

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



- Πτυχιακή Εργασία -

Σχεδιασμός και ανάπτυξη διαδικτυακής εφαρμογής για αξιολόγηση εκπαιδευτικών
ιδρυμάτων

Αιμίλιος Παυλίδης Π2019053

Επιβλέπων: - Αριστείδης Βραχάτης -

27 Δεκεμβρίου 2023

Επιβλέπων

Αριστείδης Βραχάτης, Επίκουρος Καθηγητής,
Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή

Γεώργιος Δημητρακόπουλος, Ακαδημαϊκός Υπότροφος,
Ιόνιο Πανεπιστήμιο

Ανδρέας Καναβός, Αναπληρωτής Καθηγητής,
Ιόνιο Πανεπιστήμιο

ΠΕΡΙΛΗΨΗ

Η συγκεκριμένη πτυχιακή εργασία περιλαμβάνει την ανάπτυξη και τον σχεδιασμό μίας διαδικτυακής εφαρμογής, με σκοπό την αξιολόγηση πανεπιστημιακών ιδρυμάτων από φοιτητές. Η έλλειψη μίας τέτοιας ιστοσελίδας στον Ελληνικό ιστό, προκάλεσε την επιθυμία για τη δημιουργία αυτής της ιστοσελίδας. Οι χρήστες κάνουν εγγραφή στη πλατφόρμα επιλέγοντας σημαντικά χαρακτηριστικά, όπως το πανεπιστήμιο και το τμήμα που ανήκουν. Έπειτα, συνδέονται στην ιστοσελίδα και υποβάλλουν μία κριτική για το τμήμα τους. Η συνολική βαθμολογία αντιστοιχεί τόσο στο τμήμα, όσο και στο ίδιο το πανεπιστήμιο. Οι χρήστες αυτοί έχουν τη δυνατότητα να διαγράψουν ή να τροποποιήσουν αυτή τη κριτική σε περίπτωση που το θελήσουν. Οι αξιολογήσεις των φοιτητών συντελούν στη δημιουργία των μέσων όρων για τα εκπαιδευτικά αυτά ιδρύματα. Επιπρόσθετη λειτουργία αποτελεί, η επιλογή παρουσίασης των πανεπιστημίων με βάση τη τοποθεσία τους (Βόρεια, Κεντρικά και Νότια).

Λέξεις κλειδιά: διαδικτυακή εφαρμογή, αξιολόγηση, ιδρύματα, φοιτητές, ιστοσελίδα, πανεπιστήμιο, τμήμα, κριτική, βαθμολογία

ABSTRACT

This particular thesis includes the development and design of a web-based application for the evaluation of university institutions by students. The lack of such a website in the Greek web, caused the desire to create this website. The users register on the platform by selecting important features, such as the university and the department, that they belong to. Then, they log in to the website and submit a review of their department. The overall score corresponds to both the department and the university. These users have the option to delete or modify their review, if they wish to do so. Student reviews contribute to the creation of averages for these educational institutions. An additional function, is the option of presenting universities by location (North, Central and South).

Keywords: web-based application, evaluation, institutions, students, website, university, department, review

ΕΥΧΑΡΙΣΤΙΕΣ

Η υλοποίηση της πτυχιακής εργασίας ήταν μία μακροσκελής διαδικασία που χρειάστηκε πίστη, αφοσίωση και υποστήριξη. Πρώτον, θα επιθυμούσα να ευχαριστήσω τους καθηγητές κ. Αριστείδη Βραχάτη και τον κ. Ανδρέα Καναβό που αποτέλεσαν μέλη της τριμελούς επιτροπής και οφείλω ένα ιδιαίτερο ευχαριστώ στον κ. Γεώργιο Δημητρακόπουλο που δέχτηκε να υλοποιηθεί η πτυχιακή εργασία κάτω από τη δική του εποπτεία και εκτιμήθηκε ιδιαιτέρως η συνεπής καθοδήγηση που μου παρείχε. Έπειτα, θα ήθελα να ευχαριστήσω τους συμφοιτητές μου Θωμά Σπυρίδων Πασχάλη και Αποστόλη Αντωνακάκη που σε μία κατάσταση κρίσης, έσπευσαν να με βοηθήσουν να την αντιμετωπίσω. Τέλος, ένα ακόμη άτομο που θα επιθυμούσα να ευχαριστήσω είναι ο Γεώργιος Παπουτσάκης που με φιλοξένησε για ένα μήνα στο σπίτι του, ώστε να έχω την ησυχία να ολοκληρώσω τη πτυχιακή μου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	4
ΕΥΧΑΡΙΣΤΙΕΣ	5
ΔΙΑΓΡΑΜΜΑΤΑ	9
ΕΙΚΟΝΕΣ	10
1 ΘΕΩΡΗΤΙΚΟ ΠΛΑΙΣΙΟ	11
1.1 Διαδικτυακές εφαρμογές	11
1.2 Cloud εφαρμογές (νέφους)	11
1.3 Μοντέρνα προσέγγιση.....	12
1.3.1 3-tier Architecture (αρχιτεκτονική 3 επιπέδων).....	12
1.3.2 REST API	13
1.3.3 JavaScript Frameworks (σκελετοί JavaScript)	13
1.4 Άλλες πιθανές προσεγγίσεις	14
1.4.1 Ruby on Rails.....	14
1.4.2 MySQL & PHP	14
1.4.3 MEAN stack	15
1.4.4 Flask (σκελετός της Python).....	15
1.4.5 Spring Boot (σκελετός της Java)	15
1.4.6 Συστήματα CMS (Content Management Systems).....	15
2 ΕΡΕΥΝΗΤΙΚΗ ΑΝΑΣΚΟΠΗΣΗ	17
2.1 Εισαγωγή	17
2.2 Πρόβλημα προς Επίλυση.....	17
2.3 Έρευνα Ιστοσελίδων	18
2.3.1 Παρουσίαση πανεπιστημίων.....	18
2.3.2 Σύστημα αξιολόγησης.....	18
2.3.3 Πεδία αξιολόγησης.....	19
3 ΕΡΓΑΛΕΙΑ ΥΛΟΠΟΙΗΣΗΣ.....	20
3.1 Visual Studio Code.....	20
3.2 Node.js.....	20
3.2.1 Express	20
3.2.2 Passport.js & Passport-local	21
3.2.3 Bcrypt.js.....	21

3.3	MongoDB	21
3.3.1	Mongoose	21
3.3.2	MongoDB Atlas	21
3.3.3	MongoDB Compass	21
3.4	Postman	22
3.5	React.....	22
3.5.1	React Router	22
3.6	CSS.....	22
3.6.1	Flexbox.....	22
4	ΥΛΟΠΟΙΗΣΗ ΠΙΣΩ ΜΕΡΟΥΣ (BACKEND).....	24
4.1	Διάγραμμα σχέσης οντοτήτων (ERD).....	24
4.1.1	Περιγραφή των χαρακτηριστικών.....	26
4.2	Συλλογές & έγγραφα MongoDB	30
4.3	RESTful API	32
4.3.1	Αρχείο index.js.....	32
4.3.2	Δημιουργία των models (μοντέλων)	32
4.3.3	Αρχείο passportConfig.js.....	33
4.3.4	Αρχείο routes.js.....	33
4.4	Βάση συστήματος εισόδου/εγγραφής	35
4.4.1	Bcrypt.js.....	35
4.4.2	Αρχείο passportConfig.js.....	35
4.4.3	Διαδρομές στο routes.js.....	35
4.5	Σελιδοποίηση (pagination)	36
4.6	Δημιουργία μέσου όρου βαθμολογιών.....	37
4.6.1	Μέσος όρος τμήματος.....	37
4.6.2	Μέσος όρος πανεπιστημίου.....	37
4.7	Εισαγωγή πανεπιστημίων & τμημάτων.....	39
5	ΥΛΟΠΟΙΗΣΗ ΜΠΡΟΣΤΙΝΟΥ ΜΕΡΟΥΣ (FRONTEND).....	40
5.1	Υλοποίηση frontend	40
5.1.1	Χρήση react-router-dom.....	40
5.1.2	Πρόβλημα 1: πακέτο react-router-dom	41
5.2	Εμφάνιση πανεπιστημίων.....	42
5.3	Φίλτρο τοποθεσίας.....	43
5.4	Εμφάνιση τμημάτων.....	44
5.5	Μπροστινό μέρος συστήματος εισόδου/εγγραφής	45

5.5.1	Αρχείο loginSignup.js	45
5.5.2	Αρχείο loginPage.js.....	45
5.5.3	Αρχείο signupPage.js	46
5.5.4	Επιπρόσθετες λειτουργίες.....	47
5.6	Star System (σύστημα αστεριών)	48
5.7	Review System (σύστημα αξιολόγησης).....	49
5.8	Εμφάνιση κριτικών τμημάτων	51
5.8.1	Για αρχείο routes.js	51
5.8.2	Αρχείο DepartmentReviews.js.....	51
5.9	Τροποποίηση και διαγραφή κριτικής	52
5.9.1	Δημιουργία κουμπιών τροποποίησης και διαγραφής.....	52
5.9.2	Διαγραφή κριτικής	53
5.9.3	Τροποποίηση κριτικής.....	53
5.10	Ανταποκρισιμότητα (responsiveness)	54
5.10.1	Προσαρμογή για tablets.....	54
5.10.2	Προσαρμογή για smartphones	55
5.11	Φιλοξενία ιστοσελίδας (hosting).....	57
5.11.1	Ανέβασμα εργασίας στο GitHub.....	57
5.11.2	Φιλοξενία ιστοσελίδας στο Render.....	57
5.11.3	Πρόβλημα 2: λανθάνουσα χρήση GitHub.....	58
6	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	59
7	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	60

ΔΙΑΓΡΑΜΜΑΤΑ

Διάγραμμα 1: ERD.....	25
Διάγραμμα 2: Οντότητα student	26
Διάγραμμα 3: Οντότητα university	27
Διάγραμμα 4: Οντότητα department	28
Διάγραμμα 5: Οντότητα review	28

ΕΙΚΟΝΕΣ

Εικόνα 1: Συλλογές MongoDB.....	30
Εικόνα 2: Έγγραφα MongoDB	31
Εικόνα 3: Κώδικας μοντέλου department.....	33
Εικόνα 4: Εμφάνιση σελιδοποίησης.....	36
Εικόνα 5: Εμφάνιση μέσου όρου πανεπιστημίου	38
Εικόνα 6: Κώδικας για Link	40
Εικόνα 7: Κώδικας για useLocation	41
Εικόνα 8: Εμφάνιση πανεπιστημίων	42
Εικόνα 9: Εμφάνιση φίλτρου τοποθεσίας.....	43
Εικόνα 10: Εμφάνιση τμημάτων	44
Εικόνα 11: Εμφάνιση κουμπιών login/signup	45
Εικόνα 12: Εμφάνιση φόρμας σύνδεσης	46
Εικόνα 13: Εμφάνιση φόρμας εγγραφής (1).....	47
Εικόνα 14: Εμφάνιση φόρμας εγγραφής (2).....	47
Εικόνα 15: Εμφάνιση συνδεδεμένος και κουμπί αποσύνδεσης.....	47
Εικόνα 16: Εμφάνιση συστήματος αστεριών	49
Εικόνα 17: Εμφάνιση φόρμας αξιολόγησης.....	50
Εικόνα 18: Εμφάνιση κριτικών τμήματος	52
Εικόνα 19: Εμφάνιση κουμπιών edit & delete.....	53
Εικόνα 20: Εμφάνιση φόρμας τροποποίησης κριτικής	54
Εικόνα 21: Εμφάνιση ιστοσελίδας σε οθόνη tablet	55
Εικόνα 22: Εμφάνιση κριτικών τμήματος σε οθόνη κινητού	56
Εικόνα 23: Εμφάνιση πίνακα ελέγχου της πλατφόρμας Render.....	58

1 ΘΕΩΡΗΤΙΚΟ ΠΛΑΙΣΙΟ

Η εργασία που υλοποιήθηκε αποτελεί μία διαδικτυακή εφαρμογή και συγκεκριμένα χρησιμοποιήθηκε το μοντέλο MERN stack (MongoDB, Express, React, Node). Σε αυτό το κεφάλαιο θα αναλυθούν διάφορα μονοπάτια γύρω από την ανάπτυξη εφαρμογών. Συγκεκριμένα θα γίνουν αναφορές στις διαδικτυακές εφαρμογές (σαν αυτή που φτιάξαμε), αλλά και στις cloud applications (εφαρμογές νέφους) που είναι πιο σύγχρονες. Έπειτα θα ακολουθήσει το κεφάλαιο “Μοντέρνα προσέγγιση”, όπου θα επεξηγηθούν έννοιες που συνάπτουν με την εργασία που δομήθηκε. Τέλος, θα γίνει μία σύντομη αναφορά σε άλλες πιθανές προσεγγίσεις που θα μπορούσαν να ακολουθηθούν. Το κεφάλαιο αυτό αποτελεί τα θεμέλια της εργασίας, αφού εισάγει τον αναγνώστη στην φιλοσοφία της ανάπτυξης εφαρμογών.

1.1 Διαδικτυακές εφαρμογές

Οι διαδικτυακές εφαρμογές έχουν ως έδρα τον ιστό και ο χρήστης έχει πρόσβαση σε αυτές μέσω προγραμμάτων περιήγησης ιστού (web browsers). Οι διαδικτυακές εφαρμογές διαφέρουν αρκετά από τις ιστοσελίδες, αφού οι ιστοσελίδες συχνά είναι στατικές, ενώ μία διαδικτυακή εφαρμογή είναι δυναμική. Οι χρήστες έχουν την δυνατότητα καταχώρησης δεδομένων προς την βάση, αλλά και αίτηση για λήψη δεδομένων από τη βάση. Για την πραγματοποίηση της δοσοληψίας συμβάλλει το πρόγραμμα περιήγησης και ο διακομιστής ιστού (Web Server) [1]. Η δομή των διαδικτυακών εφαρμογών αποτελείται συχνά από το 3-tier architecture (αρχιτεκτονική 3 επιπέδων). Η αρχιτεκτονική τριών επιπέδων χρησιμοποιεί μία βάση δεδομένων, έναν backend server (για επικοινωνία με τη βάση) και το μπροστινό μέρος της εφαρμογής που είναι ορατό προς τον χρήστη [2]. Αυτήν την αρχιτεκτονική ακολουθήσαμε και εμείς.

1.2 Cloud εφαρμογές (νέφους)

Οι εφαρμογές νέφους λειτουργούν παρόμοια με τις διαδικτυακές εφαρμογές, απλώς έχουν μία μεγάλη διαφορά, που είναι η χρήση απομακρυσμένων διακομιστών (remote servers) [3]. Σε μία εφαρμογή νέφους, η πρόσβαση των χρηστών μπορεί να επιτευχθεί συχνά μέσω ενός προγράμματος περιήγησης ιστού (όπως και στις διαδικτυακές εφαρμογές) [4], αλλά κάποιες βασικές λειτουργίες, όπως είναι η αποθήκευση αρχείων, η συλλογή δεδομένων ή η διαχείριση αποθεμάτων γίνονται διαδικτυακά και όχι τοπικά. Παρατηρείται ραγδαία αύξηση χρήσης εφαρμογών νέφους για ποικίλους λόγους. Πολύ βασικός λόγος είναι η ενισχυμένη ασφάλεια και η ευκολότερη δημιουργία αντιγράφων ασφαλείας (backups). Ένας ακόμη λόγος είναι τα μειωμένα κόστη, λόγω του μεγάλου ανταγωνισμού μεταξύ των παρόχων και το μέγεθος των εγκαταστάσεων που διαθέτουν.

1.3 Μοντέρνα προσέγγιση

Η μοντέρνα προσέγγιση αποτελεί ένα κεφάλαιο όπου θα εξηγηθούν έννοιες γύρω από τη κατασκευή διαδικτυακών εφαρμογών και συγκεκριμένα κοντά στο μοντέλο που δομήθηκε στα πλαίσια της εργασίας. Θα γίνει μία αναλυτικότερη επεξήγηση του 3-tier architecture, των REST API, των JavaScript Frameworks και του Model-View-Controller architecture (αρχιτεκτονική μοντέλου-προβολής-ελεγκτή). Οι έννοιες αυτές είναι πολύ σημαντικές και χωρίς την κατανόησή τους δε μπορεί να υπάρξει ομαλή μετάβαση στο πιο πρακτικό κομμάτι της εργασίας.

1.3.1 3-tier Architecture (αρχιτεκτονική 3 επιπέδων)

Παραπάνω έχουμε αναφέρει συνοπτικά τα τρία επίπεδα της αρχιτεκτονικής [5]. Τα επίπεδα αυτά αποτελούν τους πυλώνες του έργου και είναι σημαντική η αλληλεπίδραση συνεργασία μεταξύ τους για μία λειτουργική εφαρμογή.

Το πρώτο επίπεδο είναι η βάση δεδομένων (database), η οποία είναι ο χώρος όπου αποθηκεύονται όλα τα δεδομένα και οι πληροφορίες σε μία οργανωμένη μορφή [6]. Για την διαχείριση και τακτοποίηση των δεδομένων, συχνά χρειάζεται ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS) [7]. Για λειτουργίες που απαιτούν επικοινωνία με τη βάση όπως τροποποίηση, ανάρτηση ή αίτηση για παρουσίαση δεδομένων, υπαίτιο είναι το σύστημα διαχείρισης βάσεων δεδομένων. Όσον αφορά την μοντελοποίηση των δεδομένων, υπάρχουν δύο βασικές κατηγορίες. Η πρώτη και πιο συνηθισμένη είναι οι γραμμές και οι πίνακες που χρησιμοποιούνται από SQL βάσεις και έπειτα υπάρχουν τα έγγραφα και οι συλλογές που χρησιμοποιεί για παράδειγμα το MongoDB που είναι μοντέλο NoSQL.

Το δεύτερο επίπεδο είναι ο backend server, που η κύρια αρμοδιότητά του είναι η επικοινωνία με τη βάση δεδομένων. Ο server (διακομιστής) ζητάει και δίνει στοιχεία στη βάση δεδομένων. Μια καλή πρακτική για την περαιτέρω κατανόηση της έννοιας αυτής, είναι να οραματιστούμε το μεσοδιάστημα μεταξύ της άντλησης των στοιχείων (από την βάση) και της εμφάνισης τους στην οθόνη. Στο κομμάτι αυτό ανήκει και το REST API το οποίο διαχειρίζεται τις λειτουργίες που επιθυμούμε μέσω των μεθόδων POST (ανέβασμα), GET (τράβηξε), PATCH (ενημέρωσε) και DELETE (διέγραψε). Για τη χρήση των μεθόδων είναι απαραίτητη η συγγραφή διαδρομών (routes) που θα αναλυθούν παρακάτω στο κεφάλαιο “REST API”.

Τέλος, το τρίτο επίπεδο είναι το frontend (μπροστινό μέρος της εφαρμογής). Το μπροστινό μέρος αποτελείται από αυτά που ο χρήστης βλέπει και αλληλεπιδράει [8]. Το βασικό τρίπτυχο εργαλείων για την πραγματοποίηση του μπροστινού μέρους είναι οι γλώσσες HTML, CSS και JavaScript. Η HTML είναι υπαίτια για τη δομή και την προσθήκη των στοιχείων. Η CSS ασχολείται με την εμφάνιση των στοιχείων (στοίχιση, χρώματα, μέγεθος, διάταξη). Η JavaScript χρησιμοποιείται περισσότερο στο λειτουργικό κομμάτι, όπως για παράδειγμα όταν γίνεται κλικ σε ένα κουμπί ή σε μία σελίδα της ιστοσελίδας. Το 2023 υπάρχουν πολλές εκσυγχρονισμένες βιβλιοθήκες που βοηθούν στην δημιουργία γραφικών διεπαφών φιλικές προς τον χρήστη (user-friendly) και σχεδιαστικά ελκυστικές στο μάτι.

1.3.2 REST API

Για τα REST APIs έχει γίνει ήδη μια συνοπτική αναφορά, αλλά είναι σημαντικό να επεξηγηθούν λίγο παραπάνω. Το REST API ανήκει στο backend (πίσω μέρος) και το δημιουργεί ο προγραμματιστής για να υλοποιούνται υπηρεσίες ιστού εύκολα και απλά [9]. Το API αυτό λειτουργεί αυτοματοποιημένα έπειτα από τον προγραμματισμό του και διαχειρίζεται τη καταχώρηση και το τράβηγμα δεδομένων από και προς τον χρήστη. Για την επιτυχής αυτή διαδικασία χρησιμοποιούνται οι μέθοδοι POST, GET, PATCH και DELETE που έχουν αναφερθεί και παραπάνω. Η μέθοδος POST, χρησιμοποιείται για την υποβολή δεδομένων. Η μέθοδος GET για τράβηγμα (fetch) δεδομένων από τη βάση δεδομένων. Η PATCH για την τροποποίηση/ενημέρωση επιλεγμένων στοιχείων. Η DELETE για τη διαγραφή δεδομένων από τη βάση. Οι μέθοδοι αυτοί είναι απαραίτητοι για την ύπαρξη ενός ολοκληρωμένου REST API. Αξίζει να αναφερθεί πως οι αιτήσεις που γίνονται μέσω των μεθόδων έχουν τη μορφή HTTP.

1.3.3 JavaScript Frameworks (σκελετοί JavaScript)

Οι σκελετοί (frameworks), όπως εκφράζει και το όνομά τους βασίζονται σε μία γλώσσα προγραμματισμού και διευκολύνουν τη διαδικασία συγγραφής κώδικα. Ο κώδικας που συγγράφεται από τον προγραμματιστή είναι πιο απλός και αυτό οδηγεί στην ευκολότερη κατανόησή του από άλλους προγραμματιστές, αλλά και στην αποφυγή λαθών [10]. Ακόμη, ένα άλλο μεγάλο θετικό είναι πως αποτελούν ένα πρότυπο, όπου οτιδήποτε κατασκευαστεί βασίζεται πάνω σε αυτό. Οι δυνατότητες είναι άπειρες, αλλά η δομή και η διαδικασία ακολουθάει ένα συγκεκριμένο μοτίβο. Στο κομμάτι αυτό θα αναφερθούμε σε σκελετούς (frameworks) της JavaScript [11], αφού είναι η κυρίαρχη γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την εργασία μας.

1.3.3.1 Angular

Η Angular είναι μία από τις δημοφιλέστερες επιλογές, όσον αφορά την επιλογή σκελετού βασισμένη στη JavaScript. Κύριο κριτήριο επιλογής αποτελεί η δημιουργία μίας εφαρμογής μόνης σελίδας [12]. Βασίζεται στην TypeScript και υπαίτια εταιρεία ανάπτυξης είναι η Google. Ένα άλλο μεγάλο πλεονέκτημα είναι η χρήση των PWA και SPA. Το PWA (progressive web app) χαμηλώνει τις απαιτήσεις του δίκτυο για την ομαλή λειτουργία της εφαρμογής και βοηθάει τις ιστοσελίδες να λειτουργούν σαν εφαρμογές, προσδίδοντας ευκολία πλοήγησης στους χρήστες. Σε παρόμοιο πλαίσιο, το SPA (single-page web application), προσφέρει γρηγορότερη φόρτωση της σελίδας και διαχειρίζεται πολύ αποδοτικά τους πόρους της συσκευής. Η Angular αξιοποιεί την αρχιτεκτονική μοντέλου-προβολής-ελεγκτή (model-view-controller), η οποία θα αναλυθεί παρακάτω.

1.3.3.2 Vue.js

Για το Vue.js δε θα γίνει μεγάλη επεξήγηση, αφού έχει ήδη αναλυθεί η έννοια των JavaScript Frameworks (σκελετούς JavaScript). Το Vue έχει την δυνατότητα ενίσχυσης της HTML, το οποίο σημαίνει πως μπορεί να επιλεγεί από τον προγραμματιστή στη θέση της jQuery [13]. Το Vue μπορεί να χρησιμοποιηθεί για τη δημιουργία SPA (single-page web application) και να αξιοποιήσει ένα μεγάλο εύρος βιβλιοθηκών. Η εισαγωγή του στην εφαρμογή μας μπορεί να γίνει σταδιακά και να απασχολήσει συγκεκριμένα μέρη του κώδικα. Τέλος, ένα πολύ δυνατό χαρακτηριστικό του σκελετού (framework) αυτού, είναι η ευκολία στην χρήση του και το γεγονός ότι είναι πολύ ελαφρύ συγκριτικά με τη React για παράδειγμα [14].

1.4 Άλλες πιθανές προσεγγίσεις

Στον χώρο του προγραμματισμού και της πληροφορικής υπάρχουν ποικίλες επιλογές εργαλείων, εξειδικευμένες να ταιριάζουν στις γνώσεις και τις αρέσκειες του προγραμματιστή. Εμείς, όπως έχει αναφερθεί και παραπάνω, επιλέξαμε το μοντέλο MERN stack. Υπάρχουν επιλογές πασίγνωστες όπως JavaScript για το μπροστινό μέρος μίας ιστοσελίδας και SQL για το πίσω μέρος (τη βάση της ιστοσελίδας). Παρ' όλα αυτά όμως δεν υπάρχουν μόνο επιλογές με πρωταρχικό κριτήριο τον ενεργό προγραμματισμό από τον δημιουργό. Υπάρχουν πολλά περιβάλλοντα που βασίζονται σε άλλες μορφές δημιουργίας ιστοσελίδων, κυρίως επιλογής χαρακτηριστικών, διάταξης αντικειμένων με drag and drop (σέρνω και αφήνω) και άλλες τέτοιου είδους λειτουργίες. Εργαλεία που καλύπτουν αυτά τα κριτήρια είναι το αξιόπιστο, γνωστό σε όλους WordPress και άλλες λιγότερο γνωστές πλατφόρμες όπως το Joomla.

1.4.1 Ruby on Rails

Ο συνδυασμός της Ruby με την Rails είναι πολύ ισχυρός όσον αφορά την ανάπτυξη ιστοσελίδων. Η Ruby είναι μία γλώσσα προγραμματισμού αφιερωμένη στο backend (πίσω μέρος) μίας ιστοσελίδας ή εφαρμογής. Η Rails είναι ένα framework (σκελετός) της Ruby που επιτρέπει Full stack (και για το μπροστινό μέρος και για το πίσω μέρος της ιστοσελίδας ή εφαρμογής) ανάπτυξη [15]. Αξίζει να αναφερθεί ότι όταν δημιουργείται ένα project με τη χρήση της Ruby on Rails, δημιουργείται και η αυτόματη βάση δεδομένων SQLite3 [16].

1.4.2 MySQL & PHP

Η MySQL ανήκει καθαρά στο πίσω μέρος της ιστοσελίδας και ασχολείται με τη διαχείριση της βάσης δεδομένων. Οι βάσεις δεδομένων που δημιουργούνται στον χώρο αυτό είναι σχεσιακές, που σημαίνει πως η οργάνωση των δεδομένων συμβαίνει με τη χρήση πινάκων και υπάρχει συσχέτιση μεταξύ των δεδομένων [17]. Η PHP αποτελεί έναν πυλώνα της ανάπτυξης ιστοσελίδων, λόγω της μεγάλης επιλογής της από προγραμματιστές. Η PHP έχει πολλές χρησιμότητες, με μία από αυτές να είναι η βοήθεια στη δημιουργία γραφικών διεπαφών [18], δηλαδή στο κομμάτι που είναι ορατό προς τον χρήστη και μπορεί να αλληλεπιδράσει.

1.4.3 MEAN stack

Η MEAN stack είναι μία επιλογή εργαλείων, πολύ κοντινή σε αυτή που επιλέχθηκε, για την ανάπτυξη διαδικτυακών ιστοσελίδων. Τα αρχικά της αποτελούνται από τις τεχνολογίες MongoDB, Express, Angular και Node. Για τις τρεις από αυτές έχουμε ήδη κάνει μία εισαγωγή, οπότε θα ήταν επιθυμητό να αναφερθούμε αποκλειστικά στην Angular. Η Angular, όπως και η React, είναι framework (σκελετός) της JavaScript, αλλά με τη θεμελιώδη διαφορά ότι είναι γραμμένη σε TypeScript [19]. Σε γενικές γραμμές η διαφορά της Angular από την React είναι οι περισσότερες δυνατότητες που έχει. Σε σύγκριση με τη React που είναι κυρίως μια βιβλιοθήκη, η Angular έχει μεγαλύτερη υπόσταση και αυτοδυναμία.

1.4.4 Flask (σκελετός της Python)

Το Flask είναι ένας σκελετός (framework) γραμμένος σε γλώσσα Python και χρησιμοποιείται για την ανάπτυξη εφαρμογών διαδικτύου. Το Flask είναι micro-framework (μικροσκελετός) που σημαίνει πως ένας προγραμματιστής εύκολα και γρήγορα, χωρίς πολλές ρυθμίσεις και προσαρμογές, μπορεί να κατασκευάσει μία μικρή εφαρμογή [20]. Ο σκελετός αυτός μπορεί να οδηγήσει σε αποφυγή σπατάλης χρόνου και χρημάτων, λόγω των χαμηλών απαιτήσεών του. Το Flask χρησιμοποιεί τεχνολογία WSGI (Web Server Gateway Interface) που βοηθάει στην επικοινωνία του server (διακομιστή) με την εφαρμογή. Επίσης, χρησιμοποιεί τη μηχανή Jinja 2, η οποία δέχεται το περιεχόμενο που έχει συγγράψει ο προγραμματιστής και το μεταδίδει στον διακομιστή.

1.4.5 Spring Boot (σκελετός της Java)

Ο σκελετός (framework) Spring Boot, που προέρχεται από τον σκελετό Spring, είναι γραμμένος σε κώδικα Java. Η Java είναι μία σύνθετη και πολύπλοκη γλώσσα προγραμματισμού, ενώ ο σκελετός Spring Boot είναι πολύ πιο απλός και εύκολος στη χρήση του. Τα έργα που μπορεί να υλοποιήσει ένας προγραμματιστής με τον σκελετό αυτό είναι μικρουπηρεσίες και εφαρμογές διαδικτύου [21]. Τέλος, αξίζει να αναφερθεί η χρήση της dependency injection (έγχυσης εξαρτήσεων). Η Java, ως μία αντικειμενοστραφής γλώσσα, χρησιμοποιεί εξαρτήσεις και στην ουσία το μοντέλο έγχυσης εξαρτήσεων αυτοματοποιεί την επικοινωνία, αφού ο εγχυτήρας επιλέγει την κατάλληλη υπηρεσία για τον πελάτη.

1.4.6 Συστήματα CMS (Content Management Systems)

Τα CMS (συστήματα διαχείρισης περιεχομένου), αποτελούνται από πλατφόρμες που επιτρέπουν την δημιουργία ιστοσελίδων, χωρίς να είναι απαραίτητη η συγγραφή κώδικα [22]. Οι πλατφόρμες αυτές επιλέγονται πολύ από τον κόσμο τα τελευταία χρόνια, λόγω της εύκολης χρήσης τους και των μεγάλων δυνατοτήτων τους.

Παρ' όλα αυτά για την υλοποίηση μίας δυναμικής εφαρμογής, όπως είναι η δικιά μας, δεν αποτελούν καλή επιλογή. Στο κεφάλαιο αυτό θα αναφερθούμε στις πλατφόρμες WordPress και Joomla.

Το WordPress είναι μία πλατφόρμα δημιουργίας ιστοσελίδων, όπου η χρήση κώδικα είναι τελείως προαιρετική. Οι δυνατότητες και η προσαρμοστικότητα που προσφέρει η πλατφόρμα είναι πολύ μεγάλες και ιδιαίτερα με τη χρήση κώδικα μπορούν να δημιουργηθούν κορυφαίες ιστοσελίδες. Οι κύριες προγραμματιστικές γλώσσες και γλώσσες σήμανσης που χρησιμοποιούνται είναι η PHP και η CSS. Το διαδίκτυο πολιορκείται από ιστοσελίδες WordPress αφού αποτελεί τη κυρίαρχη πλατφόρμα. Αξίζει να αναφερθεί και η χρήση των plugins (πρόσθετων), όπου επιτρέπουν επιπρόσθετες λειτουργίες και καλύπτουν σχεδόν κάθε ανάγκη. Για παράδειγμα το WooCommerce αποτελεί ένα plugin (πρόσθετο) που επιτρέπει τη δημιουργία και διαχείριση διαδικτυακών μαγαζιών (e-shop).

Το Joomla είναι μία αρκετά διαδεδομένη λύση που μοιάζει σε έναν βαθμό με το WordPress. Όπως και το WordPress ανήκει στην κατηγορία CMS (σύστημα διαχείρισης περιεχομένου) [23] και ως κύρια αρμοδιότητα έχει την δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών [24]. Η συγγραφή κώδικα δεν αποτελεί απαραίτητη προϋπόθεση, αλλά σίγουρα επεκτείνει τις δυνατότητες της πλατφόρμας και προτείνεται. Όσον αφορά την βάση δεδομένων συχνή επιλογή αποτελεί η σύνδεση του Joomla με MySQL ή PostgreSQL.

2 ΕΡΕΥΝΗΤΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

Στην ενότητα αυτή αναγράφεται μία αναλυτική εισαγωγή για την ομαλή ένταξη του αναγνώστη στην πτυχιακή εργασία που θα ακολουθήσει. Έπειτα, περιγράφεται το πρόβλημα προς επίλυση της εργασίας αυτής και τέλος η έρευνα που υλοποιήθηκε με σκοπό την παροχή ιδεών, τόσο σχεδιαστικών όσο και λειτουργικών.

2.1 Εισαγωγή

Η πτυχιακή εργασία αποτελεί το αποτύπωμα που αφήνει ο φοιτητής και σηματοδοτεί την ολοκλήρωση των σπουδών του. Για αυτόν τον λόγο, η επιλογή του θέματος είναι σημαντικό να συμβαδίζει τόσο με τα ενδιαφέροντα του φοιτητή όσο και με μελλοντικές ευκαιρίες ή μεταπτυχιακά που σκοπεύει να ακολουθήσει. Η συγκεκριμένη πτυχιακή εργασία αποτελεί μία διαδικτυακή εφαρμογή αξιολόγησης εκπαιδευτικών ιδρυμάτων. Παρατηρήθηκε έπειτα από μία μικρή έρευνα που υλοποιήθηκε, πως δεν υπάρχει μία ολοκληρωμένη ιστοσελίδα που να περιλαμβάνει κριτικές για τα πανεπιστήμια της Ελλάδας από φοιτητές.

Για την υλοποίηση της ιδέας ήταν απαραίτητη η δημιουργία μίας βάσης δεδομένων και μίας γραφικής διεπαφής, όπου οι χρήστες θα μπορούσαν να αλληλεπιδράσουν εύκολα και απλά με αυτήν. Για τη δημιουργία της βάσης επιλέχθηκε το MongoDB, λόγω της μεγάλης πληροφορίας που υπάρχει στο διαδίκτυο, των δυνατοτήτων που διαθέτει και την ευκολία στη χρήση του, αφού διαθέτει περιβάλλον NoSQL.

Το μπροστινό μέρος (frontend) της εφαρμογής δομήθηκε με τη βιβλιοθήκη και framework (σκελετό) της JavaScript, την React. Η React δεν αποτέλεσε μία τυχαία επιλογή, αφού το μοντέλο κατασκευής της ιστοσελίδας ήταν το MERN stack (MongoDB, Express, React, Node). Στο παρελθόν, υπήρξε μία εξοικείωση με τη γλώσσα προγραμματισμού JavaScript, οπότε η επιλογή αυτού του μοντέλου ήταν μία λογική απόφαση. Επίσης, το μοντέλο αυτό χρησιμοποιείται αρκετά τα τελευταία χρόνια και θα αποτελούσε μία καλή εμπειρία για μελλοντικές ευκαιρίες.

Όσον αφορά την ίδια την ιστοσελίδα, το τελικό αποτέλεσμα ήταν αρκετά ολοκληρωμένο, σε βαθμό που με λίγες περαιτέρω προσθήκες και τροποποιήσεις θα μπορούσε κάλλιστα να έχει διαδικτυακή υπόσταση. Η ιστοσελίδα διαθέτει δεκατέσσερα (14) πανεπιστήμια και εξήντα ένα (61) τμήματα. Επίσης, υπάρχει σύστημα εγγραφής και εισόδου των χρηστών. Το σύστημα αξιολόγησης λειτουργεί άψογα και η εμφάνιση των κριτικών έχει περιποιημένη δομή. Οι χρήστες μπορούν να τροποποιήσουν ή και να διαγράψουν την κριτική τους εφόσον το επιθυμούν. Παρακάτω στο κεφάλαιο έξι (6) αναγράφονται πιθανές επεκτάσεις και προσθήκες που θα έδιναν στην ιστοσελίδα μία πιο επαγγελματική πνοή.

2.2 Πρόβλημα προς Επίλυση

Η ιδέα για τη δημιουργία της ιστοσελίδας, που δημιουργήθηκε στα πλαίσια της πτυχιακής μου, είχε ως πηγαία έμπνευση ένα βίντεο που είχε παρακολουθηθεί κατά τη διάρκεια της καραντίνας. Στο βίντεο αυτό ένας νεαρός προγραμματιστής έφτιαξε έναν ιστότοπο όπου περιλάμβανε όλα τα πανεπιστήμια της περιοχής του και έδινε τη δυνατότητα στους φοιτητές να βαθμολογούν το ίδρυμα στο οποίο ανήκουν. Έπειτα από την παρακολούθηση του βίντεο αυτού υλοποιήθηκε μία μικρή έρευνα για την παρατήρηση αν υπάρχει κάτι παρόμοιο για τα Ελληνικά πανεπιστήμια. Δυστυχώς δεν βρέθηκε κάποια ιστοσελίδα που να κάλυπτε τα κριτήρια που είχαν οραματιστεί. Με αυτά κατά νου αποφασίστηκε πως θα αποτελούσε μία καλή ιδέα να χτιζόταν μία τέτοια ιστοσελίδα από εμένα.

Ο λόγος της σημαντικότητας της ιστοσελίδας, είναι η παροχή μίας πλατφόρμας όπου οι φοιτητές μπορούν να εκφράσουν την άποψη τους ελεύθερα και ανώνυμα για το πανεπιστημιακό τους ίδρυμα. Αυτό μπορεί να ωθήσει σε ανάπτυξη και καταπολέμηση κάποιων θεμελιωδών προβλημάτων του ιδρύματος, αλλά και να βοηθήσει αισθητά στην απόφαση επιλογής εκπαιδευτικού ιδρύματος και τμήματος για τους μαθητές της τρίτης (3^{ης}) λυκείου. Εν κατακλείδι, είναι σημαντικό να σημειωθεί πως με την ολοκλήρωση της διαδικτυακής εφαρμογής, το αρχικό όραμα και ο σκοπός της ιστοσελίδας επιτεύχθηκε.

2.3 Έρευνα Ιστοσελίδων

Όταν αποφασίστηκε η επιλογή αυτού του θέματος για τη πτυχιακή εργασία έπρεπε αρχικά να γίνει ένα γενικό περίγραμμα για το πώς θα ήταν η δομή του. Όταν η γενική κατεύθυνση της ιδέας τοποθετήθηκε σε μία σειρά, σηματοδοτήθηκε η έναρξη της έρευνας άλλων ιστοσελίδων παρόμοιας θεματολογίας με αυτή που ήταν επιθυμητό να υλοποιηθεί. Βρέθηκαν ποικίλες ιστοσελίδες όπου πολλές από αυτές περιλάμβαναν σύστημα αξιολόγησης (review system), ενώ κάποιες άλλες όχι. Επίσης, σημαντική έρευνα έγινε και για τα πεδία της αξιολόγησης που θα χρησιμοποιούντουσαν, για το οποίο θέμα βρέθηκαν αρκετές πηγές όπως και λήφθηκαν υπόψιν κριτήρια κατατάξεων των ιδρυμάτων.

2.3.1 Παρουσίαση πανεπιστημίων

Σημαντικό ρόλο στην αισθητική της ιστοσελίδας περιλάμβανε η απόφασή για το τρόπο παρουσίασης των πανεπιστημίων και των τμημάτων. Μία πολύ καλή ιδέα προήλθε από την ιστοσελίδα BigFuture (<https://bigfuture.collegeboard.org/college-search/filters>), στην οποία η παρουσίαση των πανεπιστημιακών ιδρυμάτων συμβαίνει με τη χρήση κουτιών (boxes). Παρόμοια αισθητική προσέγγιση ακολουθεί και η ιστοσελίδα ukuni (<https://www.ukuni.net/universities>), αφού σε μορφή τετράγωνου κουτιού συμπεριλαμβάνονται όλες οι βασικές πληροφορίες του πανεπιστημίου.

2.3.2 Σύστημα αξιολόγησης

Στην ιστοσελίδα που δημιουργήθηκε στα πλαίσια της πτυχιακής ο χρήστης κάνει κριτική στο τμήμα το οποίο ανήκει. Έπειτα από την κριτική αυτή ανανεώνεται ο μέσος όρος του τμήματος και ο μέσος όρος του πανεπιστημίου που ανήκει. Η ιδέα για να βαθμολογεί το τμήμα του και όχι απλά το πανεπιστήμιό του, προήλθε από την ιστοσελίδα EduOpinions (<https://www.eduopinions.com/universities/universities-in-united-kingdom/manchester/the-university-of-manchester/computer-science/computer-science-reviews>). Επίσης είναι επιθυμητό να σημειωθεί πως εκτός της ιδέας αυτής, η ιστοσελίδα EduOpinions βοήθησε και στη σχεδίαση των κριτικών. Η δημιουργία των δύο στηλών που εφαρμόστηκε στην ιστοσελίδα μου για την αποτύπωση των πεδίων αξιολόγησης έχει αρκετές ομοιότητες. Παρόμοια εμφάνιση κριτικών παρατηρήθηκε και στο HotCoursesAbroad (<https://www.hotcoursesabroad.com/study/student-reviews-ratings/curtin-university/72214/rankings.html>).

2.3.3 Πεδία αξιολόγησης

Τα πεδία αξιολόγησης ήταν ένα κομμάτι της πτυχιακής που έπρεπε να προσεχθούν ιδιαίτερα. Τα πεδία αυτά όφειλαν να είναι εύστοχα και σχετικά με τις προτεραιότητες και τα άγχη των φοιτητών. Η έρευνα που διεξάχθηκε για τον εντοπισμό τους περιλάμβανε τόσο πεδία αξιολόγησης όσο και κριτήρια κατατάξεων. Όσον αφορά ιστοσελίδες που βοήθησαν στη διαδικασία αυτή ήταν σίγουρα η EduOpinions, το HotCoursesAbroad και το StudentCrowd (https://www.studentcrowd.com/university-11055200-s1008250-university_of_greenwich-south-east-london#booking) και έπειτα ερευνήθηκαν μεγάλοι κολοσσοί στον χώρο των κατατάξεων όπως το Times Higher Education (<https://www.timeshighereducation.com/world-university-rankings/university-cambridge>) και το Top Universities (<https://www.topuniversities.com/universities/stanford-university#p2-rankings>). Εν κατακλείδι, τα κριτήρια κατατάξεων δεν χρησιμοποιήθηκαν, λόγω του γεγονότος ότι η ιστοσελίδα απευθύνεται σε φοιτητές.

3 ΕΡΓΑΛΕΙΑ ΥΛΟΠΟΙΗΣΗΣ

Η λίστα των εργαλείων που συνέβαλαν για την δημιουργία της ιστοσελίδας ήταν μεγάλη. Το βασικό μοντέλο υλοποίησης ήταν το MERN stack [25]. Το μοντέλο MERN αποτελείται από την βάση δεδομένων MongoDB, το framework (σκελετό) Express, τη React για το μπροστινό μέρος και την Node.js για τη συγγραφή του κώδικα της βάσης δεδομένων. Το σύστημα που υποστήριξε και επέτρεψε την ανάπτυξη της πτυχιακής εργασίας ήταν τα Windows 10 Home. Το σύστημα αυτό επιλέχθηκε λόγω της μακροχρόνιας εξοικείωσης που υπήρχε και την χρήση λογισμικών που ανήκουν στην εταιρεία Microsoft. Για παράδειγμα, το Visual Studio Code και το Word που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας λειτουργούν άρτια στο περιβάλλον εργασίας της ίδιας εταιρείας. Τέλος, είναι σημαντικό να αναφερθεί συνοπτικά πως το μοντέλο που ακολουθήθηκε αποτελείται από το 3-tier architecture (αρχιτεκτονική 3 επιπέδων). Η αρχιτεκτονική αυτή περιλαμβάνει τη βάση δεδομένων (database), τον backend server (το RESTful API που δημιουργήθηκε για την επικοινωνία με τη βάση) και το frontend (μπροστινό μέρος). Για τη βάση δεδομένων επιλέχθηκε το MongoDB και για το μπροστινό μέρος της εφαρμογής επιλέχθηκε η React.

3.1 Visual Studio Code

Αρχικά αναφέρουμε την πλατφόρμα δημιουργίας και επεξεργασίας κώδικα που χρησιμοποιήθηκε (Integrated Development Environment- IDE). Η πλατφόρμα αυτή ήταν το Visual Studio Code. Ο λόγος επιλογής της πλατφόρμας αυτής ήταν η παρελθοντική εξοικείωση που υπήρξε και η μεγάλη δημοτικότητα του εργαλείου. Το Visual Studio Code περιλαμβάνει επίσης ποικίλες δυνατότητες μέσω extensions (επεκτάσεων), αλλά και την αυτόματη λειτουργία υπογράμμισης κώδικα όταν εντοπίζεται κάποιο συντακτικό λάθος.

3.2 Node.js

Το Node.js (έκδοση 18.14.1) αποτέλεσε βασικό πυλώνα όσον αφορά την υλοποίηση του backend (της βάσης δεδομένων) της ιστοσελίδας μου. Η βασική λειτουργικότητά του είναι το γεγονός ότι επιτρέπει στον προγραμματιστή, να γράψει όλα τα αρχεία της βάσης του σε JavaScript και να τα συνδέσει μεταξύ τους εύκολα και απλοϊκά [26].

3.2.1 Express

Αξίζει να αναφερθεί και το framework (σκελετός) Express (έκδοση 4.18.2) το οποίο δίνει σαν γάντι με την Node.js όσον αφορά τη δημιουργία RESTful APIs [27]. Τα RESTful APIs στην ουσία χρησιμοποιούν βασικές λειτουργίες όπως POST (ανέβασμα), GET (τράβηξε), PATCH (ενημέρωσε) και DELETE (διέγραψε) για την επικοινωνία με τη βάση του προγραμματιστή και την υλοποίηση διαφόρων λειτουργιών [28]. Για το συγκεκριμένο project η Express αποτελεί μία πολύ συνηθισμένη επιλογή.

3.2.2 Passport.js & Passport-local

Τα πακέτα αυτά ήταν ιδιαίτερος σημαντικά για την ιστοσελίδα. Ο λόγος ήταν η ανάγκη τόσο για την αποθήκευση της πιστοποιημένης εισόδου του χρήστη στην ιστοσελίδα, όσο και για την πιστοποίηση με βάση του ακαδημαϊκού του ταχυδρομικού λογαριασμού και του κωδικού πρόσβασής του [29].

3.2.3 Bcrypt.js

Το Bcrypt ήταν ένα πακέτο που βοήθησε πάρα πολύ στο κομμάτι της ασφάλειας των χρηστών και της αυθεντικοποίησης του χρήστη [30]. Η αυθεντικοποίηση λαμβάνει χώρα μέσω της σύνδεσης του ακαδημαϊκού email με τον κωδικό πρόσβασής του. Για την ασφάλεια χρησιμοποιήθηκε κατακερματισμός του κωδικού πρόσβασης και για την σύνδεση του email με το password, η λειτουργία compare (σύγκριση).

3.3 MongoDB

Το MongoDB (έκδοση 6.0.11) αποτελεί τη βάση δεδομένων που χρησιμοποιήθηκε για την πραγματοποίηση της πτυχιακής εργασίας. Το MongoDB [31] αποτελεί μοντέλο NoSQL, όπου χρησιμοποιεί έγγραφα και συλλογές στη θέση των γραμμών και πινάκων. Τα έγγραφα αποτελούνται από καταχωρήσεις μεταβλητών, ενώ οι συλλογές είναι ένα σύνολο από έγγραφα [32].

3.3.1 Mongoose

Το Mongoose είναι ένα εργαλείο που χρησιμοποιείται για την μοντελοποίηση αντικειμένων (όπως συλλογών και εγγράφων) για το MongoDB και την Node.js [33]. Όσον αφορά τη προκείμενη εργασία, το Mongoose βοήθησε στην σύνδεση από τον τοπικό υπολογιστή με τη MongoDB βάση και στην δημιουργία schemas (σχημάτων) για τις συλλογές και τα έγγραφα.

3.3.2 MongoDB Atlas

Το εργαλείο Atlas χρησιμοποιήθηκε για πολύ απλές λειτουργίες. Συγκεκριμένα, ήταν χρήσιμο στη σύνδεση με τη βάση του MongoDB, αφού ήταν απαραίτητη η εφαρμογή login (σύνδεση) και έπειτα από αυτήν, στο Atlas να τεθεί το IP Address (διεύθυνση IP).

3.3.3 MongoDB Compass

Το MongoDB Compass ήταν η εφαρμογή που κατέβηκε στον υπολογιστή για να υπάρχει πρόσβαση στα έγγραφα και τις συλλογές της βάσης. Στον χώρο αυτό υπάρχει η δυνατότητα παρατήρησης των αλλαγών, των προσθηκών και των διαγραφών των ενεργειών των χρηστών. Επίσης όταν προστεθούν δεδομένα από τον προγραμματιστή, μπορεί να παρατηρηθεί αν εισάχθηκαν σωστά.

3.4 Postman

Το Postman ήταν ένα ιδιαιτέρως χρήσιμο εργαλείο που διευκόλυνε πολύ τα πρωταρχικά στάδια της εργασίας. Το πλαίσιο λειτουργίας του εργαλείου αυτού ήταν για τον έλεγχο της εισαγωγής, τροποποίησης, διαγραφής και τραβήγματος δεδομένων από και προς τη βάση. Επίσης αξίζει να σημειωθεί πως όλα τα πανεπιστήμια και τμήματα περάστηκαν στη βάση από την πλατφόρμα του Postman.

3.5 React

Σε αυτό το σημείο της αναφοράς των εργαλείων φτάσαμε στο μπροστινό μέρος της εφαρμογής (frontend). Ως κύριο εργαλείο για την ανάπτυξη της γραφικής διεπαφής [34], χρησιμοποιήθηκε η ευρέως γνωστή στον χώρο του προγραμματισμού React (έκδοση 17.0.2) [35]. Η React αποτελεί ένα framework (σκελετό) και βιβλιοθήκη (library) για τη πολύ ισχυρή γλώσσα προγραμματισμού JavaScript [36]. Ο κύριος λόγος επιλογής της React ήταν η μεγάλη πληροφορία που υπάρχει στο διαδίκτυο, αλλά και λόγου του μοντέλου MERN stack που χρησιμοποιήσα.

3.5.1 React Router

Για τη δημιουργία μίας θετικής εμπειρίας στον χρήστη και την πλοήγησή του αβίαστα στην ιστοσελίδα ήταν απαραίτητη η χρήση δρομολόγησης. Χρησιμοποιήθηκε η βιβλιοθήκη react-router-dom (έκδοση 6.14.2) για την εφαρμογή των Route (διαδρομών), Link (συνδέσμων), useNavigate (πλοηγήσεων) και άλλων λειτουργιών.

3.6 CSS

Η CSS αποτέλεσε έναν ακόμη σημαντικό πυλώνα, και συγκεκριμένα τον πυλώνα αισθητικής της ιστοσελίδας. Η βασική επιθυμία ήταν να υπάρχει μία ιδιαίτερη εμφάνιση με κυρίαρχο χρώμα το πράσινο. Αυτό επιτεύχθηκε επιτυχώς με τη χρήση της CSS. Επίσης με τη χρήση του @media προσαρμόστηκε η ιστοσελίδα για όλες τις σύγχρονες διαστάσεις των τεχνολογικών συσκευών.

3.6.1 Flexbox

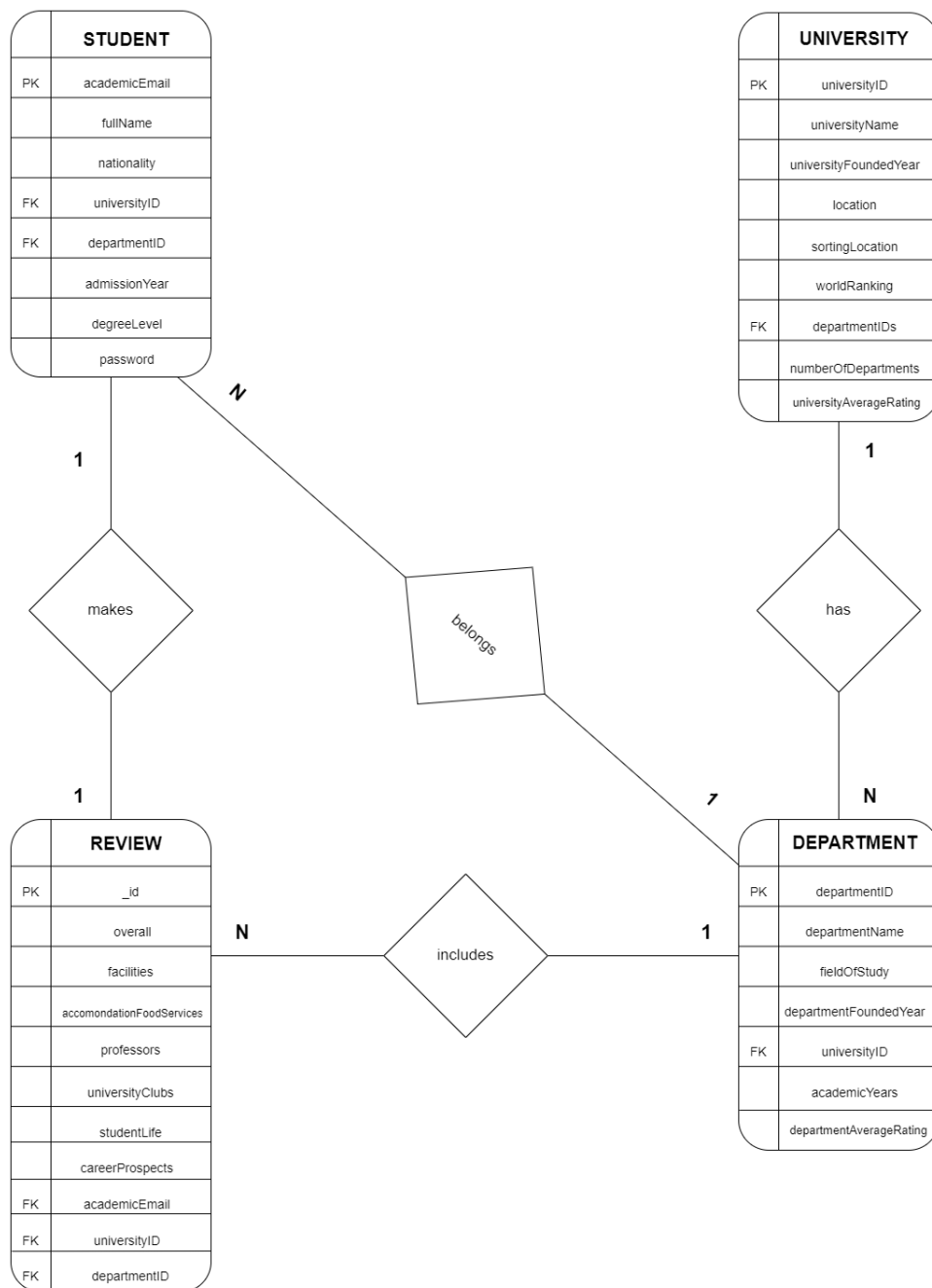
Το Flexbox είναι ένα μοντέλο διάταξης που βοηθάει τον προγραμματιστή να τοποθετήσει τα αντικείμενα σε γραμμές ή στήλες [37]. Οι δυνατότητες του Flexbox είναι τεράστιες όσον αφορά την εμφάνιση των αντικειμένων και αποτέλεσε σημαντικό εργαλείο στην παρουσίαση της ιστοσελίδας. Βασικές λειτουργίες που χρησιμοποιήθηκαν ήταν το flex-direction, justify-content και align-items.

4 ΥΛΟΠΟΙΗΣΗ ΠΙΣΩ ΜΕΡΟΥΣ (BACKEND)

Το πίσω μέρος της εφαρμογής αποτελείται κυρίως από το αρχείο `routes.js`. Φυσικά, θα γίνουν αναφορές και σε άλλα αρχεία, αλλά αυτό είναι το πιο σημαντικό. Ο λόγος είναι πως το αρχείο αυτό διαχειρίζεται την επικοινωνία της βάσης δεδομένων με την εμφάνιση των δεδομένων αυτών στον χρήστη. Επίσης, σημαντικό κομμάτι αποτελεί και η σχεδίαση των βάσεων, που δημιουργήθηκε πριν την υλοποίηση του κώδικα.

4.1 Διάγραμμα σχέσης οντοτήτων (ERD)

Τα βήματα για την επιτυχής υλοποίηση μίας βάσης δεδομένων είναι αρκετά περισσότερα από αυτά που πολύς κόσμος οραματίζεται. Έπειτα από την έρευνα που έγινε για την επιλογή των βάσεων και των χαρακτηριστικών (πεδίων) που θα περιέχονται, ήταν σημαντική η αποτύπωση της σχέσης που θα έχουν μεταξύ τους [38]. Αυτό ήταν ένα απαραίτητο βήμα για να υπάρχει ένα ξεκάθαρο σχέδιο υλοποίησης. Στην διαδικασία αυτή δέχτηκα βοήθεια από τα δύο αυτά βίντεο [39] και [40].



Διάγραμμα 1: ERD

Όπως μπορεί να παρατηρηθεί και από το διάγραμμα που δημιουργήθηκε οι οντότητες που χρησιμοποιήθηκαν ήταν τέσσερεις και αποτελούνται από την Student (Μαθητής), University (Πανεπιστήμιο), Department (Τμήμα) και Review (Κριτική). Το επόμενο βήμα αφού επιλέχθηκαν οι οντότητες αυτές ήταν η σχέση που θα υπήρξε μεταξύ τους. Ένας μαθητής ανήκει σε ένα πανεπιστήμιο και κατά συνέπεια σε ένα τμήμα, ένα τμήμα όμως έχει πολλούς μαθητές για αυτό και το N προς 1. Έπειτα ήταν σωστό να αποτυπωθεί η σχέση του πανεπιστημίου με τα τμήματα. Για την ομαλή και απλή λειτουργία της βάσης αποφασίστηκε πως ένα τμήμα ανήκει μόνο σε ένα πανεπιστήμιο. Αυτό με βάση το όνομα δεν αληθεύει, αφού πολλά πανεπιστήμια έχουν

τμήματα που μοιράζονται το ίδιο όνομα. Παρ' όλα αυτά για την ευδιάκριτη διαφοροποίηση των κριτικών και την κατανομή τους στο σωστό τμήμα, ταίριαζε περισσότερο αυτή η λογική. Έπειτα ακολούθησε η σύνδεση των κριτικών με τα τμήματα. Ένα τμήμα μπορεί να έχει πολλές κριτικές, ενώ μία κριτική απευθύνεται αυστηρά μόνο σε ένα τμήμα. Τέλος, έπρεπε να ληφθεί η σχέση μεταξύ του μαθητή και της κριτικής που δημιουργεί. Για να μην μπορεί να επηρεάσει άδικα τη βαθμολογία ένας μαθητής, του επιτρέπεται να κάνει μία μόνο κριτική. Αυτή τη κριτική φυσικά μπορεί αν θέλει να τη τροποποιήσει.

4.1.1 Περιγραφή των χαρακτηριστικών

Στο παραπάνω κείμενο περιγράφηκαν οι οντότητες που αναγράφονται στο διάγραμμα σχέσης οντοτήτων και την σχέση που έχουν μεταξύ τους. Σε αυτό το σημείο της συγγραφής θα αναφερθούμε στα χαρακτηριστικά της κάθε οντότητας. Πολλά από τα χαρακτηριστικά αυτά πηγάζουν από εμένα και κάποια άλλα είναι εμπνευσμένα από πηγές που έχουν αναφερθεί παραπάνω στην έρευνα που συντελέστηκε.

STUDENT	
PK	academicEmail
	fullName
	nationality
FK	universityID
FK	departmentID
	admissionYear
	degreeLevel
	password

Διάγραμμα 2: Οντότητα student

Θα ήταν επιθυμητή η έναρξη με την οντότητα Student (Μαθητής). Τα χαρακτηριστικά που συντελούν την οντότητα αυτή είναι πρώτον και πιο σημαντικό το academicEmail (ακαδημαϊκό email). Το ακαδημαϊκό email, όπως και το password (κωδικός πρόσβασης) που θα αναφερθούν και παρακάτω, αποτελούν τα δύο χαρακτηριστικά που ο χρήστης συμπληρώνει για να κάνει login (σύνδεση) στη πλατφόρμα. Έπειτα έχουμε το πεδίο FullName (ονοματεπώνυμο) και την εθνικότητα. Για ξένα κλειδιά έχουμε το universityID (πανεπιστημιακό id) και το departmentID (id του τμήματος) τα οποία είναι ιδιαίτερος σημαντικά, αφού χωρίς αυτά δε μπορούμε να γνωρίζουμε σε ποιο τμήμα ο μαθητής μπορεί να κάνει τη κριτική του. Με βάση αυτά τα δύο γνωρίσματα επιτρέπεται ή δεν επιτρέπεται στον χρήστη να αφήσει κριτική. Έπειτα ακολουθεί το admissionYear (έτος εισαγωγής), το degreeLevel (επίπεδο πτυχίου) και ο κωδικός (password) που αναφέρθηκε παραπάνω. Ο κωδικός πρόσβασης

αξιίζει να αναφερθεί πως δέχεται κατακερματισμό για την ασφάλεια των δεδομένων των χρηστών.

UNIVERSITY	
PK	universityID
	universityName
	universityFoundedYear
	location
	sortingLocation
	worldRanking
FK	departmentIDs
	numberOfDepartments
	universityAverageRating

Διάγραμμα 3: Οντότητα university

Επόμενη οντότητα είναι αυτή του πανεπιστημίου. Για το πανεπιστήμιο ορίστηκε ένα id με όνομα universityID (πανεπιστημιακό id) που είναι και το βασικό κλειδί. Έπειτα ακολουθεί το universityName (όνομα πανεπιστημίου) και το έτος ίδρυσης του πανεπιστημίου (universityFoundedYear). Σημαντική είναι και η τοποθεσία (location) για την οποία τέθηκε η βασική έδρα του πανεπιστημίου. Πολλά πανεπιστήμια έχουν τμήματα σε πάνω από ένα νομούς, για αυτόν τον λόγο επιλέχθηκε ο κυρίαρχος νομός του πανεπιστημίου. Το sortingLocation (ταξινόμηση τοποθεσίας) είναι ένα χαρακτηριστικό που προστέθηκε για την δημιουργία φίλτρων με βάση τη τοποθεσία. Το χαρακτηριστικό αυτό λαμβάνει μία από τις τρεις τιμές North (Βόρεια), Central (Κεντρικά) και South (Νότια). Το worldRanking (θέση στον κόσμο) είναι η θέση που καταλαμβάνει το πανεπιστήμιο με βάση τα πανεπιστήμια όλου του κόσμου. Ως ξένο κλειδί έχει τεθεί το departmentIDs (id των τμημάτων) το οποίο έχει μορφή πίνακα και δέχεται τα τμήματα που ανήκουν στο πανεπιστήμιο αυτό. Τα τελευταία χαρακτηριστικά της οντότητας είναι τα numberOfDepartments (αριθμός των τμημάτων) και universityAverageRating (μέση βαθμολογία του πανεπιστημίου).

DEPARTMENT	
PK	departmentID
	departmentName
	fieldOfStudy
	departmentFoundedYear
FK	universityID
	academicYears
	departmentAverageRating

Διάγραμμα 4: Οντότητα department

Όσον αφορά την οντότητα department (τμήμα) η διαδικασία επιλογής των χαρακτηριστικών ήταν πολύ απλή. Αρχικά τέθηκε το πρωτεύον κλειδί departmentID (id του τμήματος) και έπειτα το departmentName (όνομα τμήματος). Ακολούθησε το χαρακτηριστικό fieldOfStudy (πεδίο σπουδών) το οποίο περιγράφει το πεδίο που ανήκει το συγκεκριμένο τμήμα που φοιτά ο φοιτητής. Το departmentFoundedYear (έτος ίδρυσης του τμήματος) ήταν το επόμενο και έπειτα εισήχθηκε το universityID (πανεπιστημιακό id) για τη σύνδεση των δύο αυτών οντοτήτων. Τέλος, ακολούθησαν τα academicYears (ακαδημαϊκά χρόνια) τα οποία περιγράφουν τα ελάχιστα χρόνια για την ολοκλήρωση των σπουδών του φοιτητή και το departmentAverageRating (μέση βαθμολογία τμήματος).

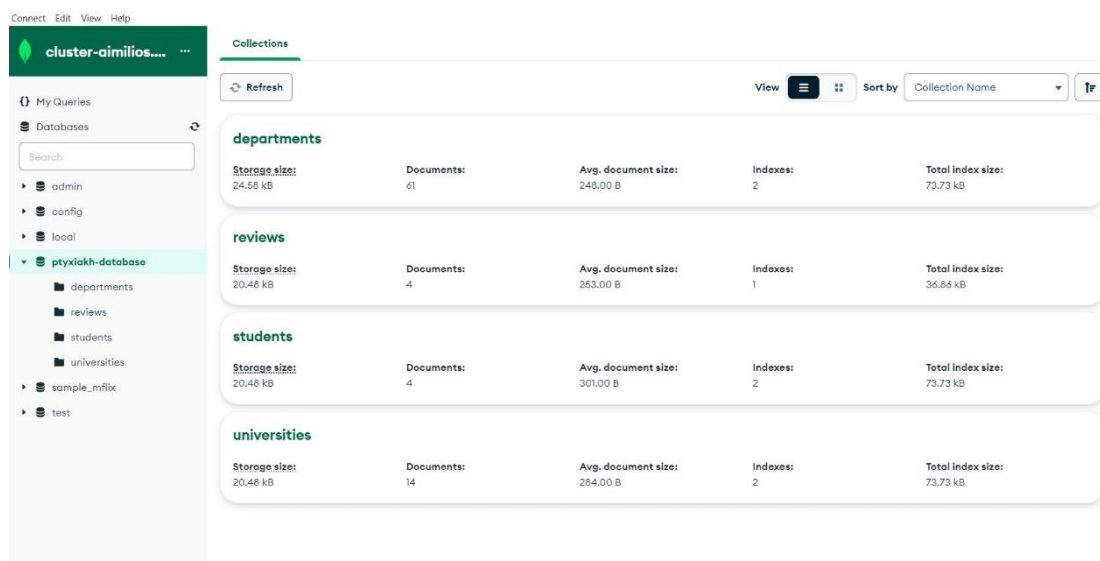
REVIEW	
PK	_id
	overall
	facilities
	accommodationFoodServices
	professors
	universityClubs
	studentLife
	careerProspects
FK	academicEmail
FK	universityID
FK	departmentID

Διάγραμμα 5: Οντότητα review

Η τελευταία, αλλά πολύ σημαντική οντότητα ήταν αυτή της κριτικής. Για την οντότητα αυτή έχουν αναφερθεί ήδη αρκετά στο κομμάτι της έρευνας που περιγράφεται παραπάνω στο κεφάλαιο **2.1** και τώρα είναι ώρα να παρουσιαστούν τα χαρακτηριστικά που τελικά χρησιμοποιήθηκαν. Αρχικά, για πρωτεύον κλειδί τέθηκε το `id` που δημιουργεί το Mongo (`_id`) γιατί με εξυπηρετούσε πολύ όσον αφορά την συμπλήρωση της φόρμας της κριτικής από τους φοιτητές. Επέτρεπε τη σύνδεση της οντότητας `review` (κριτικής) με άλλες οντότητες, χωρίς ο μαθητής να θέτει ένα `review id` (`id` κριτικής). Έπειτα ακολούθησε το χαρακτηριστικό `overall` (συνολικά) που χρησιμοποιείται ως το κριτήριο για τη παραγωγή βαθμολογίας του `departmentAverageRating` (μέση βαθμολογία τμήματος). Τα `facilities` (εγκαταστάσεις), `accommodationFoodServices` (υπηρεσίες στέγασης και φαγητού), `professors` (καθηγητές), `universityClubs` (πανεπιστημιακές λέσχες), `studentLife` (φοιτητική ζωή) και `careerProspects` (προοπτικές καριέρας) αποτελούν τα κριτήρια βαθμολόγησης του φοιτητή. Ο φοιτητής επιλέγει τη συνολική του βαθμολογία και έπειτα βαθμολογεί αυτά τα επιμέρους χαρακτηριστικά. Τέλος, ενσωματώθηκαν και τα ξένα κλειδιά `academicEmail` (ακαδημαϊκό email), `universityID` (πανεπιστημιακό id) και `departmentID` (id του τμήματος) για την επικοινωνία των οντοτήτων.

4.2 Συλλογές & έγγραφα MongoDB

Έχει αναφερθεί παραπάνω πως το MongoDB χρησιμοποιεί συλλογές και έγγραφα. Σε ένα γενικό πλαίσιο για την κατανόηση της λειτουργίας του, μπορεί κανείς να φανταστεί πως οι συλλογές αποτελούν τις οντότητες και τα έγγραφα το αποτέλεσμα εισαγωγής των χαρακτηριστικών. Η αναπαράσταση των δεδομένων έπειτα από την εισαγωγή, τροποποίηση και διαγραφή τους είναι ιδιαίτερος σημαντικές για να δίνουν την απαραίτητη πληροφορία στον προγραμματιστή. Για αυτόν τον λόγο σε αυτό το κεφάλαιο θα επισυναφθούν εικόνες που περιγράφουν τη διεπαφή χρήστη του MongoDB για την απεικόνιση των συλλογών και εγγράφων. Αξίζει να αναφερθεί πως οι εικόνες αυτές προέρχονται από το λογισμικό MongoDB Compass.



Εικόνα 1: Συλλογές MongoDB

Αυτή είναι η γραφική διεπαφή του MongoDB Compass όσον αφορά την αναπαράσταση των συλλογών. Όπως μπορεί να παρατηρηθεί από το αριστερό μέρος του menu (μενού) βρισκόμαστε στη βάση δεδομένων **ptyxiakh-database** (βάση δεδομένων πτυχιακής), όπου περιλαμβάνονται όλες οι απαραίτητες συλλογές σύμφωνα με το διάγραμμα σχέσης οντοτήτων. Στην κάθε συλλογή αναπαρίσταται το μέγεθος αποθηκευτικού χώρου, ο αριθμός των εγγράφων, το μέσο μέγεθος των εγγράφων, τα ευρετήρια και το συνολικό μέγεθος ευρετηρίου. Συγκεκριμένα, ο αριθμός των εγγράφων είναι πολύ σημαντικός για την εργασία αυτή, αφού είναι εύκολη η παρατήρηση νέων χρηστών ή κριτικών στην πλατφόρμα μας.

The screenshot shows the MongoDB Compass web interface. At the top, there's a header with 'Documents' and 'ptyxiakh-databa...'. Below that, the collection name 'ptyxiakh-database.universities' is displayed, along with counts for 'DOCUMENTS' (14) and 'INDEXES' (2). A navigation bar includes 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. A search bar with a 'Filter' icon and a query input field is present, with 'Reset', 'Find', and 'Options' buttons. Below the search bar, there are 'ADD DATA' and 'EXPORT DATA' buttons. The main area displays two documents in a JSON-like format. The first document is for 'Ionian University' and the second is for 'Aristotle University'.

```
{
  "_id": ObjectId("64a006736d443b16b905ba61"),
  "universityID": "IU",
  "universityName": "Ionian University",
  "universityFoundedYear": 1984,
  "location": "Ionian Islands",
  "worldRanking": 5946,
  "departmentIDs": Array,
  "numberOfDepartments": 12,
  "___V": 0,
  "sortingLocation": "North",
  "universityAverageRating": 3.5
}
```

```
{
  "_id": ObjectId("64a03d61aed5b4e9d0b07b47"),
  "universityID": "AUTH",
  "universityName": "Aristotle University",
  "universityFoundedYear": 1925,
  "location": "Thessaloniki",
  "worldRanking": 530,
  "departmentIDs": Array,
  "numberOfDepartments": 40,
  "___V": 0
}
```

Εικόνα 2: Έγγραφα MongoDB

Στο παραπάνω στιγμιότυπο αναπαρίστανται τα έγγραφα από τη συλλογή universities (πανεπιστήμια). Όπως μπορεί να παρατηρηθεί πάνω δεξιά αποτυπώνονται ο αριθμός των εγγράφων και ο αριθμός των ευρετηρίων. Το πρώτο έγγραφο είναι το Ιόνιο Πανεπιστήμιο, το οποίο περιλαμβάνει όλα τα χαρακτηριστικά που αναφέρθηκαν παραπάνω στο υποκεφάλαιο 2.2.1. Το departmentIDs (id των τμημάτων) έχει μορφή πίνακα, το sortingLocation (ταξινόμηση τοποθεσίας) έχει τη τιμή “North” (Βόρεια) και το universityAverageRating (μέση βαθμολογία του πανεπιστημίου) έχει τη τιμή 3.5 τη δεδομένη χρονική στιγμή. Επίσης το MongoDB έχει παράξει αυτόματα το _id όπως φαίνεται και στο στιγμιότυπο. Στην οθόνη αυτή μπορούν να γίνουν τροποποιήσεις και διαγραφές εγγράφων, αλλά δεν προτείνεται. Η διαδικασία που ακολουθήθηκε περιλάμβανε εισαγωγή, τροποποίηση και διαγραφή δεδομένων με τη χρήση του εργαλείου Postman.

4.3 RESTful API

Για τη λειτουργικότητα των RESTful APIs [41] έχει γίνει μία περιεκτική αναφορά στην υποενότητα **1.1.2.1**. Όπως αναφέρθηκε και προηγουμένως οι μέθοδοι POST (ανέβασμα), GET (τράβηξε), PATCH (ενημέρωσε) και DELETE (διέγραψε) αποτελούν τις θεμελιώδεις λειτουργίες των API αυτών. Σε αυτό το κεφάλαιο είναι σημαντικό να ειπωθεί η διαδικασία που ακολουθήθηκε για την υλοποίηση της βάσης με όλα τα απαραίτητα βήματα. Βασική πηγή που χρησιμοποιήθηκε ήταν ο guide (οδηγός) της αναφοράς [42]. Όπως θα παρατηρηθεί και παρακάτω, θα αναπτύξουμε τη διαδικασία παραγωγής των εγγράφων `index.js` (του εξυπηρετητή αρχείου), `routes.js` (που συμπεριλαμβάνονται οι μέθοδοι `post`, `get`, `patch`, `delete`) και τα `models` (όπου συμπεριλαμβάνονται οι τέσσερις συλλογές `student`, `university`, `department` και `review`).

4.3.1 Αρχείο `index.js`

Το αρχείο `index.js` είναι υπαίτιο για το τρέξιμο (`run`) της βάσης δεδομένων μας. Σε αυτό το αρχείο συμπεριλήφθηκαν όλα τα απαραίτητα πακέτα, βιβλιοθήκες και frameworks (σκελετοί) για την ορθή λειτουργία της βάσης δεδομένων, όπως η `express`, το `mongoose`, το `cors` και το `passport`. Στο έγγραφο αυτό υλοποιείται η σύνδεση της βάσης του MongoDB με το τοπικό σύστημα του υπολογιστή. Αρχικά, δημιουργήθηκε ένα αρχείο με ονομασία `file.env`, το οποίο περιέχει τον σύνδεσμο επικοινωνίας με τη βάση του Mongo και την επιλογή της πύλης (`port`) που θέλουμε να τρέξει. Έπειτα ο σύνδεσμος αυτός χρησιμοποιήθηκε από το `mongoose` για να υλοποιηθεί επιτυχής σύνδεση.

4.3.2 Δημιουργία των `models` (μοντέλων)

Τα μοντέλα τα οποία κατά βάση είναι `schemas` (σχήματα) είναι πολύ μεγάλης σημασίας για την δημιουργία του RESTful API. Για ακόμη μία φορά με τη χρήση του `mongoose` υλοποιήθηκαν οι οντότητες και τα χαρακτηριστικά τους, σε μία μορφή που να είναι κατανοητή από το MongoDB και το τοπικό υπολογιστικό σύστημα. Τα μοντέλα αυτά ήταν πολύ απλοϊκά όταν πρώτο δημιουργήθηκαν, άλλα έπειτα από ενσωματώσεις νέων μηχανισμών, προστέθηκε επιπρόσθετος κώδικας. Όσον αφορά το `universitySchema` (σχήμα πανεπιστημίου) και `departmentSchema` (σχήμα τμήματος) η υλοποίηση ήταν πολύ εύκολη και βασίστηκε σχεδόν πλήρως στον οδηγό που ακολουθήθηκε. Για το `reviewSchema` (σχήμα κριτικής) και `studentSchema` (σχήμα μαθητή) η διαδικασία ήταν επίσης η ίδια, απλώς έπρεπε να προστεθούν κάποια πράγματα. Στο μοντέλο του `student` (μαθητή) υλοποιείται ο κατακερματισμός του κωδικού, όπως και γίνεται ο έλεγχος αν ο κωδικός αντιστοιχεί στον κωδικό που εισάγεται. Για το μοντέλο `review` (κριτικής) προστέθηκε μόνο ένα μικρό κομμάτι κώδικα που ελέγχει αν το ακαδημαϊκό email που εισάγεται υπάρχει, το οποίο δεν είναι ιδιαίτερος χρήσιμο. Ο λόγος για αυτό είναι επειδή ενσωματώθηκε η λειτουργία `login` (σύνδεση) του χρήστη στη πλατφόρμα, όπου το ακαδημαϊκό του email συμπληρώνεται αυτόματα στη φόρμα κριτικής, οπότε δε μπορεί να εισαχθεί λάθος email.

```

ptyxiakh2 > model > JS department.js > ...
1  const mongoose = require('mongoose');
2
3  const departmentSchema = new mongoose.Schema({
4    departmentID: {
5      type: String,
6      required: true,
7      unique: true
8    },
9    departmentName: {
10     type: String,
11     required: true
12   },
13   fieldOfStudy: {
14     type: String,
15     required: true
16   },
17   departmentFoundedYear: {
18     type: Number,
19     required: true
20   },
21   universityID: {
22     type: mongoose.Schema.Types.ObjectId,
23     ref: 'University', // Reference to the University model
24     required: true
25   },
26   academicYears: {
27     type: Number,
28     required: true
29   },
30   departmentAverageRating: { // New field to store average rating
31     type: Number,
32     default: 0 // Default value is 0
33   }
34 });
35
36 module.exports = mongoose.model('Department', departmentSchema);

```

Εικόνα 3: Κώδικας μοντέλου department

4.3.3 Αρχείο passportConfig.js

Για το αρχείο αυτό θα γίνει μία περιεκτική αναφορά γιατί ανήκει κυρίως στην ενότητα του login/signup system (σύστημα εισόδου/εγγραφής). Η πηγή που βοήθησε για την δημιουργία της επιτυχούς σύνδεσης και εγγραφής του χρήστη θα υπάρξει παρακάτω στο ανάλογο κεφάλαιο. Το μόνο που θέλω να αναφέρω είναι πως στο κομμάτι της σύνδεσης βοήθησε ενεργά ο κώδικας που συγγράφηκε. Ο λόγος ήταν πως έβρισκε το email του χρήστη και έπειτα πήγαινε στο μοντέλο του μαθητή για να ελέγξει αν ο κωδικός πρόσβασης που έγραψε ο χρήστης είναι ο σωστός για τον συγκεκριμένο χρήστη.

4.3.4 Αρχείο routes.js

4.3.4.1 Διαδρομές για το μοντέλο μαθητή

Το τελικό μέρος της δημιουργίας του RESTful API ήταν το αρχείο `routes.js`. Το αρχείο αυτό είναι υπαίτιο για όλες τις διαδρομές (routes) των λειτουργιών που επιθυμούμε. Οι συνολικές γραμμές του αρχείου ανέρχονται στις 447, το οποίο εξηγεί πως για τη δημιουργία μίας ολοκληρωμένης ιστοσελίδας υπάρχουν πολλές διαδρομές που οφείλουν να υλοποιηθούν. Οι βασικοί μέθοδοι διαδρομών έχουν περιγραφεί περιεκτικά παραπάνω και αποτελούνται από τις μεθόδους `post`, `get`, `update` και `delete`. Για την εγγραφή και την σύνδεση του χρήστη στη πλατφόρμα χρησιμοποιήθηκε το `Post` (ανέβασμα). Δεν θα απευθυνθούμε πολύ σε αυτό το κομμάτι ακόμη, αλλά θα πούμε ότι χρησιμοποιήθηκε το πακέτο `passport` για την ορθή αυθεντικοποίηση των εισαγόμενων δεδομένων. Έπειτα οι διαδρομές `getOneStudent` (τράβα τον συγκεκριμένο μαθητή) και `getAllStudents` (τράβα όλους τους μαθητές) ακολούθησαν. Αξίζει να αναφερθεί πως τα `get all` και `get one` χρησιμοποιήθηκαν και για όλες τις υπόλοιπες συλλογές. Έπειτα έπρεπε να συμπεριληφθεί σε περίπτωση ανάγκης ή αλλαγής το `updateStudent` (ενημέρωση μαθητή) και `deleteStudent` (διαγραφή μαθητή). Εννοείται, πως στη διαδικασία που περιγράφεται ο οδηγός βοήθησε πολύ δραστικά αφού δεν υπήρχαν προ υπάρχουσες γνώσεις στο θέμα αυτό. Τα `update` (ενημέρωση) και `delete` (διαγραφή) προστέθηκαν και για όλες τις υπόλοιπες συλλογές.

4.3.4.2 Διαδρομές για το μοντέλο πανεπιστημίου

Έπειτα είναι σημαντική η αναφορά στις διαδρομές που αφορούν τα πανεπιστήμια. Στο πεδίο αυτό όπως και για τα τμήματα, προστέθηκε κώδικας για τη δημιουργία μέσης βαθμολογίας. Οι μέσες βαθμολογίες θα αναλυθούν παρακάτω στην αναφορά. Το `getAllUniversities` (τράβα όλα τα πανεπιστήμια) αποτέλεσε μία πολύ σημαντική διαδρομή, όπου προστέθηκε κώδικας για το φίλτρο τοποθεσίας και για τη σελιδοποίηση (pagination). Η σελιδοποίηση περιλαμβάνει τη παρουσίαση τριών πανεπιστημίων για κάθε σελίδα. Οι υπόλοιπες διαδρομές για τα πανεπιστήμια έχουν ήδη αναφερθεί.

4.3.4.3 Διαδρομές για το μοντέλο τμήματος

Για τα τμήματα είναι σημαντικό να γίνει αναφορά σχετικά με τη διαδρομή `getDepartmentsByUniversity` (τράβα τα τμήματα ενός συγκεκριμένου πανεπιστημίου). Η διαδρομή αυτή εξυπηρέτησε πολύ τη κατασκευή της ιστοσελίδας και ακολουθεί παρόμοια λογική με τη διαδρομή `getAllUniversities`. Ο λόγος είναι ότι περιέχει και αυτή σελιδοποίηση με αποτέλεσμα ο κώδικας να μοιάζει σε μεγάλο βαθμό. Επίσης όπως περιγράφεται και από το όνομά της, χρησιμοποιεί το `id` του πανεπιστημίου.

4.3.4.4 Διαδρομές για το μοντέλο κριτικής

Για τις διαδρομές των κριτικών υπάρχουν κάποια αρκετά ιδιαίτερα χαρακτηριστικά. Αρχικά, οφείλει να αναφερθεί πως στις διαδρομές `postReview` (ανέβασμα κριτικής), `updateReview` (ενημέρωση κριτικής) και `deleteReview` (διαγραφή κριτικής) καλείται η συνάρτηση υπολογισμού της μέσης βαθμολογίας του τμήματος. Έπειτα, υπάρχει και η διαδρομή `getReviewsByDepartment` (τράβα όλες τις κριτικές ενός συγκεκριμένου τμήματος), όπου χρησιμοποιείται για την εμφάνιση των κριτικών στην ιστοσελίδα. Εννοείται πως για την ορθή λειτουργία της διαδρομής αυτής χρησιμοποιείται το `id` του τμήματος.

4.4 Βάση συστήματος εισόδου/εγγραφής

Το σύστημα αυτό αποτέλεσε ένα πολύ μεγάλο κομμάτι της εργασίας. Για την επιτυχής υλοποίηση ενός τέτοιου συστήματος έπρεπε να γίνει το backend (πίσω μέρος), τα κουμπιά login (σύνδεση) και signup (εγγραφή) και οι φόρμες συμπλήρωσης. Στο backend ανήκουν πολλές υλοποιήσεις όπως το αρχείο passportConfig.js, οι διαδρομές στο routes.js και η χρήση του Bcrypt. Το σύστημα αυτό ήταν απαραίτητο για την αυθεντικοποίηση των χρηστών, την ομαλή λειτουργία της ιστοσελίδας και την παροχή κατάλληλης ασφάλειας. Τέλος, βοήθησε πολύ ο εξής οδηγός [43] που ακολουθήθηκε για την επιτυχία της ενέργειας αυτής.

4.4.1 Bcrypt.js

Για το πακέτο Bcrypt έχουμε ήδη αναφερθεί απλώς θα ήταν σημαντικό να προστεθεί πως ο κατακερματισμός του κωδικού πρόσβασης ήταν πολύ σημαντικός για την διασφάλιση των δεδομένων των χρηστών. Επίσης, με τη χρήση της λειτουργίας compare (σύγκριση) του Bcrypt έγινε έλεγχος αν ο εισαγόμενος κωδικός πρόσβασης ταιριάζει με τον κατοχυρωμένο κωδικό πρόσβασης.

4.4.2 Αρχείο passportConfig.js

Για το αρχείο αυτό έχει γίνει ήδη μία περιεκτική αναφορά στο υποκεφάλαιο 3.1.3. Παρ' όλα αυτά αξίζει να επεξηγηθεί περαιτέρω η διαδικασία δημιουργίας του αρχείου. Ο οδηγός βοήθησε πολύ στο σημείο αυτό. Για το login (σύνδεση) του χρήστη ζητείται μόνο το academicEmail (ακαδημαϊκό email) και το password (κωδικός πρόσβασης). Για το signup (εγγραφή) η διαδικασία είναι λίγο πιο σύνθετη, για αυτόν τον λόγο χρησιμοποιήθηκε το req.body για τον προσδιορισμό των επιπρόσθετων μεταβλητών. Για την εγγραφή επίσης, συμβαίνει και ένας έλεγχος σε περίπτωση που ο χρήστης εισάγει ένα ακαδημαϊκό email που υπάρχει ήδη. Τέλος, για το κομμάτι κώδικα του login (σύνδεσης), αποθηκεύεται το email του μαθητή και χρησιμοποιείται για την σωστή αντιστοίχιση με τον κωδικό πρόσβασης.

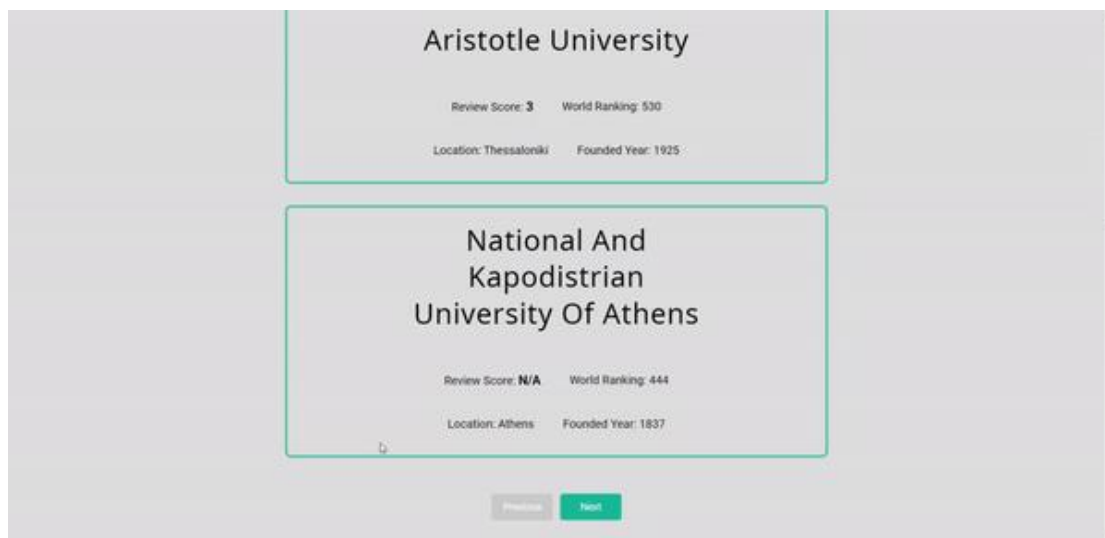
4.4.3 Διαδρομές στο routes.js

Για την ομαλή λειτουργία της εγγραφής και της σύνδεσης έπρεπε να υλοποιηθούν οι αντίστοιχες διαδρομές. Η μέθοδος που εφαρμόστηκε ήταν η post (ανέβασμα) και χρησιμοποιήθηκε το passport.authenticate για την επικοινωνία με το αρχείο passportConfig.js. Έπειτα, σημαντική ήταν και η συγγραφή μηνύματος σε περίπτωση επιτυχίας, η οποία χρησιμοποιήθηκε για τη σύνδεση του πίσω μέρους με του μπροστινού και είχε ως αποτέλεσμα την εμφάνιση ενός κατάλληλου μηνύματος στον χρήστη.

4.5 Σελιδοποίηση (pagination)

Η σελιδοποίηση είναι μία λειτουργία που εφαρμόζεται σε όλες τις αξιοπρεπείς και προσεγμένες ιστοσελίδες. Η ενσωμάτωσή της στην ιστοσελίδα που δημιουργήθηκε ήταν απαραίτητη, λόγω του μεγάλου όγκου των πανεπιστημίων και των τμημάτων που εκχωρήθηκαν. Τα πανεπιστήμια στο άθροισμά τους ήταν δεκατέσσερα (14), ενώ τα τμήματα ήταν εξήντα ένα (61). Για την υλοποίηση της σελιδοποίησης ακολουθήθηκε η λογική που αναπαρίσταται στις πηγές [44] και [45]. Ειδικότερα ο κώδικας της δεύτερης πηγής και η μεθοδολογία συγγραφής του, βοήθησε πολύ στην δημιουργία του δικού μας.

Το αρχείο το οποίο περιλάμβανε τη σελιδοποίηση ήταν το routes.js και συγκεκριμένα οι διαδρομές getAllUniversities (τράβα όλα τα πανεπιστήμια) και getDepartmentsByUniversity (τράβα τα τμήματα ενός συγκεκριμένου πανεπιστημίου). Για τη σελιδοποίηση των πανεπιστημίων τις βασικές μεταβλητές αποτέλεσαν ο αριθμός πανεπιστημίων ανά σελίδα (universitiesPerPage), η τιμή της σελίδας (page) και η τιμή skip (παράλειψη). Έπειτα με τη μεταβλητή count (καταμέτρηση) βρίσκουμε τον συνολικό αριθμό των πανεπιστημίων και τον διαιρούμε με τον αριθμό των πανεπιστημίων ανά σελίδα για να βρούμε τον συνολικό αριθμό σελιδών που χρειαζόμαστε. Όσον αφορά τη skip (παράλειψη) και limit (όριο), η skip βοηθάει στο να εμφανιστούν τα κατάλληλα πανεπιστήμια με βάση τη σελίδα που βρισκόμαστε και το limit θέτει ως ανώτατο όριο παρουσίασης τη τιμή που θέσαμε στη μεταβλητή universitiesPerPage. Η διαδικασία που ακολουθήθηκε για τη σελιδοποίηση των τμημάτων βασίζεται ακριβώς στην ίδια μεθοδολογία.



Εικόνα 4: Εμφάνιση σελιδοποίησης

4.6 Δημιουργία μέσου όρου βαθμολογιών

Οι μέσοι όροι για τις βαθμολογίες των τμημάτων και πανεπιστημίων αποτελούν το τελευταίο κομμάτι του συστήματος αξιολόγησης στην ιστοσελίδα. Η δημιουργία τους ήταν απαραίτητη για να θεωρείται ολοκληρωμένη η ιστοσελίδα, ώστε οι επισκέπτες να γνωρίζουν με μία γρήγορη ματιά την άποψη των φοιτητών για τα ιδρύματα. Η υλοποίηση συνέβη καθαρά στο πίσω μέρος της εφαρμογής, για αυτόν τον λόγο θα αναλυθεί σε αυτό το κεφάλαιο. Όσον αφορά τη διαδικασία υλοποίησης μεταξύ των τμημάτων και των πανεπιστημίων, υπάρχουν διαφορές, για αυτόν τον λόγο θα χωριστούν σε δύο διαφορετικά μέρη.

4.6.1 Μέσος όρος τμήματος

Όλη η συγγραφή της λειτουργίας του υπολογισμού του μέσου όρου έγινε στο αρχείο `routes.js`. Δημιουργήθηκε η συνάρτηση `calculateAverageDepartmentRating` (υπολόγισε τη μέση βαθμολογία του τμήματος), όπου αρχικά ψάχνει όλες τις κριτικές που ανήκουν στο συγκεκριμένο τμήμα (με βάση το `id` του τμήματος) και τις αποθηκεύει. Έπειτα με τη χρήση της μεθόδου `.reduce`, προσθέτουμε όλες τις βαθμολογίες (τις `overall` βαθμολογίες) και τις περνάμε στη μεταβλητή `totalDepartmentRating` (συνολική βαθμολογία τμήματος). Αν οι κριτικές για το τμήμα είναι περισσότερες από μηδέν (0), τότε διαίρεσε τη συνολική βαθμολογία (όλων των κριτικών του τμήματος) με τον αριθμό των κριτικών. Αν δεν υπάρχουν κριτικές, τότε πέρασε ως μέση βαθμολογία το μηδέν. Ακολουθεί η ενημέρωση της βαθμολογίας του τμήματος με τη χρήση δύο δεκαδικών σε περίπτωση που χρειαστούν. Τέλος, γίνεται εύρεση του τμήματος με βάση το `id` και περνιέται ως παράμετρος το πανεπιστήμιο που ανήκει το τμήμα αυτό για τον υπολογισμό του μέσου όρου του πανεπιστημίου.

Με τη μέθοδο `.reduce` ήρθα σε πρώτη επαφή μέσω της συγκεκριμένης δημοσίευσης [46]. Επίσης, ο κώδικας που ενσωματώθηκε στο μπροστινό μέρος για την εμφάνιση της βαθμολογίας ήταν πολύ απλοϊκός. Στην ουσία, αν η μέση βαθμολογία είναι μηδέν (δηλαδή δεν υπάρχουν κριτικές), τότε δείχνει στον χρήστη την ένδειξη `N/A`. Αλλιώς (αν υπάρχουν κριτικές), δείχνει το `departmentAverageRating` (τη μέση βαθμολογία του τμήματος).

4.6.2 Μέσος όρος πανεπιστημίου

Όπως αναφέρθηκε παραπάνω, έπειτα από την ενημέρωση του υπολογισμού του μέσου όρου τμήματος, ακολουθεί η κλήση της συνάρτησης `calculateAverageUniversityRating` (υπολόγισε τη μέση βαθμολογία του πανεπιστημίου). Η συνάρτηση αυτή δέχεται ως παράμετρο το `id` του πανεπιστημίου και η διαδικασία που ακολουθεί μοιάζει σε μεγάλο βαθμό με αυτή του μέσου όρου τμήματος. Μία σημαντική σημείωση, είναι πως ο μέσος όρος του πανεπιστημίου προκύπτει από το άθροισμα των μέσων όρων των τμημάτων που έχουν βαθμολογηθεί προς το πλήθος των τμημάτων αυτών. Ο τρόπος αυτός ήταν πολύ πιο εύκολος από το να αθροίζονται όλες οι κριτικές από όλα τα τμήματα και έπειτα να διαιρούνται με το

άθροισμά τους. Επίσης, θεωρήθηκε σημαντικό το κάθε τμήμα να διαθέτει την ίδια δύναμη, για αυτόν τον λόγο υλοποιήθηκε η συνάρτηση με αυτόν τον τρόπο. Για το μπροστινό μέρος, η διαδικασία εμφάνισης του μέσου όρου των πανεπιστημίων ήταν η ίδια με αυτή των τμημάτων.



Εικόνα 5: Εμφάνιση μέσου όρου πανεπιστημίου

4.7 Εισαγωγή πανεπιστημίων & τμημάτων

Έχουν γίνει κάποιες αναφορές στο κομμάτι αυτό, αλλά θεωρήθηκε σωστό να αναπτυχθεί λίγο παραπάνω η μεθοδολογία της εισαγωγής των δεδομένων. Αρχικά, όπως έχει ήδη επισημανθεί, τα πανεπιστήμια στο άθροισμά τους είναι δεκατέσσερα (14) και τα τμήματα είναι εξήντα ένα (61).

Για την επιλογή των πανεπιστημίων, έγινε προσπάθεια να ενσωματωθούν τα πιο γνωστά πανεπιστήμια της Ελλάδας και τα χαρακτηριστικά του καθενός να είναι όσο πιο ακριβής γίνεται. Φυσικά αξίζει να σημειωθεί πως το πρώτο πανεπιστήμιο που εισάχθηκε ήταν το Ιόνιο, αφού αποτελεί και το πανεπιστήμιο που φοιτώ εγώ ο ίδιος. Επίσης, ενσωματώθηκαν όλα τα τμήματα του πανεπιστημίου αυτού. Έπειτα μία σημαντική αναφορά που πρέπει να ειπωθεί είναι η βασική πηγή άντλησης των δεδομένων. Σε γενικές γραμμές έγινε έρευνα από εμένα τον ίδιο ατομικά για το κάθε πανεπιστήμιο και τμήμα, αλλά βοήθησε πάρα πολύ και η συγκεκριμένη ιστοσελίδα (<https://studyinggreece.edu.gr/universities/>). Η ιστοσελίδα αυτή περιλαμβάνει όλα τα πανεπιστήμια της Ελλάδας και αναγράφει σημαντικές πληροφορίες όπως το όνομα του πανεπιστημίου, το έτος ίδρυσης και τη τοποθεσία, που χρησιμοποιήθηκαν και για τη δική μας ιστοσελίδα. Τέλος, όσον αφορά το χαρακτηριστικό worldRanking (θέση στον κόσμο), οι κατατάξεις λήφθηκαν από δύο ιστοσελίδες. Η πρώτη είναι η (<https://www.topuniversities.com/>), ενώ η δεύτερη είναι η (<https://www.timeshighereducation.com/>).

Για την επιλογή των τμημάτων, ήταν επιθυμητό να υπάρχει ένα ευρύ φάσμα πεδίων σπουδών για μια πιο ολοκληρωμένη ιστοσελίδα. Στην ιστοσελίδα αυτή υπάρχουν τμήματα που ανήκουν στις θετικές επιστήμες, στις θεωρητικές, στις επιστήμες υγείας και σε άλλες πολλές. Το κάθε πανεπιστήμιο περιλαμβάνει τουλάχιστον τρία (3) τμήματα. Για την καταγραφή των χαρακτηριστικών βοήθησε για ακόμη μία φορά η ιστοσελίδα “study in Greece”. Με την επιλογή Learn More (μάθε περισσότερα), παρουσιάζονται όλα τα πεδία σπουδών (fieldOfStudy) του συγκεκριμένου πανεπιστημίου τα οποία αποτελούν και χαρακτηριστικό για την ιστοσελίδα που δημιουργήσαμε. Έπειτα, όταν ένας χρήστης πατάει στο πεδίο σπουδών, εμφανίζονται τα τμήματα που ανήκουν σε εκείνη τη κατηγορία. Συχνά στο τμήμα αυτό, περιλαμβάνεται και το έτος ίδρυσής του, το οποίο είναι πολύ χρήσιμο στην περίπτωσή μας. Τέλος, ενσωματώνεται και ο σύνδεσμος για το τμήμα, όπου γρήγορα και εύκολα είναι δυνατή η πρόσβαση σε αυτό. Κάτι ακόμη που οφείλει να σημειωθεί, είναι πως για την εύρεση των ελάχιστων ακαδημαϊκών χρόνων φοίτησης, χρειάστηκε η προβολή οδηγών σπουδών και προγραμμάτων μαθημάτων του τμήματος. Αυτό ήταν αρκετά χρονοβόρο, λόγω του μεγάλου όγκου τμημάτων που έπρεπε να καλυφθεί.

5 ΥΛΟΠΟΙΗΣΗ ΜΠΡΟΣΤΙΝΟΥ ΜΕΡΟΥΣ (FRONTEND)

5.1 Υλοποίηση frontend

Για την ολοκληρωμένη υλοποίηση της πτυχιακής ήταν απαραίτητη η προσθήκη γραφικής διεπαφής, όπου οι χρήστες θα μπορούσαν να αλληλεπιδρούν δίχως δυσκολίες. Για την επίτευξη του μπροστινού μέρους της ιστοσελίδας χρησιμοποιήθηκε ο σκελετός (framework) και βιβλιοθήκη React. Αξίζει να αναφερθεί πως υπήρξε η πρώτη και πολύ μεγάλη δυσκολία στην επιτυχής χρήση του πακέτου react-router-dom. Για αυτό το θέμα θα γίνει μία εκτενής επεξήγηση στο κεφάλαιο αυτό.

Μία λειτουργία που ήταν πάρα πολύ σημαντική ήταν τα useEffect. Το useEffect αποτελεί hook (άγκιστρο) και ήταν πολύ βοηθητικό στη δημιουργία των fetch (τράβηγμα) δεδομένων από την βάση του backend (πίσω μέρος της ιστοσελίδας). Τα useEffect είναι μία σύνθετη λειτουργία της react, όπου ο συγκεκριμένος κώδικας τρέχει αφού γίνουν οι απαραίτητες αναλύσεις και ενημερώσεις [47]. Έπειτα θα ήταν σωστό να γίνει αναφορά και στα useState. Τα useState ανήκουν και αυτά στη κατηγορία των hooks (άγκιστρων) και επιτρέπουν την εισχώρηση μίας κατάστασης για τη μεταβλητή που απευθυνόμαστε [48]. Για παράδειγμα στην δικιά μας ιστοσελίδα πολλές μεταβλητές θέλαμε να αρχειοθετηθούν με κενές τιμές ή με άδειους πίνακες. Το useState επέτρεψε την διαδικασία αυτή με εύκολο και ξεκάθαρο τρόπο.

5.1.1 Χρήση react-router-dom

Στην ιστοσελίδα που δημιουργήθηκε ήταν απαραίτητη η ενσωμάτωση σελίδων για την εύκολη πλοήγηση των χρηστών. Ο ευκολότερος και πιο διαδομένος τρόπος ήταν μέσω της εγκατάστασης του πακέτου react-router-dom. Το πακέτο αυτό προσφέρει ποικίλες δυνατότητες όπως τα Routes (διαδρομές), Route (διαδρομή), useParams, Link (σύνδεσμος), useLocation και useNavigate που χρησιμοποιήθηκαν για τη δημιουργία της ιστοσελίδας. Τα Routes (διαδρομές) γράφονται στον κώδικα για την ενσωμάτωση των Route. Με λίγα λόγια τα Route που αναπαριστούν τις διαδρομές που επιθυμούμε να υπάρχουν πρέπει να ενσωματώνονται μέσα στα Routes. Έπειτα πολύ σημαντικό ήταν το Link (σύνδεσμος) που επέτρεπε τη μεταπήδηση σε μία νέα σελίδα μέσω ενός κλικ του ποντικιού.

```
<Link to={` /getDepartmentsByUniversity/${university._id}`} className="university-link">
  {university.universityName}
</Link>
```

Εικόνα 6: Κώδικας για Link

Το `useParams` χρησιμοποιείται στο τράβηγμα παραμέτρων που απασχολούν τις διαδρομές [49]. Το `useLocation` χρησιμοποιήθηκε για την εμφάνιση ή μη κάποιων λειτουργιών με βάση τη σελίδα που βρισκόμαστε. Είναι σημαντικό να ειπωθεί πως τα `useParams` και `useLocation` σε γενικές γραμμές δεν χρησιμοποιήθηκαν πολύ στη διαδικασία κατασκευής της ιστοσελίδας.

```
{  
  // if you are not inside the departments page, show the filters  
  !location.pathname.startsWith("/getDepartmentsByUniversity") &&  
  <LocationFilter onSelectLocation={handleLocationFilter} />  
}
```

Εικόνα 7: Κώδικας για `useLocation`

Τέλος, το `useNavigate` χρησιμοποιήθηκε αρκετά στην ιστοσελίδα και βοήθησε πολύ στην αυτοματοποιημένη ανακατεύθυνση του χρήστη. Ένα εύστοχο παράδειγμα για την κατανόηση της λειτουργίας του εργαλείου αυτού είναι, πως όταν ο χρήστης κάνει μία επιτυχής εγγραφή, τότε πηγαίνει αυτόματα στην σελίδα του login (της σύνδεσης). Έπειτα από την περιγραφή όλων αυτών των λειτουργιών του πακέτου `react-router-dom`, είναι κατανοητή η μεγάλη βοήθεια που πρόσφερε για την δημιουργία μίας ολοκληρωμένης ιστοσελίδας.

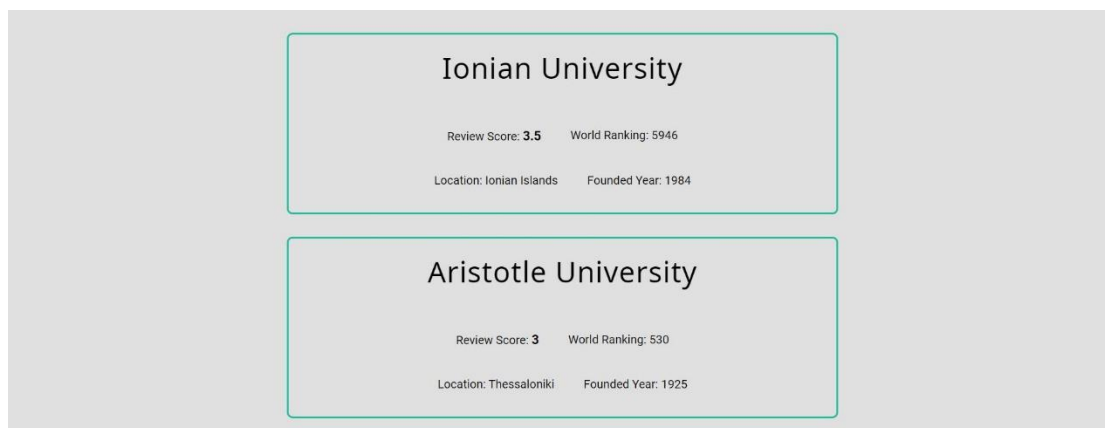
5.1.2 Πρόβλημα 1: πακέτο `react-router-dom`

Αρχικά, η διαδικασία ήταν αρκετά απλοϊκή, αφού εφαρμόστηκε η εντολή `create-react-app` για τη δημιουργία της `react` εφαρμογής. Έπειτα από την υλοποίηση των βασικών εντολών και τον πειραματισμό με την `react`, ήταν η ώρα να δοκιμαστεί η δημιουργία διαδρομών (routes). Για να έρθει εις πέρας η λειτουργία των διαδρομών έπρεπε να εγκατασταθεί το πακέτο `react-router-dom`. Όταν λοιπόν εγκαταστάθηκε το πακέτο αυτό επιτυχώς και έγινε προσπάθεια χρήσης του, το terminal γέμισε με errors (λάθη).

Σε αυτό το σημείο έγινε μία έρευνα σχετικά με το τι μπορεί να φταίει και παρατηρήθηκε ότι το `react-router-dom` δέχτηκε πρόσφατα ενημέρωση και το πακέτο ήταν πολύ καινούργιο. Πιθανά ο λόγος που δε λειτουργούσε σωστά να ήταν αυτός. Στο τέλος της προσπάθειας επίλυσης του προβλήματος εντοπίστηκε το εξής βίντεο (<https://www.youtube.com/watch?v=59IXY5IDrBA>) το οποίο επισήμανε την εγκατάσταση ενός zip αρχείου από ένα GitHub repository (GitHub αποθετήριο) [50]. Όταν έγινε η επιτυχής εγκατάσταση του αρχείου και εκτελέστηκε η εντολή `npm install && npm start`, το πρόβλημα επιλύθηκε. Το zip αυτό περιλάμβανε μία παλαιότερη έκδοση του `react-router-dom` και αυτός πρέπει να ήταν ο λόγος που λειτούργησαν όλα ομαλά.

5.2 Εμφάνιση πανεπιστημίων

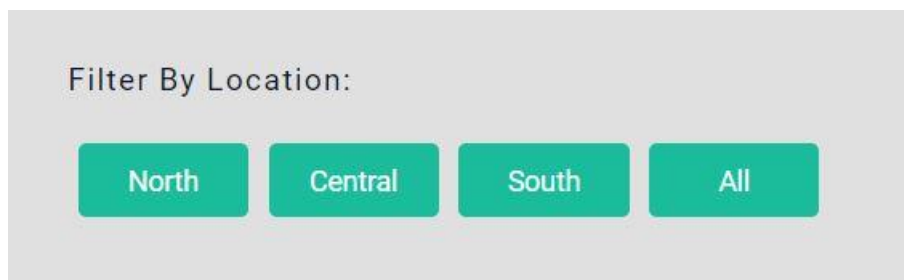
Έπειτα από το 3.1 και το 3.2 ήρθε η ώρα για την ανάδειξη μεθοδολογιών που εφαρμόστηκαν για την αισθητική και τη λειτουργικότητα της γραφικής διεπαφής. Ο σχεδιασμός της ιστοσελίδας δεν έγινε στη τύχη, αφού βασική πηγή έμπνευσης προέκυψε από την έρευνα που συγγράφηκε παραπάνω. Τα πανεπιστήμια καλούνται με τη μέθοδο `fetch` (τράβηγμα) η οποία χρησιμοποιεί τις διαδρομές που δημιουργήθηκαν στο αρχείο `routes.js` (κεφάλαιο **3.1.4**). Η διαδρομή που χρησιμοποιείται είναι η `getAllUniversities` (τράβα όλα τα πανεπιστήμια) και στο συγκεκριμένο `useEffect` ενσωματώνεται και η τοποθεσία σε περίπτωση που έχει επιλεγθεί από τον χρήστη. Στην περιοχή αυτή αποθηκεύονται όλα τα πανεπιστήμια που πληρούν τα κριτήρια σε έναν πίνακα. Έπειτα, θα ήταν σωστή η αναφορά και στη `return` (επιστροφή). Σε εκείνο το μέρος του κώδικα με τη χρήση της μεθόδου `map()` (χάρτης) επιστρέφονται και παρουσιάζονται όλα τα πανεπιστήμια με τα ανάλογα πεδία (χαρακτηριστικά) που επιθυμούμε. Στη `return` επίσης, ο κώδικας γράφεται σε JSX (JavaScript XML), αλλά η αλήθεια είναι πως μοιάζει πάρα πολύ με τη γνωστή HTML [51]. Τα χαρακτηριστικά που επιλέχθηκαν να παρουσιάζονται στο πανεπιστήμιο ήταν τα Review Score (βαθμολογία κριτικής), World Ranking (παγκόσμια κατάταξη), Location (τοποθεσία) και Founded Year (έτος ίδρυσης).



Εικόνα 8: Εμφάνιση πανεπιστημίων

5.3 Φίλτρο τοποθεσίας

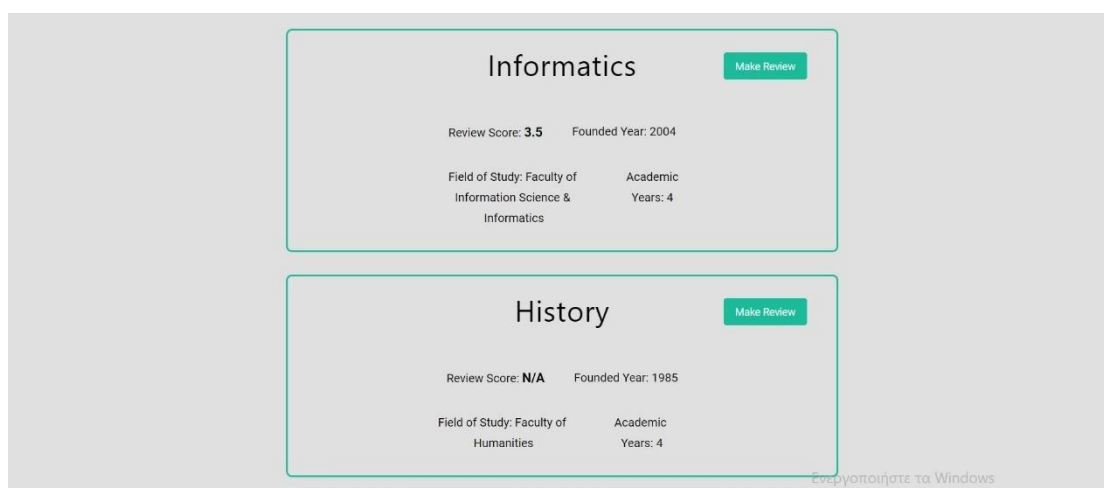
Στις σύγχρονες ιστοσελίδες προσφέρεται εξατομικευμένη επιλογή με βάση τις προτιμήσεις του χρήστη. Για την δημιουργία μίας φιλικής προς τον χρήστη ιστοσελίδας και την επένδυση σε μία ευκολότερη πλοήγηση, επιλέχθηκε να ενσωματωθεί το φίλτρο τοποθεσίας. Η υλοποίηση του φίλτρου ήταν αρκετά απλοϊκή, με πρώτο βήμα να αποτελεί η ενσωμάτωση του χαρακτηριστικού `sortingLocation` (ταξινόμηση τοποθεσίας) στο μοντέλο του πανεπιστημίου. Έπειτα στη διαδρομή `getAllUniversities` (τράβα όλα τα πανεπιστήμια) στο αρχείο `routes.js`, έπρεπε να τεθεί η υπόθεση ότι αν επιλεγεί κάποιο φίλτρο, κατέγραψε τα πανεπιστήμια μόνο αυτού του φίλτρου. Τέλος, έπρεπε να εφαρμοστεί παρόμοια λογική και στο βασικό αρχείο της `React App.js`. Στο `App.js` έγινε ένα `fetch` (τράβηγμα) της διαδρομής `getAllUniversities` όπου λήφθηκε υπόψιν η περίπτωση επιλογής φίλτρου από τον χρήστη. Για την διαδικασία επιλογής φίλτρου δημιουργήθηκε ένα αρχείο με όνομα `locationFilter.js` (τοποθεσία φίλτρου) το οποίο περιέχει όλες τις πιθανές τοποθεσίες (`North`, `Central`, `South` και `All`). Αν ο χρήστης επιλέξει κάποια από αυτές, η επιλογή του αποθηκεύεται και χρησιμοποιείται για να τραβηχθούν τα ανάλογα πανεπιστήμια. Επίσης, στο αρχείο `App.js` γίνονται και έλεγχοι όπου εφόσον έχει επιλεγεί τοποθεσία να φανούν μόνο τα πανεπιστήμια με αυτή τη τοποθεσία, ενώ αν δεν έχει επιλεγεί τοποθεσία να φανούν όλα τα πανεπιστήμια.



Εικόνα 9: Εμφάνιση φίλτρου τοποθεσίας

5.4 Εμφάνιση τμημάτων

Για την εμφάνιση των τμημάτων δε θα αναφερθούν πάρα πολλά πράγματα, διότι η διαδικασία μοιάζει σε μεγάλο βαθμό με τη παρουσίαση των πανεπιστημίων. Παρ' όλα αυτά κάποιες διαδικασίες αξίζει να αναφερθούν. Αρχικά, για πρώτη φορά στην ιστοσελίδα μας χρησιμοποιήθηκε το Link (σύνδεσμος) από το πακέτο react-router-dom. Όταν ο χρήστης πατάει στο όνομα του πανεπιστημίου, χρησιμοποιείται το ανάλογο Route (διαδρομή), σε αυτήν την περίπτωση η διαδρομή με path (μονοπάτι) το getDepartmentsByUniversity. Έπειτα ο κώδικας μεταβαίνει στη συνάρτηση UniversityDepartments (τμήματα πανεπιστημίου), όπου γίνεται και το κατάλληλο fetch. Όταν έρθει η ώρα για τη παρουσίαση των πανεπιστημίων ο κώδικας μεταβαίνει στο αρχείο DepartmentDetails.js για την εμφάνιση των ονομάτων και των υπόλοιπων χαρακτηριστικών των τμημάτων. Τέλος, θα ήταν σωστό να αναφερθεί πως το κάθε πανεπιστήμιο περιλαμβάνει τουλάχιστον τρία (3) τμήματα.



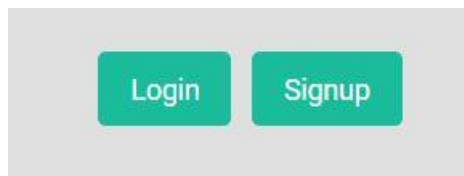
Εικόνα 10: Εμφάνιση τμημάτων

5.5 Μπροστινό μέρος συστήματος εισόδου/εγγραφής

Στο κεφάλαιο 4 που γίνονται αναφορές στην υλοποίηση του πίσω μέρους της εφαρμογής, περιλαμβάνεται και το σύστημα εισόδου/εγγραφής (login/signup). Σε αυτό το κεφάλαιο θα αναφερθούμε στο μπροστινό μέρος του συστήματος που ήταν απαραίτητο για την προβολή και ομαλή λειτουργικότητα των διαδρομών μας. Τα αρχεία που δημιουργήθηκαν στο σύνολό τους ήταν τρία (3). Αρχικά, δομήθηκε το loginSignup.js το οποίο περιέχει τα δύο κουμπιά για την πρόσβαση στη φόρμα της σύνδεσης και της εγγραφής. Έπειτα έπρεπε να δημιουργηθούν αυτές οι φόρμες. Τα αρχεία λοιπόν που προέκυψαν από τη διαδικασία αυτή ήταν το loginPage.js (σελίδα σύνδεσης) και το signupPage.js (σελίδα εγγραφής). Η δημιουργία των αρχείων περιλάμβανε δύο βασικές λειτουργίες. Η πρώτη ήταν αυτή του fetch (τραβήγματος) που βοήθησε πολύ το άρθρο [52], ενώ η δεύτερη ήταν η φόρμα συμπλήρωσης που βοήθησε πολύ το άρθρο [53].

5.5.1 Αρχείο loginSignup.js

Για τη δημιουργία των κουμπιών τα πράγματα ήταν αρκετά ξεκάθαρα. Με τη χρήση του Link (συνδέσμου), έθεσα το μονοπάτι της σύνδεσης και της εγγραφής. Όταν ο χρήστης πατάει το κουμπί “Login”, μεταβαίνει στο σύνδεσμο /login. Το αντίστοιχο ισχύει και για την εγγραφή με τον σύνδεσμο /signup. Έχουν φτιαχτεί και τα routes (διαδρομές) στο αρχείο App.js (το κεντρικό αρχείο) για την ομαλή άφιξη στις ανάλογες φόρμες συμπλήρωσης.

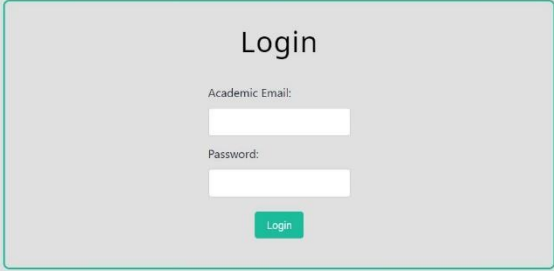


Εικόνα 11: Εμφάνιση κουμπιών login/signup

5.5.2 Αρχείο loginPage.js

Η ανάλυση θα ξεκινήσει πρώτα με το loginPage.js αρχείο μιας και η υλοποίηση του ήταν σαφώς αρκετά ευκολότερη από αυτή του signupPage.js. Οι δύο οδηγοί ακολουθήθηκαν σε μεγάλο βαθμό για την επιτυχής πραγματοποίηση της φόρμας σύνδεσης. Αρχικά, έγινε το fetch (τράβηγμα) της διαδρομής για τη σύνδεση του χρήστη από το αρχείο routes.js. Εκεί στη περιοχή body (σώματος) χρειάστηκε η διατύπωση των πεδίων του ακαδημαϊκού email και του κωδικού πρόσβασης. Έπειτα, αν το αποτέλεσμα του μηνύματος είναι ίδιο με το μήνυμα που θέσαμε στο αρχείο routes.js, ο χρήστης συνδέεται επιτυχώς. Σε εκείνη τη περιοχή κιόλας γίνεται μία αποθήκευση του ακαδημαϊκού email του μαθητή, του πανεπιστημίου που ανήκει και του τμήματός του. Αυτό το κομμάτι είναι πολύ σημαντικό για την δημιουργία κριτικής που θα ακολουθήσει σε μετέπειτα κεφάλαιο.

Έπειτα ακολούθησε η φόρμα συμπλήρωσης, όπου η δομή της είναι εμπνευσμένη από το άρθρο που αναφέρθηκε παραπάνω. Με τη χρήση του label (ετικέτας) καταγράφεται το όνομα του πεδίου συμπλήρωσης και έπειτα ακολουθούν ο τύπος, το όνομα του στοιχείου, η τιμή, το onChange που αποθηκεύει τις αλλαγές στο πεδίο όταν ο χρήστης πληκτρολογεί και το required δηλώνει πως είναι απαραίτητη η συμπλήρωση.



Εικόνα 12: Εμφάνιση φόρμας σύνδεσης

5.5.3 Αρχείο signupPage.js

Για το αρχείο signupPage.js η διαδικασία ήταν η ίδια, απλώς συνέβησαν κάποιες σημαντικές προσθήκες. Όταν ο μαθητής κάνει την εγγραφή του θα πρέπει να συμπληρώσει το πανεπιστήμιο του και το τμήμα του. Αν το πεδίο ήταν σε μορφή text (κειμένου) τότε ο μαθητής θα μπορούσε να έγραφε διαφορετικά το τμήμα του ή το πανεπιστήμιο του και να μην γινόταν σωστή αντιστοίχιση. Για αυτόν τον λόγο συγγράφηκε κώδικας για την επιλογή του πανεπιστημίου και του τμήματος σε μορφή dropdown menu (αναδυόμενου μενού).

Για το fetch (τράβηγμα) των πανεπιστημίων και των τμημάτων η διαδικασία ήταν λίγο ιδιαίτερη γιατί υπάρχει σελιδοποίηση. Για το κομμάτι των πανεπιστημίων, στη συνάρτηση που συμβαίνει το fetch περνάμε τη παράμετρο currentPage (παρούσα σελίδα). Έπειτα, γίνεται έλεγχος αν έχουν τραβηχτεί πανεπιστήμια και αν ναι περνιούνται στο πίνακα allUniversities (όλα τα πανεπιστήμια) με τη μέθοδο push (σπρώξιμο). Ακολουθεί έλεγχος αν τα πανεπιστήμια που τραβήχτηκαν ήταν λιγότερα από τρία, που αν αληθεύει σηματοδοτείται η λήξη της επανάληψης. Αυτός ο έλεγχος υπάρχει, διότι ο μέγιστος αριθμός πανεπιστημίων ανά σελίδα είναι τρία. Τέλος, αν υπάρχουν και άλλα πανεπιστήμια που δεν έχουμε τραβήξει, μεταβαίνουμε στην επόμενη σελίδα (currentPage +1).

Η ίδια διαδικασία ισχύει και για τα τμήματα, με μόνη διαφορά να αποτελεί πως περνιέται μία ακόμη παράμετρος, το universityId (το id του πανεπιστημίου). Αυτή η παράμετρος χρησιμοποιείται για το τράβηγμα μόνο των τμημάτων του συγκεκριμένου πανεπιστημίου που επέλεξε ο χρήστης. Κάτι ακόμη που αξίζει να αναφερθεί, είναι πως αν ο χρήστης αλλάξει την επιλογή του πανεπιστημίου από το μενού, τότε ο πίνακας των τμημάτων αδειάζει και γίνεται επιστροφή σε ένα νέο τράβηγμα δεδομένων με βάση το νέο πανεπιστήμιο ως παράμετρο.

Sign Up

Academic Email:

Password:

Full Name:

Nationality:

University:
Ionian University ▼

Εικόνα 13: Εμφάνιση φόρμας εγγραφής (1)

Department:
Select a Department ▼

- Select a Department
- Informatics**
- History
- Department of Foreign Languages, Translation and Interpreting
- Archives, Library Science and Museum Studies
- Digital Media and Communications
- Audio and Visual Arts
- Department of Music Studies
- Department of Ethnomusicology
- Department of Environment
- Food Science and Technology
- Regional Development
- Tourism

Ενεργοποιήστε τα Windows
Μεταβείτε στις ρυθμίσεις για ενεργοποίηση των Windows.

Εικόνα 14: Εμφάνιση φόρμας εγγραφής (2)

5.5.4 Επιπρόσθετες λειτουργίες

Οι βασικές λειτουργίες έχουν ήδη αναλυθεί, αλλά αξίζει να επισημανθούν και κάποιες επιπρόσθετες που βοήθησαν πολύ σε μια πιο φιλική προς τον χρήστη εμπειρία. Η πρώτη λειτουργία είναι το μήνυμα “Logged In” (συνδεδεμένος), το οποίο ενεργοποιείται όταν ο χρήστης συνδέεται επιτυχώς στην ιστοσελίδα μας. Η δεύτερη λειτουργία ενεργοποιείται με τον ίδιο τρόπο και εμφανίζει το κουμπί logout (αποσύνδεση). Όταν ο χρήστης πατάει το κουμπί αυτό, αποσυνδέεται από την ιστοσελίδα. Ο τρόπος που επιτυγχάνονται οι λειτουργίες αυτές είναι μέσω ενός απλού κώδικα που συγγράφηκε. Όταν ο χρήστης συνδέεται στο σύστημα, η μεταβλητή `setIsLoggedIn` γίνεται `true` (αλήθεια). Αν λοιπόν το `setIsLoggedIn` είναι `true`, τότε εμφανίσε το μήνυμα Logged In (συνδεδεμένος) και το κουμπί Logout (αποσύνδεση).

Login Signup

Logged In Logout

Εικόνα 15: Εμφάνιση συνδεδεμένος και κουμπί αποσύνδεσης

Τέλος, θα ήταν σωστό να γίνει και μία σύντομη αναφορά στο timer (χρονόμετρο) που κατασκευάστηκε. Για την ασφάλεια του χρήστη και των δεδομένων του, δημιουργήθηκε ένα χρονόμετρο όπου έπειτα από δεκαπέντε (15) λεπτά σύνδεσης ο χρήστης αποσυνδέεται αυτόματα. Δημιουργήθηκε ένα useEffect, στο οποίο αν ο χρήστης είναι συνδεδεμένος και τα δεκαπέντε λεπτά ολοκληρωθούν, τότε αποσυνδέεται από τη πλατφόρμα. Αυτό εμφανίζεται και μέσω της μορφής μηνύματος προς τον χρήστη.

5.6 Star System (σύστημα αστεριών)

Στα επόμενα κεφάλαια θα αναλυθεί το κυρίως μέρος της εργασίας, το οποίο είναι το σύστημα αξιολόγησης που δομήθηκε. Όλη η ιδεολογία της ιστοσελίδας είναι γύρω από την αξιολόγηση τμημάτων και πανεπιστημίων. Η ιδεολογία αυτή λοιπόν, δε θα μπορούσε να είναι ολοκληρωμένη χωρίς την ενσωμάτωση ενός συστήματος κριτικής μέσω της επιλογής αστεριών. Για τη βέλτιστη αισθητική της ιστοσελίδας και την κατανόηση της διαδικασίας από τους χρήστες, το σύστημα κριτικής με την επιλογή αστεριών απασχολεί μόνο τη συνολική βαθμολογία του τμήματος. Για την υλοποίηση του συστήματος αυτού, ακολουθήθηκαν οι οδηγίες του άρθρου [54].

Όσον αφορά τον κώδικα που συγγράφηκε, η συνάρτηση StarRating (βαθμολόγηση με αστέρια) δέχεται τρεις (3) παραμέτρους. Η πρώτη είναι το rating (βαθμολόγηση) η οποία δέχεται το χαρακτηριστικό της φόρμας της κριτικής που επιθυμούμε. Στη συγκεκριμένη περίπτωση το στοιχείο overall (συνολικά). Έπειτα είναι το setRating (θέσε βαθμολογία) που δέχεται τη τιμή setOverall (θέσε συνολικά). Το setRating χρησιμοποιείται στη μέθοδο onClick (όταν γίνεται κλικ) και αλλάζει τη βαθμολογία του χρήστη με βάση το αστέρι που επιλέγει. Τρίτη και τελευταία παράμετρος είναι το interactive (διαδραστικό), το οποίο δέχεται τη τιμή true (αλήθεια) από προεπιλογή. Παρ' όλα αυτά σε περίπτωση της εμφάνισης των κριτικών το interactive (διαδραστικό) γίνεται false (ψέμα), ώστε ο χρήστης μόνο να παρατηρεί τη βαθμολόγηση που έχει τεθεί.

Submit Your Review

Overall: ★ ★ ★ ★ ★

Student Life:

Facilities:

Career Prospects:

Accommodation & Food Services:

Academic Email:

Professors:

University ID:

University Clubs:

Department ID:

Εικόνα 16: Εμφάνιση συστήματος αστεριών

5.7 Review System (σύστημα αξιολόγησης)

Έπειτα από την αναφορά όλων των προηγούμενων κεφαλαίων, ήρθε η ώρα να αναλυθεί το σύστημα αξιολόγησης. Το σύστημα αξιολόγησης είναι αρκετά σύνθετο, διότι περιλαμβάνει ποικίλες λειτουργίες. Στο μέρος αυτό θα μιλήσουμε για το αρχείο `createReview.js` που αποτελεί και τη φόρμα συμπλήρωσης της κριτικής. Το σύστημα αξιολόγησης στην ευρύτερη έννοιά του αποτελεί το τελευταίο μεγάλο κεφάλαιο της εργασίας.

Ένα ενδιαφέρον γεγονός για τη δημιουργία του αρχείου `createReview.js` (δημιουργία κριτικής), είναι πως δεν υπήρξε κάποια βασική πηγή έμπνευσης. Ο λόγος είναι, επειδή η διαδικασία έμοιαζε πολύ με το `loginPage.js` (υποκεφάλαιο 3.7.2.2). Το αρχείο που κατασκευάστηκε περιλάμβανε το `fetch` (τράβηγμα) του `postReview` (ανέβασμα κριτικής) με τα πεδία `method` (μέθοδος), `headers` (κεφαλίδες) και `body` (σώμα), όπως στη σελίδα σύνδεσης.

Έπειτα, στο `route` (διαδρομή) `/postReview` του αρχείου `routes.js` έχουν ενσωματωθεί δύο μηνύματα που εμφανίζονται στους χρήστες. Το πρώτο εμφανίζεται σε περίπτωση που ο χρήστης έχει κάνει ήδη κριτική και έχει `status` (κατάσταση) 409 που υποδηλώνει σύγκρουση. Το δεύτερο είναι ένα μήνυμα επιτυχίας και εξηγεί στον χρήστη πως η κριτική του προστέθηκε στο σύστημα. Τα δύο αυτά μηνύματα χρησιμοποιούνται στο αρχείο `createReview.js`.

Ακολουθεί η φόρμα συμπλήρωσης που έχει την ίδια ιδεολογία με τη φόρμα της σύνδεσης. Η διαφορά είναι πως χρησιμοποιείται η συνάρτηση `StarRating` για την εμφάνιση του πεδίου συνολικής βαθμολόγησης (`overall`) με τη μορφή αστεριών. Στα υπόλοιπα πεδία έχει προστεθεί κώδικας για την εφαρμογή ως χαμηλότερου ορίου τη βαθμολογία ένα (1) και υψηλότερου ορίου τη βαθμολογία πέντε (5).

Τέλος, θα ήθελα να αναφερθώ στα πεδία ακαδημαϊκού email, id πανεπιστημίου και id τμήματος. Επιστημάνθηκε παραπάνω πως όταν ένας χρήστης συνδέεται στη πλατφόρμα, αποθηκεύονται αυτά τα τρία χαρακτηριστικά. Στο κομμάτι της

δημιουργίας κριτικής τραβώ τα δεδομένα αυτά και συμπληρώνονται αυτόματα στα κατάλληλα πεδία. Αυτό υλοποιήθηκε με την εντολή `localStorage.getItem`. Σε περίπτωση που ο χρήστης δεν είναι συνδεδεμένος, στα πεδία αυτά δίνονται κενές τιμές. Για να μην μπορεί ο χρήστης να αλλάξει τα δεδομένα που συμπληρώθηκαν αυτόματα, στο κομμάτι `input` (είσοδο) έχουν την ιδιότητα `readOnly` (μόνο διάβασμα).

Submit Your Review

Overall: ★ ★ ★ ★ ★	Student Life:
Facilities: 	Career Prospects:
Accommodation & Food Services: 	Academic Email: p19pavi@ionio.gr
Professors: 	University ID: 64a006736d443b16b905ba61
University Clubs: 	Department ID: 64a0097f340a663b267a85d7

Ενεργοποιήστε τα Windows
Μεταβείτε στις ρυθμίσεις για ενεργοποίηση των Windows.

Εικόνα 17: Εμφάνιση φόρμας αξιολόγησης

5.8 Εμφάνιση κριτικών τμημάτων

Η εμφάνιση των κριτικών αποτέλεσε μία ενδιαφέρουσα διαδικασία που αξίζει να αναφερθεί στη συγγραφή της πτυχιακής. Για την επιτυχής υλοποίηση της λειτουργικότητας αυτής έπρεπε αρχικά να δομηθεί η κατάλληλη διαδρομή στο αρχείο `routes.js` και έπειτα ένα αρχείο για το frontend (μπροστινό μέρος) της ιστοσελίδας.

5.8.1 Για αρχείο `routes.js`

Στο αρχείο `routes.js` προστέθηκε μία νέα διαδρομή με όνομα `getReviewsByDepartment` (τράβηξε τις κριτικές με βάση το τμήμα), όπου γίνεται αυτό που υποδηλώνει το όνομα της διαδρομής. Στην ουσία βρίσκουμε τις κριτικές με βάση το συγκεκριμένο `departmentId` (`id` του τμήματος) και τις αποθηκεύουμε. Σε περίπτωση που ένα τμήμα δε διαθέτει κριτικές, δημιουργήθηκε μήνυμα που ενημερώνει τον χρήστη.

5.8.2 Αρχείο `DepartmentReviews.js`

Το αρχείο αυτό διαχειρίζεται την εμφάνιση των πανεπιστημίων στο οπτικό κομμάτι της σελίδας. Η βασική λειτουργία που ήταν επιθυμητό να προστεθεί για το κομμάτι αυτό ήταν, όταν ένας χρήστης είναι συνδεδεμένος να μπορεί να τροποποιήσει τη κριτική του εφόσον το επιθυμεί. Για αυτόν τον λόγο, για να μην αναζητάει τη κριτική του μεταξύ όλων των υπολοίπων, συγγράφηκε κώδικας που εμφανίζει τη κριτική του πρώτη στην οθόνη. Η λογική του κώδικα ήταν πολύ απλή. Με τη χρήση του `getItem` τραβιέται το ακαδημαϊκό email του μαθητή. Έπειτα, γίνεται έλεγχος αν η κριτική που αναλύεται εκείνη τη στιγμή έχει το ίδιο ακαδημαϊκό email με αυτό του μαθητή. Αν αυτό αληθεύει, τότε η κριτική αυτή είναι του μαθητή και αποθηκεύεται στη μεταβλητή `studentReview` (κριτική μαθητή). Έπειτα σε μία άλλη μεταβλητή με όνομα `otherReviews` (άλλες κριτικές), αποθηκεύονται όλες οι υπόλοιπες κριτικές, χωρίς αυτή του μαθητή.

Όταν έρχεται η ώρα παρουσίασης των κριτικών του τμήματος, η κριτική του μαθητή είναι πρώτη στον κώδικα και έπειτα ακολουθούν όλες οι υπόλοιπες κριτικές. Ο κώδικας μεταξύ των δύο αυτών μερών μοιάζει, απλώς στο κομμάτι του μαθητή υπάρχουν περισσότερες λειτουργίες. Εφόσον είναι συνδεδεμένος μπορεί να τροποποιήσει ή και να διαγράψει τη κριτική του. Για αυτές τις λειτουργίες θα γίνει εκτενής επεξήγηση αργότερα στην αναφορά.

Τέλος, αξίζει να σημειωθεί περιεκτικά πως όταν γίνεται το `fetch` (τράβηγμα) των αξιολογήσεων του τμήματος, σε περίπτωση που δεν βρεθούν κριτικές εμφανίζεται το ανάλογο μήνυμα στον χρήστη, γίνεται μετάβαση πίσω στα τμήματα και ο πίνακας με τις κριτικές παίρνει κενή τιμή. Επίσης, εφόσον υπάρχουν κριτικές, τα πεδία που τις αποτελούν είναι αυτά που καταγράφηκαν στη φόρμα συμπλήρωσης της κριτικής. Δεν αναφέρονται το τμήμα και το πανεπιστήμιο γιατί είναι ήδη γνωστά και το ακαδημαϊκό email για λόγους ανωνυμότητας.



Εικόνα 18: Εμφάνιση κριτικών τμήματος

5.9 Τροποποίηση και διαγραφή κριτικής

Για μια πιο ολοκληρωμένη ιστοσελίδα ήταν σημαντική η εισαγωγή ενός συστήματος τροποποίησης και διαγραφής της κριτικής. Η διαδικασία των διαδρομών στο αρχείο `routes.js` ήταν αρκετά ξεκάθαρη και δε θα γίνουν αναφορές σε αυτό το κομμάτι της εργασίας, αλλά λίγο παρακάτω. Στο κεφάλαιο 3.11 θα αναλυθεί η δημιουργία των κουμπιών `edit` (τροποποίησης) και `delete` (διαγραφής), η μεθοδολογία της διαγραφής και η σελίδα που κατασκευάστηκε για τη τροποποίηση της κριτικής.

5.9.1 Δημιουργία κουμπιών τροποποίησης και διαγραφής

Στο προηγούμενο κεφάλαιο (3.10) έγινε μία εκτενής επεξήγηση των λειτουργιών της εμφάνισης των αξιολογήσεων. Τα δύο κουμπιά που κατασκευάστηκαν, ενσωματώθηκαν στο αρχείο `DepartmentReviews.js` (κριτικές τμήματος). Για να μπορέσουν όμως να εμφανιστούν στην κριτική του χρήστη πρέπει να πληρούνται κάποιες προϋποθέσεις. Αρχικά, ο χρήστης θα πρέπει να είναι συνδεδεμένος και επίσης, θα πρέπει φυσικά να έχει κάνει κριτική. Για αυτόν τον λόγο τα δύο κουμπιά εντάχθηκαν στο κομμάτι εμφάνισης της κριτικής του `studentReview` (κριτική μαθητή), και έχουν τη συνθήκη `isLoggedIn` (να είναι συνδεδεμένος).



Εικόνα 19: Εμφάνιση κουμπιών edit & delete

5.9.2 Διαγραφή κριτικής

Όσον αφορά τη διαγραφή της κριτικής δε χρειάστηκε να δημιουργηθεί σελίδα για τη λειτουργία αυτή. Στα πλαίσια της διαγραφής φτιάχτηκε μία συνάρτηση, όπου γίνεται και το αντίστοιχο fetch (τράβηγμα) με βάση το id της κριτικής του μαθητή. Αν όλα πάνε καλά και υλοποιηθεί η μέθοδος delete (διαγραφή), τότε εμφανίζεται το ανάλογο μήνυμα στον χρήστη και ακολουθεί η δημιουργία ενός πίνακα που δε περιλαμβάνει αυτή τη κριτική. Έπειτα, χρησιμοποιείται αυτός ο πίνακας για να εμφανιστούν οι ανανεωμένες κριτικές.

5.9.3 Τροποποίηση κριτικής

Για τη τροποποίηση της κριτικής ακολουθήθηκαν διάφορες σημαντικές μεθοδολογίες για την επιτυχής υλοποίησή της. Αρχικά, είναι σημαντικό να αναφερθεί ότι η διαδικασία μοιάζει αρκετά με το createReview.js (δημιουργία κριτικής). Στο αρχείο routes.js η διαδρομή είναι η '/updateReview/:id' και το id χρησιμοποιείται για την τροποποίηση της συγκεκριμένης κριτικής που επιθυμούμε. Με βάση αυτή τη λογική στο DepartmentReviews.js έχει δημιουργηθεί ένα Link (σύνδεσμος), όπου στέλνεται το studentReview._id (το id της κριτικής του μαθητή).

Έπειτα μέσω της λειτουργίας useParams του πακέτου react-router-dom, τραβάμε το id της συγκεκριμένης κριτικής. Δημιουργήθηκε και ένα useEffect με σκοπό να αποκτηθούν όλα τα δεδομένα της κριτικής που θα γίνει η τροποποίηση. Μετά από αυτές τις προσθήκες η διαδικασία του υπόλοιπου κώδικα είναι σχεδόν η ίδια με αυτή του αρχείου createReview.js.

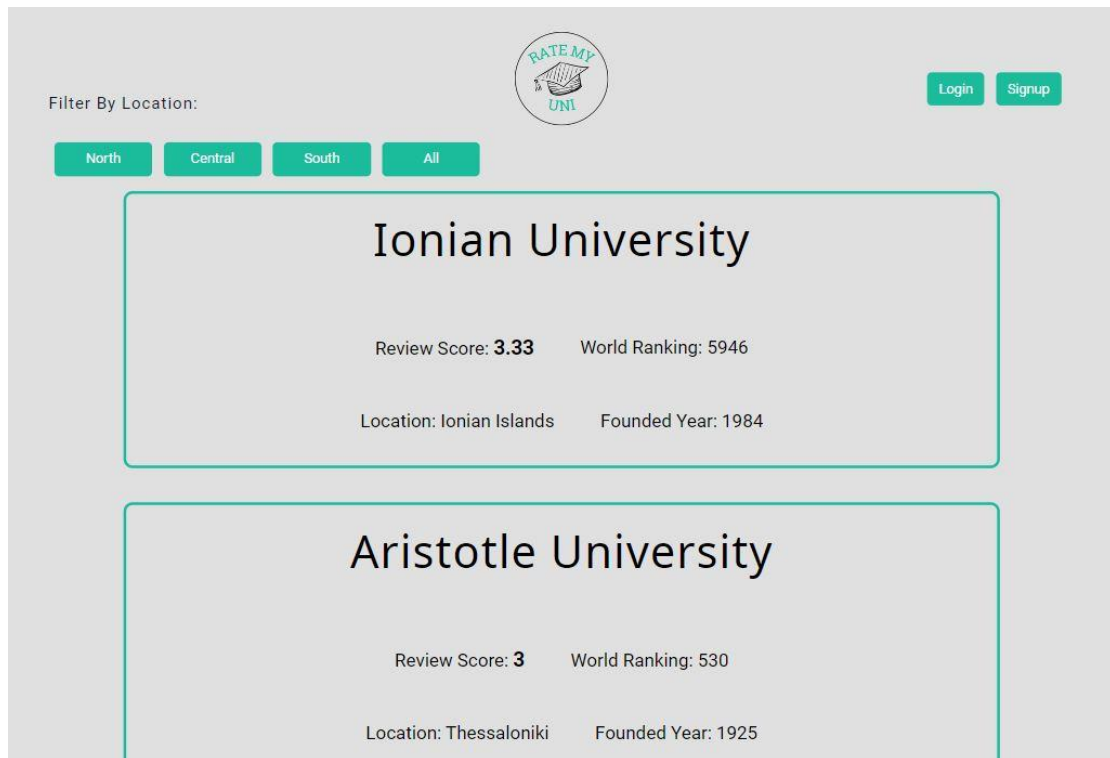
Εικόνα 20: Εμφάνιση φόρμας τροποποίησης κριτικής

5.10 Ανταποκρισιμότητα (responsiveness)

Η ανταποκρισιμότητα (responsiveness) [55] είναι η προσαρμογή των διαστάσεων της ιστοσελίδας για τις σύγχρονες οθόνες. Οι τρεις βασικές οθόνες είναι αυτή του υπολογιστή, του tablet (τάμπλετ) και του smartphone (κινητού τηλεφώνου). Η ιστοσελίδα δομήθηκε αρχικά με βάση την οθόνη του laptop (λάπτοπ) που χρησιμοποιήθηκε στα πλαίσια της εργασίας. Για την προσαρμοστικότητα των άλλων δύο οθονών χρησιμοποιήθηκε η @media της CSS.

5.10.1 Προσαρμογή για tablets

Οι ταμπλέτες κατά γενικό κανόνα έχουν για ελάχιστο πλάτος τα 768 pixels (εικονοστοιχεία) και μέγιστο πλάτος τα 1024 pixels (εικονοστοιχεία). Με τη χρήση της λειτουργίας @media τέθηκαν αυτές οι διαστάσεις στο αρχείο της CSS, όπου ότι συγγράφεται σε εκείνο το μέρος του κώδικα εφαρμόζεται για τις διαστάσεις αυτές. Οποιαδήποτε κλάση (className) ή id που δε συμπεριλαμβάνεται μέσα σε αυτό το μέρος του κώδικα χρησιμοποιείται με τον ίδιο τρόπο όπως στην προεπιλεγμένη οθόνη. Η εμφάνιση και αισθητική της ιστοσελίδας για τις διαστάσεις αυτές μοιάζει σε πολύ μεγάλο βαθμό με αυτές της οθόνης του υπολογιστή.



Εικόνα 21: Εμφάνιση ιστοσελίδας σε οθόνη tablet

5.10.2 Προσαρμογή για smartphones

Έπειτα, ήταν επιθυμητό να υλοποιηθεί η προσαρμογή της ιστοσελίδας για οθόνες κινητών συσκευών (smartphones). Η προσαρμογή αυτή ήταν αρκετά πιο δύσκολη από αυτή του tablet (τάμπλετ), λόγω του μικρού πλάτους της οθόνης. Για τις διαστάσεις τέθηκε ως μέγιστο πλάτος τα 767 pixels (εικονοστοιχεία). Μία πολύ σημαντική σημείωση είναι, πως πολλές λειτουργίες της ιστοσελίδας τροποποιήθηκαν για να ταιριάζουν με τις διαστάσεις της οθόνης του χρήστη. Η φόρμα δημιουργίας και τροποποίησης της κριτικής έχει τη μορφή μίας στήλης αντί για δύο. Επίσης, οι κριτικές του τμήματος και αυτές παρουσιάζουν όλα τα πεδία σε μία μόνο στήλη. Ακόμη, το μέγεθος των κουμπιών και των γραμματοσειρών τροποποιήθηκε για να ταιριάζει με την ανάλυση της οθόνης.



Εικόνα 22: Εμφάνιση κριτικών τμήματος σε οθόνη κινητού

5.11 Φιλοξενία ιστοσελίδας (hosting)

Η ιστοσελίδα, για όλη τη διαδικασία κατασκευής της λειτουργούσε με localhost (τοπική φιλοξενία στο υπολογιστικό μου σύστημα). Όταν ολοκληρώθηκε η ανάπτυξη, το localhost ήταν πολύ περιοριστικό, αφού κανείς πέρα από εμένα δε θα μπορούσε να κάνει κριτική. Για αυτόν τον λόγο, έγινε μία μικρή έρευνα για να βρεθούν τρόποι να φιλοξενηθεί η ιστοσελίδα σε κάποια πλατφόρμα. Πολύ γνωστή επιλογή αποτελούσε το Heroku (<https://www.heroku.com/>), όμως πρόσφατα το δωρεάν πλάνο που είχε, καταργήθηκε. Για αυτόν τον λόγο, φιλοξενήθηκε η ιστοσελίδα στην πλατφόρμα Render (<https://render.com/>), η οποία περιλάμβανε δωρεάν πλάνο.

5.11.1 Ανέβασμα εργασίας στο GitHub

Με το ανέβασμα της εργασίας στο GitHub υπήρξε ένα μεγάλο πρόβλημα που θα αναλυθεί περεταίρω παρακάτω. Αυτό το πρόβλημα ήταν δύσκολο να επιλυθεί από εμένα και αν δεν υπήρχε η κατάλληλη υποστήριξη, πιθανά να χανόταν σχεδόν όλη η εργασία μου.

Για το ανέβασμα του αρχείου βοήθησε πολύ ο συγκεκριμένος οδηγός [56]. Ο οδηγός αυτός ακολουθήθηκε σε μεγάλο βαθμό και βοήθησε πολύ σε μικρά πράγματα που δεν είχα σκεφτεί, όπως τη μεταφορά και τη τροποποίηση του .gitignore αρχείου. Επίσης, είχαν ξεχασθεί κάποιες βασικές εντολές του GitHub και το άρθρο αυτό βοήθησε στο φρεσκάρισμά τους. Την εργασία της πτυχιακής την πέρασα στο branch (κλαδί) thesis (διπλωματική εργασία). Ο σύνδεσμος για το GitHub repository (αποθετήριο) μου είναι: <https://github.com/AimiliosPavlidis2001/RateMyUni>.

5.11.2 Φιλοξενία ιστοσελίδας στο Render

Η πλατφόρμα που επιλέχθηκε τελικά ήταν το Render. Δυστυχώς επειδή η φιλοξενία είναι δωρεάν υπάρχουν πολλοί περιορισμοί και η ιστοσελίδα αργεί να φορτώσει. Επίσης συχνά γίνεται bad gateway error (σφάλμα κακής πύλης), όπου πρέπει να περιμένει ο χρήστης κάποια λεπτά για να έχει πρόσβαση στην ιστοσελίδα. Παρ' όλα αυτά όμως, όλες οι λειτουργίες της ιστοσελίδας δουλεύουν δίχως κανένα πρόβλημα. Το ανέβασμα των αρχείων δεν ήταν εύκολο και για αυτό αξιοποιήθηκε αυτό το βίντεο [57]. Στην πλατφόρμα Render ανέβηκε ξεχωριστά η βάση και ξεχωριστά το μπροστινό μέρος της εργασίας. Όσον αφορά τη βάση δεδομένων, το βίντεο αυτό βοήθησε πολύ στη ρύθμιση των παραμέτρων, αφού ήταν σημαντική η σύνδεση του MongoDB με τη πλατφόρμα αυτή. Έπειτα για το μπροστινό μέρος η διαδικασία ήταν πιο εύκολη και βασίστηκε στην ιδεολογία της βάσης. Τέλος, είναι σημαντικό να προστεθεί πως τροποποιήθηκαν οι σύνδεσμοι στα fetch (τραβήγματα) για την εμφάνιση των δεδομένων. Συνέβη αλλαγή από localhost:3000 σε ratemyuni.onrender.com. Το μονοπάτι ratemyuni.onrender.com είναι αυτό της βάσης που δημιούργησα στη πλατφόρμα του Render. Ο σύνδεσμος για την ιστοσελίδα μου είναι: <https://ratemyuni-frontend.onrender.com/>.

render						
Dashboard Blueprints Env Groups Docs Community Help						
New +  Aimiilis Pavlidis						
Overview						
Search services						
Active 2 Suspended 0 All 2						
NAME	STATUS	TYPE	RUNTIME	REGION	LAST DEPLOYED ↓	
 ratemyuni-frontend	 Deployed	Web Service	Node	Frankfurt	13 days ago	...
 ratemyuni	 Deployed	Web Service	Node	Frankfurt	13 days ago	...
 Feedback  Invite a friend  Contact support						
Ενεργοποιήστε τα Windows Μεταβείτε στις ρυθμίσεις για ενεργοποίηση των Windows.						

Εικόνα 23: Εμφάνιση πίνακα ελέγχου της πλατφόρμας Render

5.11.3 Πρόβλημα 2: λανθάνουσα χρήση GitHub

Το δεύτερο και τελευταίο πρόβλημα, είναι πολύ μεγαλύτερο από το πρώτο. Το GitHub είναι μία πλατφόρμα που είχε ξαναχρησιμοποιηθεί τοπικά στον υπολογιστή μου για εργασίες του πανεπιστημίου, αλλά είχαν ξεχασθεί βασικές λειτουργίες του. Αρχικά, δημιουργήθηκε ένα repository (αποθετήριο) χειροκίνητα και έπειτα ακολούθησαν οι εντολές add (πρόσθεσε), commit (δέσμευσε) και push (σπρώξε) για το ανέβασμα των αρχείων στο αποθετήριο αυτό. Παρ' όλα αυτά, παρατηρήθηκε πως χρειαζόντουσαν περαιτέρω αλλαγές σε κάποια αρχεία και για αυτό διαγράφηκε το αποθετήριο αυτό και ξανά δημιουργήθηκε.

Όταν ήρθε η ώρα να ξανά επαναληφθεί η διαδικασία για το σπρώξιμο των αρχείων στο κενό αυτό αποθετήριο, διαγράφηκαν όλα τα αρχεία που δεν τροποποιήθηκαν (δηλαδή σχεδόν όλα). Πιθανά αυτό να συνέβη, επειδή τοπικά ο υπολογιστής θυμάται να είχε περάσει όλα τα αρχεία, αλλά αυτά τα αρχεία δεν βρέθηκαν. Έπειτα από αυτό το γεγονός, έγινε μία αναζήτηση για τη καταπολέμηση του προβλήματος και βρέθηκε η εντολή reflog. Η εντολή reflog στην ουσία παρουσιάζει το ιστορικό των τροποποιήσεων και ενσωματώσεων του αποθετηρίου. Έπειτα από το χρονικό διάστημα που απαιτήθηκε για τη κατανόηση και εφαρμογή της εντολής αυτής, υλοποιήθηκε η επαναφορά των αρχείων, μέσω της κλήσης του σωστού commit. Για την επίτευξη της επαναφοράς χρησιμοποιήθηκαν λίγες απλές εντολές.

6 ΣΥΜΠΕΡΑΣΜΑΤΑ

Το τελικό κομμάτι της εργασίας αποτελείται από τις εντυώσεις και γνώσεις που αποκομίστηκαν. Πολλές γνώσεις είχαν τεχνικό χαρακτήρα και άλλες είχαν πιο κοινωνικό χαρακτήρα. Αρχικά, όσον αφορά τις τεχνικές γνώσεις, κατά τη διάρκεια υλοποίησης της εργασίας ήρθα σε επαφή με νέα εργαλεία που έχουν μεγάλη αναγνώριση στον χώρο εργασίας. Για την επιτυχής ολοκλήρωση της εργασίας απαιτήθηκε full stack [58] ανάπτυξη, δηλαδή υλοποιήθηκε και το μπροστινό μέρος και το πίσω μέρος της εφαρμογής από εμένα. Αυτό είχε ως αποτέλεσμα την εξοικείωση και εύκολη χρήση σύνθετων εργαλείων και την απόκτηση μίας μεγαλύτερης κατανόησης για την φύση του κώδικα.

Στον προσωπικό τομέα, χαρακτηριστικά όπως η αφοσίωση και η πειθαρχία ήταν απαραίτητα για την αντιμετώπιση ενός τόσο μεγάλου έργου. Η εργασία διήρκεσε πολλούς μήνες και προσωπικά συναισθήματα όπως βαρεμάρα ή αποθάρρυνση έπρεπε να καταπολεμηστούν. Φυσικά, η συνεχής πρόοδο της εργασίας, πρόσφερε μία μεγάλη ικανοποίηση και έδινε μεγαλύτερη ελπίδα για την συνέχεια.

Η χρήση της React, της Node.js και του MongoDB ήταν κάτι καινούργιο για εμένα και μέχρι να αποκτηθεί εξοικείωση, υπήρχαν πολλές δυσκολίες. Η React έχει ως μητρική γλώσσα προγραμματισμού τη JavaScript με την οποία υπήρχε μία πολύ μικρή επαφή στο παρελθόν, το οποίο βοήθησε στα πλαίσια της εργασίας. Επίσης, τόσο για τη React όσο και για το MongoDB, υπάρχει άπλετη πληροφορία στο διαδίκτυο σε μορφή οδηγών, άρθρων και βίντεο που βοήθησαν αισθητά για το τελικό αποτέλεσμα. Αξίζει να σημειωθεί πως μπορεί να υπήρχε έμπνευση από τις πηγές αυτές, αλλά ήταν απαραίτητη η προσαρμογή του κώδικα για τη προσωπική μας εργασία.

Τέλος, θα ήταν σωστό να επισημανθούν και κάποιες πιθανές επεκτάσεις και προσθήκες. Μία πολύ χρήσιμη επέκταση, θα μπορούσε να είναι η επιλογή Ελληνικής γλώσσας. Πολλοί χρήστες δεν γνωρίζουν καλά Αγγλικά και μπορεί να δυσκολευτούν να πλοηγηθούν στην ιστοσελίδα. Έπειτα λειτουργίες όπως αναζήτηση, εφαρμογή tokens για μεγαλύτερη ασφάλεια των προσωπικών δεδομένων και αυθεντικοποίηση του φοιτητή μέσω του πανεπιστημιακού του ιδρύματος, θα αποτελούσαν προσθήκες που θα έκαναν πιο επαγγελματική την ιστοσελίδα. Ως τελικό αποτέλεσμα όμως, η ιστοσελίδα είναι αρκετά ολοκληρωμένη και πλήρως λειτουργική, έχοντας επιτεύξει όλους τους αρχικούς στόχους και φιλοδοξίες του δημιουργού.

7 ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] “What is a Web App? | Codecademy.” Accessed: Nov. 04, 2023. [Online]. Available: <https://www.codecademy.com/article/what-is-a-web-app>
- [2] “3 Tier Architecture - YouTube.” Accessed: Nov. 01, 2023. [Online]. Available: <https://www.youtube.com/watch?v=THPg8Hz2zZo>
- [3] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, “*Cloud computing patterns: fundamentals to design, build, and manage cloud applications*”, Springer Science & Business Media, 2014.
- [4] “What Is a Cloud Application? Here’s What You Need to Know.” Accessed: Nov. 05, 2023. [Online]. Available: <https://www.businessinsider.com/guides/tech/what-is-a-cloud-application>
- [5] T. Mens, Demenyer, S., Heckel, R., Correia, R., Matos, C., El-Ramly, M., ... & Andrade, L., “*Architectural transformations: From legacy to three-tier and services*”, Software Evolution, 139-170, 2008.
- [6] Y. Bai and S. Bhalla, “*Oracle Database Programming with Java*”, chapter: *Introduction to Databases (pp. 9-66)*, Auerbach Publications, 2022.
- [7] “Introduction of DBMS (Database Management System) - Set 1 - GeeksforGeeks.” Accessed: Nov. 05, 2023. [Online]. Available: <https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/>
- [8] “What is a Front End (In a Website) - Definition & Development.” Accessed: Nov. 05, 2023. [Online]. Available: <https://airfocus.com/glossary/what-is-a-front-end/>
- [9] “REST API Introduction - GeeksforGeeks.” Accessed: Nov. 05, 2023. [Online]. Available: <https://www.geeksforgeeks.org/rest-api-introduction/>
- [10] “What is a Framework? - GeeksforGeeks.” Accessed: Nov. 05, 2023. [Online]. Available: <https://www.geeksforgeeks.org/what-is-a-framework/>
- [11] Elar Saks, “JavaScript frameworks: Angular vs React vs Vue”.
- [12] “What is the Angular framework?” Accessed: Nov. 05, 2023. [Online]. Available: <https://mdevelopers.com/blog/what-is-the-angular-framework>
- [13] “Getting started with Vue - Learn web development | MDN.” Accessed: Nov. 05, 2023. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Vue_getting_started
- [14] “What Is Vue JS? (Definition, Uses, Advantages) | Built In.” Accessed: Nov. 05, 2023. [Online]. Available: <https://builtin.com/software-engineering-perspectives/vue-js>

- [15] “Getting started with Ruby on Rails: Build a web app with Ruby.” Accessed: Oct. 19, 2023. [Online]. Available: <https://www.educative.io/blog/ruby-on-rails>
- [16] “Rails And Sqlite3 Database.” Accessed: Oct. 19, 2023. [Online]. Available: <https://rubyonwindowsguides.github.io/book/ch03-01.html>
- [17] P. DuBois, “*MySQL*”, Addison-Wesley, 2013.
- [18] “What is PHP? The PHP Programming Language Meaning Explained.” Accessed: Oct. 19, 2023. [Online]. Available: <https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/>
- [19] “What is Angular? | Simplilearn.” Accessed: Oct. 19, 2023. [Online]. Available: <https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular>
- [20] “Everything you need to know about Flask for beginners. - Great Learning.” Accessed: Nov. 06, 2023. [Online]. Available: <https://www.mygreatlearning.com/blog/everything-you-need-to-know-about-flask-for-beginners/>
- [21] “What is Java Spring Boot?—Intro to Spring Boot | Microsoft Azure.” Accessed: Nov. 06, 2023. [Online]. Available: <https://azure.microsoft.com/en-gb/resources/cloud-computing-dictionary/what-is-java-spring-boot>
- [22] “What Is a Content Management System (CMS)?” Accessed: Nov. 06, 2023. [Online]. Available: <https://kinsta.com/knowledgebase/content-management-system/>
- [23] “What is Joomla - Definition, meaning and examples.” Accessed: Oct. 19, 2023. [Online]. Available: <https://www.arimetrics.com/en/digital-glossary/joomla>
- [24] “What is Joomla? | Key features & Uses | Career And Advantages.” Accessed: Oct. 19, 2023. [Online]. Available: <https://www.educba.com/what-is-joomla/>
- [25] V. Subramanian, *Pro MERN Stack*. Apress, 2019. doi: 10.1007/978-1-4842-4391-6.
- [26] “What Is Node.js and Why You Should Use It.” Accessed: Dec. 27, 2023. [Online]. Available: <https://kinsta.com/knowledgebase/what-is-node-js/>
- [27] “What Is Express.js? Everything You Should Know.” Accessed: Dec. 27, 2023. [Online]. Available: <https://kinsta.com/knowledgebase/what-is-express-js/>
- [28] “What is REST API (RESTful API)?” Accessed: Oct. 18, 2023. [Online]. Available: <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>
- [29] “Node JS with Passport Authentication simplified | by Prashant Ram | Medium.” Accessed: Oct. 18, 2023. [Online]. Available: <https://medium.com/@prashantramnyc/node-js-with-passport-authentication-simplified-76ca65ee91e5>

- [30] P. Sriramya and R. A. Karthika, "PROVIDING PASSWORD SECURITY BY SALTED PASSWORD HASHING USING BCrypt ALGORITHM", *ARPN journal of engineering and applied sciences*, vol. 10, no. 13, pp. 5551-5556, 2015.
- [31] Cornelia Gyrödi, Robert Gyrödi, George Pecherle, and Andrada Olah, *A comparative study MongoDB vs. MySQL*.
- [32] "Collections — MongoDB Compass." Accessed: Oct. 18, 2023. [Online]. Available: <https://www.mongodb.com/docs/compass/current/collections/>
- [33] "What is Mongoose?. Mongoose is an object modeling tool for... | by Monib Bormon | Medium." Accessed: Oct. 18, 2023. [Online]. Available: <https://monib-bormon.medium.com/what-is-mongoose-c1bc3031cc08>
- [34] S. Chen, U. R. Thaduri, and V. K. R. Ballamudi, "Front-End Development in React: An Overview", *Engineering International*, 7(2), pp. 117–126, 2019.
- [35] P. Rawat and A. N. Mahajan, "ReactJS: A Modern Web Development Framework," 2020. [Online]. Available: www.ijisrt.com
- [36] "What is React.js? Uses, Examples, & More." Accessed: Oct. 18, 2023. [Online]. Available: <https://blog.hubspot.com/website/react-js>
- [37] "Flexbox - Learn web development | MDN." Accessed: Oct. 18, 2023. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox
- [38] I.-Y. Song, M. Evans, and U. E. K. Park, "A Comparative Analysis of Entity-Relationship Diagrams," 1995.
- [39] "(1) Entity Relationship Diagram (ERD) Tutorial - Part 1 - YouTube." Accessed: Oct. 22, 2023. [Online]. Available: <https://www.youtube.com/watch?v=QpdhBUYk7Kk>
- [40] "(1) Entity Relationship Diagram (ERD) Tutorial - Part 2 - YouTube." Accessed: Oct. 22, 2023. [Online]. Available: <https://www.youtube.com/watch?v=-CuY5ADwn24>
- [41] V. Bojinov, *RESTful web API design with Node.js: design and implement comprehensive RESTful solutions in Node.js*.
- [42] "How to Build a RESTful API Using Node, Express, and MongoDB." Accessed: Oct. 21, 2023. [Online]. Available: <https://www.freecodecamp.org/news/build-a-restful-api-using-node-express-and-mongodb/>
- [43] "User Authentication in NodeJS Using Passport and MongoDB." Accessed: Oct. 23, 2023. [Online]. Available: <https://www.makeuseof.com/user-authentication-in-nodejs/>
- [44] "(1) Complete MongoDB Tutorial #23 - Pagination - YouTube." Accessed: Oct. 23, 2023. [Online]. Available:

https://www.youtube.com/watch?v=zOI6W0DiYPc&list=PLnTyhvkIHJa_QGRP_Sy0_OmNyn5FVO_HML

- [45] “(1) Easy Pagination With MongoDB & React.js - YouTube.” Accessed: Oct. 23, 2023. [Online]. Available: https://www.youtube.com/watch?v=ja4yIn2pCzw&list=PLnTyhvkIHJa_QGRP_Sy0_OmNyn5FVO_HML
- [46] “reactjs - average score in react jsx - Stack Overflow.” Accessed: Oct. 28, 2023. [Online]. Available: <https://stackoverflow.com/questions/67337993/average-score-in-react-jsx>
- [47] “Using the Effect Hook – React.” Accessed: Oct. 22, 2023. [Online]. Available: <https://legacy.reactjs.org/docs/hooks-effect.html>
- [48] “How to Use the useState() Hook in React – Explained with Code Examples.” Accessed: Oct. 22, 2023. [Online]. Available: <https://www.freecodecamp.org/news/usestate-hook-3-different-examples/>
- [49] “How to use the useParams hook in React Router | refine.” Accessed: Oct. 22, 2023. [Online]. Available: <https://refine.dev/blog/react-router-useparams/#introduction>
- [50] “john-smilga/react-router-6-tutorial.” Accessed: Oct. 22, 2023. [Online]. Available: <https://github.com/john-smilga/react-router-6-tutorial>
- [51] “React JSX.” Accessed: Oct. 22, 2023. [Online]. Available: https://www.w3schools.com/react/react_jsx.asp
- [52] “Post Request Using Fetch API for Simple Login Page | by Jessica Lee | Medium.” Accessed: Oct. 24, 2023. [Online]. Available: <https://jessywee.medium.com/apply-http-post-method-and-fetch-function-to-user-login-js-react-d74a2c19ab7d>
- [53] “Login Page in React JS - Coding Ninjas.” Accessed: Oct. 24, 2023. [Online]. Available: <https://www.codingninjas.com/studio/library/login-page-in-react-js>
- [54] “Create a custom React star rating component - DEV Community.” Accessed: Oct. 25, 2023. [Online]. Available: <https://dev.to/michaelburrows/create-a-custom-react-star-rating-component-5o6>
- [55] M. H. Baturay and M. Birtane, “Responsive Web Design: A New Type of Design for Web-based Instructional Content,” *Procedia Soc Behav Sci*, vol. 106, pp. 2275–2279, Dec. 2013, doi: 10.1016/j.sbspro.2013.12.259.
- [56] “MERN stack project setup to push to Github - DEV Community.” Accessed: Oct. 29, 2023. [Online]. Available: <https://dev.to/birdy/mern-stack-project-setup-to-push-to-github-4l5e>
- [57] “Deploying your FREE web service using Render, MongoDB Atlas, and Github - YouTube.” Accessed: Oct. 29, 2023. [Online]. Available:

https://www.youtube.com/watch?v=RMNWI-_8SUU&list=PLnTyhvkIHJa_QGRPSy0_OmNyn5FVO_HML

- [58] A. Mardan, *Full stack JavaScript : learn Backbone.js, Node.js and MongoDB*.