόδους που θα αναλυθούν παρακάτω.

### 1.4 Συνεισφορά

Για τις ανάγκες της εργασίας έγινε συλλογή όλων των αρχείων σχετικών με τις βάσεις εισαγωγής και τα στατιστικά προτιμήσεων από το 2013 και το 2016 αντίστοιχα έως και το 2020 από την ιστοσελίδα του ΥΠΑΙΘ. Ύστερα έγινε ανάλυση της δομής των αρχείων αυτών, για να αποφασιστεί ποια δεδομένα θα αξιοποιηθούν. Δημιουργήθηκαν δύο αρχεία πρότυπα, ένα για τα δεδομένα σχετικά με τις βάσεις εισαγωγής και ένα για τα στατιστικά προτιμήσεων. Αυτό

έγινε με σκοπό να μειωθεί ο αριθμός των αρχείων στα δύο για κάθε έτος, με το σύνολο όλων των δεδομένων να συγκεντρώνεται σε αυτά.

Η ανάπτυξη του λογισμικού ξεκίνησε με τη σχεδίαση της βάσης δεδομένων, το σχήμα της οποίας θα αναλυθεί παρακάτω. Στη συνέχεια, αναπτύχθηκε μια ιστοσελίδα για τη μεταφόρτωση αυτών των αρχείων στον διακομιστή και ανάλυση αυτών μέσω αλγορίθμου που αναπτύχθηκε, έτσι ώστε να καταχωρηθούν σωστά στη βάση δεδομένων. Η σελίδα περιέχει κάποιες πληροφορίες για τη μορφή των αρχείων και δίνει τη δυνατότητα της μεταφόρτωσης 2 αρχείων τη φορά (ένα για τα δεδομένα των βάσεων και ένα για τα στατιστικά). Το πρόβλημα που λύθηκε με αυτόν τον τρόπο, ήταν η επίτευξη της οργανωμένης συγκέντρωσης όλων των δεδομένων σε μια πηγή και η δυνατότητας εύκολης ενημέρωσης της πηγής με νέα δεδομένα.

Ως επόμενο βήμα τέθηκε η δημιουργία ενός δημόσιου διαδικτυακού API. Στόχος της ανάπτυξης του API, ήταν η εύκολη ανάκτηση και συλλογή δεδομένων για προγραμματιστές και ερευνητές. Συγκεκριμένα αναπτύχθηκαν μέθοδοι ανάκτησης του συνόλου ή υποσυνόλων δεδομένων, όπως για παράδειγμα τα τριτοβάθμια ιδρύματα, τα τμήματα που αποτελούν μέρος τους, τις βάσεις εισαγωγής και τα στατιστικά. Επιπλέον, δίνεται η δυνατότητα αναζήτησης και ανάκτησης μέρος των δεδομένων μέσω συγκεκριμένων μεθόδων αναζήτησης. Τέλος, για την καθοδήγηση των χρηστών του API και για την ευκολότερη προσαρμογή αυτών, δημιουργήθηκε μια ιστοσελίδα που δρα ως οδηγός χρήσης του, στην οποία παρουσιάζονται αναλυτικά όλες οι λειτουργίες του API και παραδείγματα αυτών. Μέσω του API, λύνεται το πρόβλημα της δυσκολίας αξιοποίησης και αναζήτησης των δεδομένων των Πανελλαδικών Εξετάσεων.

Τέλος, αναπτύχθηκε μια διαδικτυακή εφαρμογή που είχε ως στόχο την παρουσίαση των δεδομένων αυτών μέσω γραφικού περιβάλλοντος. Με αυτή την εφαρμογή δίνεται η δυνατότητα στον χρήστη να αναζητήσει, να παρατηρήσει και να αναλύσει τα δεδομένα αυτά με εύκολο τρόπο και χωρίς τη γνώση προγραμματισμού. Προσφέρεται η αναζήτηση και επιλογή τμημάτων και η παρουσίαση τους με χρήση συγκριτικών διαγραμμάτων και οργανωμένης παρουσίασης τους με βάση το έτος και την κατηγορία εξέτασης.

## 1.5 Οργάνωση της εργασίας

Στο πρώτο κεφάλαιο της εργασίας έγινε αναφορά στο σύστημα εξετάσεων των Πανελλαδικών Εξετάσεων, στη μορφή και τη μέθοδο που δημοσιεύονται τα δεδομένα από το ΥΠΑΙΘ. Επίσης, αναπτύχθηκαν οι λόγοι που οδήγησαν στην εκπόνηση της εργασίας, καθώς και μια σύντομη αναφορά στον τρόπο με τον οποίο έγινε η συνεισφορά για την επίλυση των προβλημάτων.

Στα επόμενα κεφάλαια θα αναπτυχθούν πληροφορίες σχετικά με τις τεχνολογίες και τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την ανάπτυξη του απαραίτητου λογισμικού. Θα γίνει παρουσίαση του back-end που αναπτύχθηκε και συγκεκριμένα στη σχεδίαση της βάσης δεδομένων, του διαδικτυακού API που αναπτύχθηκε, της σελίδας αναφοράς αυτού και κάποιων

## Κεφάλαιο 1

παραδειγμάτων κλήσεων προς αυτό. Επίσης, θα γίνει αναλυτική αναφορά στη διαδικτυακή εφαρμογή που αναπτύχθηκε και αξιοποιεί το API, θα παρουσιαστούν αποσπάσματα από τον κώδικα που την αποτελεί και ιστορίες χρήστη. Τέλος, θα αναφερθούν τα συμπεράσματα που προέκυψαν από την εκπόνηση της εργασίας και οι μελλοντικές επεκτάσεις που μπορούν να γίνουν.

## Κεφάλαιο 2ο: Τεχνολογίες

### 2.1 Εισαγωγή

Στο κεφάλαιο αυτό θα γίνει μία εισαγωγή στις τεχνολογίες και τα εργαλεία που απαιτούνται για την ανάπτυξη του λογισμικού που αφορά την εργασία. Συγκεκριμένα θα γίνει αναφορά στις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την ανάπτυξη του διαδικτυακού API και σε αυτές για τη web εφαρμογή, καθώς και για τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη και δοκιμή τους.

### 2.2 HTML και CSS

#### 2.2.1 HTML

Η HTML (HyperText Markup Language) είναι μία γλώσσα σήμανσης, η οποία χρησιμοποιείται για τη δημιουργία δομημένων εγγράφων. Αποτελεί το πρότυπο για την παρουσίαση εγγράφων στο διαδίκτυο. Η HTML περιλαμβάνει ένα σύνολο στοιχείων, με τα οποία μέρη του περιεχομένου του εγγράφου επεξεργάζονται προκειμένου να παρουσιαστούν με διαφορετική μορφή (π.χ. επικεφαλίδες κειμένου, παράγραφοι κ.α.).

Η σύνταξη της HTML αποτελείται αποκλειστικά από ετικέτες οι οποίες περιβάλλουν το περιεχόμενο που θα παραμετροποιηθεί. Αυτές, ξεκινούν με το σύμβολο του “μικρότερο από“ (<) ακολουθούμενες από τον τίτλο της ετικέτας και το σύμβολο “μεγαλύτερο από“ (>). Επιπρόσθετα, οι ετικέτες περιλαμβάνουν ένα σύνολο υποχρεωτικών ή μη χαρακτηριστικών που ορίζουν τις παραμέτρους αυτής (π.χ. το χαρακτηριστικό `method="post"` στην ετικέτα `form`, ορίζει ότι η HTTP μέθοδος που θα χρησιμοποιηθεί για την αποστολή δεδομένων, είναι η `post`).

```

16 <form method="post" enctype="multipart/form-data">
17   <h3>Αρχείο Βάσεων</h3>
18   <input type="file" name="vaseis" id="">
19   <h3>Αρχείο Στατιστικών</h3>
20   <input type="file" name="stats" id="">
21   <br><br>
22   <input type="submit" value="Υποβολή" name="submit">
23 </form>
24

```

Σχήμα 2.1: Παράδειγμα κώδικα HTML

Υπεύθυνο για την ανάπτυξη και προτυποποίηση της HTML είναι το W3C (World Wide Web Consortium), ενώ από το 2004 για τη δημοσίευση νέων προτύπων εργάζεται και το WHATWG (Web Hypertext Application Technology Working Group). Μέλη του WHATWG μεταξύ άλλων είναι εταιρίες όπως η Apple, η Mozilla Corporation και η Opera Software. [1] Αν και το W3C και το WHATWG συνεργάζοντουσαν, σε πολλές περιπτώσεις υπήρχαν διαφωνίες με αποτέλεσμα

να δημοσιεύονται πρότυπα και από τους δύο, τα οποία είχαν διαφορετικά χαρακτηριστικά. Για την αντιμετώπιση το προβλήματος της δημιουργίας δύο διαφορετικών εκδόσεων της HTML το 2019 αποφασίστηκε η από κοινού εργασία για ένα μοναδικό πρότυπο της HTML. [2]

Η πρώτη έκδοση της HTML εκδόθηκε το 1993 και έκτοτε έχουν δημοσιευτεί αναθεωρήσεις και νέες εκδόσεις. Η τελευταία έκδοση της HTML είναι η HTML5, η ανάπτυξη της οποίας ξεκίνησε το 2004 και ολοκληρώθηκε το 2014 με αναθεωρήσεις αυτής να έχουν γίνει το 2016 (HTML5.1) και το 2017 (HTML5.2). [3] Οι νέες εκδόσεις είναι πλήρως συμβατές με τις παλαιότερες.

Η HTML5 προσθέτει ετικέτες με σημασιολογική σημασία για χρήση αντί της ετικέτας γενικής χρήσης “div“, ενώ καταργεί άλλες ετικέτες και προτείνει τη χρήση της CSS, για επίτευξη των στόχων αυτών των ετικετών. Ακόμα δίνεται η δυνατότητα αναπαραγωγής οπτικοακουστικού περιεχομένου χωρίς την ανάγκη χρήσης κάποιου πρόσθετου (π.χ. Adobe Flash). Τέλος, προσθέτει έναν μεγάλο αριθμό από καινούρια API τα οποία συνεργάζονται με τη JavaScript. Κάποια παραδείγματα αυτών είναι το Canvas, με το οποίο γίνεται ευκολότερη η δημιουργία γραφικών και κινούμενων σχεδίων για το διαδίκτυο, αλλά και η υποστήριξη του “drag and drop“. [4]

Η HTML αποτελεί το κυριότερο εργαλείο για την ανάπτυξη μίας διαδικτυακής εφαρμογής. Έτσι και στην περίπτωση της διαδικτυακής εφαρμογής που αναπτύχθηκε, αποτελεί το κυριότερο στοιχείο, καθώς προσφέρει όλη τη δομή της κάθε σελίδας.

### 2.2.2 CSS

Η CSS (Cascading Style Sheets) είναι μια γλώσσα φύλλων ύφους, σκοπός της οποίας είναι η επεξεργασία του τρόπου παρουσίασης των ετικετών της HTML. Προσφέρει τη δυνατότητα επεξεργασίας των χρωμάτων που χρησιμοποιούνται, των γραμματοσειρών και της διάταξης του περιεχομένου.

Η σύνταξη της CSS είναι αρκετά απλή, καθώς αποτελείται από τον επιλογέα. Ο επιλογέας μπορεί να είναι μια ετικέτα HTML (π.χ. p), ένα χαρακτηριστικό τύπου class (π.χ. .class-example) ή ένα χαρακτηριστικό τύπου id (π.χ. #id-example). Επιπρόσθετα δίνεται η δυνατότητα προσθήκης ψευδοκλάσεων, με σκοπό την πιο προχωρημένη επεξεργασία της παρουσίασης του εγγράφου (π.χ. η ψευδοκλάση hover εφαρμόζει τα χαρακτηριστικά της, όταν ο κέρσορας του χρήστη βρίσκεται πάνω από την ετικέτα που εφαρμόζεται). Ύστερα από τον επιλογέα ακολουθεί ένα ζεύγος αγκίστρων, τα οποία περιέχουν όλες τις παραμέτρους. Οι παράμετροι έχουν τη μορφή “όνομα παραμέτρου : τιμή παραμέτρου“.

Όπως η HTML, έτσι και η CSS αποτελεί ένα ανοιχτό πρότυπο το οποίο βρίσκεται υπό τη διαχείριση του W3C. Η πρώτη έκδοση της, δημοσιεύτηκε το 1996 και η τελευταία η οποία ονομάστηκε CSS3 το 2016. [5] Από τότε αποφασίστηκε να μην εκδίδονται νέες εκδόσεις, αλλά να δημοσιεύονται αναθεωρήσεις που ονομάζονται “στιγμιότυπα“.

```

102  .landing-text .subtitle {
103      font-size: 1.2em;
104      margin: 10px;
105  }
106
107  .landing .form-container form {
108      display: flex;
109      flex-direction: row;
110      align-items: center;
111      justify-content: center;
112      flex-wrap: wrap;
113  }
114

```

Σχήμα 2.2: Παράδειγμα κώδικα CSS

Στις τελευταίες αναθεωρήσεις της CSS προστέθηκαν χαρακτηριστικά τα οποία αλλάζουν τον τρόπο, που αναπτύσσονται οι ιστοσελίδες. Ο τρόπος σχεδίασης της διάταξης ενός περίπλοκου εγγράφου έγινε ευκολότερος με την προσθήκη των Flexible Box και Grid. [6] [7] Ακόμα δίνεται η δυνατότητα σύνθετων επιλογών με τη χρήση κανονικών εκφράσεων και η χρήση μεταβλητών για τιμές παραμέτρων που επαναχρησιμοποιούνται. Τέλος, προσφέρονται περισσότερες επιλογές για τη σύνταξη media queries, με σκοπό την πιο αποδοτική ανάπτυξη αποκριτικού σχεδιασμού (Responsive Design).

Η CSS χρησιμοποιήθηκε για τη σχεδίαση της διαδικτυακής ιστοσελίδας. Οι νέες δυνατότητες που προσφέρουν τα πρόσφατα στιγμιότυπα που έχουν κυκλοφορήσει έδωσαν τη δυνατότητα ανάπτυξης της διαδικτυακής εφαρμογής, χωρίς την ανάγκη χρησιμοποίησης επιπλέον βιβλιοθηκών για θέματα δομής και εμφάνισης.

### 2.3 PHP: Hypertext Preprocessor

Η PHP είναι μία scripting γλώσσα για την ανάπτυξη διαδικτυακών εφαρμογών που δημιουργήθηκε το 1995 από τον Rasmus Lerdorf. Από την πρώτη έκδοση της έως σήμερα, ο κώδικας της έχει επαναγραφτεί πολλές φορές με σκοπό την καλύτερη απόδοση της γλώσσας και την προσθήκη νέων χαρακτηριστικών που δεν υπήρχαν στις πρώτες εκδόσεις. [8]

Η PHP είναι μία γλώσσα ανοιχτού κώδικα, στην οποία ο καθένας μπορεί να συμβάλλει. Η μεγάλη συμμετοχή της κοινότητας έχει ως αποτέλεσμα να μπορεί να τρέξει στα περισσότερα λειτουργικά συστήματα και στους περισσότερους διαδικτυακούς διακομιστές, μέσω πρόσθετων που είναι εγκατεστημένα σε αυτούς. Ένα από τα σημαντικότερα χαρακτηριστικά της, είναι η υποστήριξη ενός μεγάλου πλήθους συστημάτων διαχείρισης βάσεων δεδομένων, από τις πολύ διαδεδομένες (mariaDB, postgresSQL κ.α.) έως και κάποιες πιο σύγχρονες (mongoDB κ.α.), χωρίς την ανάγκη για εγκατάσταση κάποιου πρόσθετου.

Ο κώδικας που θα εκτελεστεί από τον διακομιστή, θα πρέπει να βρίσκεται μέσα σε αρχεία με

κατάληξη .php και επιπλέον θα πρέπει να προηγείται του κώδικα ο οριοθέτης “<?php” και στο τέλος αυτού ο οριοθέτης “>”. Οι μεταβλητές θα πρέπει να ξεκινούν με το σύμβολο \$ και ύστερα θα πρέπει το όνομα της μεταβλητής να ξεκινά είτε με πεζό ή κεφαλαίο χαρακτήρα είτε με κάτω παύλα. Οι μεταβλητές δεν είναι αυστηρού τύπου, δηλαδή μπορεί να αλλάξει ο τύπος αυτών, σε αντίθεση με άλλες γλώσσες προγραμματισμού.

Τα τελευταία χρόνια έχει γίνει μια προσπάθεια προτυποποίησης της PHP από δημιουργούς βιβλιοθηκών. Η προσπάθεια αυτή οδήγησε στη δημιουργία του PHP-FIG, το οποίο εκδίδει κάποιες προτάσεις για την προτεινόμενη υλοποίηση συγκεκριμένων λειτουργιών με τη χρήση της PHP. [9]

Μία ακόμα προσπάθεια εκσυγχρονισμού της PHP, αποτελεί το Composer, το οποίο πρόκειται για έναν διαχειριστή εξαρτήσεων. Δίνει τη δυνατότητα ενός κεντρικού σημείου διαχείρισης των βιβλιοθηκών και των εκδόσεων των βιβλιοθηκών που θα χρησιμοποιηθούν στην ανάπτυξη ενός λογισμικού. Έμπνευση για τη δημιουργία του αποτέλεσε το οικοσύστημα npm που αναπτύχθηκε για τη γλώσσα JavaScript. [10]

```
} elseif ($_GET["type"] == "gel-ime-ten") {
    if($details) {
        $query = "SELECT b.*, d.name, u.full_title, u.title FROM " . $this->tableName . " AS b " .
            "LEFT JOIN dept AS d ON b.code = d.code LEFT JOIN university AS u on d.uni_id = u.id " .
            "WHERE b.code=? and b.year=? and b.title like '10\% ΓΕΛ%' " .
            "ORDER BY b.year ASC";
    } else {
        $query = "SELECT * FROM " . $this->tableName . " WHERE code=? and year=? AND title like '10\
    }
    $stmt = $this->conn->prepare($query);
    $stmt->bind_param('ii', $dept, $year);
    $stmt->execute();
}
```

Σχήμα 2.3: Παράδειγμα κώδικα PHP

Η PHP αποτελεί την κύρια γλώσσα προγραμματισμού που χρησιμοποιήθηκε και από το διαδικτυακό API και από τη διαδικτυακή εφαρμογή. Το API υλοποιείται πλήρως από την PHP και η διαδικτυακή εφαρμογή αποτελείται σε μεγάλο βαθμό από κώδικα γραμμένο σε αυτή γλώσσα.

## 2.4 SQL και MariaDB

### 2.4.1 SQL

Η SQL (Structured Query Language) είναι μία γλώσσα για τη δημιουργία, αποθήκευση και επεξεργασία δεδομένων. Υλοποιεί το σχεσιακό μοντέλο, καθώς τα αποτελέσματα των εντολών που εκτελούνται είναι πίνακες. Το σχεσιακό μοντέλο οργανώνει τα δεδομένα σε πίνακες. Οι πίνακες αποτελούνται από στήλες και γραμμές και μπορούν να είναι αποθηκευμένοι είτε προσωρινά είτε μόνιμα. Οι στήλες ορίζουν τον τύπο των δεδομένων, ενώ οι γραμμές αποτελούν τα

ίδια τα δεδομένα. Οι πίνακες πρέπει να έχουν μία ή ένα σύνολο στηλών οι οποίες θα αποτελούν το κύριο κλειδί και θα είναι μοναδικές τιμές. Προαιρετικά μπορούν να περιέχουν στήλες οι οποίες αποτελούν ξένο κλειδί. Το ξένο κλειδί αποτελεί τιμή που ανήκει σε γραμμή άλλου πίνακα και χρησιμοποιείται για συσχέτιση των δεδομένων διαφορετικών πινάκων. [11]

```

1 SELECT b.*, d.`name`, u.`full_title`, u.`title`
2 FROM `base` AS b
3 LEFT JOIN `dept` AS d ON b.`code` = d.`code`
4 LEFT JOIN `university` AS u ON d.`uni_id` = u.`id`
5 WHERE b.`code`=1625 AND b.`year`=2020 AND b.`title` LIKE '10\% ΓΕΛ%'
6 ORDER BY b.`year` ASC
7

```

Σχήμα 2.4: Παράδειγμα κώδικα SQL

Η SQL αποτελεί πρότυπο πιστοποιημένο από τον οργανισμό ISO από το 1987. [12] Έκτοτε το πρότυπο ανανεώνεται συνεχώς με την τελευταία έκδοση να εκδίδεται το 2016. [13]

### 2.4.2 MariaDB

Η MariaDB είναι ένα RDBMS (Σχεσιακό Σύστημα Διαχείρισης Βάσης Δεδομένων) το οποίο αξιοποιεί την SQL για τη διαχείριση της βάσης δεδομένων. Αποτελεί ένα ανοιχτού κώδικα RDBMS, το οποίο δημιουργήθηκε το 2009 ως κόμβος της MySQL, ύστερα από την εξαγορά της από την Oracle Corporation. Η MySQL αποτελεί το δημοφιλέστερο RDBMS στον κόσμο. [14]

Η MariaDB αρχικά δημιουργήθηκε ως μια εναλλακτική της MySQL που θα ήταν πλήρως συμβατή με αυτή, αλλά με την πάροδο του χρόνου έχουν προστεθεί νέα χαρακτηριστικά, τα οποία είναι μη συμβατά με αυτή. [15] Η SQL και η MariaDB αποτελούν τους πυλώνες του διαδικτυακού API, καθώς η δυνατότητα που προσφέρουν για δομημένη αποθήκευση πληροφορίας, επέτρεψε τον ευκολότερο σχεδιασμό της βάσης δεδομένων.

## 2.5 JavaScript

Η JavaScript είναι μία γλώσσα προγραμματισμού που δημιουργήθηκε το 1995 για τον φυλλομετρητή Netscape Navigator, με σκοπό την προσθήκη της δυνατότητας αλληλεπίδρασης με μια ιστοσελίδα. Η JavaScript αποτελεί υλοποίηση του προτύπου ECMAScript που παρέχεται από την ECMA International. [16]

Η αρχική πρόθεση κατά τη δημιουργία της JavaScript ήταν να αποτελεί μια γλώσσα προγραμματισμού σεναρίων, αλλά πλέον αποτελεί μια γλώσσα προγραμματισμού γενικού σκοπού. Αυτό συμβαίνει, γιατί πλέον δίνεται η δυνατότητα προγραμματισμού εξυπηρετών [17], εφαρμογών υπολογιστών [18] και εφαρμογών έξυπνων συσκευών [19] με την χρήση της JavaScript.

Η Javascript είναι ταυτόχρονα μια αντικειμενοστραφής και δομημένη γλώσσα. Οι μεταβλητές της δεν είναι σταθερού τύπου, συνεπώς κατά την εκτέλεση του προγράμματος ο τύπος μπορεί να

αλλάξει. Αυτό είναι κάτι που επιδιώκει να αλλάξει η TypeScript, μία επέκταση της JavaScript, η οποία μεταξύ άλλων επιβάλλει την αυστηρότητα των τύπων των δεδομένων. [20]

```
430 async function changeBaseData() {
431     statsId = document.querySelector( selectors: '.base-details').id;
432     let year = document.querySelector( selectors: '.base-year-select').value;
433     let selectValue = document.querySelector( selectors: '.type-select').value;
434     let type;
435     if (selectValue === "ΓΕΛΑ") type = 0;
436     else if (selectValue === "ΕΠΑΛ") type = 1;
437     await loadBaseData(year, statsId, type);
438     if (type === 0) type = 2;
439     else type = 3;
440     await loadBaseData(year, statsId, type);
441 }
```

Σχήμα 2.5: Παράδειγμα κώδικα JavaScript

Η υλοποίηση της JavaScript στους φυλλομετρητές αποτελεί ευθύνη των προγραμματιστών των φυλλομετρητών. Αυτό έχει οδηγήσει σε υλοποίηση διαφορετικών χαρακτηριστικών στις διαφορετικές υλοποιήσεις, με αποτέλεσμα να υπάρχουν προβλήματα ασυμβατότητας. Κάποιες γνωστές υλοποιήσεις της JavaScript είναι οι SpiderMonkey (Mozilla Firefox), V8 (Google Chrome και node.js), JavaScriptCore (Apple Safari) και η JScript (Internet Explorer). [21]

Η JavaScript χρησιμοποιήθηκε αποκλειστικά από τη διαδικτυακή εφαρμογή. Ήταν απαραίτητη η χρήση της για τη διαδραστικότητα της ιστοσελίδας και για την υποστήριξη που προσέφερε στην CSS για την προσαρμογή της εφαρμογής σε κάθε τύπο οθόνης. Επιπρόσθετα, η JavaScript χρησιμοποιήθηκε για το μεγαλύτερο μέρος των αιτημάτων προς το API, με τη χρήση του Fetch API.

### 2.5.1 JSON

Το JSON (JavaScript Object Notation) είναι πρότυπο παράστασης και οργάνωσης δεδομένων. Θεωρείται εύκολο στην ανάγνωση από ανθρώπους και από μηχανές. Η δομή του θυμίζει αυτή των αντικειμένων της γλώσσας JavaScript, αλλά υποστηρίζεται από τις περισσότερες γλώσσες προγραμματισμού. [22]

Αποτελείται από ζευγάρια ονομάτων και τιμών. Τα ονόματα θα πρέπει να περικλείονται από εισαγωγικά, ενώ οι τιμές μπορούν να είναι αλφαριθμητικά, τύποι δεδομένων αλήθειας, null ή πίνακες. Επιπρόσθετα οι τιμές μπορούν να αποτελούν οι ίδιες αντικείμενα, τα οποία αντίστοιχα περιέχουν τα δικά τους ζευγάρια ονομάτων και τιμών. [23]

Το JSON χρησιμοποιήθηκε ως ο μοναδικός τύπος δεδομένων που προσφέρει το διαδικτυακό API. Παράλληλα χρησιμοποιήθηκε από τη διαδικτυακή εφαρμογή, η οποία βασίζεται πλήρως στο API για τη λήψη των απαραίτητων δεδομένων. Η εφαρμογή θα έπρεπε να επεξεργαστεί

```

1 [
2   {
3     "code": 101,
4     "deptName": "ΘΕΟΛΟΓΙΑΣ (ΑΘΗΝΑ)",
5     "uniTitle": "Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών",
6     "uniTitleShort": "ΕΚΠΑ",
7     "bases": [
8       {
9         "examType": "10% ΓΕΛ, ΕΠΑΛΒ 2012",
10        "specialCat": "10% ΓΕΝΙΚΗ ΣΕΙΡΑ",
11        "positions": 22,
12        "baseFirst": 16010,
13        "baseLast": 12106,
14        "year": 2013
15      },
16      {
17        "examType": "ΓΕΛ, ΕΠΑΛΒ ΕΣΠΕΡΙΝΑ",
18        "specialCat": "90% ΓΕΝΙΚΗ ΣΕΙΡΑ ΕΣΠΕΡΙΝΑ",
19        "positions": 2,
20        "baseFirst": 8733,
21        "baseLast": 8394,
22        "year": 2013
23      }
24    ]
25  }
26 ]

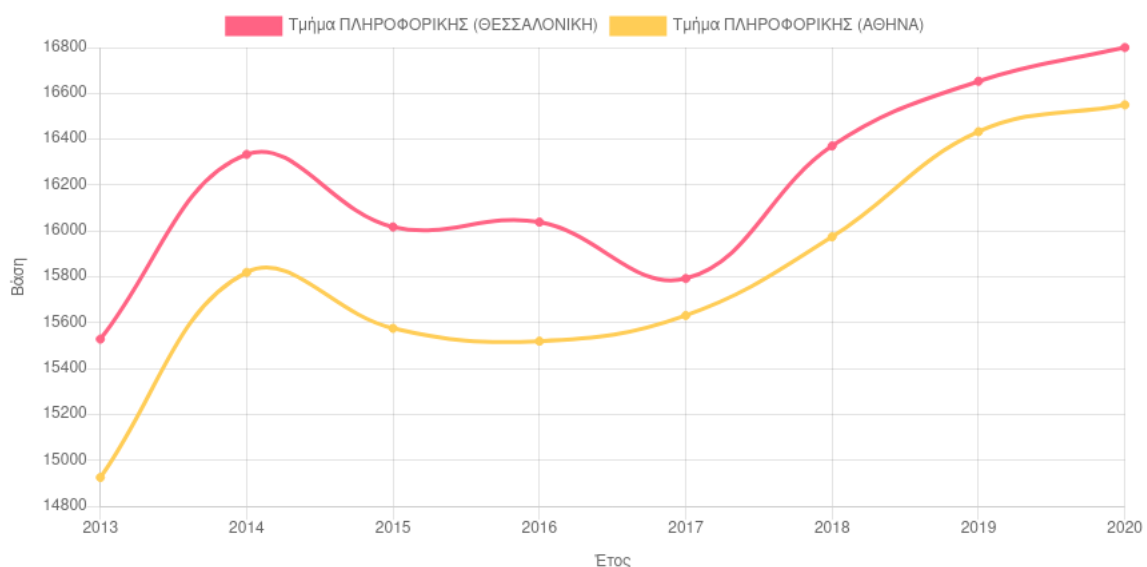
```

Σχήμα 2.6: Παράδειγμα δεδομένων σε μορφή JSON

σωστά τα δεδομένα με τον κατάλληλο τρόπο έτσι ώστε να εμφανίσει τις σωστές πληροφορίες στον χρήστη.

## 2.5.2 Chart.js

Η Chart.js είναι μία βιβλιοθήκη της JavaScript που βοηθά στην ευκολότερη απεικόνιση γραφημάτων στο διαδίκτυο. Δημιουργήθηκε το 2013, αποτελεί ανοιχτό λογισμικό και συντηρείται από την ομάδα της Chart.js και από συνεισφορές τρίτων. [24]



Σχήμα 2.7: Παράδειγμα γραφήματος που παράχθηκε από την Chart.js

Η βιβλιοθήκη δίνει τη δυνατότητα απεικόνισης 8 τύπων γραφημάτων και συνδυασμών αυτών. Επιπρόσθετα ο χρήστης μπορεί να παραμετροποιήσει την εμφάνιση των γραφημάτων και την

πηγή των δεδομένων κατά τη διάρκεια της εκτέλεσης της εφαρμογής.

Η Chart.js μπορεί να ενσωματωθεί σε μία ιστοσελίδα με την προσθήκη της ετικέτας HTML script, η οποία θα φορτώνει το αρχείο JavaScript της βιβλιοθήκης ή μέσω του npm και ενός φορτωτή JavaScript. [25] [26] Για τη δημιουργία ενός γραφήματος, η βιβλιοθήκη απαιτεί τη χρήση του Canvas της HTML ως βάση. Με τη χρήση της κλάσης Chart που περιλαμβάνει, αρχικοποιείται το διάγραμμα ως ένα αντικείμενο της JavaScript και περνώντας τις απαραίτητες παραμέτρους (π.χ. τύπος γραφήματος, τίτλοι αξόνων, σύνολο δεδομένων) ρυθμίζεται το τελικό διάγραμμα που θα παρουσιαστεί. [27]

```
002 let config = {
003     type: 'line',
004     data: {
005         labels: [],
006         datasets: []
007     },
008     options: {
009         maintainAspectRatio: true,
010         legend: {
011             display: true
012         },
013         responsive: true,
014         tooltips: {
015             mode: 'index',
016             intersect: false,
017         },
018         hover: {
019             mode: 'nearest',
020             intersect: true
021         },
022         scales: {
023             xAxes: [{
024                 display: true,
025                 scaleLabel: {
026                     display: true,
027                     labelString: 'Έτος'
```

Σχήμα 2.8: Παράδειγμα κώδικα για τη δημιουργία γραφήματος με τη χρήση της Chart.js

Η βιβλιοθήκη όντας ανοιχτού κώδικα, είναι εύκολα επεκτάσιμη με αποτέλεσμα να έχουν δημιουργηθεί διάφορα πρόσθετα για αυτή από τρίτους. Αυτά περιλαμβάνουν νέα διαγράμματα και επιπλέον ρυθμίσεις και επιλογές για τα ήδη υπάρχοντα. Επιπρόσθετα, είναι διαθέσιμη και για άλλες γλώσσες προγραμματισμού εκτός της JavaScript, αφού προγραμματιστές έχουν κατασκευάσει υλοποιήσεις της για αυτές. [28]

Η βιβλιοθήκη της JavaScript Chart.js χρησιμοποιήθηκε αποκλειστικά στην διαδικτυακή εφαρμογή που αναπτύχθηκε. Έδωσε τη δυνατότητα παρουσίασης ευανάγνωστων και αποκριτικών γραφημάτων, τα οποία ήταν εύκολα στη σχεδίαση, ενώ παράλληλα προσφέρουν πολλές δυνατότητες παραμετροποίησης.

### 2.5.3 Prism

Η Prism είναι μια βιβλιοθήκη της JavaScript, η οποία με τη βοήθεια της CSS, επιτρέπει την απεικόνιση κώδικα με επισημάνσεις. Είναι μια ανοιχτού κώδικα βιβλιοθήκη που δημιουργήθηκε το 2012 και χρησιμοποιείται από εκατομμύρια ιστοσελίδες παγκοσμίως (π.χ. Mozilla Development Network, React.js Documentation κ.α.). [29]

```

Μέρος Τυπικής Απάντησης
{
  "records": [
    {
      "id": 96087,
      "title": "ΔΙ.ΠΑ.Ε.",
      "full-title": "Διεθνές Πανεπιστήμιο της Ελλάδος"
    },
    {
      "id": 96118,
      "title": "ΔΠΘ",
      "full-title": "Δημοκρίτειο Πανεπιστήμιο Θράκης"
    }
  ]
}

```

Σχήμα 2.9: Παράδειγμα παρουσίασης κώδικα με τη χρήση της Prism

Η βιβλιοθήκη υποστηρίζει 249 διαφορετικές γλώσσες προγραμματισμού και δίνεται η δυνατότητα επέκτασης αυτών των γλωσσών μέσω προσθέτων. Ακόμα, υπάρχουν διαθέσιμα πρόσθετα που προσθέτουν επιπλέον δυνατότητες στην βιβλιοθήκη (π.χ. αριθμός γραμμών). [30] Για την ενσωμάτωση της Prism σε μια ιστοσελίδα, μπορούν να χρησιμοποιηθούν οι μέθοδοι που είναι διαθέσιμοι και στην Chart.js και αναλύθηκαν παραπάνω. Για την απεικόνιση κώδικα με επισήμανση σύνταξης, απαιτείται μια ετικέτα HTML pre στην οποία περιέχεται μία ετικέτα code, που θα έχει ως παράμετρο class τη γλώσσα που θα ακολουθήσει. Αυτή η παράμετρος πρέπει να είναι της μορφής language-x (π.χ. language-css). [31]

```

149 <pre><code class="language-json">{
150   "records": [
151     {
152       "id": 96087,
153       "title": "ΔΙ.ΠΑ.Ε.",
154       "full-title": "Διεθνές Πανεπιστήμιο της Ελλάδος"
155     },
156     {
157       "id": 96118,
158       "title": "ΔΠΘ",
159       "full-title": "Δημοκρίτειο Πανεπιστήμιο Θράκης"
160     }
161   ]
162 }
163 </code></pre>

```

Σχήμα 2.10: Παράδειγμα κώδικα για την υλοποίηση επισήμανσης σύνταξης με τη χρήση της Prism

Η βιβλιοθήκη Prism αξιοποιήθηκε κατά τη διάρκεια της ανάπτυξης του εγχειριδίου χρήσης του διαδικτυακού API. Επέτρεψε μία ευκολότερη και περισσότερο ευανάγνωστη παρουσίαση παραδειγμάτων χρήσης των συνδέσμων του API.

## 2.6 RESTful API

Το REST (Representation State Transfer) είναι μια αρχιτεκτονική σχεδίασης λογισμικού για τη δημιουργία διαδικτυακών εφαρμογών. Χρησιμοποιεί ένα υποσύνολο από τις υπάρχουσες μεθόδους του πρωτοκόλλου HTTP και υλοποιεί τις λειτουργίες της αρχιτεκτονικής CRUD. Το CRUD αποτελεί ακρωνύμιο των λέξεων “Create, Read, Update, Delete”.

Η αρχιτεκτονική του REST υλοποιεί το CRUD με τη χρήση τεσσάρων μεθόδων του HTTP. Με τη μέθοδο POST υλοποιείται το κομμάτι του “Create”, δηλαδή χρησιμοποιείται για την προσθήκη νέων δεδομένων. Η μέθοδος GET χρησιμοποιείται για το “Read” και αποτελεί μια ασφαλή μέθοδο, καθώς ο αποκλειστικός της σκοπός είναι η ανάγνωση δεδομένων και όχι η προσθήκη ή η επεξεργασία. Η μέθοδος PUT (μπορεί να γίνει χρήση της PATCH, αντί της PUT) υλοποιεί το “Update” και έχει σκοπό την ανανέωση κάποιων ήδη υπαρχόντων δεδομένων και η DELETE τη διαγραφή δεδομένων. [32]

Ένα API, για να μπορεί να θεωρηθεί RESTful θα πρέπει να πληρεί κάποια συγκεκριμένα κριτήρια που ορίζονται από την αρχιτεκτονική. Συγκεκριμένα θα πρέπει να υλοποιεί την αρχιτεκτονική CRUD, όπως αναφέρθηκε παραπάνω. Όλοι σύνδεσμοι του API, θα πρέπει να βασίζονται σε ένα αρχικό API και να προσφέρεται σε κάθε απόκριση του ο τύπος δεδομένων που επιστρέφεται. Ακόμα, κατά την επικοινωνία πελάτη εξυπηρετή θα η κατάσταση των πόρων αποθηκεύεται στον εξυπηρετή και είναι διαθέσιμη σε όλους τους πελάτες, ενώ η κατάσταση εφαρμογής αποθηκεύεται στον πελάτη και αφορά αποκλειστικά αυτόν. [33]

Η αρχιτεκτονική του REST ακολουθήθηκε εν μέρει κατά τη διάρκεια ανάπτυξης του διαδικτυακού API. Όπως θα αναλυθεί παρακάτω, το API υλοποιεί ένα μέρος του CRUD, χρησιμοποιώντας τη μέθοδο GET και πρόκειται για υλοποίηση χωρίς κατάσταση (stateless).

## 2.7 Επίλογος

Στο Κεφάλαιο 2 παρουσιάστηκαν οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του διαδικτυακού API και της εφαρμογής. Αναφέρθηκαν κάποιες πληροφορίες σχετικά με αυτές και ο τρόπος με τον οποίο αξιοποιήθηκαν στα πλαίσια της εκπόνησης αυτής της εργασίας.

## Κεφάλαιο 3ο: Backend και Διαδικτυακό API

### 3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση και ανάλυση του κομματιού σχετικό με το backend αυτής της εργασίας. Συγκεκριμένα θα παρουσιαστεί η βάση δεδομένων και ο σχεδιασμός της, η δημιουργία των αρχείων δεδομένων και η κατασκευή του ιστότοπου για τη μεταφόρτωση, επεξεργασία και αποθήκευση στη βάση δεδομένων. Ακόμα θα γίνει αναλυτική παρουσίαση του διαδικτυακού API, με παραδείγματα χρήσης του.

### 3.2 Σχεδιασμός Βάσης Δεδομένων

Η δομή των δεδομένων των Πανελλαδικών Εξετάσεων απαιτούσε μια σχεσιακή βάση δεδομένων, για αυτό το λόγο ως Σύστημα Διαχείρισης Βάσεων Δεδομένων επιλέχθηκε η mariaDB. Η βάση δεδομένων αποτελείται από έξι πίνακες, ο πίνακας university που περιέχει τα δεδομένα των τριτοβάθμιων ιδρυμάτων, ο πίνακας dept ο οποίος περιέχει τα δεδομένα των τμημάτων των τριτοβάθμιων ιδρυμάτων, ο πίνακας base που περιέχει τα δεδομένα των βάσεων των πανελλαδικών εξετάσεων, ο πίνακας examtype ο οποίος περιέχει τους τύπους εξετάσεων, ο πίνακας specialcat που περιέχει τις ειδικές κατηγορίες και ο statistics ο οποίος περιέχει τα δεδομένα των στατιστικών των πανελλαδικών εξετάσεων. Παρακάτω θα γίνει μία παρουσίαση της δομής κάθε πίνακα, καθώς και των σχέσεων μεταξύ αυτών.

Ο πίνακας university αποτελείται από τρία πεδία. Το πεδίο id, το οποίο είναι το κύριο κλειδί του πίνακα, αποτελεί την ταυτότητα του τριτοβάθμιου ιδρύματος και παράγεται αυτόματα από τη βάση δεδομένων. Το πεδίο title είναι το όνομα του ιδρύματος με τη μορφή που δίνεται στα αρχεία του Υπουργείου Παιδείας, πολλές φορές δίνεται με τη μορφή ακρωνυμίου. Το τελευταίο πεδίο είναι το full\_title το οποίο λύνει το πρόβλημα του προηγούμενο πεδίου, καθώς είναι πλήρεις. Το πεδίο full\_title συμπληρώθηκε χειροκίνητα από τον επιμελητή.

university	
id	int(255)
title	varchar(255)
full_title	varchar(500)

Σχήμα 3.1: Η δομή του πίνακα university

Ο πίνακας dept αποτελείται και αυτός από τρία πεδία. Το πεδίο code αντιστοιχεί στο μοναδικό κωδικό τμήματος που δίνεται από το Υπουργείο Παιδείας και είναι το κύριο κλειδί του πίνακα. Το πεδίο name περιλαμβάνει τα ονόματα των τμημάτων των ιδρυμάτων. Τέλος, το πεδίο uni\_id

αντιστοιχεί στον κωδικό του πανεπιστημίου που ανήκει κάθε τμήμα, αποτελεί ξένο κλειδί καθώς οι τιμές του uni\_id του πίνακα depts αντιστοιχούν στις τιμές του πεδίου id του πίνακα university.

dept	
<b>code</b>	int(255)
name	varchar(500)
uni_id	int(255)

Σχήμα 3.2: Η δομή του πίνακα dept

Ο πίνακας examtype αποτελείται από δύο πεδία. Το πεδίο title που περιλαμβάνει τον τίτλο των τύπων εξετάσεων (π.χ. 10% ΓΕΛ ΗΜΕΡΗΣΙΑ) και αποτελεί το κύριο κλειδί του πίνακα. Το δεύτερο πεδίο του πίνακα είναι το description, το οποίο περιλαμβάνει μια πιθανή επεξήγηση ή περιγραφή του τίτλου του τύπου εξέτασης. Το πεδίο description είναι προαιρετικό και συμπληρώνεται από τον επιμελητή.

examtype	
<b>title</b>	varchar(500)
description	varchar(5000)

Σχήμα 3.3: Η δομή του πίνακα examtype

Ο πίνακας specialcat αποτελείται επίσης από δύο πεδία. Το πεδίο code το οποίο αντιστοιχεί στο μοναδικό κωδικό τμήματος και αποτελεί ξένο κλειδί του πίνακα καθώς οι τιμές του αντιστοιχούν σε αυτές του πίνακα dept που αναφέρθηκε παραπάνω. Το δεύτερο πεδίο είναι το title το οποίο είναι ο τίτλος της ειδικής κατηγορίας (π.χ. ειδική κατηγορία για πολύτεκνους) και αποτελεί το κύριο κλειδί του πίνακα.

specialcat	
code	int(255)
<b>title</b>	varchar(500)

Σχήμα 3.4: Η δομή του πίνακα specialcat

Ο πίνακας base έχει 8 πεδία. Το πρώτο πεδίο είναι το code που όπως και προηγουμένως αντιστοιχεί στο μοναδικό κωδικό τμήματος και αποτελεί ξένο κλειδί του πίνακα dept. Το επόμενο είναι το title, το οποίο περιλαμβάνει στον τίτλο του είδους εξέτασης, αποτελεί ξένο κλειδί του πίνακα examtypes, καθώς οι τιμές αυτού του πεδίου αντιστοιχούν στο αντίστοιχο του πίνακα examtypes. Το πεδίο year είναι έτος που αντιστοιχεί η συγκεκριμένη βάση εισαγωγής. Τα τρία προηγούμενα πεδία δημιουργούν το κύριο κλειδί του πίνακα. Το πεδίο cat\_title αποτελεί αντίστοιχα με το title τον τίτλο της ειδικής κατηγορίας και είναι ξένο κλειδί του specialcat. Οι

θέσεις που είναι διαθέσιμες για κάθε κατηγορία αποθηκεύονται στο `positions`. Τα επιστημονικά πεδία στα οποία ανήκει κάθε τμήμα δεν περιλαμβάνεται σε όλα τα αρχεία που δημοσιοποιεί το Υπουργείο Παιδείας, για αυτό και το πεδίο `field` στο οποίο αποθηκεύεται αυτή η τιμή, επιτρέπει τις τιμές `null`. Τέλος, τα πεδία `vasiprotou` και `vasitel` αντιστοιχούν στη βάση εισαγωγής του πρώτου και του τελευταίου για κάθε περίπτωση.

base	
<b>code</b>	int(255)
<b>title</b>	varchar(500)
cat_title	varchar(500)
positions	int(255)
field	varchar(10)
<b>year</b>	int(4)
vasiprotou	int(5)
vasitel	int(5)

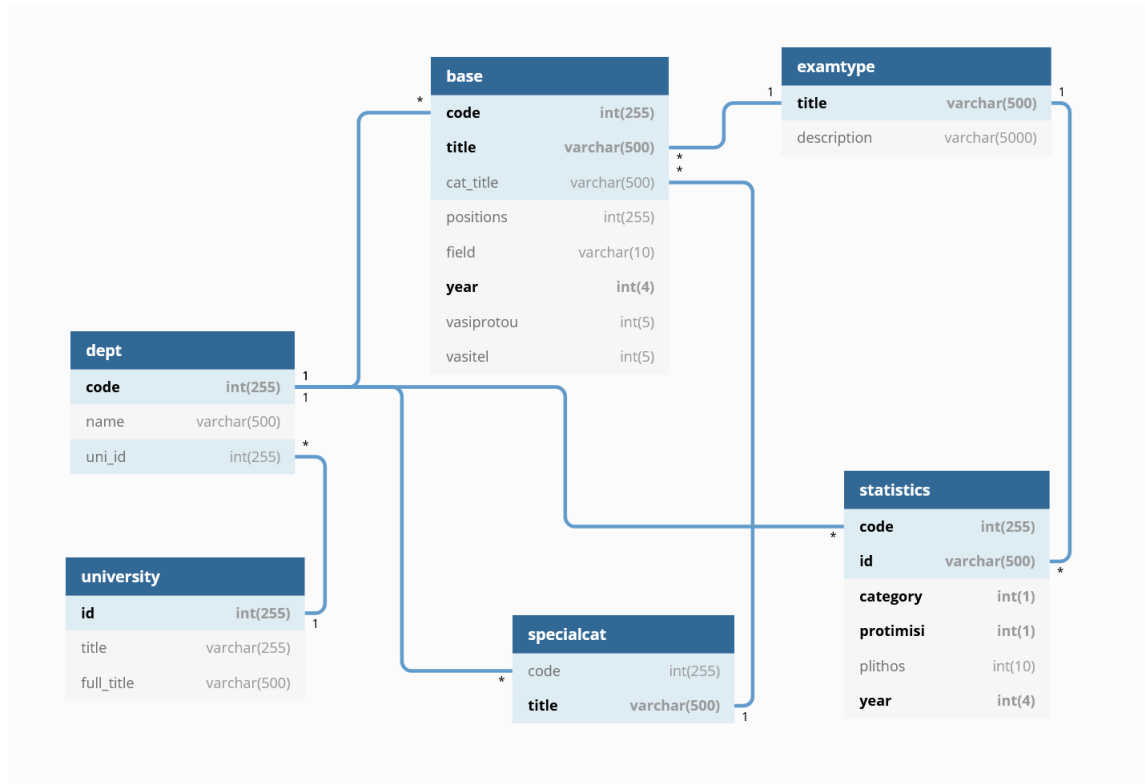
Σχήμα 3.5: Η δομή του πίνακα base

Ο τελευταίος πίνακας είναι ο `statistics` και αποτελείται από 6 πεδία. Τα πεδία `code` αποτελούν και `id` αποτελούν ξένα κλειδιά των πινάκων `dept` και `examtype` αντίστοιχα. Το πεδίο `category` μπορεί να πάρει τις τιμές μηδέν ή ένα και αναλογούν στους υποψήφιους και τους επιτυχόντες αντίστοιχα. Το πεδίο `protimisi` μπορεί να πάρει τιμές από ένα έως επτά (για τους υποψήφιους οι πιθανές τιμές είναι ένα έως τρία και επτά) και δηλώνουν τη σειρά προτίμηση ενός συγκεκριμένου τμήματος από τους υποψηφίους. Ο αριθμός επτά δηλώνει ότι η προτίμηση ήταν χαμηλότερη από τρίτη για τους υποψηφίους και χαμηλότερη από έκτη για τους επιτυχόντες. Το `year` όπως και στο `base` δηλώνει το έτος που αντιστοιχούν αυτά τα στατιστικά. Τα πεδία που αναφέρθηκαν δημιουργούν το κύριο κλειδί του πίνακα `statistics`. Το τελευταίο πεδίο είναι το `plithos`, το οποίο αντιστοιχεί στο πλήθος των υποψηφίων ή επιτυχόντων που είχαν δηλώσει στο μηχανογραφικό τους δελτίο, στη συγκεκριμένη σειρά προτίμησης κάποιο τμήμα.

statistics	
<b>code</b>	int(255)
<b>id</b>	varchar(500)
<b>category</b>	int(1)
<b>protimisi</b>	int(1)
plithos	int(10)
<b>year</b>	int(4)

Σχήμα 3.6: Η δομή του πίνακα statistics

Παρακάτω ακολουθεί ένα διάγραμμα οντοτήτων - συσχετίσεων σε UML (Unified Modeling Language) το οποίο παρουσιάζει με γραφικό τρόπο τις πληροφορίες που αναλύθηκαν παραπάνω. Οι γραμμές δείχνουν τη συσχέτιση των οντοτήτων, στις άκρες των οποίων ο αριθμός ένα και ο αστερίσκος μας δείχνουν τα όρια των συσχετίσεων τα οποία είναι ένα έως πολλά (π.χ. για ένα συγκεκριμένο code του πίνακα dept, πολλές γραμμές στον πίνακα base με αυτό το code).



Σχήμα 3.7: Το διάγραμμα οντοτήτων - συσχετίσεων της βάσης δεδομένων

### 3.3 Ιστότοπος Μεταφόρτωσης Αρχείων Δεδομένων

Με την ολοκλήρωση της σχεδίασης της βάσης δεδομένων, επόμενο βήμα ήταν η δημιουργία ενός ιστότοπου για τη μεταφόρτωση, επεξεργασία και αποθήκευση των δεδομένων των Πανελλαδικών Εξετάσεων. Αυτό το έργο χωρίστηκε σε δύο βήματα, το πρώτο ήταν η επιλογή της μορφής των δεδομένων και η επεξεργασία αυτών προτού μεταφορτωθούν στην ιστοσελίδα και το δεύτερο ήταν η υλοποίηση του αλγορίθμου που θα υλοποιεί τις ενέργειες που αναλύθηκαν παραπάνω.

Η αρχική πρόταση ήταν τα δεδομένα να δίνονται αυτούσια, δηλαδή χωρίς καμία επεξεργασία των αρχείων του Υπουργείου Παιδείας και ο χρήστης που είναι υπεύθυνος για τη μεταφόρτωση να πρέπει να ορίσει κάποιες παραμέτρους στην ιστοσελίδα για τη σωστή επεξεργασία τους. Αυτή η ιδέα εν τέλει απορρίφθηκε καθώς κρίθηκε επικίνδυνο το ενδεχόμενο να δοθεί από λάθος ή απροσεξία μία μη ορθή παράμετρος, κάτι που θα είχε ως αποτέλεσμα είτε να εμφανιστεί ένα σφάλμα είτε στο χειρότερο σενάριο να αποθηκευτούν στη βάση δεδομένων δεδομένα σε λάθος

πίνακες ή πεδία.

Η υλοποίηση που τελικά επιλέχθηκε ήταν ότι για κάθε έτος θα πρέπει να δημιουργηθούν δύο αρχεία, ένα για τα δεδομένα των βάσεων και ένα για τα δεδομένα των στατιστικών των Πανελλαδικών Εξετάσεων, τα οποία θα περιέχουν το σύνολο όλων των δεδομένων για ένα έτος. Τα αρχεία αυτά θα είναι τύπου csv (Comma Separated Values file) και η ονομασία τους θα πρέπει να ακολουθεί αυστηρά τη μορφή “vaseis.έτος.csv” και “stats.έτος.csv” για τις βάσεις και τα στατιστικά αντίστοιχα. Με αυτό τον τρόπο γίνεται η διαφοροποίηση μεταξύ των δύο τύπων δεδομένων και εξάγεται το έτος που αναφέρονται τα αρχεία. Επίσης, και η δομή αυτών των αρχείων θα πρέπει να ακολουθεί αυστηρή δομή. Οι στήλες που θα περιλαμβάνονται στο αρχείο των βάσεων θα πρέπει να είναι συγκεκριμένες και να ακολουθούν τη σειρά που έχει οριστεί, αυτές είναι ο κωδικός του τμήματος, το πανεπιστήμιο, το όνομα του τμήματος, η ειδική κατηγορία, το επιστημονικό πεδίο (αν λείπει θα πρέπει η στήλη να είναι κενή), οι διαθέσιμες θέσεις, η βάση εισαγωγής του πρώτου και τελευταίου επιτυχόντα και ο τύπος εξέτασης. Για το αρχείο των στατιστικών αντίστοιχα οι στήλες είναι ο κωδικός του τμήματος, επτά στήλες για την αντίστοιχη σειρά προτίμησης (για τους υποψηφίους οι θέσεις τέσσερα, πέντε και έξι μένουν κενές), ο τύπος εξέτασης και η κατηγορία των στατιστικών (ένα για τους επιτυχόντες, μηδέν για τους υποψήφιους).

Έπειτα από τον σχεδιασμό των αρχείων δεδομένων, κατασκευάστηκε ένας ιστότοπος που έχει ως σκοπό τη μεταφόρτωση των αρχείων δεδομένων, την επεξεργασία τους και την αποθήκευση των δεδομένων στη βάση δεδομένων. Ο ιστότοπος χωρίζεται σε δύο μέρη, το πρώτο είναι μία απλή φόρμα μεταφόρτωσης αρχείων και το δεύτερο κάποιες βασικές οδηγίες για τη μορφή των αρχείων. Ο χρήστης έχει τη δυνατότητα να μεταφορτώσει ένα ή δύο αρχεία τη φορά, δίχως να χρειάζεται υποχρεωτικά να είναι δεδομένα του ίδιου έτους, αρκεί να ακολουθηθεί η σωστή μορφή στην ονομασία των αρχείων.

## Εισαγωγή Βάσεων και Στατιστικών Πανελλαδικών Εξετάσεων

Το αρχείο των βάσεων θα πρέπει να είναι της μορφής vaseis.{έτος}.csv και των στατιστικών stats.{έτος}.csv

### Αρχείο Βάσεων

No file selected.

### Αρχείο Στατιστικών

No file selected.

### Μορφή Αρχείων

#### Αρχείο Βάσεων

Το csv αρχείο θα πρέπει να είναι της μορφής code,uni,dept,specialcat,scientificField,theseis,vasiprotou,vasitel,examtype

#### Αρχείο Στατιστικών

Το csv αρχείο θα πρέπει να της μορφής code, protimisi1,protimisi2,protimisi3,protimisi4,protimisi5,protimisi6,promisiAllo,examtype,category(0/1)

Για τα στατιστικά των υποψηφίων τα protimisi4,protimisi5,protimisi6 μένουν κένα.

Σχήμα 3.8: Ο ιστότοπος μεταφόρτωσης των αρχείων δεδομένων

Επιλέγοντας το κουμπί “Υποβολή” εκκινείται ο αλγόριθμος υπεύθυνος για την επεξεργασία των δεδομένων. Ύστερα από τον έλεγχο του αν έχουν μεταφορτωθεί αρχεία και αν τηρούν τη δομή που χρειάζεται στον τίτλο τους, εξάγεται το έτος που αφορούν τα δεδομένα αυτά. Στη συνέχεια ξεκινώντας από το αρχείο των βάσεων εισαγωγής, για κάθε γραμμή του αρχείου ελέγχεται η ύπαρξη κόμματος στους τιμές των βάσεων εισαγωγής και αφαιρείται για τη σωστή επεξεργασία τους και ύστερα καλείται η συνάρτηση υπεύθυνη για την αποθήκευση των δεδομένων στη βάση δεδομένων.

```

if (isset($fileName["vaseis"]))
{
    if($handle = fopen($_FILES["vaseis"]["tmp_name"], mode: 'r'))
    {
        $error = false;
        $conn->begin_transaction();
        while ($data = fgetcsv($handle))
        {
            //fix bases number, by removing comma
            if (strpos($data[6], needle: ',') || strpos($data[7], needle: ','))
            {
                $data[6] = str_replace( search: ',', replace: '', $data[6]);
                $data[7] = str_replace( search: ',', replace: '', $data[7]);
            }

            $error = uploadVaseis($conn, $data, $year["vaseis"]);
        }
        $conn->commit();
        if ($error) {
            echo 'Υπήρξε σφάλμα, το αρχείο των βάσεων δεν ανέβηκε';
        } else {
            echo 'Το αρχείο των βάσεων ανέβηκε.';
        }
    }
}

```

Σχήμα 3.9: Ο αλγόριθμος επεξεργασίας δεδομένων

Η συνάρτηση καλεί μια σειρά από εντολές, οι οποίες εκτελούν τα κατάλληλα ερωτήματα SQL. Ξεκινώντας από το πανεπιστήμιο, γίνεται έλεγχος αν υπάρχει ήδη και αν όχι τότε προστίθεται στον πίνακα university. Αντίστοιχα ελέγχεται για τα τμήματα των πανεπιστημίων, αν υπάρχει άλλη εγγραφή με τον ίδιο κωδικό και αν όχι αποθηκεύεται προσθέτοντας και την αυτόματα δημιουργημένη ταυτότητα πανεπιστημίου από το προηγούμενο ερώτημα. Την ίδια λογική ελέγχου ακολουθούν και τα ερωτήματα για τους τύπους εξετάσεων και τις ειδικές κατηγορίες και τέλος προστίθενται τα ανάλογα δεδομένα στον πίνακα base χωρίς κάποιο περαιτέρω έλεγχο.

Ο αντίστοιχος αλγόριθμος εκτελείται και για το αρχείο δεδομένων στατιστικών. Σε κάθε γραμμή του αρχείου εκτελείται ένα βρόχος επανάληψης για κάθε στήλη των σειρών προτίμησης όπου και εξάγονται οι τιμές και εκτελείται ένα ερώτημα SQL για την αποθήκευση των δεδομένων. Αυτό έχει ως αποτέλεσμα για κάθε γραμμή του αρχείου να δημιουργούνται 7 νέες γραμμές στον πίνακα statistics. Αξίζει να σημειωθεί, ότι από τα αρχεία στατιστικών προστίθενται επιπλέον

τύποι εξετάσεων, διότι τα δεδομένα που δίνονται συνήθως είναι το σύνολο των ημερήσιων και εσπερινών λυκείων.

```

123 function uploadVaseis($conn, $data, $year)
124 {
125     try {
126         $stmt = $conn->prepare("INSERT INTO university (`title`) VALUES (?) ON DUPLICATE KEY UPDATE title=?");
127         $stmt->bind_param('ss', $data[1], $data[1]);
128         $stmt->execute();
129
130         $stmt = $conn->prepare("INSERT INTO dept (`code`, `name`, `uni_id`)
131             VALUES (?,?,(SELECT id FROM university WHERE title=?))
132             ON DUPLICATE KEY UPDATE code=?");
133         $stmt->bind_param('ssss', $data[0], $data[2], $data[1], $data[0]);
134         $stmt->execute();
135
136         $stmt = $conn->prepare("INSERT INTO `examtype` (`title`)
137             VALUES (?) ON DUPLICATE KEY UPDATE `title`=?");
138         $stmt->bind_param('ss', $data[8], $data[8]);
139         $stmt->execute();
140
141         $stmt = $conn->prepare("INSERT INTO specialcat (`code`, `title`)
142             VALUES (?,?) ON DUPLICATE KEY UPDATE title=?");
143         $stmt->bind_param('sss', $data[0], $data[3], $data[3]);
144         $stmt->execute();
145
146         $stmt = $conn->prepare("INSERT INTO base
147             (`code`, `title`, `cat_title`, `positions`, `field`, `year`, `vasiprotou`, `vasitel`)
148             VALUES (?,?,,?,?,,?,?)");
149         $stmt->bind_param('ssssssss', $data[0], $data[8], $data[3], $data[5], $data[4], $year, $data[6], $data[7]);
150         $stmt->execute();
151         return false;
152     } catch (mysqli_sql_exception $e) {
153         $conn->rollback();
154         echo $e->getMessage();
155         return true;
156     }
157 }

```

Σχήμα 3.10: Η αλληλουχία εκτέλεσης των ερωτημάτων SQL για το αρχείο δεδομένων των βάσεων εισαγωγής

### 3.4 Διαδικτυακό API

Με τα βήματα που περιγράφηκαν παραπάνω, έχει επιτευχθεί ο στόχος της οργάνωσης και συγκέντρωσης των δεδομένων των Πανελλαδικών Εξετάσεων σε μία πηγή. Το επόμενο βήμα που επιλέχθηκε ήταν η κατασκευή ενός διαδικτυακού API, έτσι ώστε να δοθεί η δυνατότητα σε οποιοδήποτε προγραμματιστή ή ερευνητή να εξάγει όσα και όποια δεδομένα επιθυμεί για τη κατασκευή μίας εφαρμογής ή για την υλοποίηση μίας έρευνας.

Το διαδικτυακό API έχει προγραμματιστεί έτσι ώστε να επιστρέφει τα δεδομένα σε μορφή JSON, καθώς θεωρήθηκε ότι πρόκειται για ένας από τους πιο διαδεδομένους τύπους μορφοποίησης δεδομένων και υποστηρίζεται από τις περισσότερες γλώσσες προγραμματισμού. Ακόμη οι απαντήσεις στα ερωτήματα που θα γίνονται σε αυτό, θα περιλαμβάνουν την κατάλληλη κατάσταση HTTP, οι πιθανές καταστάσεις είναι η 200 αν το ερώτημα είναι εντάξει και επιστραφούν δεδομένα, η 400 αν είναι λανθασμένη η σύνταξη του ερωτήματος, η 404 αν το ερώτημα είναι εντάξει αλλά δεν υπάρχουν δεδομένα να επιστραφούν, η 405 αν γίνει προσπάθεια χρήσης ενός HTTP ρήματος που δεν υποστηρίζεται και η 500 σε περίπτωση που προκύψει ένα σφάλμα στον διακομιστή. Το HTTP ρήματα που επιτρέπονται είναι τα HEAD, OPTIONS και GET, διότι το API υποστηρίζει μόνο τη λήψη δεδομένων.

Για κάθε ερώτημα που γίνεται στο API, καλείται ο χειριστής (handler), ο ρόλος του οποίου είναι να αναγνωρίσει τον τύπο των δεδομένων που αναζητά ο χρήστης και να ανακατευθύνει το αίτημα στον ανάλογο ελεγκτή (controller). Αυτό γίνεται εξετάζοντας το URI που του δίνεται

και ελέγχοντας τη θέση που αντιστοιχεί η κατηγορία δεδομένων. Στο ενδεχόμενο που δε δίνεται μία τέτοια κατηγορία επιστρέφεται η κατάσταση 400, ενώ αν κατηγορία δεν αντιστοιχεί σε μία από τις υποστηριζόμενες επιστρέφει την κατάσταση 405. Οι υποστηριζόμενες κατηγορίες είναι ουσιαστικά οι ίδιες με τους πίνακες που αποτελούν τη βάση δεδομένων.

Εφόσον το ερώτημα πληροί τις παραπάνω προϋποθέσεις το αίτημα ανακατευθύνεται στο αντίστοιχο ελεγκτή της κατηγορίας. Σε εκείνο το σημείο γίνεται περαιτέρω έλεγχος του URI, για να διερευνηθεί η σωστή σύνταξη του με βάση τα κριτήρια του ελεγκτή. Δηλαδή ελέγχεται ο αριθμός των μερών του URI και το περιεχόμενό τους. Στην περίπτωση που εντοπιστεί κάποιο λάθος στη δομή επιστρέφεται η κατάσταση 400, εφόσον ο συνδυασμός αυτών είναι επιτρεπτός καλείται μία συνάρτηση η οποία αρχικοποιεί το αντικείμενο της συγκεκριμένης κατηγορίας και καλεί τη μέθοδο που αντιστοιχεί στον συνδυασμό αυτό. Αυτή η μέθοδος εκτελεί ένα ερώτημα SQL με τις παραμέτρους που δόθηκαν και επιστρέφει στη συνάρτηση ένα αντικείμενο τύπου “mysqli\_stmt“ το οποίο περιέχει τα δεδομένα. Ύστερα καλείται μία συνάρτηση η οποία διαβάζει τα δεδομένα από το αντικείμενο και μέσω ενός βρόχου επανάληψης τα αποθηκεύει με μια συγκεκριμένη δομή που έχει αποφασιστεί σε έναν πίνακα, ο οποίος και επιστρέφεται στην αρχική συνάρτηση. Τέλος, εφόσον δεν προέκυψε κάποιο σφάλμα, ο πίνακας κωδικοποιείται σε μορφή JSON και επιστρέφεται στον χρήστη που έκανε το αίτημα.

```

14 function getDeptResults($uri) {
15     if (isset($uri[5])) http400();
16     if (isset($uri[4])) {
17         if ($uri[3] == "university") getDepartmentsByUni($uri[4]);
18         else http400();
19     }
20     elseif (isset($uri[3])) {
21         if (isset($_GET)) {
22             getDepartments();
23         } else {
24             getDepartment($uri[3]);
25         }
26     }
27     else getDepartments();
28 }

```

Σχήμα 3.11: Μέρος του εγχειριδίου οδηγιών του διαδικτυακού API.

Έκτος των υποχρεωτικών παραμέτρων, το API υποστηρίζει και κάποιες προαιρετικές παραμέτρους. Εφόσον δοθούν αυτές οι παράμετροι, ο αλγόριθμος αφού ελέγξει αν πράγματι υποστηρίζονται θα προχωρήσει σε παραμετροποίηση της δομής των δεδομένων ή θα επιστρέψει ένα υποσύνολο των δεδομένων που ζητήθηκαν. Παραδείγματα προαιρετικών παραμέτρων θα παρουσιαστούν παρακάτω.

Στα πλαίσια της υλοποίησης του API και αφού ολοκληρώθηκαν όλες οι βασικές κλήσεις του, αποφασίστηκε να γίνει προσθήκη νέων ερωτημάτων, για λειτουργίες που είχαν ήδη υλοποιηθεί. Αυτά τα ερωτήματα δίνουν περισσότερη έμφαση στην οργάνωση και δομή των δεδομένων που επιστρέφονται, έτσι ώστε να είναι ευκολότερο να τα διαχειριστεί ένας προγραμματιστής. Σε αυτά, δόθηκε η κωδική ονομασία “1.0“, καθώς θεωρήθηκε ότι πρόκειται για μια πιο ώριμη προσέγγιση του API, που παύει να είναι σε πειραματικό στάδιο.

```

3 function getResults($stmt): array
4 {
5     $result = $stmt->get_result();
6     $num = $result->num_rows;
7
8     if ($num>0) {
9         $deptArray = array();
10        if ($_GET["details"] == "full") {
11            while ($row = $result->fetch_assoc()) {
12                extract($row);
13                $deptItem = array(
14                    "code" => $code,
15                    "name" => $name,
16                    "uni-id" => $uni_id,
17                    "title" => $title,
18                    "fullTitle" => $full_title
19                );
20                array_push( &array: $deptArray, $deptItem);
21            }
22        } else {
23            while ($row = $result->fetch_assoc()) {
24                extract($row);
25                $deptItem = array(
26                    "code" => $code,
27                    "name" => $name,
28                    "uni-id" => $uni_id,
29                );
30                array_push( &array: $deptArray, $deptItem);
31            }
32        }
33        http200();
34        return $deptArray;
35    } else {
36        return http404();
37    }
38 }

```

Σχήμα 3.12: Αλγόριθμος μορφοποίησης δεδομένων.

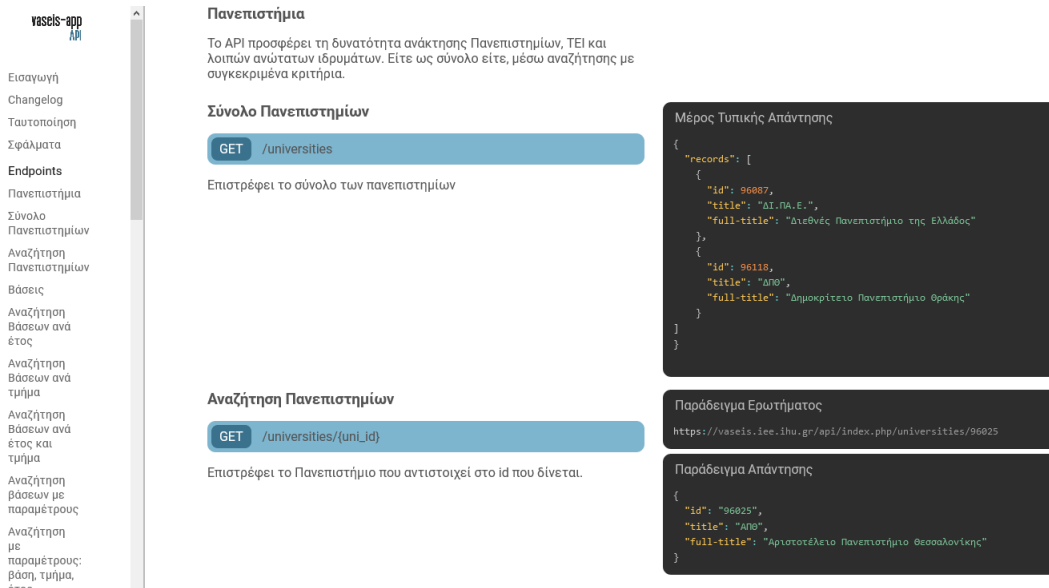
Η πολυπλοκότητα κάποιων ερωτημάτων, αλλά κυρίως το πλήθος αυτών οδήγησε στην απόφαση δημιουργίας ενός ιστοτόπου ο οποίος θα λειτουργεί ως εγχειρίδιο χρήσης του API.<sup>1</sup> Ο ιστοτόπος περιλαμβάνει κάποιες βασικές πληροφορίες για το API και στη συνέχεια μία λίστα με όλα τα αιτήματα που μπορεί να κάνει κάποιος στο API. Για κάθε αίτημα δίνεται ένας τίτλος, η δομή του ερωτήματος, μια περιγραφή και ένα παράδειγμα πραγματικού ερωτήματος με την απάντηση του API που του αντιστοιχεί. Αν κάποιο ερώτημα υποστηρίζει προαιρετικές παραμέτρους, γίνεται αναφορά και σε αυτές και δίνεται μία περιγραφή τους.

### 3.5 Παραδείγματα Κλήσεων του API

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση κάποιων επιλεγμένων κλήσεων του API. Θα παρουσιαστεί ο σύνδεσμος που απαιτείται για την λήψη των δεδομένων και δομή αυτών. Στα παραδείγματα παρακάτω δίνεται μόνο το μέρος του συνδέσμου που αφορούν την κάθε κλήση. Το πρώτο μέρος της συνδέσμου παραμένει σταθερό και ονομάζεται αρχικός σύνδεσμος.<sup>2</sup>

<sup>1</sup>Το εγχειρίδιο χρήσης του API είναι διαθέσιμο στη διεύθυνση <https://www.vaseis.iee.ihu.gr/api>

<sup>2</sup>Ο αρχικός σύνδεσμος του API είναι ο <https://vaseis.iee.ihu.gr/api/index.php>



Σχήμα 3.13: Μέρος του εγχειριδίου οδηγιών του διαδικτυακού API.

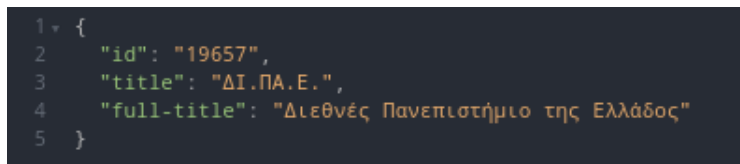
### 3.5.1 Αναζήτηση Πανεπιστημίων με χρήση μοναδικού κωδικού

Το API δίνει τη δυνατότητα λήψης πληροφοριών όλων των πανεπιστημίων, δίνεται όμως και η δυνατότητα αναζήτησης κάποιου συγκεκριμένου, με τη χρήση του μοναδικού κωδικού του. Στο Σχήμα 3.14 δίνεται η δομή του συνδέσμου για να εκτελεστεί η κλήση.



Σχήμα 3.14: Η δομή του συνδέσμου για την αναζήτηση πανεπιστημίων.

Για τη χρήση του απαιτείται η αντικατάστασή του “uni\_id” με έναν κωδικό που αντιστοιχεί σε πανεπιστήμιο. Στο σχήμα 3.15 χρησιμοποιήθηκε ως παράδειγμα ο αριθμός 19657, που αντιστοιχεί στο Διεθνές Πανεπιστήμιο της Ελλάδος. Το API επιστρέφει τον κωδικό του πανεπιστημίου, τη σύντομη και πλήρη ονομασία του Πανεπιστημίου.



Σχήμα 3.15: Παράδειγμα αναζήτησης Πανεπιστημίου.

### 3.5.2 Αναζήτηση βάσεων

Η αναζήτηση βάσεων εισαγωγής θεωρήθηκε ως το σημαντικότερο μέρος του API, για αυτό και δόθηκε ιδιαίτερη σημασία στο να καλυφθούν οι περισσότερες περιπτώσεις. Δίνεται η δυνατότητα αναζήτησης του συνόλου των βάσεων για ένα συγκεκριμένο έτος, για ένα συγκεκριμένο

τμήμα ή και συνδυασμούς αυτών. Στο Σχήμα 3.16 παρουσιάζεται η δομή του αιτήματος ανάκτησης δεδομένων των βάσεων εισαγωγής με κριτήρια το έτος και το τμήμα.

GET /bases/{year}/dept/{dept\_id}

Σχήμα 3.16: Παράδειγμα αναζήτησης βάσεων εισαγωγής με κριτήρια το έτος και το τμήμα.

Προκειμένου να γίνει χρήση του θα πρέπει να συμπληρωθούν οι τιμές “year“ και “dept\_id“ με το έτος και τον κωδικό του τμήματος αντίστοιχα. Ως παράδειγμα, δόθηκε η τιμή 2020 για το έτος και 1625 για τον κωδικό τμήματος, τα οποία αντιστοιχεί στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΠΠΑΕ. Στο Σχήμα 3.17, φαίνεται ένα τμήμα των αποτελεσμάτων που προέκυψαν. Τα δεδομένα περιλαμβάνουν τον κωδικό του τμήματος, τον τύπο εξέτασης και την ειδική κατηγορία, τις διαθέσιμες θέσεις, τη βάση εισαγωγής του πρώτου και του τελευταίου επιτυχόντα και το έτος το οποίο αφορούν αυτά τα δεδομένα.

```

48  {
49    "code": 1625,
50    "examType": "ΓΕΛ ΝΕΟ ΗΜΕΡΗΣΙΑ",
51    "specialCat": "ΓΕΛ ΓΕΝΙΚΗ ΣΕΙΡΑ ΗΜ. (ΝΕΟ)",
52    "positions": 145,
53    "baseFirst": 16125,
54    "baseLast": 13325,
55    "year": 2020
56  },
57  {
58    "code": 1625,
59    "examType": "ΓΕΛ ΠΑΛΑΙΟ ΗΜΕΡΗΣΙΑ",
60    "specialCat": "ΓΕΛ ΓΕΝΙΚΗ ΣΕΙΡΑ ΗΜ. (ΠΑΛΑΙΟ)",
61    "positions": 15,
62    "baseFirst": 12774,
63    "baseLast": 8569,
64    "year": 2020
65  },
66  {
67    "code": 1625,
68    "examType": "ΕΠΑΛ ΕΣΠΕΡΙΝΑ",
69    "specialCat": "ΕΠΑΛ ΓΕΝΙΚΗ ΣΕΙΡΑ ΕΣΠ.",
70    "positions": 2,
71    "baseFirst": 14865,
72    "baseLast": 14505,
73    "year": 2020
74  },

```

Σχήμα 3.17: Τμήμα των αποτελεσμάτων από την εκτέλεση της κλήσης του API.

### 3.5.3 Αναζήτηση βάσεων εισαγωγής με χρήση φίλτρων

Με τα ερωτήματα που αναφέρθηκαν παραπάνω είναι δυνατή η ανάκτηση όλων των βάσεων εισαγωγής που υπάρχουν διαθέσιμες. Όμως εύκολα γίνεται αντιληπτό, ότι η παροχή όλων των δεδομένων δίχως περιορισμούς μπορεί να δημιουργήσει δυσκολία στον προγραμματιστή ως προς την επεξεργασία.

Τα δεδομένα περιλαμβάνουν βάσεις εισαγωγής για όλους του τύπους εξετάσεων για τα Γενικά και Επαγγελματικά Λύκεια. Τα φίλτρα που θεωρήθηκαν τα σημαντικότερα, ήταν αυτά που θα έκαναν δυνατό τον διαχωρισμό των δεδομένων για τους τύπους εξετάσεων 10% και 90%, τα Γενικά και Επαγγελματικά Λύκεια, καθώς και τα ημερήσια και εσπερινά Λύκεια.

Επιπλέον, δημιουργήθηκε ένα ακόμα φίλτρο, σκοπός του οποίου είναι στα δεδομένα να περιλαμβάνονται το τμήμα και το τριτοβάθμιο ίδρυμα που αντιστοιχούν τα δεδομένα που επιστρέφονται. Αυτό, δημιουργήθηκε με σκοπό τη χρησιμοποίηση του σε συνδυασμό με τα παραπάνω φίλτρα. Στόχος αυτού του φίλτρου είναι η αποτροπή των χρηστών από τις επιπλέον κλήσεις στο API για τη λήψη των δεδομένων των Πανεπιστημίων και τμημάτων. Στο Σχήμα 3.18, μέσω ενός στιγμιότυπου από το εγχειρίδιο χρήσης του API, γίνεται μια προσπάθεια σύντομης παρουσίασης των φίλτρων.

- type=gel-ime-gen  
Επιστρέφει τις βάσεις που αντιστοιχούν στα αποτελέσματα σχετικά με την κατηγορία 90% ημερήσιων ΓΕΛ
- type=epal-ime-gen  
Επιστρέφει τις βάσεις που αντιστοιχούν στα αποτελέσματα σχετικά με την κατηγορία 90% ημερήσιων ΕΠΑΛ.
- type=gel-ime-ten  
Επιστρέφει τις βάσεις που αντιστοιχούν στα αποτελέσματα σχετικά με την κατηγορία 10% ημερήσιων ΓΕΛ
- type=epal-ime-ten  
Επιστρέφει τις βάσεις που αντιστοιχούν στα αποτελέσματα σχετικά με την κατηγορία 10% ημερήσιων ΕΠΑΛ
- details=full  
Με αυτό το φίλτρο για κάθε βάση επιστρέφεται και το όνομα του πανεπιστημίου στο οποίο ανήκει το κάθε τμήμα.

Σχήμα 3.18: Στιγμιότυπο από το εγχειρίδιο χρήσης του API, που εξηγούνται τα φίλτρα.

Η δομή μίας κλήσης στο API με τη χρήση φίλτρων δεν απαιτεί ιδιαίτερη εξήγηση, καθώς πρόκειται για την ίδια δομή που ακολουθείται στα προηγούμενα ερωτήματα. Η προσθήκη των φίλτρων γίνεται στο τέλος του συνδέσμου και προστίθενται ως παράμετροι τύπου GET.

```
https://vaseis.iee.ihu.gr/api/index.php/bases/2020/department/1625?type=gel-ime-gen&details=full
```

Σχήμα 3.19: Παράδειγμα συνδέσμου με χρήση φίλτρων.

### 3.5.4 Αναζήτηση βάσεων εισαγωγής με χρήση κριτηρίων

Μία ακόμη σημαντική μέθοδος αναζήτησης των βάσεων εισαγωγής θεωρήθηκε πως ήταν η χρήση κριτηρίων. Για αυτό το λόγο το API δίνει τη δυνατότητα αναζήτησης με την παροχή της βάσης εισαγωγής, ενός όρου αναζήτησης, του έτους ή και συνδυασμών αυτών.

Με αυτόν τον τρόπο δίνεται η δυνατότητα αναζήτησης τμημάτων τα οποία έχουν κοινή ονομασία, οπότε μπορεί να γίνει και ομαδοποίηση των δεδομένων με ευκολία. Ακόμα η προσθήκη ορίου για τη βάση εισαγωγής επιτρέπει στον προγραμματιστή να λαμβάνει δεδομένα τα οποία χρειάζονται λιγότερη επεξεργασία και στον διακομιστή στον οποίο φιλοξενείται το API να μεταδίδει μικρότερο όγκο πληροφοριών. Στο Σχήμα 3.20 παρουσιάζεται η δομή που απαιτείται για ένα τέτοιο ερώτημα. Στο πεδίο “base“ θα πρέπει να συμπληρωθεί η μέγιστη βάση εισαγωγής, στο “search\_query“ ο όρος αναζήτησης και στο “year“ το έτος που θα αντιστοιχούν τα δεδομένα.

```
GET /bases/search/?base={base}&department={search_query}&
year={year}&=details=full
```

Σχήμα 3.20: Η δομή αναζήτησης βάσεων εισαγωγής με χρήση κριτηρίων (μέγιστη βάση εισαγωγής, όρος αναζήτησης, έτος).

Στο παράδειγμα κλήσης του API στο Σχήμα 3.21 δόθηκαν ως κριτήρια ο αριθμός “17000” ως μέγιστη βάσης εισαγωγής, το “ιατρική” ως όρος αναζήτησης και το 2020 ως έτος. Με αυτά τα δεδομένα η αναζήτηση του API επιστρέφει δύο τμήματα που αντιστοιχούν στα κριτήρια που δόθηκαν.

```
3+  {
4    "code": 369,
5    "examType": "ΠΑΝ. ΘΕΣΣΑΛΙΑΣ",
6    "specialCat": "ΓΕΛ ΓΕΝΙΚΗ ΣΕΙΡΑ ΗΜ.",
7    "positions": 162,
8    "baseFirst": 17653,
9    "baseLast": 14815,
10   "year": 2018,
11   "deptName": "ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΪΑΤΡΙΚΗ (ΛΑΜΙΑ)",
12   "uniTitle": "Πανεπιστήμιο Θεσσαλίας"
13  },
14+  {
15   "code": 388,
16   "examType": "ΠΑΝ. ΔΥΤ. ΑΤΤΙΚΗΣ",
17   "specialCat": "ΓΕΛ ΓΕΝΙΚΗ ΣΕΙΡΑ ΗΜ.",
18   "positions": 91,
19   "baseFirst": 17428,
20   "baseLast": 14930,
21   "year": 2018,
22   "deptName": "ΜΗΧΑΝΙΚΩΝ ΒΙΟΪΑΤΡΙΚΗΣ (ΑΙΓΑΛΕΩ)",
23   "uniTitle": "Πανεπιστήμιο Δυτικής Αττικής"
24  }
```

Σχήμα 3.21: Παράδειγμα απόκρισης του API μετά από αναζήτηση βάσεων εισαγωγής με χρήση κριτηρίων.

### 3.5.5 Αναζήτηση Βάσεων με ερωτήματα της έκδοσης 1.0

Όπως προαναφέρθηκε η κατηγορία των βάσεων εισαγωγής θεωρήθηκε το σημαντικότερο κομμάτι του API, για αυτό το λόγο τα πρώτα ερωτήματα που δημιουργήθηκαν στην έκδοση 1.0 ήταν αυτά των βάσεων εισαγωγής. Η σύνταξη του συνδέσμου κλήσης παραμένει σχεδόν ο ίδιος και συνήθως η μόνη προσθήκη είναι του “v1.0” στην αρχή του συνδέσμου.

Τα ερωτήματα που έχουν δημιουργηθεί για τις βάσεις εισαγωγής μέχρι και τη στιγμή της συγγραφής αυτής της εργασίας είναι η αναζήτηση βάσεων με μόνο κριτήριο το τμήμα και οι αναζήτηση βάσεων με κριτήριο τους κωδικούς ενός ή πολλών τμημάτων και το έτος. Το σημαντικότερο από αυτά τα νέα ερωτήματα, είναι η δυνατότητα ανάκτησης δεδομένων πολλών τμημάτων με τη χρήση των μοναδικών κωδικών τους. Η σημαντικότητα του πηγάζει από το γεγονός ότι με αυτό τον τρόπο οι κλήσεις που πραγματοποιούνται στο API, μειώνονται σημαντικά. Με τα προηγούμενα ερωτήματα αν υπήρχε μία λίστα από κωδικούς για τους οποίους θα έπρεπε να ανακτηθούν δεδομένα, θα έπρεπε να γίνει μία κλήση για κάθε τμήμα, ενώ τώρα αρκεί μόλις μία. Επίσης, η υποστήριξη φίλτρων σε αυτά τα ερωτήματα, μειώνει και τον όγκο δεδομένων που με-

ταφέρονται, με αποτέλεσμα η χρήση του API από εφαρμογές που είτε θα εκτελούνται σε κινητές συσκευές είτε θα περιορίζονται από τη διαθέσιμη ταχύτητα πρόσβασης στο διαδίκτυο να είναι περισσότερο λειτουργική. Στο Σχήμα 3.22 παρουσιάζεται η δομή του ερωτήματος αναζήτησης των βάσεων εισαγωγής με κριτήρια πολλαπλά τμήματα και το έτος. Οι κωδικοί τμημάτων θα πρέπει να δίνονται χωρισμένοι από κόμμα.

```
GET /v1.0/bases/{year}?departments={dept_id1,dept_id2,...}
```

Σχήμα 3.22: Η δομή του συνδέσμου αναζήτησης βάσεων της έκδοσης 1.0

Ως παράδειγμα κλήσης αυτού ερωτήματος του API δόθηκαν οι τιμές 2020 για το έτος και οι κωδικοί που αντιστοιχούν στο Τμήμα Πληροφορικής του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης και στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙΠΑΕ. Επιπλέον, χρησιμοποιήθηκε το κατάλληλο φίλτρο για την επιλογή το δεδομένων που αντιστοιχούν μόνο στην κατηγορία 90% των ημερήσιων Γενικών Λυκείων. Στο Σχήμα 3.23 φαίνεται η δομή των δεδομένων που επεστράφησαν, πλέον τα δεδομένα αποτελούνται από έναν πίνακα του οποίου κάθε στοιχείο είναι ένα τμήμα. Στο κάθε στοιχείο περιλαμβάνονται τα στοιχεία του τμήματος και περιλαμβάνεται ένα πίνακας, όπου περιέχονται όλες τα δεδομένα των βάσεων εισαγωγής για το τμήμα αυτό.

```

1 - [
2 - {
3 -   "code": 338,
4 -   "deptName": "ΠΛΗΡΟΦΟΡΙΚΗΣ (ΘΕΣΣΑΛΟΝΙΚΗ)",
5 -   "uniTitle": "Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης",
6 -   "uniTitleShort": "ΑΠΘ",
7 -   "bases": [
8 -     {
9 -       "examType": "ΓΕΛ ΝΕΟ ΗΜΕΡΗΣΙΑ",
10 -      "specialCat": "ΓΕΛ ΓΕΝΙΚΗ ΣΕΙΡΑ ΗΜ. (ΝΕΟ)",
11 -      "positions": 105,
12 -      "baseFirst": 19200,
13 -      "baseLast": 16800,
14 -      "year": 2020
15 -     }
16 -   ]
17 - },
18 - {
19 -   "code": 1625,
20 -   "deptName": "ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
(ΘΕΣΣΑΛΟΝΙΚΗ)",
21 -   "uniTitle": "Διεθνές πανεπιστήμιο της Ελλάδος",
22 -   "uniTitleShort": "ΔΙ.ΠΑ.Ε.",
23 -   "bases": [
24 -     {
25 -       "examType": "ΓΕΛ ΝΕΟ ΗΜΕΡΗΣΙΑ",
26 -       "specialCat": "ΓΕΛ ΓΕΝΙΚΗ ΣΕΙΡΑ ΗΜ. (ΝΕΟ)",
27 -       "positions": 145,
28 -       "baseFirst": 16125,
29 -       "baseLast": 13325,
30 -       "year": 2020
31 -     }
32 -   ]
33 - }
34 - ]

```

Σχήμα 3.23: Παράδειγμα απόκρισης του API μετά από αναζήτηση βάσεων εισαγωγής με χρήση ερωτημάτων της έκδοσης 1.0

### 3.5.6 Αναζήτηση Στατιστικών

Η κατηγορία των στατιστικών αποτελεί και αυτή ένα σημαντικό μέρος του API. Υπάρχει μεγάλο πλήθος παραμέτρων που μπορούν να χρησιμοποιηθούν για την αναζήτηση δεδομένων.

Συγκεκριμένα μπορεί να γίνει αναζήτηση με παραμέτρους το έτος που αφορούν τα δεδομένα, το τριτοβάθμιο ίδρυμα, το τμήμα και την κατηγορία των στατιστικών (συμμετέχοντες ή επιτυχόντες), καθώς και τους πιθανούς συνδυασμούς αυτών.

Ο όγκος των δεδομένων για κάθε ερώτημα είναι μεγάλος, καθώς για κάθε τμήμα και για κάθε τύπο εξέτασης υπάρχουν διαφορετικές προτιμήσεις και διότι η κατηγορία των επιτυχόντων αποτελούνται από επτά θέσεις προτίμησης, ενώ η κατηγορία των υποψηφίων από τέσσερις. Όπως και στα δεδομένα των βάσεων εισαγωγής, έτσι και στα δεδομένα των στατιστικών υποστηρίζονται φίλτρα που περιορίζουν τα δεδομένα σε αυτά των Γενικών και Επαγγελματικών Λυκείων.

Στο Σχήμα 3.24 παρουσιάζεται η δομή του ερωτήματος με κριτήρια το έτος εισαγωγής, τον μοναδικό κωδικό τμήματος και την κατηγορία στατιστικών.

GET /statistics/{year}/department/{dept\_id}/category/{category}

Σχήμα 3.24: Η δομή του συνδέσμου αναζήτησης στατιστικών

Η απόκριση του API σε κάποιο ερώτημα στατιστικών περιλαμβάνει ένα σύνολο στοιχείων, που το κάθε ένα από αυτά περιέχει τις πληροφορίες σχετικές με το τμήμα και το έτος που αφορούν. Επιπρόσθετα, κάθε στοιχείο περιέχει την κατηγορία, τη σειρά προτίμησης και το πλήθος υποψηφίων που αντιστοιχούν τα δεδομένα.

```

1 v [
2 + {
3   "code": 1625,
4   "examType": "ΓΕΛ ΝΕΟ ΗΜΕΡΗΣΙΑ & ΕΙΣΠΕΡΙΝΑ",
5   "category": 1,
6   "preference": 1,
7   "count": 18,
8   "year": 2020
9 } ,
10 + {
11  "code": 1625,
12  "examType": "ΓΕΛ ΝΕΟ ΗΜΕΡΗΣΙΑ & ΕΙΣΠΕΡΙΝΑ",
13  "category": 1,
14  "preference": 2,
15  "count": 16,
16  "year": 2020
17 } ,
18 + {
19  "code": 1625,
20  "examType": "ΓΕΛ ΝΕΟ ΗΜΕΡΗΣΙΑ & ΕΙΣΠΕΡΙΝΑ",
21  "category": 1,
22  "preference": 3,
23  "count": 20,
24  "year": 2020
25 } ,

```

Σχήμα 3.25: Παράδειγμα απόκρισης του API σε αναζήτηση στατιστικών

Στο Σχήμα 3.25 παρουσιάζεται ένα μέρος από την απόκριση του API σε αναζήτηση στατιστικών για το τμήμα με κωδικό 1625, για το έτος 2020 και την κατηγορία 1 (επιτυχόντες), ενώ γίνεται και η χρήση φίλτρου για περιορισμό των δεδομένων σε αυτά που αφορούν τα Γενικά Λύκεια.

### 3.5.7 Αναζήτηση Στατιστικών με ερωτήματα της έκδοσης 1.0

Η κατηγορία των στατιστικών αποτέλεσε εύκολα το σημαντικότερο κομμάτι του API, που θα έπρεπε να αναθεωρηθεί η δομή του. Από αυτά που παρουσιάστηκαν παραπάνω, μπορεί να γίνει εύκολα αντιληπτό ότι σε ένα ερώτημα που δεν περιορίζονται σε μεγάλο βαθμό τα δεδομένα από παραμέτρους και φίλτρα, ο όγκος των δεδομένων γίνεται αρκετά μεγάλος. Κάτι τέτοιο, μπορεί να οδηγήσει εφαρμογές που θα βασίζονται στο API, να υπολειτουργούν, καθώς θα πρέπει να μεταφορτωθούν όλα αυτά τα δεδομένα προτού υποστούν επεξεργασία.

Για τη δημιουργία της νέας δομής, αποφασίστηκε να δημιουργηθεί μία όψη (view) στη βάση δεδομένων η οποία θα προσομοιάζει μία νέα δομή για τον πίνακα statistics. Η δομή της όψης αντικαθιστά τα πεδία που αντιστοιχούσαν στη σειρά προτίμησης και στο πλήθος ατόμων με πεδία για κάθε σειρά προτίμησης που θα περιέχουν το πλήθος. Τα πεδία της όψης φαίνονται στο Σχήμα 3.26.

statistics_v1			
code	int	s_first	int
id	string	s_second	int
category	int	s_third	int
plithos	int	s_fourth	int
year	int	s_fifth	int
c_first	int	s_sixth	int
c_second	int	s_other	int
c_third	int		
c_other	int		

Σχήμα 3.26: Η δομή της όψης “statistics\_v1”

Η επιλογή αυτή έγινε, με σκοπό αργότερα όταν θα γίνονται ερωτήματα SQL, με την χρήση του “GROUP BY” να ομαδοποιηθούν τα δεδομένα ενός τμήματος σε μία γραμμή. Με αυτόν τον τρόπο επιτεύχθηκε η μείωση των παραγομένων γραμμών για κάθε τμήμα και κατηγορία εξέτασης από επτά σε μία.

Για το μέρος του ερωτήματος στο API ή νέα δομή που σχεδιάστηκε ακολουθεί την ίδια φιλοσοφία με αυτή των βάσεων εισαγωγής. Η σύνταξη του συνδέσμου παραμένει η ίδια με την προσθήκη του “v1.0” στο αρχικό μέρος. Το κάθε στοιχείο της απόκρισης αποτελείται από τις πληροφορίες του τμήματος και για κάθε τμήμα υπάρχει ένας πίνακας με όνομα statistics που περιλαμβάνει τα δεδομένα των στατιστικών. Αυτός ο πίνακας περιλαμβάνει το έτος, το είδος

εξέτασης και το πλήθος των θέσεων που υπήρχαν και στο προηγούμενο. Η διαφοροποίηση γίνεται στην προσθήκη του αθροίσματος των υποψηφίων και επιτυχόντων. Η κυριότερη προσθήκη είναι ενός πεδίου που περιλαμβάνει τον αριθμό των ατόμων για κάθε σειρά προτίμησης για τους υποψήφιους και για τους επιτυχόντες.

```

1 [
2   {
3     "code": 1625,
4     "statistics": [
5       {
6         "year": 2019,
7         "examType": "10% ΓΕΛ ΗΜΕΡΗΣΙΑ & ΕΙΣΠΕΡΙΝΑ 2017",
8         "positions": 6,
9         "totalSuccessful": null,
10        "totalCandidates": null,
11        "candidatePreferences": {
12          "first": "6",
13          "second": "1",
14          "third": "9",
15          "other": "288"
16        },
17        "successfulPreferences": {
18          "first": "1",
19          "second": "1",
20          "third": "0",
21          "fourth": "1",
22          "fifth": "1",
23          "sixth": "2",
24          "other": "1"
25        }
26      },
27      {
28        "year": 2020,
29        "examType": "10% ΓΕΛ ΗΜΕΡΗΣΙΑ & ΕΙΣΠΕΡΙΝΑ 2018",
30        "positions": 5,
31        "totalSuccessful": null,
32        "totalCandidates": null,
33        "candidatePreferences": {
34          "first": "5",

```

Σχήμα 3.27: Παράδειγμα απόκρισης του API σε αναζήτηση στατιστικών με χρήση ερωτημάτων της έκδοσης 1.0

Στο Σχήμα 3.27 παρουσιάζεται η απόκριση του API για ερώτημα που έγινε χρησιμοποιώντας ως παράμετρο μόνο τον κωδικό του τμήματος. Ένα πλεονέκτημα που παρατηρείται είναι η καλύτερη οργάνωση των δεδομένων με αποτέλεσμα και την καλύτερη αξιοποίηση τους από τον προγραμματιστή. Το κυριότερο πλεονέκτημα είναι η μείωση του όγκου των δεδομένων, καθώς για το παράδειγμα που παρουσιάστηκε το μέγεθος των δεδομένων ήταν της τάξεως των 4.8 KB, ενώ το αντίστοιχο ερώτημα με την παλιά δομή δημιουργούσε δεδομένα μεγέθους 16 KB. Παρατηρείται δηλαδή μείωση 70% στην μεταφορά δεδομένων.

### 3.6 Επίλογος

Στο κεφάλαιο 3 έγινε μία παρουσίαση της σχεδίασης της βάσης δεδομένων. Παρουσιάστηκε ο ιστότοπος μεταφόρτωσης των δεδομένων των Πανελλαδικών Εξετάσεων, η μορφή των αρχείων που θα πρέπει να του δοθούν και ο αλγόριθμος που χρησιμοποιήθηκε για την ανάγνωση

## Κεφάλαιο 3

των δεδομένων και την αποθήκευση τους στη βάση δεδομένων. Έγινε περιγραφή των κατηγοριών που αποτελούν το API, των τεχνολογιών που χρησιμοποιεί και παρουσίαση κάποιων σημείων του αλγορίθμου που τον αποτελεί. Τέλος, παρουσιάστηκαν κάποια παραδείγματα από τα σημαντικότερα μέρη του API και αναθεωρήσεις που έγιναν σε αυτό.

## Κεφάλαιο 4ο: Διαδικτυακή Εφαρμογή

### 4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση της ανάπτυξης της σχεδίασης και της τελικής κατάστασης της διαδικτυακής εφαρμογής που αναπτύχθηκε στα πλαίσια της εργασίας. Συγκεκριμένα, θα παρουσιαστούν μία σειρά από ιστορίες χρήστη που θα περιγράφουν τις πιθανές περιπτώσεις χρήσης της εφαρμογής. Θα γίνει ανάλυση επιλεγμένων σημείων του κώδικα που γράφτηκε για την υλοποίηση και θα παρουσιαστούν οι λειτουργίες της εφαρμογής στο τελικό στάδιο.

### 4.2 Ιστορίες Χρήστη

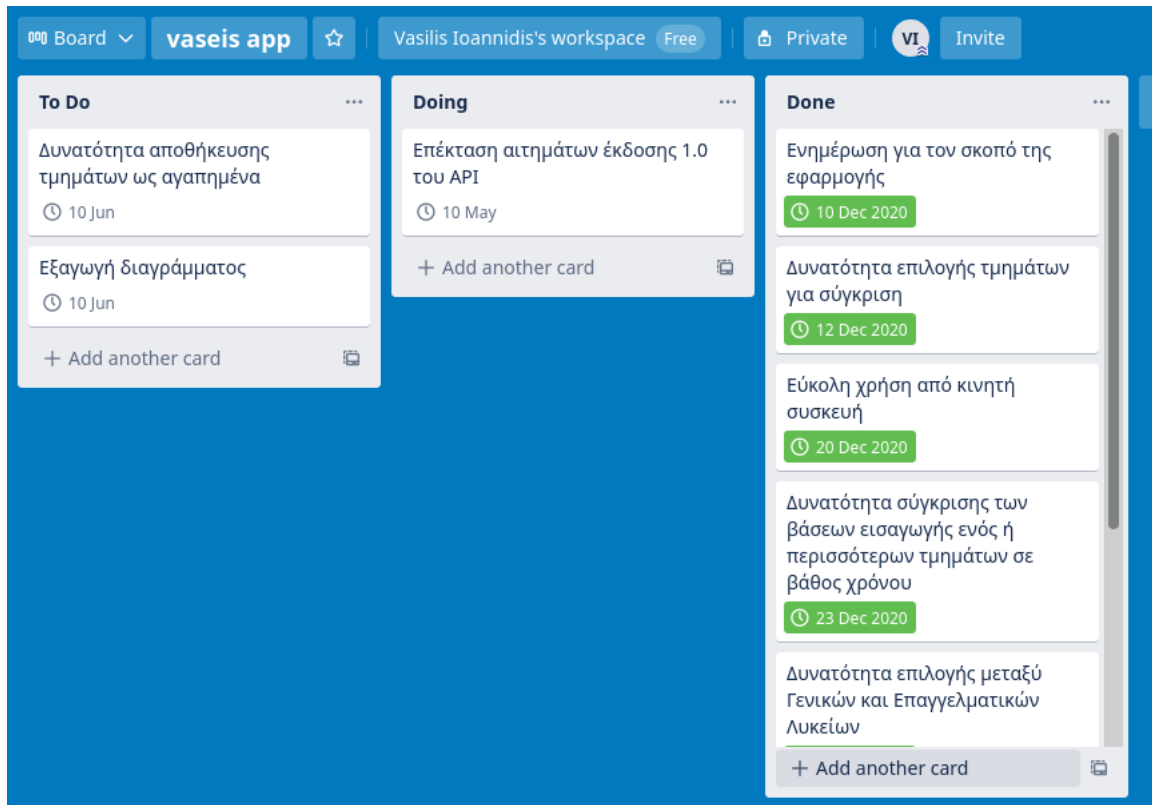
Οι ιστορίες χρήστη (user stories) αποτελούν μία μέθοδος που χρησιμοποιείται στην ανάπτυξη λογισμικού για την περιγραφή των χαρακτηριστικών που θα υλοποιηθούν. Συνήθως γράφονται με ανεπίσημο τρόπο και συνηθίζεται να γράφονται είτε σε αυτοκόλλητα χαρτάκια ή σε λογισμικό για αυτή τη χρήση. Οι ιστορίες χρήστη θα πρέπει να είναι σύντομες και διαχωρίζονται σε κατηγορίες ανάλογα το πόσο σημαντικό είναι το κάθε χαρακτηριστικό και στο τι αποσκοπεί. [34]

Προτού ξεκινήσει η ανάπτυξη της εφαρμογής θεωρήθηκε ότι θα πρέπει να γραφούν ένας αριθμός ιστοριών χρήστη, που θα καλύπτουν τις περιπτώσεις χρήσης και τα χαρακτηριστικά που ήταν επιθυμητό να υποστηρίζονται. Η κατηγοριοποίηση που επιλέχθηκε κατά την ανάπτυξη της εφαρμογής ήταν με βάση την κατάσταση του κάθε χαρακτηριστικού. Συγκεκριμένα η κάθε ιστορία χρήστη μπορεί να τοποθετηθεί στην κατηγορία “μελλοντική υλοποίηση“, στην “υλοποιείται“ και στην “υλοποιημένο“.

Ως εργαλείο για τη δημιουργία και οργάνωση των ιστοριών χρήστη επιλέχθηκε το Trello.<sup>3</sup> Επιπρόσθετα, για κάθε μία ιστορία επιλέχθηκε και μία άτυπη επιθυμητή ημερομηνία ολοκλήρωσης ενός χαρακτηριστικού, για την βέλτιστη οργάνωση του χρόνου.

Μερικές από τις ιστορίες χρήστη αφορούν τη δυνατότητα του χρήστη να ενημερωθεί για τις δυνατότητες που προσφέρει η εφαρμογή και την εύκολη χρήση της εφαρμογής από συσκευές με οθόνες μικρού μεγέθους. Ακόμα ο χρήστης θα πρέπει να έχει τη δυνατότητα να επιλέγει τα τριτοβάθμια ιδρύματα που επιθυμεί να συγκρίνει. Με την επιλογή των τμημάτων θα πρέπει μπορεί να έχει πρόσβαση σε ένα γράφημα γραμμής που θα παρουσιάζει την πορεία των βάσεων εισαγωγής των επιλεγμένων τμημάτων, δύο γραφήματα ράβδων για τα στατιστικά προτιμήσεων. Επιπλέον, θα πρέπει να προσφέρεται στον χρήστη η δυνατότητα επιλογής ενός τμήματος και να εμφανίζεται μία αναλυτική παρουσίαση δεδομένων για αυτό. Η επόμενη ιστορία χρή-

<sup>3</sup>Η ιστοσελίδα του Trello: <https://trello.com/>



Σχήμα 4.1: Στιγμιότυπο από τη σελίδα διαχείρισης των ιστοριών χρήστη του Trello

στη που αποτελεί συνέχεια της προηγούμενης αποτελεί η δυνατότητα του χρήστη να επιλέξει την προβολή δεδομένων είτε των Γενικών είτε των Επαγγελματικών Λυκείων και μεταξύ των κατηγοριών 90% και 10%. Τέλος, ο χρήστης θα πρέπει να έχει τη δυνατότητα να προσθέτει και να αφαιρεί τμήματα και να μπορεί να μοιράζεται τον σχετικό σύνδεσμο που θα περιλαμβάνει τα επιλεγμένα τμήματα.

Εκτός από τις ιστορίες χρήστη που ανήκουν στην κατηγορία “υλοποιημένα“, στην κατηγορία “μελλοντική υλοποίηση“ βρίσκονται η δυνατότητα αποθήκευσης τμημάτων ως αγαπημένα για την ευκολότερη εύρεση τους σε μελλοντική χρήση και η προσθήκη του χαρακτηριστικού εξαγωγής ενός διαγράμματος για χρήση εκτός της διαδικτυακής εφαρμογής. Τέλος, στην κατηγορία “υλοποιείται“ βρίσκεται η επιπλέον επέκταση της έκδοσης 1.0 του διαδικτυακού API.

## 4.3 Ανάλυση Κώδικα

### 4.3.1 Αρχική Σελίδα

Η ολοκλήρωση των ιστοριών χρήστη, επιτρέπει πλέον την εκκίνηση της ανάπτυξης της διαδικτυακής εφαρμογής. Αυτή χωρίζεται ουσιαστικά σε δύο σελίδες οι οποίες είναι αρχική σελίδα και η σελίδα πληροφοριών. Η αρχική σελίδα χωρίζεται σε τρία μέρη. Το πρώτο μέρος το οποίο καταλαμβάνει το πάνω μέρος της σελίδας, είναι και το σημείο που γίνεται η πρώτη επαφή του χρήστη με την εφαρμογή. Εδώ δίνεται η δυνατότητα στον χρήστη μέσω ενός παραθύρου να

επιλέξει τα τμήματα που επιθυμεί για ανάλυση.

Η υλοποίηση της λειτουργίας της αρχικής σελίδα μοιράζεται στην PHP και στην JavaScript. Όταν ο χρήστης φορτώνει την σελίδα η PHP είναι υπεύθυνη για τη φόρτωση των απαραίτητων δεδομένων. Συγκεκριμένα, κατά τη φόρτωση της σελίδας καλείται η συνάρτηση fillList η οποία κάνει δύο αιτήματα προς το API, ένα για τη λίστα όλων των πανεπιστημίων και ένα για τη λίστα όλων των τμημάτων. Στην συνέχεια έχοντας διαθέσιμους τους δύο πίνακες με τα δεδομένα, παράγει μία λίστα που περιλαμβάνει οργανωμένα τα δεδομένα αυτά με τη χρήση της HTML.

```

41 function fillList() {
42     $url = "https://vaseis.iee.ihu.gr/api/index.php/universities";
43     $unis = apiCall($url);
44     foreach ($unis["records"] as $uni) {
45         $url = "https://vaseis.iee.ihu.gr/api/index.php/departments/university/" . $uni["id"];
46         $depts = apiCall($url);
47         echo '<li><input type="checkbox" class="uni-checkbox"> ' .
48             '<label for="uni[]" class="uni-label"> ' . $uni["full-title"] . '</label>';
49         echo "<ul>";
50         foreach ($depts as $dept) {
51             echo '<li><input type="checkbox" name="dept[]" class="dept-checkbox" value=" ' . $dept['code'] .
52                 '><label for="dept[]" class="dept-label"> ' . $dept["name"] . '</label></li>';
53         }
54         echo "</li></ul>";
55     }
56 }

```

Σχήμα 4.2: Η συνάρτηση fillList

Η συνάρτηση fillList χρησιμοποιεί τη συνάρτηση apiCalls, η οποία αποτελεί μέρος ενός συνόλου συναρτήσεων που αναπτύχθηκαν κατά την υλοποίηση του διαδικτυακού API. Αυτές οι συναρτήσεις περιλαμβάνουν την apiCalls, που υλοποιεί μία κλήση προς το API και τις συναρτήσεις, που υλοποιούν τους κωδικούς απάντησης του API (200, 404 κ.ο.κ.). Αυτές οι συναρτήσεις έχουν σκοπό την ευκολότερη και περισσότερο ομοιόμορφη διαχείριση απαντήσεων του API.

```

12 function http404(): array
13 {
14     http_response_code( response_code: 404);
15     return array("error" => "Δεν βρέθηκαν αποτελέσματα.");
16 }
17
18 function http405() {
19     http_response_code( response_code: 405);
20     echo json_encode(array("error" => "Δεν επιτρέπεται αυτό το αίτημα."));
21 }
22
23 function apiCall($url) {
24     $curl = curl_init();
25     curl_setopt($curl, option: CURLOPT_URL, $url);
26     curl_setopt($curl, option: CURLOPT_RETURNTRANSFER, value: 1);
27     $result = curl_exec($curl);
28     $result = json_decode($result, associative: true);
29     return $result;
30 }

```

Σχήμα 4.3: Μέρος των βοηθητικών συναρτήσεων για τη διαχείριση του API

Η διαδραστικότητα της σελίδας υλοποιείται με τη χρήση της JavaScript. Συγκεκριμένα έχουν

δημιουργηθεί ακροατές γεγονότων (event listeners), οποίοι εκτελούν το κατάλληλο κομμάτι κώδικα ανάλογα την περίπτωση.

Μεταξύ άλλων έχουν υλοποιηθεί οι λειτουργίες για το εμφάνιση και απόκρυψη του παραθύρου επιλογής τμημάτων, η ανανέωση της λεζάντας που ενημερώνει τον αριθμό επιλεγμένων τμημάτων και η παύση αποστολής των δεδομένων της φόρμας σε περίπτωση που δεν υπάρχουν επιλεγμένα τμήματα.

Ένα σημαντικό μέρος της αρχικής σελίδας και συγκεκριμένα του παραθύρου επιλογής τμημάτων είναι η δυνατότητα αναζήτησης. Η συνάρτηση αναζήτησης εκτελείται μετά το πάτημα ενός πλήκτρου από τον χρήστη. Η αναζήτηση δεν πραγματοποιείται κάνοντας κάποιο αίτημα στο API, αλλά φιλτράροντας την λίστα που είχε παραχθεί νωρίτερα από την PHP.

Ο κώδικας που υλοποιεί την παραπάνω λειτουργία μοιράζεται σε τρεις συναρτήσεις γραμμένες σε JavaScript. Μόλις εντοπιστεί το γεγονός πληκτρολόγησης (key up) εκτελείται η συνάρτηση “filterDepts“ η οποία αποθηκεύει την είσοδο του χρήστη και καλεί τη συνάρτηση “clearFilter“ η οποία επιστρέφει τη λίστα των τμημάτων στην αρχική της κατάσταση. Ύστερα καλείται η “removeAccents“ περνώντας ως παράμετρο αυτή την αποθηκευμένη τιμή. Αυτή η συνάρτηση μετατρέπει τους χαρακτήρες σε κεφαλαίους, αφαιρεί τους τόνους και επιστρέφει την παραμετροποιημένο κείμενο. Αφού ολοκληρωθεί η προηγούμενη διαδικασία, η συνάρτηση δημιουργεί έναν βρόγχο επανάληψης στην λίστα των τμημάτων και στην περίπτωση που η ονομασία ενός τμήματος δεν περιλαμβάνει το φίλτρο του χρήστη, τότε αυτό κρύβεται και επιπρόσθετα στην περίπτωση που κρυφτούν όλα τα τμήματα ενός πανεπιστημίου τότε κρύβεται και το όνομα αυτού. Τέλος, εκτελείται ένας ακόμα βρόγχος επανάληψης επαναλαμβάνοντας την ίδια λειτουργία για τα πανεπιστήμια. Στην περίπτωση που ο τίτλος ενός πανεπιστημίου περιλαμβάνει το φίλτρο εκτελείται ένας επιπλέον βρόγχος επανάληψης για τα τμήματα του συγκεκριμένου πανεπιστημίου, έτσι ώστε να εμφανίσει οποιοδήποτε τμήμα είχε κρυφτεί νωρίτερα.

Η υποβολή της φόρμας εκτελεί μία συνθήκη η οποία ελέγχει αν έχουν αρχικοποιηθεί οι μεταβλητές τύπου GET, που απαιτούνται για την εκτέλεση του αλγορίθμου. Τα περιεχόμενα της λίστας των τμημάτων έχουν ως παράμετρο “name“ την τιμή “dept“, η οποία δίνει την οδηγία στην PHP να χειριστεί όλα αυτά τα αντικείμενα ως ένα πίνακα. Επίσης, κάθε αντικείμενο έχει ορισμένη την παράμετρο “value“, στην οποία δίνεται ως τιμή ο μοναδικός κωδικός του κάθε τμήματος.

Κατά τη διάρκεια της ανάπτυξης της διαδικτυακής εφαρμογής, εκτός από τη δυνατότητα επιλογής τμημάτων από λίστα, υλοποιήθηκε και η δυνατότητα αναζήτησης συγκεκριμένου τμήματος. Αυτή η λειτουργία υλοποιείται με τη χρήση της HTML ετικέτας datalist, που δίνει τη δυνατότητα επιλογής προκαθορισμένων επιλογών κατά την πληκτρολόγηση του χρήστη. Οι πιθανές επιλογές διατίθενται με την παραγωγή ετικετών τύπου option από την PHP. Αυτές ακολουθούν τη μορφή “κωδικός - όνομα τμήματος“. Αν και τελικά αποφασίστηκε η αφαίρεση της συγκεκριμένης λειτουργίας η προγραμματιστική υλοποίηση παραμένει ενεργή, για την ευκολότερη

```

48 function filterDepts() {
49   let filterValue = document.querySelector( selectors: '.filter-input').value;
50   clearFilter();
51   filterValue = removeAccents(filterValue);
52   for (let i = 0; i < deptLabel.length; i++) {
53     if(deptLabel[i].innerText.includes(filterValue)) {
54     } else {
55       let listEl = deptLabel[i].parentElement;
56       let uniEl = listEl.parentElement;
57       let parentEl = uniEl.parentElement;
58       listEl.style.display = 'none';
59       let count = 0;
60       for (let y = 0; y < uniEl.children.length; y++) {
61         if (uniEl.children[y].style.display === 'none') {
62           count++;
63         }
64       }
65       if (count === uniEl.children.length) {
66         parentEl.style.display = 'none';
67       }
68     }
69   }
70   for (let i = 0; i < uniLabel.length; i++) {
71     let uniName = removeAccents(uniLabel[i].innerText);
72     if(uniName.includes(filterValue)) {
73       let listEl = uniLabel[i].parentElement;
74       for (let y = 0; y < listEl.children.length; y++) {
75         listEl.style.display = 'block';
76         switch (y) {
77           case 0:
78             listEl.children[y].style.display = 'inline-block';
79             break;
80           case 1:
81             listEl.children[y].style.display = 'block';
82             break;
83           default:
84             break;
85         }
86       }
87     }
88     for (let y = 0; y < listEl.children[2].children.length; y++) {
89       listEl.children[2].children[y].style.display = 'block';
90     }
91   }
92 }

```

Σχήμα 4.4: Η συνάρτηση filterDepts

επανάταξη της.

Ο αλγόριθμος χωρίζεται σε τρία μέρη και συγκεκριμένα τρεις συνθήκες ελέγχου. Αρχικά ελέγχεται αν τα πεδία “dept” και “dept-search” της μεταβλητής GET έχουν οριστεί. Ο συγκεκριμένος έλεγχος γίνεται για την περίπτωση που χρησιμοποιηθούν και η λίστα των τμημάτων και το πεδίο αναζήτησης ταυτόχρονα. Εφόσον η συνθήκη είναι αληθής εκτελείται ένα βρόγχος επανάληψης, ο οποίος απαριθμεί τα πεδία του πίνακα “dept”. Αυτά, περιλαμβάνουν τους κωδικούς των τμημάτων και προστίθενται σε μία μεταβλητή τύπου string, διαχωρισμένα από κόμμα. Με την ολοκλήρωση της επανάληψης και εφόσον το πεδίο “dept-search” της μεταβλητής GET δεν είναι κενό, η συμβολοσειρά του “dept-search” διασπάται σε πίνακα χρησιμοποιώντας ως διαχωριστή τον χαρακτήρα “-“. Έπειτα προστίθενται το πρώτο πεδίο του πίνακα στην μεταβλητή που αναφέρθηκε νωρίτερα και ο χρήστης ανακατευθύνεται στη σελίδα δεδομένων, δίνοντας ως παράμετρο GET το “id” με τη συμβολοσειρά που παράχθηκε. Η λειτουργία των επόμενων δύο συνθηκών ελέγχου, υλοποιούν χωριστά τις δύο περιπτώσεις αλληλεπίδρασης του χρήστη με την

εφαρμογή και τελικά ανακατευθύνουν τον χρήστη στη σελίδα δεδομένων.

```

4  if (isset($_GET['dept']) && isset($_GET['dept-search'])) {
5      $depts = "";
6      foreach ($_GET['dept'] as $dept) {
7          $depts .= $dept . ',';
8      }
9      $depts = substr($depts, offset: 0, length: strlen($depts) -1);
10     if($_GET['dept-search'] != '') {
11         $dept = explode( separator: '-', $_GET['dept-search']);
12         $depts .= ',' . $dept[0];
13     }
14     header( header: 'Location: view.php?id=' . $depts);
15 }
16 elseif(isset($_GET['dept'])) {
17     $depts = "";
18     foreach ($_GET['dept'] as $dept) {
19         $depts .= $dept . ',';
20     }
21     $depts = substr($depts, offset: 0, length: strlen($depts) -1);
22     header( header: 'Location: view.php?id=' . $depts);
23 }

```

Σχήμα 4.5: Μέρος του αλγορίθμου διαχείρισης της υποβολής φόρμας

#### 4.3.2 Σελίδα Δεδομένων

Κατά τη φόρτωση της σελίδας δεδομένων εκτελείται μια αλληλουχία συναρτήσεων της PHP, για την αρχικοποίηση της σελίδας. Αρχικά ελέγχονται τα περιεχόμενα του πεδίου id της μεταβλητής GET και εφόσον δεν είναι ορισμένο, τότε ο χρήστης ανακατευθύνεται στην αρχική σελίδα. Εφόσον είναι ορισμένη η μεταβλητή, τότε η συμβολοσειρά των κωδικών διασπάται σε πίνακα, εκτελείται ερώτημα στο διαδικτυακό API για το ελάχιστο και μέγιστο έτος των δεδομένων που είναι αποθηκευμένα και αρχικοποιούνται οι μεταβλητές που θα χρησιμοποιηθούν αργότερα.

Η τελευταία ομάδα εντολών που εκτελείται αυτόματα κατά τη φόρτωση της σελίδας είναι ένας βρόγχος επανάληψης που απαριθμεί τον πίνακα των κωδικών των τμημάτων. Αυτός δημιουργεί για κάθε τμήμα ένα πεδίο σε έξι πίνακες και έχει ως δείκτη των κωδικό του τμήματος. Αυτοί οι πίνακες αφορούν τη βάση εισαγωγής του πρώτου και τελευταίου, τις θέσεις που προσφέρονται από το τμήμα, το πανεπιστήμιο που ανήκει το τμήμα, το όνομα του τμήματος και το έτος που αφορούν τα δεδομένα. Ύστερα εκτελείται ένα αίτημα στο API για κάθε τμήμα, που αναζητούνται όλες οι βάσεις εισαγωγής του τμήματος για την κατηγορία των ημερήσιων Γενικών Λυκείων. Τέλος, εκτελείται ένα βρόγχος επανάληψης για κάθε αποτέλεσμα, που εισάγονται τα δεδομένα στα αντίστοιχα πεδία του πίνακα που δημιουργήθηκαν νωρίτερα.

Κατά τη φόρτωση της HTML της σελίδας, είναι αναγκαίο να εκτελεστεί κώδικας της PHP για να εμφανιστούν τα αρχικά δεδομένα στον χρήστη. Κατά τη διάρκεια της σχεδίασης της σελίδας, επιλέχθηκε να εμφανίζεται λεπτομερώς ένα τμήμα των δεδομένων ενός επιλεγμένου τμήματος. Κατά την αρχική φόρτωση της σελίδας επιλέχθηκε να εμφανίζονται τα δεδομένα του τμήματος του οποίου ο κωδικός βρίσκεται πρώτος στο πεδίο id της μεταβλητής. Έτσι, κατά τη φόρτωση

```

20 foreach ($id as $code) {
21     $codes[$i] = $code;
22     $i++;
23     $bases[$code] = array();
24     $basesFirst[$code] = array();
25     $places[$code] = array();
26     $unis[$code] = array();
27     $depts[$code] = array();
28     $url = 'https://vaseis.tee.ihe.gr/api/index.php/bases/department/' . $code . '?type=get-ime-gen&details=full';
29     $base = apiCall($url);
30     $base = $base['records'];
31     $years[$code] = array();
32     foreach ($base as $b) {
33         array_push( &array: $bases[$code], $b["baseLast"]);
34         array_push( &array: $basesFirst[$code], $b["baseFirst"]);
35         array_push( &array: $places[$code], $b["positions"]);
36         array_push( &array: $unis[$code], $b["uniTitle"]);
37         array_push( &array: $depts[$code], $b["deptName"]);
38         array_push( &array: $years[$code], $b["year"]);
39     }
40 }

```

Σχήμα 4.6: Ο βρόγχος επανάληψης που λαμβάνει τα αρχικά δεδομένα από το API

της σελίδας μεταξύ άλλων τυπώνονται πληροφορίες όπως το όνομα του επιλεγμένου τμήματος και το πανεπιστήμιο που ανήκει. Επιπλέον, γεμίζουν οι ετικέτες SELECT της HTML που αφορούν τα έτη για τα οποία υπάρχουν δεδομένα βάσεων εισαγωγής και στατιστικών για το τμήμα αυτό, ύστερα από κλήση στο API.

```

101 <div>
102 <?php
103     $bases = getTenPercent($codes);
104     $year = 0;
105     $count = count($bases);
106     for ($i = $count-1; $i > $count-3; $i--) {
107         $baseFirst = $bases[$i]['baseFirst'];
108         $baseLast = $bases[$i]['baseLast'];
109         $positions = $bases[$i]['positions'];
110         $special = explode( ' ', $bases[$i]['specialCat']);
111         if ($special[count($special)-1] == "ΣΕΙΡΑ") {
112             $special[count($special)-1] = "2012";
113         }
114         echo "<span>" . $special[count($special) -1] . "; ";
115         echo "<span></span><span class='year'>Βαθμός Πρώτου: </span>" . $baseFirst . "</span>";
116         echo "<span><span class='year'>Βαθμός Τελευταίου: </span>" . $baseLast . "</span>";
117         echo "<span><span class='year'>Εισακτές: </span>" . $positions . "</span>";
118         echo "</span><br>";
119     }
120     ?>
121 </div>

```

Σχήμα 4.7: Ο αλγόριθμος που αρχικοποιεί τις λεπτομερείς πληροφορίες της κατηγορίας 10% για το πρώτο τμήμα

Παράλληλα, χρησιμοποιώντας δεδομένα από το API, δημιουργείται ένα πεδίο αναζήτησης για την αναζήτηση και προσθήκη νέων τμημάτων για σύγκριση και αρχικοποιείται το εύρος τιμών για τα έτη που θα εμφανίζονται στο διάγραμμα σύγκρισης βάσεων εισαγωγής. Τέλος, δημιουργείται μία λίστα με τα τμήματα που έχει αναζητήσει ο χρήστης, με την οποία ο χρήστης μπορεί να αλληλεπιδράσει με τρόπο που θα παρουσιαστεί παρακάτω.

Αφού εκτελεστεί ο κώδικας της PHP και φορτώσει η σελίδα, έχει οριστεί ένας ακροατής γε-

γονότων της JavaScript, ο οποίος ενεργοποιείται μόλις ολοκληρωθεί η φόρτωση της. Σκοπός του είναι η περαιτέρω αρχικοποίηση της σελίδας, με την προσθήκη ακροατών γεγονότων στα στοιχεία που δημιουργήθηκαν δυναμικά νωρίτερα, για την εναλλαγή των τμημάτων και την αφαίρεση αυτών.

```

59 window.addEventListener( type: 'load', listener: () => {
60     let ctx = document.getElementById( elementId: 'myChart');
61     window.myLine = new Chart(ctx, config);
62     let statsLeftCtx = document.getElementById( elementId: 'stats-left');
63     window.myBarLeft = new Chart(statsLeftCtx, statsLeftConfig);
64     let statsRightCtx = document.getElementById( elementId: 'stats-right');
65     window.myBarRight = new Chart(statsRightCtx, statsRightConfig);
66     for (let i = yearsFrom.value; i <= yearsTo.value; i++) {
67         yearsAxis.push(i);
68     }
69     for(let i = 0; i < deptsEl.length; i++) {
70         depts.push(deptsEl[i].innerHTML);
71     }
72     let removeDeptBtns = document.querySelectorAll( selectors: '.remove-dept');
73     for (let i = 0; i < removeDeptBtns.length; i++) {
74         removeDeptBtns[i].addEventListener( type: 'click', listener: (e :Event) => removeDept(e,i));
75     }
76     let deptContainers = document.querySelectorAll( selectors: '.dept-container');
77     for (let i = 0; i < deptContainers.length; i++) {
78         deptContainers[i].addEventListener( type: 'click', listener: (e :Event) => showDetails(e));
79     }
80     let firstDept = document.querySelector( selectors: '.dept-container');
81     firstDept.classList.add('selected');
82     loadData( type: 0);
83 });

```

Σχήμα 4.8: Ο αλγόριθμος που εκτελείται μόλις ενεργοποιηθεί ο ακροατής γεγονότων για τη φόρτωση της σελίδας

Το μεγαλύτερο μέρος του κώδικα που εκτελείται αφορά την αρχικοποίηση και το γέμισμα των διαγραμμάτων της σελίδας. Συγκεκριμένα, επιλέγεται το σημείο που θα βρίσκεται το κάθε γράφημα στη σελίδα και στη συνέχεια τους δίνεται ως παράμετρος μία μεταβλητή που δρα ως αρχείο ρυθμίσεων για το κάθε διάγραμμα. Αυτές οι μεταβλητές περιέχουν πληροφορίες όπως ο τύπος του διαγράμματος, οι τίτλοι των αξόνων, τα χρώματα που θα χρησιμοποιηθούν και τα δεδομένα.

Το πεδίο των δεδομένων αρχικά είναι άδειο και για αυτό καλείται η συνάρτηση `loadData`, που έχει ως στόχο το γέμισμα των διαγραμμάτων με δεδομένα. Λειτουργεί ως διαχειριστής για τη φόρτωση των δεδομένων, καθώς η λειτουργία της περιορίζεται στην ανάκτηση των κωδικών των τμημάτων από το URL της σελίδας και η κλήση των συναρτήσεων που είναι υπεύθυνες για το κάθε τύπο διαγράμματος.

Η πρώτη συνάρτηση που καλείται είναι η `workData`, η οποία είναι υπεύθυνη για το κεντρικό διάγραμμα σύγκρισης βάσεων εισαγωγής. Η μοναδική παράμετρος που χρειάζεται είναι ο τύπος του σχολείου για το οποίο θα πρέπει να ανακτήσει τα απαραίτητα δεδομένα. Αρχικά εκτελείται ένα βρόγχος επανάληψης, που απαριθμεί τους κωδικούς των τμημάτων. Μέσα σε αυτόν καλείται μία κλήση στο διαδικτυακό API, με τη χρήση της συνάρτησης `fetchBases`, η οποία δέχεται ως παραμέτρους των κωδικό του τμήματος και τον τύπο του σχολείου. Η `fetchBases`

ορίζει το κατάλληλο URL και καλεί το API για τη λήψη των δεδομένων. Αυτή η συνάρτηση, όπως και όλες όσες αλληλεπιδρούν με το διαδικτυακό API (`fetchBasesByYear`, `fetchStats`, `fetchMinMaxBaseYear`), χρησιμοποιούν το `fetch` API που προσφέρει η JavaScript.

Η συνάρτηση μόλις λάβει την απάντηση της `fetchBases` και εφόσον δεν υπάρχει κάποιο σφάλμα, οργανώνει τα δεδομένα σε ένα πίνακα για τις βάσεις και ένα για τα έτη. Ελέγχει την περίπτωση τα παλαιότερα διαθέσιμα δεδομένα να είναι μεταγενέστερα της προ επιλεγμένης ελάχιστης τιμής του έτους του γραφήματος. Εφόσον η συνθήκη είναι αληθής, τότε προσθέτει στην αρχή του πίνακα την τιμή `null` για όσα έτη είναι αναγκαία ώστε η γραμμή του διαγράμματος να ξεκινά από το σωστό έτος. Τέλος, εισάγει αυτά τα δεδομένα στη μεταβλητή ρυθμίσεων του διαγράμματος και καλεί τη συνάρτηση ανανέωσης του.

```

115   for (let i = 0; i < ids.length; i++) {
116       let result = await fetchBases(ids[i], type);
117       if (!("error" in result)) {
118           result = result['records'];
119           let bases = result.map(x => x['baseLast']);
120           let years = result.map(x => x["year"]);
121           if (i === 0) yearsFirst = years;
122           let year = new Date();
123           year = year.getFullYear();
124           let earliestDataYear = Math.min(...years);
125           if (earliestDataYear > 2013 && yearsAxis.length > years.length) {
126               for (let y = earliestDataYear; y > yearsFrom.value; y--) {
127                   bases.unshift( items: null);
128               }
129           }
130           let dept = result[0]['deptName'];
131           let color = window.chartColors[colorNames[config.data.datasets.length % colorNames.length]];
132           data = {
133               label: `Τμήμα ${dept}`,
134               backgroundColor: color,
135               borderColor: color,
136               data: bases,
137               fill: false,
138               index: i,

```

Σχήμα 4.9: Μέρος της συνάρτησης `workData`

Με την ολοκλήρωση της φόρτωσης και εμφάνισης των δεδομένων των βάσεων εισαγωγής καλείται η συνάρτηση `loadStatsData`, της οποίας σκοπός είναι η λήψη των δεδομένων των στατιστικών προτιμήσεων και η φόρτωση τους στα αντίστοιχα διαγράμματα. Αυτή η συνάρτηση ακολουθεί την ίδια φιλοσοφία με τη `workData`, διαφοροποιείται μόνο από το γεγονός ότι γεμίζει δύο διαφορετικά γραφήματα.

Με τις παραπάνω συναρτήσεις να έχουν εκτελεστεί, η σελίδα πλέον έχει φορτωθεί πλήρως, έχοντας λάβει τα απαραίτητα δεδομένα από το διαδικτυακό API. Επιπλέον, εκτέλεση κώδικα γίνεται με την αλληλεπίδραση του χρήστη με τη σελίδα, όπως θα αναλυθεί παρακάτω.

Η προσθήκη ενός νέου τμήματος γίνεται με μία λειτουργία αναζήτησης, παρόμοια με αυτή που περιγράφηκε νωρίτερα στην αρχική σελίδα. Επιλέγοντας μία επιλογή ενεργοποιείται ο αντίστοιχος ακροατής γεγονότων, ο οποίος εκτελεί τη συνάρτηση `searchResults`. Η λειτουργία της, θυμίζει τη λειτουργία `workData`, καθώς κατά ένα μέρος επιτελούν τον ίδιο σκοπό. Η `searchResults`

διαβάζει την αναζήτηση του χρήστη, ανακτά τον κωδικό τμήματος και ύστερα λαμβάνοντας τα δεδομένα από το API, γεμίζει τα διάγραμμα. Προσθέτει το νέο τμήμα και του ακροατών γεγονότων που χρειάζονται στην λίστα των τμημάτων και τέλος επεξεργάζεται το URL του φυλλομετρητή με την προσθήκη του νέου κωδικού σε αυτό. Η λειτουργία της παραμετροποίησης του URL γίνεται με την χρήση της συνάρτησης `history.pushState()` της JavaScript, η οποία προσθέτει μία καταχώρηση στο ιστορικό του φυλλομετρητή.

Η δυνατότητα αφαίρεσης ενός τμήματος από την λίστα υλοποιείται με την ενεργοποίηση ενός ακροατή γεγονότων, η οποίος εντοπίζει το πάτημα του χρήστη στο εικονίδιο αφαίρεσης του τμήματος. Ο ακροατής γεγονότων εκτελεί την `removeDept`, η οποία αρχικά αφαιρεί από την λίστα το στοιχείο με το οποίο έγινε η αλληλεπίδραση. Κατά τη σχεδίαση της εφαρμογής, έχει προγραμματιστεί για το κάθε στοιχείο της λίστας να ορίζεται το πεδίο `id` με την τιμή του μοναδικού κωδικού του τμήματος που περιέχει. Με αυτόν τον τρόπο εξάγεται η τιμή κατά την εκτέλεση της `removeDept`, αφαιρείται από το URL από του φυλλομετρητή και γίνονται διορθώσεις που πιθανώς να χρειάζονται σε αυτό. Τέλος, ενημερώνεται το διάγραμμα χρησιμοποιώντας και πάλι τον κωδικό του τμήματος ως αναφορά.

```

268 function removeDept(e, i) {
269     e.target.parentElement.remove();
270     let code = e.target.parentElement.id;
271     let url = window.location.href;
272     url = url.replace(code, '');
273     url = url.replace(/,/g, ',');
274     url = url.replace(/=/g, '=');
275     if (url.charAt(url.length-1) === ',') url = url.substr(0, url.length -1);
276     window.history.pushState( data: '', title: 'Title', url);
277     for (let y = 0; y < config.data.datasets.length; y++) {
278         if (config.data.datasets[y].index === i) {
279             config.data.datasets = config.data.datasets.filter(data => data.index !== i);
280         }
281     }
282     window.myLine.update();
283     let deptContainers = document.querySelectorAll( selectors: '.deptContainer');
284     if (deptContainers.length === 0) {
285         document.body.scrollTop = 0;
286         document.documentElement.scrollTop = 0;
287     }
288 }

```

Σχήμα 4.10: Ο αλγόριθμος της συνάρτησης `removeDept`

Η εναλλαγή της προβολής των δεδομένων με βάση τον τύπο του σχολείου πραγματοποιείται με έναν ακροατή γεγονότων τύπου “αλλαγή” (`change`). Αυτός ορίζεται σε ένα στοιχείο HTML τύπου `select` και αφού εντοπιστεί κάποια αλλαγή στην τιμή αυτού, εκτελείται η συνάρτηση `changeType`. Η συνάρτηση, αφού λάβει τις τιμές των κωδικών των τμημάτων και τον επιλεγμένο τύπο σχολείου εκτελεί την `workData` που περιγράφηκε παραπάνω. Ύστερα εντοπίζει το επιλεγμένο τμήμα από την λίστα τμημάτων, έτσι ώστε να ανανεώσει τα δεδομένα για αυτό τμήμα των λεπτομερειών της σελίδας. Καλούνται οι `loadBaseData` και η `loadStatsData` για τις βάσεις εισαγωγής και τα στατιστικά προτίμησης αντίστοιχα. Αυτές οι δύο συναρτήσεις λαμβάνουν ως παραμέτρους το έτος, τον κωδικό τμήματος και τον τύπο του σχολείου και μέσω των

fetchBasesByYear και fetchStatsByYear, λαμβάνουν τα απαραίτητα δεδομένα με βάση αυτά τα κριτήρια από το διαδικτυακό API. Τέλος, η loadBaseData εμφανίζει οργανωμένα τα ληφθέντα δεδομένα και η loadStatsData ανανεώνει τα δύο διαγράμματα στατιστικών προτιμήσεων.

```

404 async function changeType() {
405     config.data.datasets = [];
406     let getParam = window.location.search.substr( from: 1);
407     ids = getParam.split( separator: '=');
408     ids = ids[1].split( separator: ',');
409     let selectValue = document.querySelector( selectors: '.type-select').value;
410     let type;
411     if (selectValue === "ΓΕΛ") type = 0;
412     else type = 1;
413     // loads base chart based on type selected
414     await workData(type);
415     let year = document.querySelector( selectors: '.year-select').value;
416     let deptContainers = document.querySelectorAll( selectors: '.dept-container');
417     deptContainers = Array.from(deptContainers);
418     let dept = deptContainers.filter(x => x.classList.contains('selected'));
419     let deptId = dept[0].id;
420     await loadBaseData(year, deptId, type);
421     //loads stats charts based on properties provided
422     await loadStatsData(year, deptId, type);
423 }

```

Σχήμα 4.11: Ο αλγόριθμος της συνάρτησης changeType

Η εναλλαγή μεταξύ των διαθέσιμων ετών για την προβολή λεπτομερειών των δεδομένων για το επιλεγμένο τμήμα, γίνεται όπως περιγράφηκε παραπάνω με έναν ακροατή τύπου “αλλαγή“. Οι δύο ακροατές δεδομένων για τις δύο επιλογές έτους εκτελούν δύο συναρτήσεις οι οποίες αξιοποιούν τις ίδιες συναρτήσεις που αναφέρθηκαν στην περιγραφή της changeType παραπάνω. Η λειτουργία τους αποτελεί υποσύνολο της changeType αφού η λογική του είναι η ίδια, δηλαδή η ανανέωση της εμφάνισης των δεδομένων με βάση το επιλεγμένο τμήμα, έτος και τύπο σχολείου.

Η τελευταία λειτουργία που θα αναλυθεί είναι ο αλγόριθμος επιλογής ενός τμήματος από τη λίστα τμημάτων. Με την επιλογή ενός τμήματος από τον χρήστη ενεργοποιείται ένας ακροατής γεγονότων, ο οποίος εκτελεί τη συνάρτηση showDetails. Η showDetails αφαιρεί το πεδίο τύπου class με τιμή selected από το τμήμα που ήταν επιλεγμένο νωρίτερα και το προσθέτει στο καινούργιο και καλεί τη συνάρτηση workDetails, η οποία υλοποιεί τον αλγόριθμο ανανέωσης των δεδομένων που εμφανίζονται.

Η workDetails χρησιμοποιεί την loadBaseData και loadStatsData που αναφέρθηκαν παραπάνω για να ανανεώσει τα δεδομένα που εμφανίζονται. Καλεί τη συνάρτηση resetSelect, της οποίας ο σκοπός είναι η ανανέωση των στοιχείων select της σελίδας με τις σωστές επιλογές για τα έτη των δεδομένων. Αυτό γίνεται χρησιμοποιώντας το διαδικτυακό API και τις συναρτήσεις fetchMinMaxStatYear και fetchMinMaxBaseYearByDept, προκειμένου να γίνει λήψη του μικρότερου και μεγαλύτερου έτους που υπάρχουν δεδομένα για κάθε τμήμα. Τέλος, η resetSelect καλεί τη συνάρτηση buildSelect, η οποία αφαιρεί τις επιλογές των δύο στοιχείων και τα γεμίζει με τις νέες επιλογές με βάση τα δεδομένα που λήφθηκαν από το API.

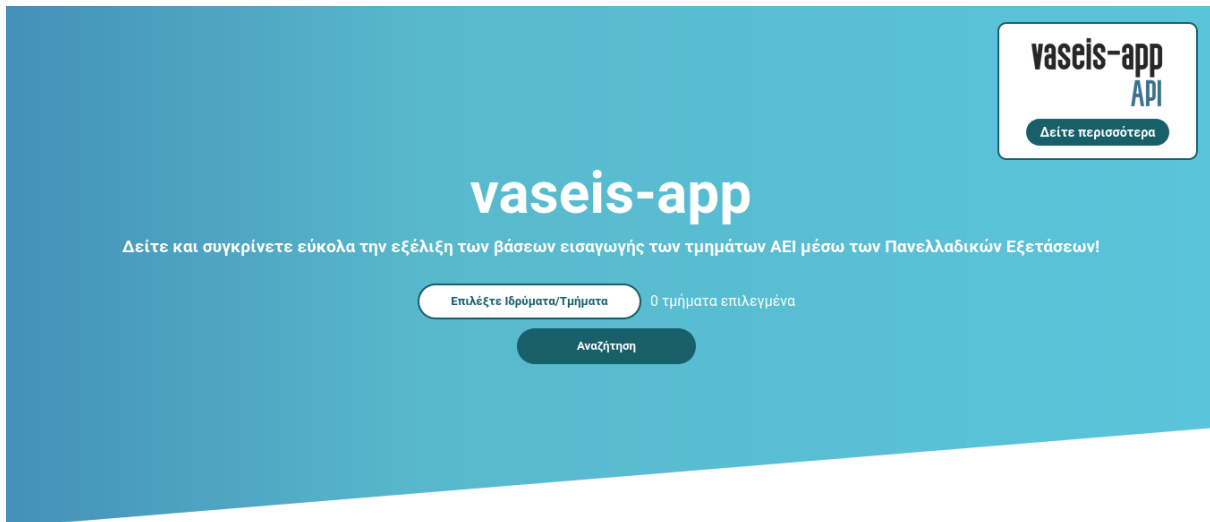
```
233 function buildSelect(select, minYear, maxYear) {
234     while (document.querySelector(selectors: `.${select}`).firstChild) {
235         document.querySelector(selectors: `.${select}`)
236             .removeChild(document.querySelector(selectors: `.${select}`).lastChild);
237     }
238     for (let i = minYear; i <= maxYear; i++) {
239         let option = document.createElement('option');
240         option.value = String(i);
241         option.innerHTML = String(i);
242         if (i === maxYear) {
243             option.selected = true;
244         }
245         document.querySelector(selectors: `.${select}`).appendChild(option);
246     }
247 }
```

Σχήμα 4.12: Ο αλγόριθμος της συνάρτησης buildSelect

## Κεφάλαιο 5ο: Παρουσίαση Εφαρμογής

### 5.0.1 Αρχική Σελίδα

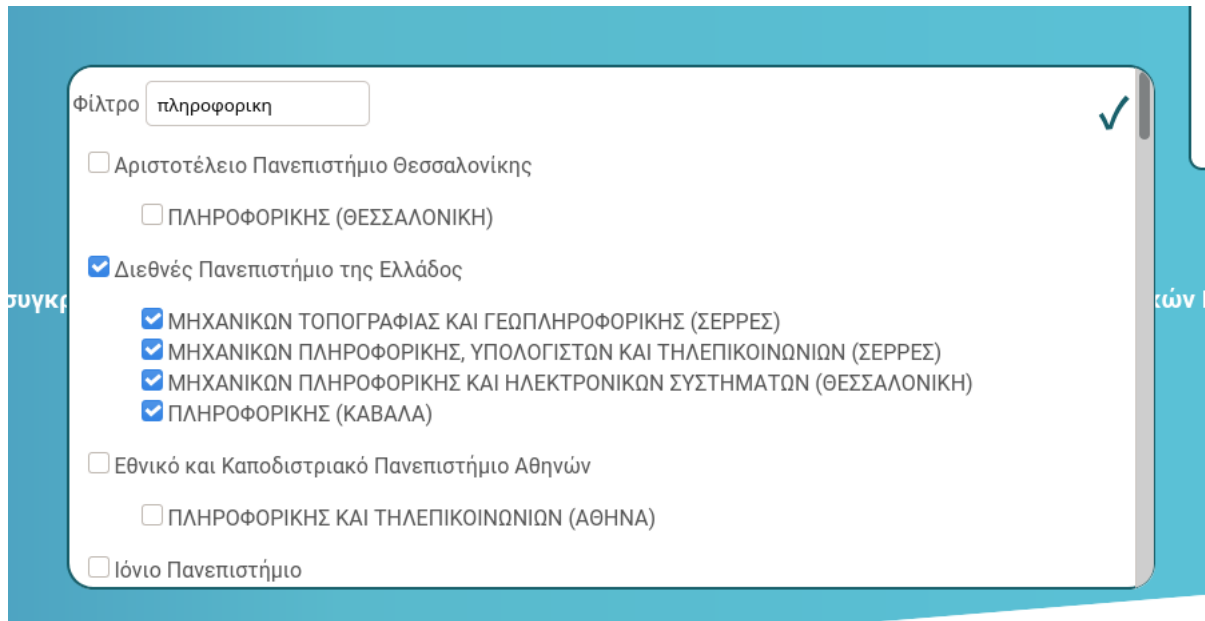
Η διαδικτυακή εφαρμογή, όπως αναφέρθηκε και παραπάνω χωρίζεται σε δύο σελίδες. Αυτές είναι η σελίδα επιλογής ή αρχική σελίδα και η σελίδα δεδομένων. Η σελίδα επιλογής χωρίζεται σε δύο τμήματα. Το πρώτο μέρος το οποίο βρίσκεται στο πάνω μέρος της σελίδας είναι το πρώτο πράγμα που αντικρίζει ο χρήστης όταν ανακτά τον ιστότοπο. Το πάνω μέρος θεωρείται και το κύριο λειτουργικά μέρος, καθώς από αυτό γίνεται η αλληλεπίδραση χρήστη και εφαρμογής για την ανάκτηση και προβολή των δεδομένων. Σε αυτό περιλαμβάνεται μια πολύ σύντομη περιγραφή με σκοπό την ενημέρωση του χρήστη, ένα παράθυρο ενημέρωσης για την ύπαρξη του διαδικτυακού API και το κουμπί επιλογής τμημάτων και το κουμπί της αναζήτησης.



Σχήμα 5.1: Το πρώτο μέρος της σελίδας επιλογής

Επιλέγοντας το κουμπί “Επιλέξτε Ιδρύματα/Τμήματα“ που φαίνεται στο κέντρο του Σχήματος 5.1, εμφανίζεται το παράθυρο επιλογής τμημάτων. Σε αυτό, ο χρήστης έχει τη δυνατότητα εφαρμογής φίλτρου χρησιμοποιώντας το πεδίο αναζήτησης, το οποίο φιλτράρει τα διαθέσιμα τμήματα και ιδρύματα. Στην περίπτωση, που το φίλτρο του χρήστη αντιστοιχεί στο όνομα ενός πανεπιστημίου τότε εμφανίζονται όλα τα τμήματα του πανεπιστημίου, διαφορετικά εμφανίζονται μόνο τα τμήματα που αντιστοιχούν στο φίλτρο. Ο χρήστης έχει την δυνατότητα επιλογής τμημάτων επιλέγοντας τα checkbox που βρίσκονται δίπλα από κάθε τμήμα και πανεπιστήμιο. Επιλέγοντας ένα πανεπιστήμιο, επιλέγονται αυτόματα και όλα τα τμήματα του. Το παράθυρο επιλογής μπορεί να κλείσει είτε επιλέγοντας το σύμβολο επιβεβαίωσης που φαίνεται στο πάνω αριστερό μέρος του Σχήματος 5.2, είτε πατώντας οπουδήποτε εκτός του παραθύρου.

Δίπλα από το κουμπί επιλογής τμημάτων, ο χρήστης μπορεί να ενημερωθεί για τον αριθμό των τμημάτων που έχει επιλεγμένα. Πατώντας το κουμπί “Δείτε περισσότερα“ που φαίνεται στο πάνω δεξιά μέρος του Σχήματος 5.1, ο χρήστης μεταφέρεται στο δεύτερο μέρος της σελίδας, το



Σχήμα 5.2: Το παράθυρο επιλογής τμημάτων της αρχικής σελίδας

οποίο θα αναλυθεί παρακάτω.

Το δεύτερο μέρος αποτελεί το σημείο ενημέρωσης του χρήστη σχετικά με την εφαρμογή. Σε αυτό το σημείο ο χρήστης μπορεί να ενημερωθεί για τα χαρακτηριστικά της διαδικτυακής εφαρμογής, τα οποία είναι η προβολή των δεδομένων των Πανελλαδικών Εξετάσεων, η σύγκριση των βάσεων εισαγωγής των τριτοβάθμιων ιδρυμάτων και τα στατιστικά προτιμήσεων. Ακόμα, ο χρήστης ενημερώνεται για την ύπαρξη και τα χαρακτηριστικά του διαδικτυακού API και του προσφέρεται ένας σύνδεσμος που οδηγεί στην ιστοσελίδα οδηγιών χρήσης του API. Τέλος, αναφέρεται ότι η διαδικτυακή εφαρμογή και το API είναι μια υπηρεσία που προσφέρεται από το Διεθνές Πανεπιστήμιο της Ελλάδος και δίνονται τρόποι επικοινωνίας με τον επιβλέποντα καθηγητή και τον επιμελητή της εργασίας.

### 5.0.2 Σελίδα Δεδομένων

Η Σελίδα Δεδομένων αποτελεί το κέντρο πληροφορίας της διαδικτυακής εφαρμογής. Περιέχει μεγάλο όγκο πληροφοριών και έγινε η προσπάθεια να οργανωθεί με τον καλύτερο δυνατό τρόπο, έτσι ώστε ο χρήστης να μπορεί να εύκολα να βρει όσες πληροφορίες χρειάζεται και χωρίς να χρειαστεί να επισκεφτεί παραπάνω σελίδες.

Η σελίδα χωρίζεται σε τρία σημεία. Το πρώτο είναι αυτό που βλέπει ο χρήστης μόλις μεταβεί σε αυτή και περιλαμβάνει το διάγραμμα σύγκρισης βάσεων. Το δεύτερο είναι αυτό που περιλαμβάνει τον μεγαλύτερο όγκο πληροφοριών για κάθε τμήμα και το τρίτο είναι η λίστα διαχείρισης τμημάτων.

Το πάνω μέρος της σελίδας αποτελείται από το διάγραμμα σύγκρισης των βάσεων εισαγωγής

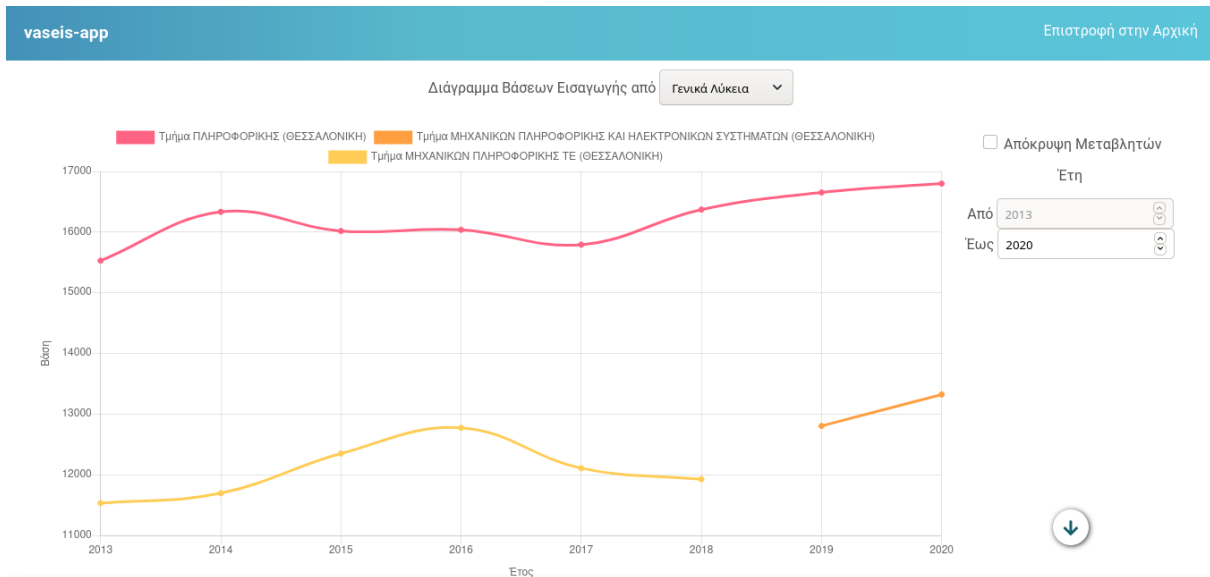


Σχήμα 5.3: Το δεύτερο μέρος της σελίδας επιλογής

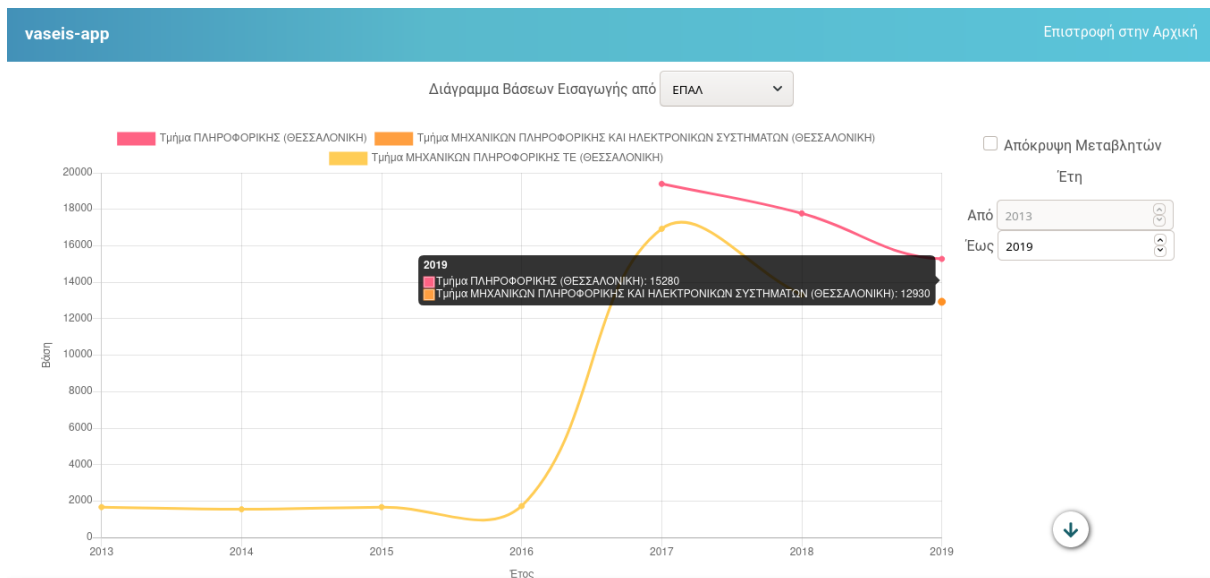
των τμημάτων που έχουν επιλεχθεί. Αυτό γίνεται με τη χρήση ενός διαγράμματος γραμμής, με άξονες το έτος και τη βάση εισαγωγής. Κάθε γραμμή αντιστοιχεί σε ένα τμήμα και δίνεται μια ποικιλία χρωμάτων για την ευκολότερη ανάγνωση του διαγράμματος. Στο πάνω μέρος του διαγράμματος προβάλλονται οι μεταβλητές του γραφήματος και το χρώμα που αντιστοιχεί σε κάθε μεταβλητή και δίπλα από αυτό δίνεται η επιλογή απόκρυψης των μεταβλητών με σκοπό την εξοικονόμηση χώρου σε περίπτωση φόρτωσης πολλών τμημάτων ή πρόσβασης από κινητό.

Ο χρήστης έχει τη δυνατότητα να αλληλεπιδράσει με το διάγραμμα σύγκρισης των βάσεων εισαγωγής με διάφορους τρόπους. Μετακινώντας τον κέρσορα πάνω από τις γραμμές του διαγράμματος εμφανίζεται ένα παράθυρο που περιλαμβάνει τις τιμές των βάσεων εισαγωγής για όλα τα τμήματα. Ακόμα μέσω των δύο εισόδων που προσφέρονται δεξιά του γραφήματος, έχει τη δυνατότητα να μεγαλώσει ή να μικρύνει των άξονα των ετών προσθέτοντας ή αφαιρώντας έτη αντίστοιχα. Τέλος, στην κορυφή της σελίδας ο χρήστης μπορεί να επιλέξει τον τύπο του σχολείου για το οποίο θα εμφανίζονται τα δεδομένα στη σελίδα. Η αλλαγή του τύπου του σχολείου μεταξύ άλλων ανανεώνει και το διάγραμμα σύγκρισης.

Ως ένα επιπλέον χαρακτηριστικό για το διάγραμμα των βάσεων, επιλέχθηκε η προσθήκη της επιλογής απόκρυψης των μεταβλητών του γραφήματος. Αυτή η επιλογή ενεργοποιείται αυτόματα, όταν ο χρήστης έχει επιλέξει για αναζήτηση δεδομένων περισσότερα από δέκα τριτοβάθμια ιδρύματα, δίνοντας του όμως την επιλογή να την απενεργοποιήσει. Η προσθήκη αυτού του χαρακτηριστικού έγινε, για λόγους οικονομίας χώρου σε μικρότερες οθόνες όπως αυτές των κινητών τηλεφώνων. Τέλος, προστέθηκε και ένα κουμπί, το οποίο περιέχει ένα βέλος που κατευθύνεται προς τα κάτω. Αυτό, λειτουργεί ως οδηγός, ενημερώνοντας τον χρήστη για το υπόλοιπο περιεχόμενο της σελίδας, αλλά και ως εύκολος τρόπος να μεταφερθεί εκεί, καθώς πατώντας τον η σελίδα αυτόματα κατεβαίνει στο τμήμα των λεπτομερών πληροφοριών.



Σχήμα 5.4: Το πάνω μέρος της σελίδας δεδομένων



Σχήμα 5.5: Το πάνω μέρος της σελίδας δεδομένων μετά από αλληλεπίδραση του χρήστη

Το δεύτερο μέρος της σελίδας των δεδομένων χωρίζεται σε δύο τμήματα. Το πρώτο είναι το τμήμα που περιλαμβάνει λεπτομερώς της πληροφορίες και τα διαγράμματα στατιστικών του επιλεγμένου τμήματος και το δεύτερο είναι το τμήμα που δρα ως κέντρο ελέγχου των δεδομένων της σελίδας.

Στο τμήμα των λεπτομερών πληροφοριών, ο χρήστης αρχικά βλέπει το όνομα του επιλεγμένου τμήματος και το τριτοβάθμιο ίδρυμα στο οποίο ανήκει. Από κάτω έχει πρόσβαση στα δεδομένα των βάσεων εισαγωγής του τμήματος. Τα δεδομένα αυτά περιλαμβάνουν τη βάση εισαγωγής του πρώτου και τελευταίου και τον αριθμό των εισακτέων για τις κατηγορίες 90% και 10%. Τα δεδομένα αφορούν το τελευταίο έτος για το οποίο υπάρχουν διαθέσιμες πληροφορίες, ενώ ο χρήστης έχει τη δυνατότητα να επιλέξει κάποιο από τα υπόλοιπα διαθέσιμα έτη για να προβάλει

τα δεδομένα εκείνης της χρονιάς. Η αλλαγή του τύπου του σχολείου στην κορυφή της σελίδας, ενημερώνει αυτόματα και τα προβαλλόμενα δεδομένα.

## Τμήμα ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ (ΘΕΣΣΑΛΟΝΙΚΗ)

Διεθνές Πανεπιστήμιο της Ελλάδος

### Βάσεις

Για το έτος

90%

Βαθμός Πρώτου: 16125 Βαθμός Τελευταίου: 13325 Εισακτέοι: 145

10%

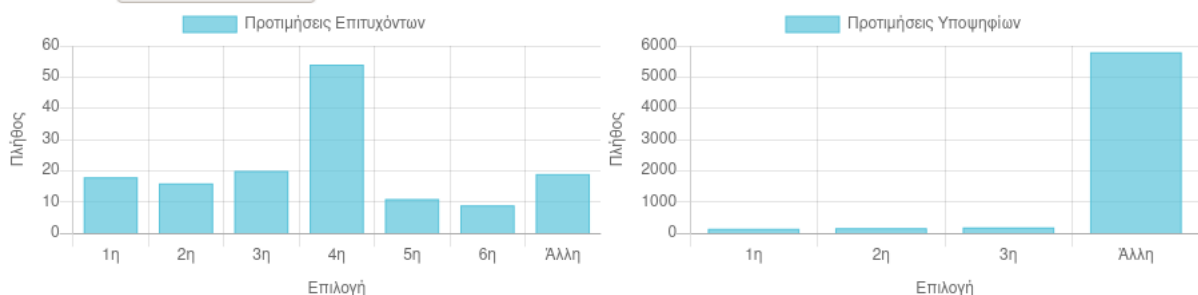
2018: Βαθμός Πρώτου: 15728 Βαθμός Τελευταίου: 12156 Εισακτέοι: 7  
2019: Βαθμός Πρώτου: 16750 Βαθμός Τελευταίου: 14051 Εισακτέοι: 11

Σχήμα 5.6: Το τμήμα πληροφοριών βάσεων εισαγωγής της σελίδας δεδομένων

Ύστερα από το τμήμα των βάσεων εισαγωγής ακολουθεί των τμήματα των στατιστικών προτιμήσεων. Εδώ, επιλέχθηκε να γίνει η προβολή των δεδομένων με τη χρήση διαγραμμάτων, σε αντίθεση με την προβολή σε μορφή κειμένου που έγινε για τις βάσεις εισαγωγής. Τα στατιστικά αντιπροσωπεύονται από δύο γραφήματα τύπου μπάρας, τα οποία έχουν στον οριζόντιο άξονα ως κλίμακα το πλήθος των επιλογών και στον κατακόρυφο τη σειρά επιλογής. Το πρώτο αφορά τα στατιστικά προτιμήσεων των επιτυχόντων, δηλαδή των επιλογών αυτών, οι οποίοι πέτυχαν στο επιλεγμένο τμήμα. Το δεύτερο αφορά τα στατιστικά των υποψηφίων, τα οποία περιλαμβάνουν όλους όσους είχαν δηλώσει το επιλεγμένο τμήμα στο μηχανογραφικό δελτίο τους, ανεξάρτητα από το αν πέτυχαν σε αυτό ή όχι. Αντίστοιχα με το τμήμα που αφορά τις βάσεις εισαγωγής, έτσι και στο τμήμα των στατιστικών προσφέρεται η δυνατότητα επιλογής του έτους που θα αφορούν τα δεδομένα που προβάλλονται και η εναλλαγή μεταξύ Γενικού ή Επαγγελματικού Λυκείου ανανεώνει αυτόματα τα δεδομένα.

### Στατιστικά

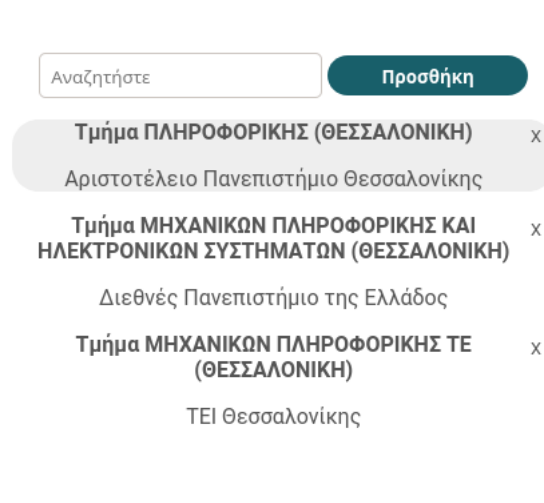
Για το έτος



Σχήμα 5.7: Το τμήμα πληροφοριών των στατιστικών προτιμήσεων της σελίδας δεδομένων

Το τελευταίο τμήμα της σελίδας δεδομένων αφορά το κέντρο διαχείρισης των τμημάτων προς αναζήτηση. Αρχικά σε αυτό προβάλλεται στον χρήστη η λίστα των τμημάτων, τα οποία επέλεξε στην αρχική σελίδα. Κατά τη φόρτωση της σελίδας, ορίζεται ως επιλεγμένο τμήμα αυτό που βρίσκεται πρώτο στη λίστα των τμημάτων.

Για κάθε στοιχείο της λίστας, ο χρήστης έχει τη δυνατότητα να το επιλέξει, έτσι ώστε να προβληθούν αναλυτικά τα δεδομένα για αυτό στο τμήμα των λεπτομερειών. Το επιλεγμένο τμήμα αποκτά διαφορετικό χρώμα παρασκηνίου, προκειμένου μπορεί να καταλάβει ποιο είναι το επιλεγμένο τμήμα και να καταστεί ευκολότερη για τον χρήστη η δυνατότητα εναλλαγής τμημάτων. Επίσης, μέσω ενός κουμπιού που υπάρχει σε κάθε στοιχείο της λίστας, ο χρήστης μπορεί να αφαιρέσει το τμήμα της επιλογής του. Σε περίπτωση που αφαιρεθούν όλα τα τμήματα από τη λίστα, τότε ο χρήστης ανακατευθύνεται στην αρχική σελίδα.

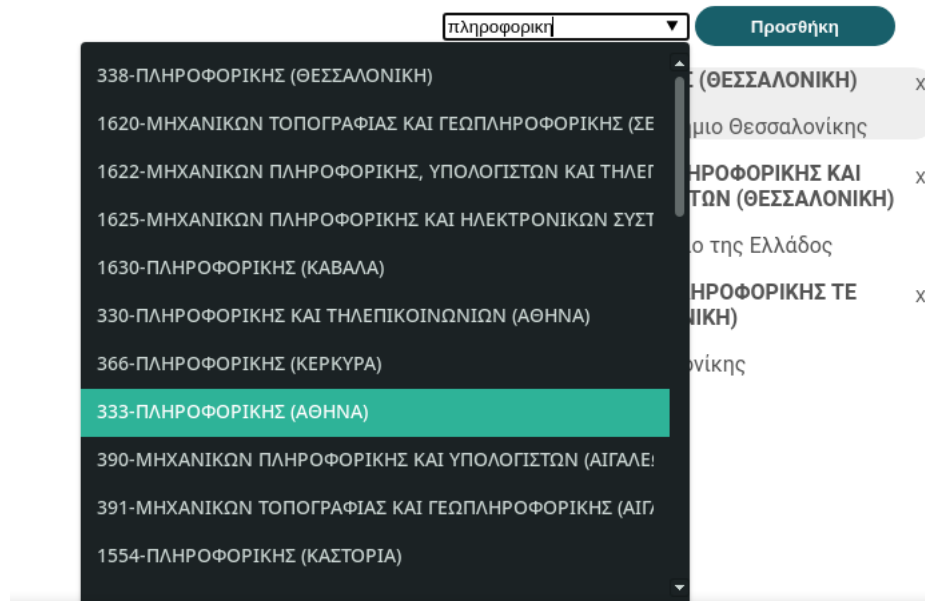


Σχήμα 5.8: Το κέντρο διαχείρισης επιλεγμένων τμημάτων

Η εφαρμογή δίνει τη δυνατότητα προσθήκης επιπλέον τμημάτων από τη σελίδα των δεδομένων, χωρίς την ανάγκη επιστροφής στην αρχική σελίδα. Αυτό έχει ως σκοπό να κάνει πιο εύκολη τη χρήση της εφαρμογής, μειώνοντας την περιττή αλληλεπίδραση του χρήστη με την εφαρμογή. Η προσθήκη τμημάτων πραγματοποιείται με τη χρήση ενός πεδίου αναζήτησης, στο οποίο με την είσοδο του κωδικού του τμήματος ή με την είσοδο του μέρους του ονόματος του τμήματος, προτείνονται τα πιθανά επιθυμητά τμήματα. Αυτά, εμφανίζονται ως λίστα και ο χρήστης με τη χρήση του πληκτρολογίου ή επιλέγοντας ένα με το ποντίκι το προσθέτει στη λίστα.

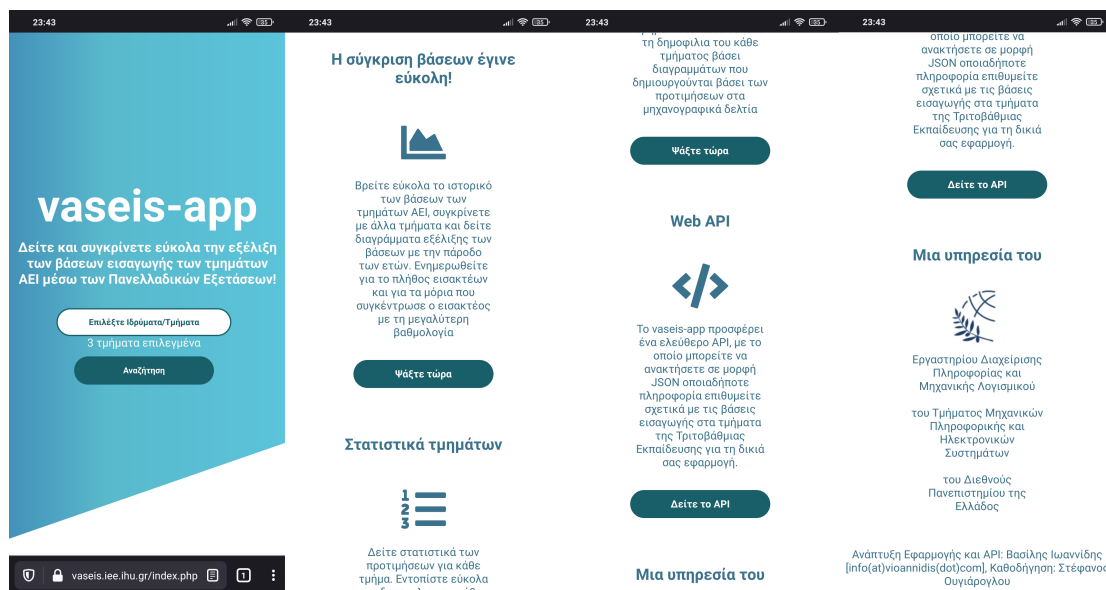
Κατά τη διάρκεια της ανάπτυξης της διαδικτυακής εφαρμογής, θεωρήθηκε απαραίτητη η υποστήριξη συσκευών με μικρό μέγεθος οθόνης, όπως τα κινητά τηλέφωνα. Για αυτό τον λόγο η εφαρμογή, μπορεί να χρησιμοποιηθεί εξίσου εύκολα σε συσκευές με κάθε είδους οθόνη. Για την επίτευξη αυτού του στόχου, η διάταξη της εφαρμογής σε ορισμένες περιπτώσεις χρειάστηκε να αλλάξει, προκειμένου να διατηρηθεί το επίπεδο της ευχρηστίας.

Στην αρχική σελίδα, η περιοχή επιλογής τμημάτων δεν επιδείχθηκε κάποιας αλλαγής. Η περιοχή ανάλυσης των χαρακτηριστικών της εφαρμογής και του διαδικτυακού API, σε συσκευές



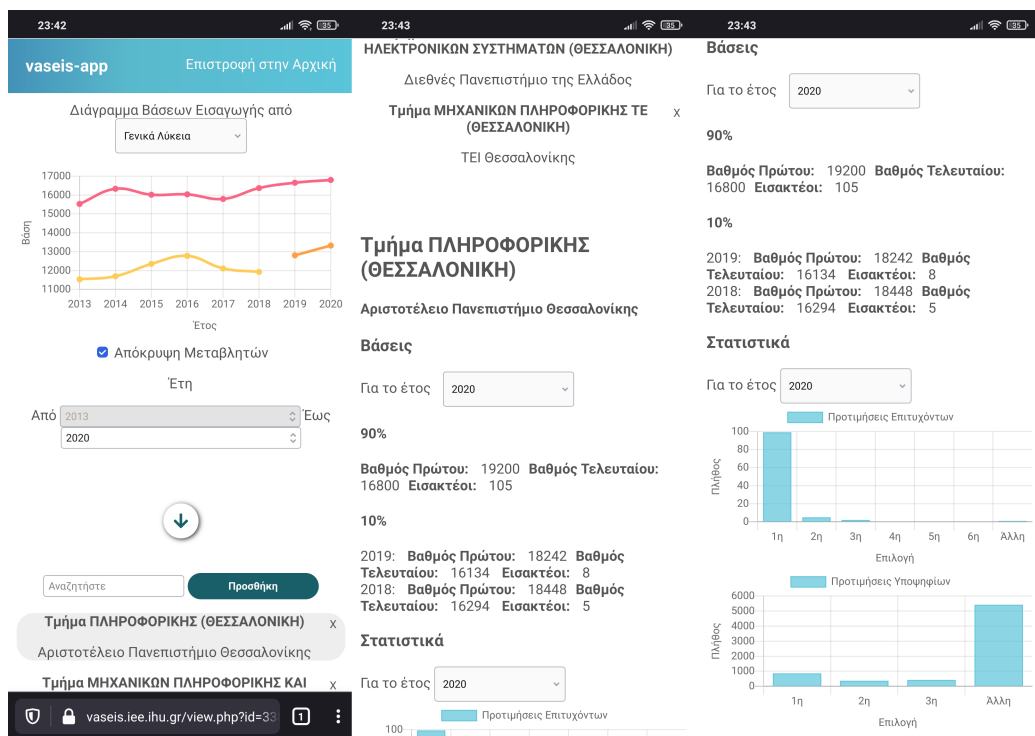
Σχήμα 5.9: Η λειτουργία αναζήτησης για την προσθήκη νέων τμημάτων

με μεγάλη οθόνη ακολουθούσε διάταξη τεσσάρων στηλών, για συσκευές με μικρές οθόνες, η διάταξη του περιεχομένου γίνεται σε μία στήλη.



Σχήμα 5.10: Η αρχική σελίδας της διαδικτυακής εφαρμογής σε συσκευή με μικρή οθόνη

Η σελίδα δεδομένων ήταν αυτή που χρειάστηκε να παραμετροποιηθεί περισσότερο η διάταξη της. Το τμήμα του διαγράμματος σύγκρισης των βάσεων εισαγωγής στις μεγαλύτερες οθόνες ακολουθεί διάταξη δύο στηλών, ενώ στις μικρότερες διάταξη μίας στήλης. Αντίστοιχα στο δεύτερο τμήμα οι δύο στήλες γίνονται πάλι μία. Επιπλέον, επιλέχθηκε να αλλάξει και η διάταξη των διαγραμμάτων παρουσίασης των στατιστικών προτίμησης, όπου και αυτά αναλόγως το μέγεθος οθόνης διατάσσονται με μία ή δύο στήλες.



Σχήμα 5.11: Η σελίδα δεδομένων της διαδικτυακής εφαρμογής σε συσκευή με μικρή οθόνη

## Κεφάλαιο 6ο: Κλίμακα Ευχρηστίας Συστήματος

Στα πλαίσια της εργασίας και με την ολοκλήρωση της ανάπτυξης της διαδικτυακής εφαρμογής, εκπονήθηκε έρευνα με σκοπό την πληροφόρηση του επιμελητή για την ευχρηστία της εφαρμογής. Για την εξαγωγή αυτών των συμπερασμάτων χρησιμοποιήθηκε η μέθοδος της κλίμακας ευχρηστίας συστήματος (System Usability Scale).

Η κλίμακα ευχρηστίας συστήματος αποτελείται από δέκα προτάσεις, στις οποίες ο χρήστης καλείται να απαντήσει δηλώνοντας τον βαθμό που συμφωνεί ή διαφωνεί. Οι απαντήσεις καλύπτουν ένα εύρος μεταξύ του “Συμφωνώ Απολύτως” και “Διαφωνώ Απολύτως”. Οι προτάσεις εναλλάσσονται μεταξύ αρνητικών και θετικών και αρνητικών με σκοπό να προτρέψει το άτομο που απαντάει να σκεφτεί περισσότερο την απάντηση του πριν την υποβάλλει. Οι προτάσεις που καλείται να απαντήσει ο χρήστης είναι οι εξής [35]:

1. Πιστεύω ότι θα χρησιμοποιούσα την εφαρμογή συχνά.
2. Πιστεύω ότι η εφαρμογή είναι αχρείαστα περίπλοκη.
3. Πιστεύω ότι η εφαρμογή ήταν εύκολη στη χρήση.
4. Πιστεύω ότι θα χρειαζόμουν βοήθεια για να χρησιμοποιήσω την εφαρμογή.
5. Πιστεύω ότι οι διάφορες λειτουργίες της εφαρμογής ήταν καλά ενσωματωμένες.
6. Πιστεύω ότι η σχεδίαση της εφαρμογής ήταν μη συνεπής.
7. Πιστεύω ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν εύκολα την εφαρμογή.
8. Πιστεύω ότι η εφαρμογή ήταν δύσχρηστη.
9. Αισθάνθηκα σίγουρος κατά τη χρήση της εφαρμογής.
10. Έπρεπε να μάθω πολλά πράγματα για να χρησιμοποιήσω την εφαρμογή.

Το ερωτηματολόγιο περιλάμβανε μία επιπλέον ερώτηση σχετικά με το κυριότερο χαρακτηριστικό που απουσιάζει από τη διαδικτυακή εφαρμογή, με σκοπό την κατανόηση των αναγκών των χρηστών όταν χρησιμοποιούν την εφαρμογή. Οι επιλογές που δόθηκαν στους χρήστες ήταν οι εξής:

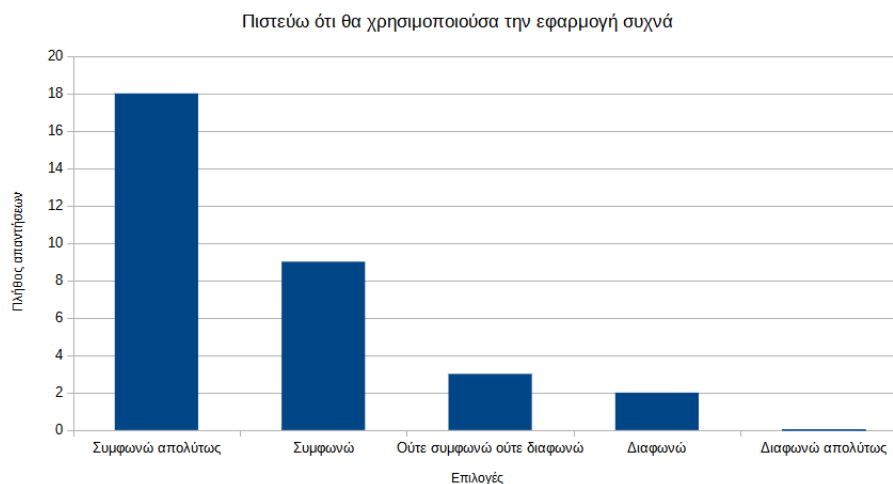
1. Ορισμός τμημάτων ως αγαπημένα.
2. Αναζήτηση τμημάτων με κριτήριο τη βάση εισαγωγής.

3. Εξαγωγή διαγραμμάτων για χρήση σε τρίτες εφαρμογές.
4. Εφαρμογή για έξυπνα τηλέφωνα.

Το ερωτηματολόγιο δημιουργήθηκε ηλεκτρονικά και διαμοιράστηκε μέσω του διαδικτύου σε ένα τυχαίο δείγμα χρηστών. Οι χρήστες ενημερώθηκαν για το περιεχόμενο της έρευνας, κλήθηκαν να χρησιμοποιήσουν την εφαρμογή και ύστερα να απαντήσουν στο ερωτηματολόγιο. Συγκεντρώθηκαν οι απαντήσεις 33 διαφορετικών χρηστών, οι οποίες εξήχθησαν σε ένα αρχείο τύπου υπολογιστικού φύλλου και παράχθηκαν διαγράμματα τα οποία παρουσιάζουν τα αποτελέσματα.

### 6.1 Πιστεύω ότι θα χρησιμοποιούσα την εφαρμογή συχνά

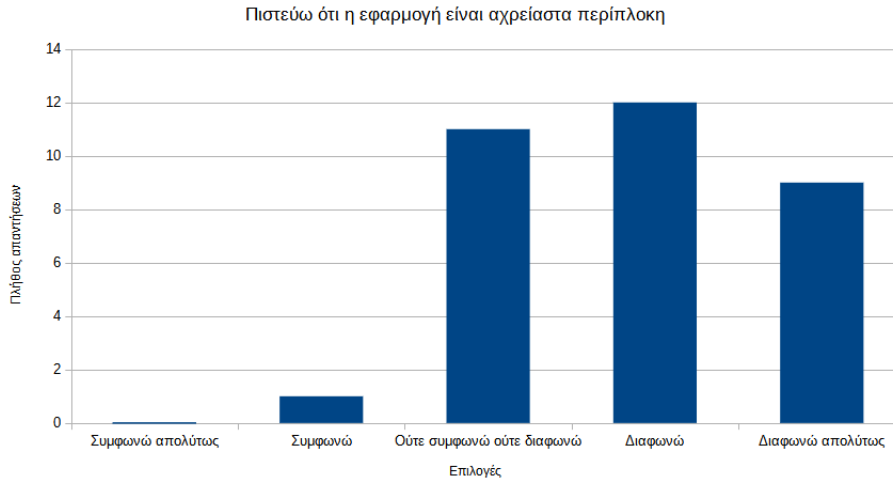
Οι πλειοψηφία των χρηστών που συμπλήρωσαν το ερωτηματολόγιο επέλεξε τις επιλογές “Συμφωνώ Απολύτως” και “Συμφωνώ” στην πρόταση “Πιστεύω ότι θα χρησιμοποιούσα την εφαρμογή συχνά”. Για τους χρήστες που απάντησαν διαφορετικά, η απάντηση τους θα μπορούσε να αποδοθεί στο γεγονός ότι μπορεί να επιθυμούν να χρησιμοποιήσουν ένα άλλο μέσο για αυτόν τον σκοπό ή ότι δεν τους απασχολεί το αντικείμενο της εφαρμογής.



Σχήμα 6.1: Οι απαντήσεις των χρηστών στη φράση “Πιστεύω ότι θα χρησιμοποιούσα την εφαρμογή συχνά”

### 6.2 Πιστεύω ότι η εφαρμογή είναι αχρείαστα περίπλοκη

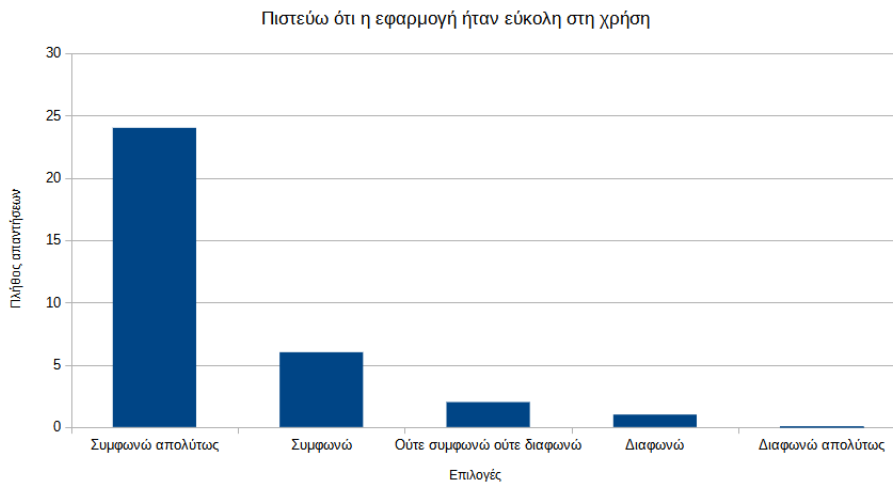
Οι απαντήσεις των χρηστών για την πρόταση “Πιστεύω ότι η εφαρμογή είναι αχρείαστα περίπλοκη” μοιράζονται στις επιλογές “Συμφωνώ απόλυτα”, “Συμφωνώ” και “Ούτε συμφωνώ ούτε διαφωνώ”. Το συμπέρασμα που μπορεί να ληφθεί από αυτό είναι ότι αν και οι περισσότεροι από τους μισούς δε βρίσκουν περίπλοκη την εφαρμογή, ένα σημαντικό μέρος θεωρεί ότι υπάρχει ένας βαθμός περιπλοκότητας.



Σχήμα 6.2: Οι απαντήσεις των χρηστών στη φράση “Πιστεύω ότι η εφαρμογή είναι αχρείαστα περίπλοκη”

### 6.3 Πιστεύω ότι η εφαρμογή ήταν εύκολη στη χρήση

Η πρόταση “Πιστεύω ότι η εφαρμογή ήταν εύκολη στη χρήση”, έλαβε από την πλειοψηφία των χρηστών την απάντηση “Συμφωνώ απολύτως”. Αυτό δείχνει ότι αν και ένα μέρος των χρηστών πιστεύει ότι υπάρχει ένας βαθμός περιπλοκότητας στην εφαρμογή, αυτό δεν επηρεάζει αρκετά ώστε να θεωρηθεί δύσκολη στη χρήση.

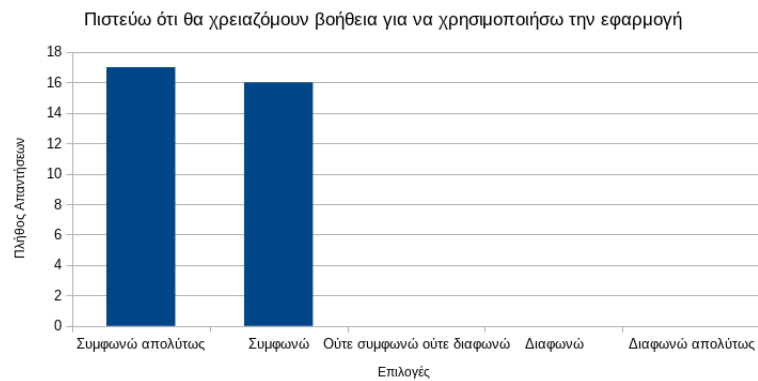


Σχήμα 6.3: Οι απαντήσεις των χρηστών στη φράση “Πιστεύω ότι η εφαρμογή ήταν εύκολη στη χρήση”

### 6.4 Πιστεύω ότι θα χρειαζόμουν βοήθεια για να χρησιμοποιήσω την εφαρμογή

Στην φράση “Πιστεύω ότι θα χρειαζόμουν βοήθεια για να χρησιμοποιήσω την εφαρμογή”, όλοι οι χρήστες που απάντησαν έδωσαν την απάντηση “Συμφωνώ απολύτως” ή “Συμφωνώ”. Αυ-

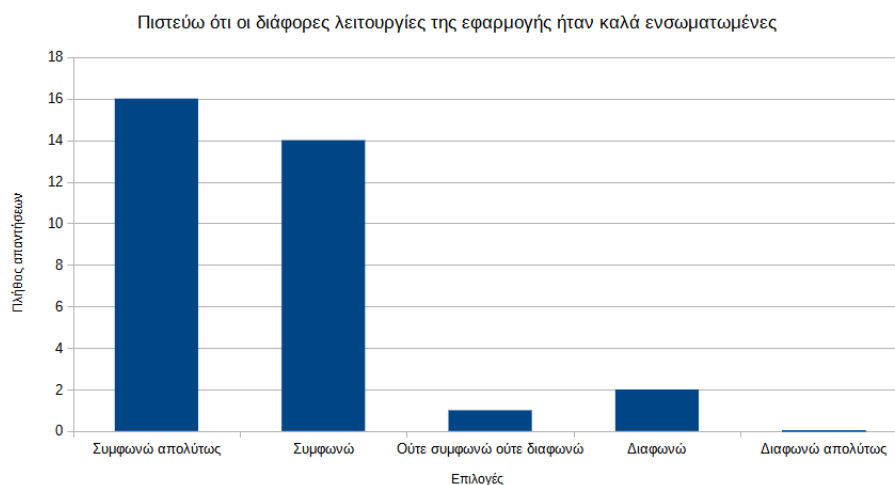
τό σηματοδοτεί ότι ο εφαρμογή καθοδηγεί από μόνη της τον χρήστη για τα ποια βήματα θα ακολουθήσει για να ολοκληρώσει τις ενέργειες που επιθυμεί.



Σχήμα 6.4: Οι απαντήσεις των χρηστών στη φράση “Πιστεύω ότι θα χρειάζομαι βοήθεια για να χρησιμοποιήσω την εφαρμογή”

## 6.5 Πιστεύω ότι οι διάφορες λειτουργίες της εφαρμογής ήταν καλά ενσωματωμένες

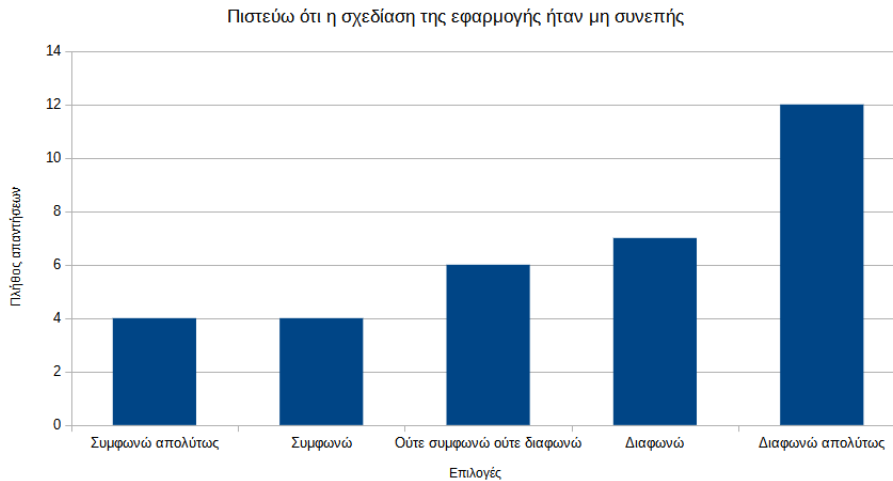
Στην πρόταση “Πιστεύω ότι οι διάφορες λειτουργίες της εφαρμογής ήταν καλά ενσωματωμένες” η πλειοψηφία των χρηστών απάντησε θετικά. Από τις απαντήσεις μπορεί κανείς να συμπεράνει ότι οι χρήστες θεωρούν ότι οι λειτουργίες της εφαρμογής λειτουργούν αρμονικά σε σχέση με το γραφικό περιβάλλον.



Σχήμα 6.5: Οι απαντήσεις των χρηστών στην πρόταση “Πιστεύω ότι οι διάφορες λειτουργίες της εφαρμογής ήταν καλά ενσωματωμένες”

## 6.6 Πιστεύω ότι η σχεδίαση της εφαρμογής ήταν μη συνεπής

Οι απαντήσεις των χρηστών σε αυτή την πρόταση ήταν αρκετά μοιρασμένες. Παρόλο που περισσότερες από τις μισές απαντήσεις διαφωνούσαν με την πρόταση, ο μεγάλος αριθμός θετικών απαντήσεων δείχνει ότι η εμφάνιση και η σχεδίαση της εφαρμογής είναι αμφιλεγόμενη και πιθανότατα να χρειάζεται αναθεωρήσεις.



Σχήμα 6.6: Οι απαντήσεις των χρηστών στην πρόταση “Πιστεύω ότι η σχεδίαση της εφαρμογής ήταν μη συνεπής”

## 6.7 Πιστεύω ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν εύκολα την εφαρμογή

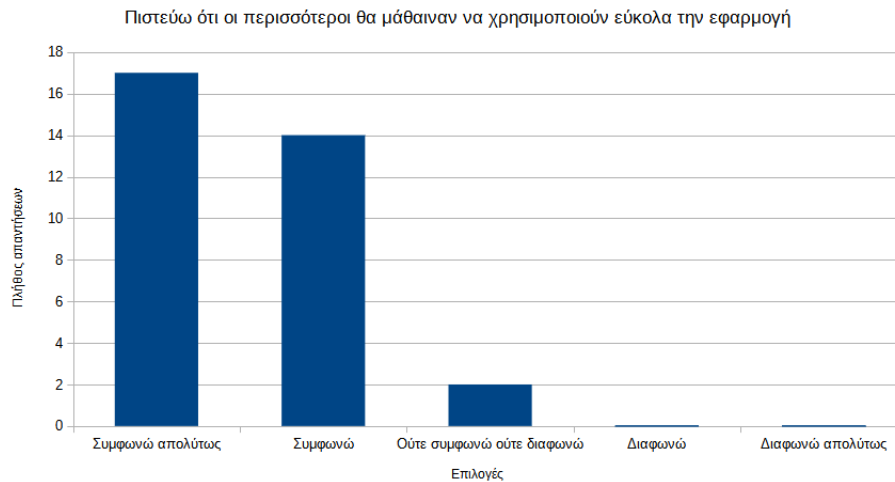
Στην πρόταση “Πιστεύω ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν εύκολα την εφαρμογή” οι απαντήσεις ήταν κατά πλειοψηφία θετικές, αντίστοιχα με την ερώτηση που αφορούσε το κατά πόσο οι χρήστες θα χρειαζόντουσαν βοήθεια για να τη χρησιμοποιήσουν.

## 6.8 Πιστεύω ότι η εφαρμογή ήταν δύσχρηστη

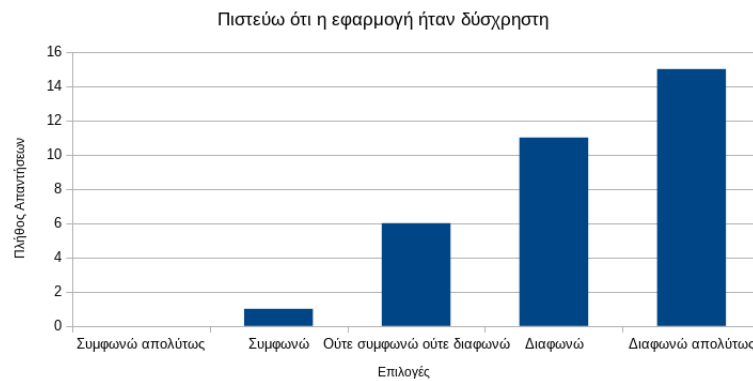
Οι απαντήσεις των χρηστών για την πρόταση “Πιστεύω ότι η εφαρμογή ήταν δύσχρηστη” ήταν, αναμενόμενα, αντιστρόφως ανάλογες αυτών της ερώτησης που αφορούσε την ευκολία χρήσης της εφαρμογής.

## 6.9 Αισθάνθηκα σίγουρος κατά τη χρήση της εφαρμογής

Οι απαντήσεις των χρηστών σε αυτή την πρόταση ήταν σχεδόν αποκλειστικά θετικές. Από τις απαντήσεις των χρηστών μπορεί να συμπεράνει κάποιος ότι η σχεδίαση της εφαρμογής βοηθά τους χρήστες να μην αμφιβάλουν για τις κινήσεις που θα κάνουν.



Σχήμα 6.7: Οι απαντήσεις των χρηστών στην πρόταση “Πιστεύω ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν εύκολα την εφαρμογή”



Σχήμα 6.8: Οι απαντήσεις των χρηστών στην πρόταση “Πιστεύω ότι η εφαρμογή ήταν δύσκολη”

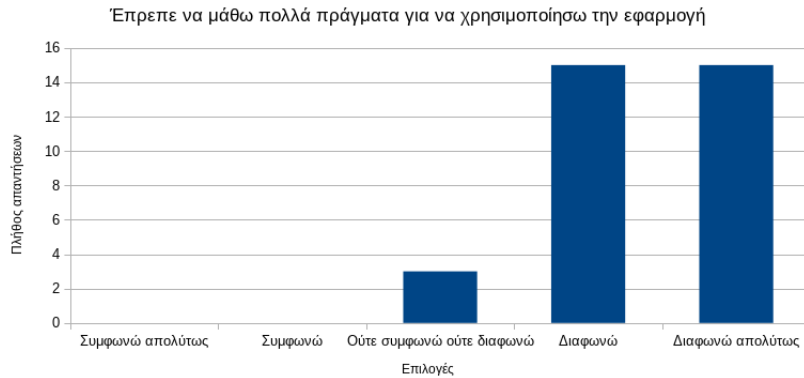


Σχήμα 6.9: Οι απαντήσεις των χρηστών στην πρόταση “Αισθάνθηκα σίγουρος κατά τη χρήση της εφαρμογής”

## 6.10 Έπρεπε να μάθω πολλά πράγματα για να χρησιμοποιήσω την εφαρμογή

Οι απαντήσεις των χρηστών για αυτήν την ερώτηση ακολουθούν το μοτίβο αντίστοιχων προτάσεων που έχουν προηγηθεί. Οι χρήστες κατά μεγάλη πλειοψηφία διαφωνούν με την παραπάνω

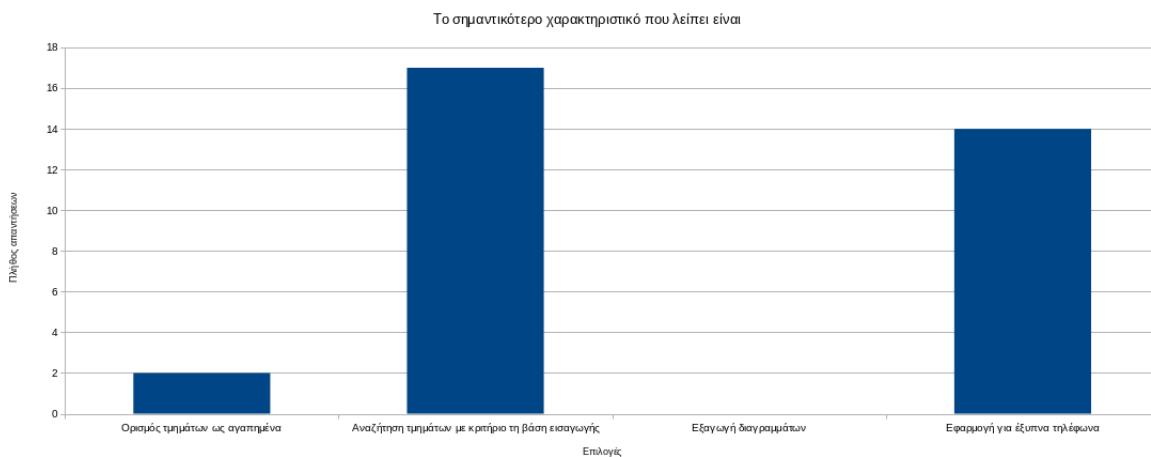
πρόταση, κάτι που οδηγεί στο συμπέρασμα ότι η εφαρμογή είναι εύκολη και απλή στη χρήση.



Σχήμα 6.10: Οι απαντήσεις των χρηστών στην πρόταση “Έπρεπε να μάθω πολλά πράγματα για να χρησιμοποιήσω την εφαρμογή”

### 6.11 Το σημαντικότερο χαρακτηριστικό της εφαρμογής που λείπει

Οι απαντήσεις των χρηστών που κλήθηκαν να απαντήσουν σε αυτή την ερώτηση, μοιράστηκαν στις επιλογές “Αναζήτηση κριτηρίων με κριτήριο τη βάση εισαγωγής” και “Εφαρμογή για έξυπνα τηλέφωνα”. Οι δύο επιλογές ήταν από τα κυριότερα χαρακτηριστικά που θα μπορούσαν να υλοποιηθούν μελλοντικά.



Σχήμα 6.11: Οι απαντήσεις των χρηστών στην πρόταση “Το σημαντικότερο χαρακτηριστικό της εφαρμογής που λείπει”

### 6.12 Επίλογος

Στο κεφάλαιο αυτό, παρουσιάστηκαν οι ιστορίες χρήστη που δημιουργήθηκαν προτού ξεκινήσει η εφαρμογή, με σκοπό τον ορισμό των χαρακτηριστικών και απαιτήσεων της εφαρμογής. Έγινε μία αναλυτική παρουσίαση επιλεγμένων σημείων του πηγαίου κώδικα της εφαρμογής και τέλος παρουσιάστηκαν οι λειτουργίες και τα χαρακτηριστικά της.

## Κεφάλαιο 7ο: Συμπεράσματα και προτάσεις βελτίωσης

### 7.1 Συμπεράσματα

Το Υπουργείο Παιδείας και Θρησκευμάτων κάθε χρόνο δημοσιεύει τα δεδομένα των βάσεων εισαγωγής και τα στατιστικά για την εισαγωγή στην τριτοβάθμια εκπαίδευση. Ο χώρος τα οποία αυτά δημοσιεύονται πρόκειται για τη σελίδα ανακοινώσεων του Υπουργείου, κάτι που καθιστά δύσκολη την αναζήτηση των αρχείων. Παράλληλα, οι πολλές κατηγορίες στις οποίες χωρίζονται τα δεδομένα καθιστά ακόμα πιο δύσκολη την οργάνωση και αναζήτηση σε αυτά τα αρχεία.

Ο σκοπός αυτής της εργασίας, ήταν να βελτιώσει όσο το δυνατόν περισσότερο για τον ερευνητή, προγραμματιστή ή και οποιοδήποτε άλλο την ανάκτηση, επεξεργασία και σύγκριση αυτών των δεδομένων. Αυτό το πραγματοποίησε με την ανάπτυξη των εργαλείων που επιτελούν αυτόν τον σκοπό.

Η συγκέντρωση όλων των δεδομένων σε ένα σημείο και η επεξεργασία τους, έτσι ώστε για κάθε έτος να αντιστοιχεί ένα αρχείο για τις βάσεις εισαγωγής και ένα αρχείο για τα στατιστικά προτιμήσεων αποτέλεσε το πρώτο βήμα. Με αυτόν τρόπο, επιτεύχθηκε ο στόχος της εύκολης ανάκτησης των αρχείων, τα οποία περιλαμβάνουν όλα τα απαραίτητα δεδομένα.

Η ανάπτυξη ενός διαδικτυακού API, αποτέλεσε ένα από τα κύρια σημεία αυτής της εργασίας. Με την υλοποίησή του, δημιουργήθηκε πλέον ένας εύκολος τρόπος, ένας ερευνητής ή προγραμματιστής να ανακτήσει τα δεδομένα που επιθυμεί με εύκολο τρόπο. Το API που αναπτύχθηκε, με τη χρήση απλών και οργανωμένων υπερσυνδέσμων προσφέρει τα δεδομένα σε δομή εύκολη προς ανάγνωση από ανθρώπους και υπολογιστές. Υλοποιήθηκαν οι κατάλληλες λειτουργίες, που επιτρέπουν την ανάκτηση δεδομένων για όλες τους πιθανούς τύπους δεδομένων που θα ήθελε κάποιος να αποκτήσει πρόσβαση. Επίσης, δόθηκε και η δυνατότητα αναζήτησης με χρήση κριτηρίων για την ευκολότερη και πιο αποδοτική ανάκτηση δεδομένων.

Η υλοποίηση μίας διαδικτυακής εφαρμογής ήταν επίσης ένα σημαντικό μέρος της εργασίας. Σκοπό της εφαρμογής ήταν να αξιοποιήσει μέρος του API, με σκοπό την ανάκτηση δεδομένων χωρίς τη χρήση εργαλείων ή γλωσσών προγραμματισμού. Με αυτό τον τρόπο, δόθηκε η δυνατότητα στον καθένα να μπορεί να αναζητήσει εύκολα τα δεδομένα που επιθυμεί, δίχως να χρειάζεται να είναι εξοικειωμένος με τις τεχνολογίες του διαδικτύου και την πληροφορική γενικότερα. Η υλοποίηση της έδωσε τη δυνατότητα επιλογής τμημάτων, σύγκρισης των βάσεων εισαγωγής τους και προβολής όλων των δεδομένων για το καθένα μέσω ενός εύχρηστου γραφικού περιβάλλοντος.

Η ανάπτυξη αυτών των εργαλείων, δημιούργησε μία πιο ελεύθερη και οργανωμένη πρόσβαση στα δεδομένα των Πανελλαδικών Εξετάσεων. Έδωσε τη δυνατότητα στον καθένα να αναπτύξει μία δικιά του εφαρμογή ή να πραγματοποιήσει μια έρευνα χρησιμοποιώντας το API. Τέλος, η

παρουσίαση των δεδομένων με τη χρήση του γραφικού περιβάλλοντος, έδωσε τη δυνατότητα στον καθένα να προβάλλει με εύκολο τρόπο και αποτελεί ένα παράδειγμα χρήσης του API.

## 7.2 Μελλοντικές Επεκτάσεις

Όπως αναφέρθηκε και παραπάνω, με την ολοκλήρωση των βασικών συνδέσμων του API, ξεκίνησε η ανάπτυξη νέων συνδέσμων οι οποίοι έχουν ως στόχο την καλύτερη δυνατή οργάνωση των δεδομένων που επιστρέφονται και τη μείωση του όγκου δεδομένων που μεταδίδει το API. Για τον ευκολότερο διαχωρισμό των νέων συνδέσμων από τους παλιούς, στους καινούργιους δόθηκε η ονομασία έκδοση 1. Οι σύνδεσμοι της έκδοσης 1, ως κύριο όγκο εργασίας έχουν αυτούς που αφορούν τις βάσεις εισαγωγής και τα στατιστικά προτιμήσεων. Περαιτέρω προσθήκες θα μπορούσαν να υλοποιηθούν και στις υπόλοιπες κατηγορίες, προσθέτοντας τη δυνατότητα αναζήτησης και δημιουργώντας όσο το δυνατόν καλύτερη οργάνωση στα δεδομένα που επιστρέφονται. Κύριος σκοπός των μελλοντικών επεκτάσεων είναι η ικανοποίηση κάθε πιθανού σεναρίου χρήσης του API, με όσο το δυνατόν ταχύτερη απόκριση και μικρότερο όγκο δεδομένων.

Επιπλέον, η περαιτέρω ανάπτυξη της διαδικτυακής εφαρμογής αποτελεί ένα στόχο για το μέλλον. Πέρα των διορθώσεων σε προβλήματα που πιθανώς να εντοπιστούν, η προσθήκη νέων λειτουργιών και χαρακτηριστικών αποτελεί ο κύριος στόχος. Οι νέες λειτουργίες αφορούν, την αποθήκευση επιλεγμένων τμημάτων από τον χρήστη για γρηγορότερη ανάκτηση των δεδομένων στο μέλλον. Επιπρόσθετα, με την αξιοποίηση της δυνατότητας αναζήτησης που προσφέρει το API, να ενσωματωθεί μια παρόμοια λειτουργία που θα υλοποιείται σε γραφικό περιβάλλον. Επίσης, να δοθεί η δυνατότητα εξαγωγής των δεδομένων που προβάλλονται είτε ως αρχείο εικόνας σε περίπτωση που πρόκειται για κάποιο διάγραμμα, είτε ως αρχείου τύπου υπολογιστικού φύλλο αν πρόκειται για απλά δεδομένα. Τέλος, με τη χρήση των νέων τεχνολογιών του διαδικτύου, η διαδικτυακή εφαρμογή θα μπορούσε να μετατραπεί σε μία εφαρμογή για κινητές συσκευές, παραμετροποιώντας την έτσι ώστε να λειτουργεί ως μία προοδευτική εφαρμογή ιστού (Progressive Web App).

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] “What is the WHATWG?.” <https://whatwg.org/faqwhat-is-the-whatwg>. [Online; accessed 27-February-2021].
- [2] C. Mercier, “W3C and the WHATWG signed an agreement to collaborate on a single version of HTML and DOM.” <https://www.w3.org/blog/news/archives/7753>, 2019. [Online; accessed 27-February-2021].
- [3] S. Faulkner, A. Eicholz, T. Leithead, A. Danilo, and S. Moon, “HTML5.2 W3C Recommendation,” tech. rep., W3C, 2017. <https://www.w3.org/TR/html52/>.
- [4] S. Pieters, “HTML5 Differences from HTML4,” tech. rep., W3C, 2014. <https://www.w3.org/TR/2014/NOTE-html5-diff-20141209/>.
- [5] “A brief history of CSS until 2016.” <https://www.w3.org/Style/CSS20/history.html>, Jan 2017. [Online; accessed 27. Feb. 2021].
- [6] “CSS Flexible Box Layout Module Level 1.” <https://www.w3.org/TR/css-flexbox-1>, Nov 2018. [Online; accessed 27. Feb. 2021].
- [7] “CSS Grid Layout Module Level 1.” <https://www.w3.org/TR/css-grid-1>, Dec 2020. [Online; accessed 27. Feb. 2021].
- [8] PHP Manual Contributors, “History of php.” <https://www.php.net/manual/en/history.php.php>, 2021. [Online; accessed 9-January-2021].
- [9] PHP-FIG, “Frequently asked questions - php-fig.” <https://www.php-fig.org/faqs>. [Online; accessed 6. Apr. 2021].
- [10] Composer, “Composer - A Dependency Manager for PHP.” <https://getcomposer.org/doc/00-intro.md>. [Online; accessed 6. Apr. 2021].
- [11] A. Beaulieu, *Learning SQL, Third Edition*, ch. 1. O’Reilly Media, Inc., 2020.
- [12] “ISO 9075:1987.” <https://www.iso.org/standard/16661.html>, Mar 2021. [Online; accessed 10. Mar. 2021].
- [13] “ISO/IEC 9075-1:2016.” <https://www.iso.org/standard/63555.html>, Mar 2021. [Online; accessed 10. Mar. 2021].
- [14] “MySQL 8.0 Reference Manual: What is MySQL?.” <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>, 2021. [Online; accessed 2. Mar. 2021].

- [15] “MariaDB versus MySQL - Compatibility.” <https://mariadb.com/kb/en/mariadb-vs-mysql-compatibility>, Mar 2021. [Online; accessed 10. Mar. 2021].
- [16] “ECMAScript® 2022 Language Specification.” <https://tc39.es/ecma262/sec-overview>, Mar 2021. [Online; accessed 10. Mar. 2021].
- [17] Node.js, “About | Node.js.” <https://nodejs.org/en/about>. [Online; accessed 10. Mar. 2021].
- [18] “Quick Start Guide | Electron.” <https://www.electronjs.org/docs/tutorial/quick-startquickstart>. [Online; accessed 10. Mar. 2021].
- [19] “React Native.” <https://reactnative.dev>. [Online; accessed 10. Mar. 2021].
- [20] “Documentation - TypeScript for JavaScript Programmers,” Mar 2021. [Online; accessed 10. Mar. 2021].
- [21] “About JavaScript - JavaScript | MDN.” [https://developer.mozilla.org/en-US/docs/Web/JavaScript/About\\_JavaScript#what\\_javascript\\_implementations\\_are\\_available](https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript#what_javascript_implementations_are_available). [Online; accessed 10. Mar. 2021].
- [22] “JSON.” <https://www.json.org/json-el.html>. [Online; accessed 11. Mar. 2021].
- [23] M. Haverbeke, *Eloquent JavaScript, Third Edition*, ch. 4. No Starch Press, 2018.
- [24] “Chart.js.” <https://github.com/chartjs/Chart.js/graphs/contributors?from=2013-03-17to=2021-03-11type=c>. [Online; accessed 15. Mar. 2021].
- [25] Chart.js, “Installation · Chart.js documentation.” <https://www.chartjs.org/docs/latest/getting-started/installation.html>. [Online; accessed 15. Mar. 2021].
- [26] Chart.js, “Integration · Chart.js documentation.” <https://www.chartjs.org/docs/latest/getting-started/integration.html>. [Online; accessed 15. Mar. 2021].
- [27] Chart.js, “Usage · Chart.js documentation.” <https://www.chartjs.org/docs/latest/getting-started/usage.html>. [Online; accessed 15. Mar. 2021].
- [28] Chart.js, “A curated list of awesome Chart.js resources and libraries.” <https://github.com/chartjs/awesome>. [Online; accessed 15. Mar. 2021].
- [29] Prism, “Prism - Used By.” <https://prismjs.com/#used-by>. [Online; accessed 15. Mar. 2021].
- [30] Prism, “Prism - Supported Languages.” <https://prismjs.com/#supported-languages>. [Online; accessed 15. Mar. 2021].
- [31] Prism, “Prism - Basic Usage.” <https://prismjs.com/#basic-usage>. [Online; accessed 15. Mar. 2021].
- [32] L. Richardson and M. Amundsen, *RESTful Web APIs*, ch. 3. O’Reilly Media, Inc., 2013.

- [33] R. Thomas, *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.
- [34] J. Patton, *User Story Mapping*, ch. 1. O'Reilly Media, Inc., 2014.
- [35] "System Usability Scale (SUS): Improving Products Since 1986." <https://digital.gov/2014/08/29/system-usability-scale-improving-products-since-1986>, Aug 2014. [Online; accessed 4. May 2021].
- [36] Kevin Tatroe and Peter MacIntyre, *Programming PHP*. O'Reilly Media, Inc.