



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΜΟΙΡΑΣΗΣ  
ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΣΥΝΕΠΙΒΑΤΙΣΜΟ ΙΔΙΩΤΙΚΩΝ ΟΧΗΜΑΤΩΝ**

Διπλωματική Εργασία

Αποστολία Ζιάκα

Επιβλέπουσα: Δασκαλοπούλου Ασπασία

Συνεπιβλέπων: Τσουκαλάς Ελευθέριος

Βόλος 2019



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΜΟΙΡΑΣΗΣ  
ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΣΥΝΕΠΙΒΑΤΙΣΜΟ ΙΔΙΩΤΙΚΩΝ ΟΧΗΜΑΤΩΝ**

Διπλωματική Εργασία

Αποστολία Ζιάκα

Επιβλέπων: Δασκαλοπούλου Ασπασία

Συνεπιβλέπων: Τσουκαλάς Ελευθέριος

Βόλος 2019



**UNIVERSITY OF THESSALY**

**SCHOOL OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**DEVELOPMENT OF A WEB-BASED DATA MANAGEMENT  
SYSTEM FOR CARPOOLING**

Diploma Thesis

Apostolia Ziaka

Supervisor: Daskalopoulou Aspasia

Supervisor: Tsoukalas Lefteris

Volos 2019

## ΕΥΧΑΡΙΣΤΙΕΣ

Ολοκληρώνοντας την διπλωματική μου αυτή εργασία, ολοκληρώνεται και η μεγάλη πορεία των σπουδών μου και αισθάνομαι ότι θα ήθελα να ευχαριστήσω όλους τους ανθρώπους που συνέβαλαν στην επίτευξη των στόχων μου.

Θα ήθελα να ευχαριστήσω την κυρία Δασκαλοπούλου για την βοήθειά της και την στήριξή της κατά την εκπόνηση την διπλωματικής αυτής εργασίας και τον κύριο Τσουκαλά για την συμβολή του ως συνεπιβλέπων καθηγητής.

Ένα μεγάλο ευχαριστώ θα ήθελα να πω στον συμφοιτητή και φίλο μου Σπύρο, οποίος με καθοδήγησε και με βοήθησε με τις γνώσεις του πάνω στο αντικείμενο της εργασίας μου, όποτε και αν το χρειάστηκα χωρίς δισταγμό. Ακόμη, ευχαριστώ πολύ την Μυρτώ για την συμβολή της στα γραφικά της εφαρμογής και για την στήριξή της καθ' όλη τη διάρκεια της εργασίας.

Τα φοιτητικά μου χρόνια σίγουρα δεν θα ήταν το ίδιο χωρίς τους φίλους μου που γνώρισα στο Βόλο. Αναπτύξαμε δυνατές σχέσεις, στηρίζοντας ο ένας τον άλλον, πορευθήκαμε μαζί στις σπουδές μας αποκτώντας κάποιες από τις καλύτερες αναμνήσεις και γιαυτό τους ευχαριστώ πολύ. Ιδιαίτερο ευχαριστώ, οφείλω στην φίλη μου Ιωάννα που εκτιμώ πολύ, καθώς με στήριξε σε μία δύσκολη και απαιτητική περίοδο κατά τη διάρκεια της εργασίας και όχι μόνο.

Ευχαριστώ πολύ αυτούς που ήταν εδώ και πολλά χρόνια εκεί για μένα, ακολουθώντας την πορεία μου, τους παιδικούς μου φίλους Νίκο, Κατερίνα, Ευαγγελία και Αλέξη, που πάντα με στήριζαν ανεξάρτητα την μεταξύ μας απόσταση, περνώντας πολλές από τις πιο καλές και κακές στιγμές μαζί.

Το πιο μεγάλο όμως ευχαριστώ θα ήθελα να το εκφράσω στην οικογένειά μου. Του γονείς μου Ευαγγελία και Βάιο και τα αδέρφια μου Σωτηρία και Θοδωρή, για την αστείρευτη αγάπη και στήριξή τους όλα αυτά τα χρόνια, χωρίς τους οποίους δεν θα είχα καταφέρει τίποτα από τους στόχους μου και δεν θα ήμουν ο ίδιος άνθρωπος.

Αποστολία Ζιάκα

Βόλος, Ιούλιος 2019



**ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ  
ΔΙΚΑΙΩΜΑΤΩΝ**

«Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής».

Η Δηλούσα

(Υπογραφή)

Ζιάκα Αποστολία

02/07/19

## ΠΕΡΙΛΗΨΗ

Η διπλωματική αυτή εργασία έχει ως αντικείμενο διαπραγμάτευσης την υλοποίηση μιας διαδικτυακής εφαρμογής συνεπιβατισμού. Η ανάπτυξη της εφαρμογής συμπεριλαμβάνει την υλοποίηση της διεπαφής του χρήστη με την εφαρμογή και την ανάπτυξη του διακομιστή της εφαρμογής. Γίνεται χρήση κυρίως νέων τεχνολογιών ανάπτυξης διαδικτυακών εφαρμογών, όπως η React.js και το Node.js και στόχος είναι μέσω της ανάπτυξης να μελετηθούν και οι συγκεκριμένες τεχνολογίες. Στην εφαρμογή ο χρήστης θα έχει την δυνατότητα να έχει λογαριασμό, να αναζητά και να προσφέρει διαδρομές από μία τοποθεσία σε μία άλλη και να αλληλοεπιδρά με άλλους χρήστες.

### Λέξεις Κλειδιά

Διαδικτυακή Εφαρμογή, Συνεπιβατισμός, React.js, Node.js, Express, PostgreSQL

## **ABSTRACT**

This diploma thesis has as subject of negotiation, the implementation of a web application for carpooling. The application's development includes the implementation of user's interface and development of the application's server. In particular, new web application development technologies such as React.js and Node.js are being used, and the goal through development is the studying of these technologies as well. In the application, the user will be able to have an account, search for and offer rides from one location to another and interact with other users.

### **Keywords**

Web Application, Carpooling, React.js, Node.js, Express, PostgreSQL



## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

<b>ΠΕΡΙΛΗΨΗ</b> .....	<b>vii</b>
<b>ABSTRACT</b> .....	<b>viii</b>
<b>ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ</b> .....	<b>ix</b>
<b>ΚΕΦΑΛΑΙΟ 1</b> .....	<b>1</b>
<b>ΕΙΣΑΓΩΓΗ</b> .....	<b>1</b>
1.1 Περιγραφή της ιδέας.....	1
1.2 Στόχος της εργασίας .....	3
1.3 Οργάνωση του τόμου .....	4
<b>ΚΕΦΑΛΑΙΟ 2</b> .....	<b>5</b>
<b>ΤΕΧΝΟΛΟΓΙΚΟ ΥΠΟΒΑΘΡΟ</b> .....	<b>5</b>
2.1 Βασικοί Ορισμοί .....	5
2.2 Web Application .....	6
2.2.1 Web Application Architecture .....	6
2.3 AJAX .....	8
<b>ΚΕΦΑΛΑΙΟ 3</b> .....	<b>9</b>
<b>ΤΕΧΝΟΛΟΓΙΚΗ ΕΠΙΣΚΟΠΗΣΗ</b> .....	<b>9</b>
3.1 Javascript.....	9
3.1.1 Full Stack JavaScript Development.....	10
3.2 Node.js .....	11
3.2.1 NPM.....	13
3.2.2 Express.....	14
3.2.3 Socket.io .....	16
3.3 React.js.....	16
3.3.1 Virtual DOM .....	17
3.3.2 Lifecycle Methods .....	19
3.3.3 JSX .....	21
3.4 PostgreSQL .....	22
3.4.1 pgAdmin .....	23

3.4.2 Knex.js .....	24
3.4.3 Schema .....	25
3.5 Εργαλεία .....	26
<b>ΚΕΦΑΛΑΙΟ 4.....</b>	<b>28</b>
<b>ΛΕΙΤΟΥΡΓΙΚΗ ΕΠΙΣΚΟΠΗΣΗ .....</b>	<b>28</b>
4.1 Λειτουργίες Εφαρμογής .....	28
4.1.1 Αρχική Σελίδα .....	29
4.1.2 Εγγραφή Χρήστη.....	29
4.1.3 Σύνδεση Χρήστη .....	31
4.1.4 Αναζήτηση Διαδρομής .....	32
4.1.5 Εμφάνιση Διαδρομών.....	32
4.1.6 Λεπτομέρειες Διαδρομής.....	33
4.1.7 Επικοινωνία με άλλον χρήστη .....	34
4.1.8 Προσφορά Διαδρομής .....	35
4.1.9 Εμφάνιση Χάρτη και προτεινόμενης διαδρομής.....	36
4.1.10 Προφίλ Χρήστη .....	37
4.1.11 Αποσύνδεση Χρήστη.....	39
4.2 Βάση Δεδομένων .....	39
<b>ΚΕΦΑΛΑΙΟ 5.....</b>	<b>42</b>
<b>ΣΥΝΟΨΗ .....</b>	<b>42</b>
5.1 Συμπεράσματα .....	42
5.2 Μελλοντικές Επεκτάσεις της εφαρμογής.....	43
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>44</b>
<b>ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ.....</b>	<b>49</b>

## Κατάλογος Σχημάτων

2.1 Web Application Architecture.....	7
3.1 Ετήσια έρευνα stackoverflow [10].....	9
3.2 Το σύστημα Node.js [12].....	12
3.3 Διαφορά λειτουργίας ενός παραδοσιακού server και ενός server Node.js [13].....	13
3.4 Κώδικας σε express, δημιουργία αντικειμένου.....	15
3.5 Αναπαράσταση διεπαφής σε ένα Virtual DOM [19].....	18
3.6 Σύγκριση Virtual DOM και πραγματικού DOM [19].....	18
3.7 Τελική μορφή του πραγματικό DOM [19].....	19
3.8 ReactJS Component’s Lifecycle [20] ].....	20
3.9 Παράδειγμα κώδικα JSX.....	21
3.10 Σύνδεση pgAdmin με τον server.....	24
3.11 Αρχείο knexfile.js – Διαχείριση σύνδεσης με τη βάση PostgreSQL.....	25
3.12 Παράδειγμα σχήματος στο pgAdmin4.....	26
4.1 Screenshot Αρχικής Σελίδας.....	29
4.2 Screenshot Εγγραφής Χρήστη.....	30
4.3 Screenshot Σύνδεσης Χρήστη.....	31
4.4 ScreenshotΑναζήτησης Διαδρομής.....	32
4.5 Screenshot Εμφάνιση Διαδρομών.....	33
4.6 Screenshot Λεπτομέρειες Διαδρομής.....	34
4.7 Screenshot Επικοινωνία με άλλο χρήστη.....	35
4.8 Screenshot Προσφοράς Διαδρομής.....	36
4.9 Screenshot Εμφάνιση Χάρτη και Προτεινόμενης Τιμής.....	37
4.10 Screenshot Προφίλ Χρήστη.....	38
4.10 Screenshot Εμφάνισης Μηνυμάτων Χρήστη.....	38
4.11 Screenshot Εμφάνισης Προσφερόμενων Διαδρομών Χρήστη.....	38
4.12 Σχήματα πινάκων βάσης δεδομένων.....	39



# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ

### 1.1 Περιγραφή της ιδέας

Ο συνεπιβατισμός ή το carpooling είναι ένας όρος που χρησιμοποιείται για να περιγράψει δύο ή περισσότερα άτομα που έχουν παρόμοιο μοτίβο μετακίνησης και μοιράζονται μια διαδρομή, χρησιμοποιώντας ιδιωτικά οχήματα, σε μια προσπάθεια να μειώσουν την κυκλοφοριακή συμφόρηση, το κόστος και τη ρύπανση [1]. Ακόμη και αν το carpooling έγινε πιο δημοφιλές τα τελευταία χρόνια, έχουν περάσει δεκαετίες που χρησιμοποιείται σαν όρος. Η ιστορία του carpooling ξεκινά στις αρχές του 1914, όταν οι ΗΠΑ αντιμετώπιζαν ύφεση. Οι ιδιοκτήτες αυτοκινήτων άρχισαν να προσφέρουν τα αυτοκίνητά τους για μαζική μετακίνηση, κάτι που εξόργισε τους επαγγελματίες μεταφορείς. Κατά τη διάρκεια του Β΄ Παγκοσμίου Πολέμου οργανώθηκαν προσπάθειες από την αμερικανική κυβέρνηση, η οποία ξόδεψε 8 εκατομμύρια δολάρια για διαφημιστικές εκστρατείες, με σκοπό να πείσει τους πολίτες να συνεπιβαίνουν σε αυτοκίνητα και να εξοικονομούν καύσιμα και χρήματα. Οι εκστρατείες είναι γνωστές ως "Car Clubs". Κατά τη δεκαετία του '70, το carpooling άκμασε και πάλι κατά τη διάρκεια της ενεργειακής κρίσης. Καθ' όλη τη διάρκεια του 1973, η κυβέρνηση του προέδρου Νixon αντιλήφθηκε ότι θα πρέπει να αναλάβει δράση για να μειώσει την κατανάλωση πετρελαίου και το 1974 υπέγραψε τον Νόμο για την Εξοικονόμηση Ενέργειας έκτακτης ανάγκης. Η πράξη αυτή απαγόρευσε τις ταχύτητες αυτοκινήτων άνω των 55 μιλίων ανά ώρα. Συνδυαστικά με αυτή την ενέργεια, η κυβέρνηση των ΗΠΑ άρχισε επίσης να προσφέρει χρηματοδοτήσεις για τις δράσεις συνεπιβατισμού. Κατά το διάστημα αυτό, οι πολιτείες των ΗΠΑ άρχισαν να εφαρμόζουν ειδικές λωρίδες οχημάτων υψηλής χωρητικότητας, γνωστές ως HOV (λωρίδες "High Occupancy Vehicle"). Οι λωρίδες HOV ήταν δρόμοι, όπου τα οχήματα που ταξιδεύουν πρέπει να φέρουν τον ελάχιστο αριθμό επιβατών που έχουν αναρτηθεί στις πινακίδες εισόδου. Οι παραβάτες έπρεπε να καταβάλουν πρόστιμο τουλάχιστον 481 δολαρίων. Μετά το εμπάργκο του πετρελαίου, η κυβέρνηση χρηματοδότησε τις πρώτες μητροπολιτικές εταιρείες που ασχολούνται με τον συνεπιβατισμό, οι οποίες προσπάθησαν να προωθήσουν το carpooling με διάφορες εκστρατείες μάρκετινγκ.

Ακολουθώντας την ηγεσία του προέδρου Νίξον, ο Πρόεδρος Carter, το 1979, διόρισε την Εθνική Ομάδα Εργασίας για τον συνεπιβατισμό. Στη δεκαετία του 1980, μετά από μια περίοδο ανθοφορίας, το carpooling παρήκμασε εξαιτίας των χαμηλών τιμών του πετρελαίου και της ισχυρής οικονομικής ανάπτυξης, ενώ τη δεκαετία του 1990 ήταν πιο δημοφιλές μεταξύ των φοιτητών. Κατά τη δεκαετία του 2000, η ανάπτυξη του Ίντερνετ συνέβαλε πολύ στην αύξηση της δημοφιλίας του carpooling και τα smartphones και το GPS το έκαναν διαθέσιμο στο εμπόριο[2][3].

Οι πιο δημοφιλείς εταιρείες στις ΗΠΑ σήμερα είναι η Lyft και η Uber, οι οποίες δημιούργησαν ένα ασφαλές σύστημα επικοινωνίας και συναλλαγών μεταξύ ταξιδιωτών και οδηγών επί πληρωμή. Στην Ευρώπη, το carpooling έχει γίνει πιο γνωστό τα τελευταία χρόνια, χάρη στη γερμανική Mitfahrgelegenheit (carpooling.com) και στη γαλλική BlaBlaCar που εξαγόρασε την παραπάνω το 2015 [4].

Πλεονεκτήματα του carpooling:

1. Συμβάλλει στη μείωση της κατανάλωσης ενέργειας και χώρου

Με την πολιτική carpooling επιδιώκεται η αύξηση της πληρότητας του αυτοκινήτου. Η πληρότητα σήμερα είναι λίγο μεγαλύτερη από ένα, δηλαδή τα περισσότερα ιδιωτικά αυτοκίνητα, όπως και οι μοτοσυκλέτες, κυκλοφορούν μόνο με τον οδηγό τους. Αυτό έχει ως αποτέλεσμα περισσότερα αυτοκίνητα στους δρόμους άρα και περισσότερη ρύπανση. Η αύξηση της πληρότητας σημαίνει ότι ένα μέρος των οδηγών δεν χρησιμοποιούν το αυτοκίνητό τους αφού εξυπηρετούνται με κάποιο άλλο. Η πολιτική του car pooling ανήκει επομένως στις πολιτικές που κάνουν λιγότερο απαραίτητη την αγορά νέου αυτοκινήτου, ή τουλάχιστον συμβάλλει στην αποφόρτιση του οδικού δικτύου. Δεν αμφισβητεί λοιπόν κανείς ότι τα οφέλη στο αστικό περιβάλλον είναι τεράστια.

2. Βοηθά στην κοινωνικοποίηση

Το car pooling είναι μια πολιτική που έχει κοινωνικές παραμέτρους. Συμμετέχοντας σημαίνει αρχικά ότι αποδέχεσαι να μοιράζεσαι το αυτοκίνητό σου με κάποιους που δεν

είναι του στενού σου περιβάλλοντος και έτσι ενισχύεται η κοινωνική αλληλεγγύη. Κυρίως όμως αποτελεί ευκαιρία για μια νέα γνωριμία και καλύτερη επικοινωνία και κατανόηση μεταξύ των μελών μιας κοινότητας.

### 3. Μειώνει το κόστος μετακινήσεων

Η πολιτική του carpooling είναι μια πολιτική που συμφέρει. Αρχικά, οι συνεπιβάτες μοιράζονται το κόστος του καυσίμου, έτσι ώστε να ωφελούνται εξίσου από την κοινή τους μετακίνηση. Επίσης, στο πλαίσιο των πολιτικών βιώσιμης μετακίνησης, σε πολλές χώρες δίνονται κίνητρα σε όσους συμμετέχουν σε πρόγραμμα car pooling ή γενικότερα σε οδηγούς που δεν μετακινούνται μόνοι τους. Τέτοια κίνητρα συνήθως είναι: ειδικές λωρίδες για οχήματα υψηλής πληρότητας (HOV lanes), παροχή δωρεάν στάθμευσης ή μείωση του υποχρεωτικού χρόνου εργασίας.

## 1.2 Στόχος της εργασίας

Δύσκολα θα μπορούσε κανείς να φανταστεί τη ζωή μας σήμερα χωρίς την χρήση πολλών διαδικτυακών εφαρμογών με στόχο την ενημέρωση, την κοινωνικοποίηση, την διασκέδαση και την κάλυψη διαφόρων αναγκών. Πλέον η ανάπτυξη εφαρμογών είναι ραγδαία και η καθημερινότητά μας είναι άμεσα συνδεδεμένη και πολλές φορές εξαρτώμενη από αυτή. Παράλληλα, παρατηρώντας την ανάπτυξη του carpooling τον τελευταίο καιρό, κυρίως στην Ευρώπη είναι αρκετά ενδιαφέρον να μελετηθεί ο συνδυασμός της χρήσης διαδικτυακών εφαρμογών και της χρήσης του carpooling με βάση πλέον τις σύγχρονες τεχνολογίες και τα δεδομένα της καθημερινότητας στην σύγχρονη κοινωνία. Στόχος λοιπόν αυτής της διπλωματικής είναι να εμβαθύνει στις τελευταίες τεχνολογίες παγκόσμιου ιστού, κατασκευάζοντας μία εφαρμογή που βοηθά τους χρήστες να προσθέσουν το carpooling στην καθημερινότητα τους και να επωφεληθούν από αυτό.

### **1.3 Οργάνωση του τόμου**

Στο **Κεφάλαιο 1**, υπάρχει η εισαγωγή της διπλωματικής, όπου αναλύεται η ιδέα και ο στόχος της διπλωματικής.

Στο **Κεφάλαιο 2**, υπάρχει η περιγραφή της αρχιτεκτονικής μιας σύγχρονης διαδικτυακής εφαρμογής.

Στο **Κεφάλαιο 3**, βρίσκεται μία αναλυτική επισκόπηση των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Στο **Κεφάλαιο 4**, περιλαμβάνεται η περιγραφή των λειτουργιών όλων των σελίδων της εφαρμογής.

Και τέλος στο **Κεφάλαιο 5**, βρίσκονται τα συμπεράσματα και οι μελλοντικές επεκτάσεις της εργασίας.



## ΚΕΦΑΛΑΙΟ 2

### ΤΕΧΝΟΛΟΓΙΚΟ ΥΠΟΒΑΘΡΟ

#### 2.1 Βασικοί Ορισμοί

Ακολουθούν μερικές λέξεις κλειδιά μαζί με τους ορισμούς τους που θα χρησιμοποιηθούν κατά τη διάρκεια της διπλωματικής και θα πρέπει να είναι από πριν κατανοητές [5].

**Software Stack** – Είναι ένα πακέτο ξεχωριστών υποσυστημάτων που όλα μαζί δημιουργούν μια ολοκληρωμένη πλατφόρμα πάνω στην οποία μπορεί να λειτουργήσει αυτόνομα ένα σύστημα.

**Back-End / Front-End Development** – Είναι ο διαχωρισμός της φάσης της δημιουργίας μίας εφαρμογής ανάμεσα στο κομμάτι που θα τρέχει στον απομακρυσμένο server και σε αυτό που θα τρέχει στον υπολογιστή του χρήστη.

**API** – Είναι η Διασύνδεση Προγραμματισμού Εφαρμογών, ή αλλιώς διεπαφή, με την οποία ένα σύστημα μπορεί να επικοινωνήσει με ένα άλλο και να ζητήσει δεδομένα ή λειτουργίες.

**Framework** – Είναι ένα πρόγραμμα, ή σειρά προγραμμάτων που σκοπός τους είναι να προσφέρουν περισσότερες επιλογές και ευκολίες στη χρήση μιας γενικότερης πλατφόρμας.

**Βιβλιοθήκη** – Είναι μια συλλογή από εφόδια που έχουν φτιαχτεί για να κάνουν πιο εύκολη την διαδικασία του Development. Μοιάζει νοητικά με το Framework αλλά συνήθως είναι πολύ μικρότερου μήκους και δεν δημιουργούνται για συγκεκριμένες τεχνολογίες.

## 2.2 Web Application

Μία διαδικτυακή εφαρμογή (web application) είναι ένα πρόγραμμα λογισμικού, που είναι αποθηκευμένο σε έναν απομακρυσμένο διακομιστή και εκτελείται σε ένα πρόγραμμα περιήγησης στο Internet, μέσω μιας ιστοσελίδας [6]. Τα δεδομένα ανταλλάσσονται μέσω του δικτύου χρησιμοποιώντας το πρωτόκολλο μεταφοράς υπερκειμένου (HTTP).

Η χρήση των web application έχει ως πλεονέκτημα ότι οι χρήστες δεν εξαρτώνται από συγκεκριμένο λειτουργικό σύστημα ή απαιτήσεις υλικού (hardware). Γεγονός που κάνει τις εφαρμογές ιστού υπηρεσίες πολλαπλών πλατφορμών.

### 2.2.1 Web Application Architecture

Μία εφαρμογή χαρακτηρίζεται από την αρχιτεκτονική της (Web Application Architecture). Η αρχιτεκτονική μιας εφαρμογής είναι ένα πλαίσιο που αποτελείται από τις σχέσεις και τις αλληλεπιδράσεις μεταξύ των συστατικών στοιχείων της εφαρμογής, όπως τα συστήματα ενδιάμεσου ιστού (middleware), τις διεπαφές χρήστη και τις βάσεις δεδομένων. Σε γενικές γραμμές, αυτό είναι ένα μοντέλο αλληλεπίδρασης μεταξύ των συστατικών εφαρμογών ιστού όπως ο διακομιστής ιστού, ο διακομιστής βάσης δεδομένων, το πρόγραμμα περιήγησης κλπ.

Υπάρχουν πολλά διαφορετικά μοντέλα αρχιτεκτονικής, τα οποία καλύπτουν διάφορες απαιτήσεις που θέτει ο χρήστης, ο προγραμματιστής ή ο ιδιοκτήτης του λογισμικού.

Τα στοιχεία από τα οποία αποτελείται η αρχιτεκτονική μιας εφαρμογής μπορούν να κατηγοριοποιηθούν σε δύο κατηγορίες:

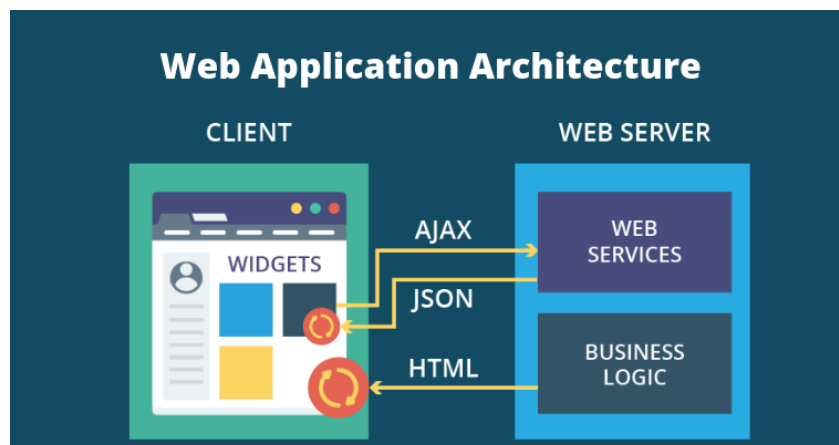
1. Τα στοιχεία διεπαφής του χρήστη
2. Τα δομικά στοιχεία

Τα στοιχεία διεπαφής του χρήστη αναφέρονται στην εμφάνιση της σελίδας, στις ειδοποιήσεις, στα αρχεία καταγραφής, στις ρυθμίσεις διαμόρφωσης και πολλά άλλα. Ασχολούνται αποκλειστικά με την εμπειρία του χρήστη και όχι με την δομική ανάπτυξη.

Τα δομικά στοιχεία αποτελούν την βάση της αναπτυξιακής διαδικασίας και είναι ο περιηγητής - πελάτης (client), ο διακομιστής εφαρμογής (application server) και ο διακομιστής βάσης δεδομένων (database server) [7].

1. Περιηγητής ή πελάτης (client): Είναι υπεύθυνος για την απόδοση της αλληλεπίδρασης των λειτουργιών της εφαρμογής με τον χρήστη. Στην ουσία το πρόγραμμα περιήγησης διαχειρίζεται πως αλληλοεπιδρούν οι χρήστες με την εφαρμογή. Το περιεχόμενο του αναπτύσσεται με τη χρήση HTML, CSS και JavaScript και το πιο σημαντικό είναι ότι δεν εξαρτάται από το λειτουργικό σύστημα του υπολογιστή.
2. Διακομιστής Εφαρμογής (application server): Διαχειρίζεται τη λογική και τα δεδομένα της εφαρμογής. Το περιεχόμενο του αναπτύσσεται σε γλώσσες όπως η PHP, Java, Node.js, Ruby και άλλες.
3. Διακομιστής Βάσεων Δεδομένων (database server): Αποθηκεύει και προσφέρει τα δεδομένα και πληροφορίες της εφαρμογής, τα οποία διαχειρίζεται ο διακομιστής.

Η αλληλεπίδραση του χρήστη με την εφαρμογή, το κομμάτι δηλαδή που ο χρήστης μπορεί να δει στον περιηγητή του ονομάζεται και fronted side, ενώ το κομμάτι του διακομιστή εφαρμογής που διαχειρίζεται τη λογική της εφαρμογής και τα δεδομένα ονομάζεται backend side [7] [8].



Σχήμα 2.1 Web Application Architecture

Η συνηθισμένη διαδικασία με την οποία δουλεύει η αρχιτεκτονική μιας εφαρμογής σύμφωνα με το διάγραμμα είναι η παρακάτω:

- Χρησιμοποιώντας τον περιηγητή, ο χρήστης στέλνει μία εντολή στον διακομιστή μέσω του Internet.
- Ο διακομιστής ιστού είναι υπεύθυνος να προωθήσει τις εντολές στον απαιτούμενο διακομιστή.
- Έπειτα από αλληλεπίδραση με τη βάση δεδομένων, ο διακομιστής βρίσκει τα απαιτούμενα αποτελέσματα των εντολών.
- Η εφαρμογή δίνει τα αποτελέσματα στον διακομιστή.
- Και αυτός με την σειρά του παρέχει στον χρήστη τις ζητούμενες πληροφορίες.

### **2.3 AJAX**

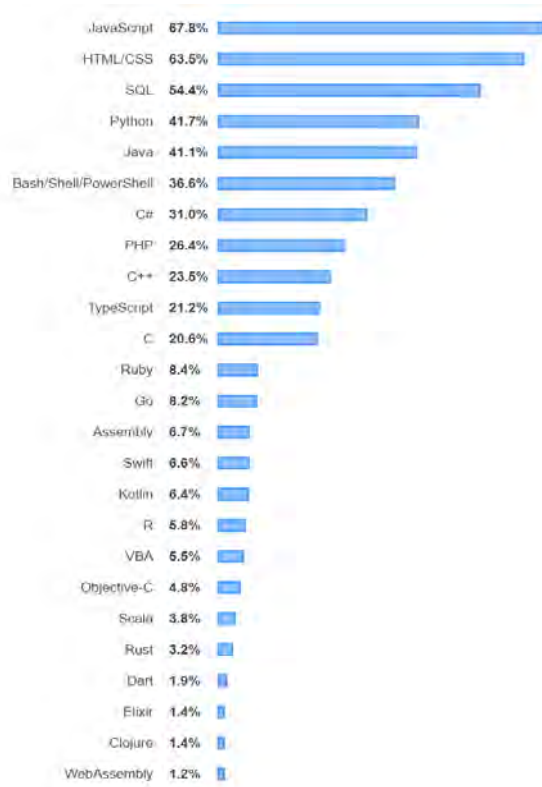
Το AJAX είναι μία μέθοδος ανάπτυξης web applications, η οποία δημιουργεί ασύγχρονες εφαρμογές. Ο περιηγητής και ο διακομιστής χρησιμοποιούν το AJAX για να επικοινωνήσουν. Το AJAX κάνει χρήση του XMLHttpRequest αντικειμένου για να επικοινωνήσει με τον διακομιστή και να στείλει και να λάβει πληροφορίες σε μορφή XML, JSON, HTML ή άλλη μορφή. Το XMLHttpRequest είναι ένα API, το οποίο επιτρέπει την ασύγχρονη επικοινωνία του περιηγητή και του διακομιστή χωρίς να χρειάζεται ανανέωση ολόκληρης της ιστοσελίδας [9].

## ΚΕΦΑΛΑΙΟ 3

### ΤΕΧΝΟΛΟΓΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

#### 3.1 Javascript

Η Javascript είναι σύμφωνα με πρόσφατες έρευνες η πιο αναπτυσσόμενη γλώσσα προγραμματισμού τα τελευταία χρόνια. Σύμφωνα με την ετήσια έρευνα του stackoverflow, η Javascript το 2019 είναι η περισσότερο χρησιμοποιούμενη γλώσσα προγραμματισμού, προτιμώμενη από το 67,8% των προγραμματιστών που συμμετείχαν στην έρευνα [10].



Σχήμα 3.1 Ετήσια έρευνα stackoverflow [10]

Είναι υπεύθυνη για την προσθήκη όλων των διαδραστικών στοιχείων και λειτουργιών μιας ιστοσελίδα ή μιας διαδικτυακής εφαρμογής. Θεωρείται μία από τα τρεις βασικές frontend τεχνολογίες μαζί με την HTML και CSS και χρησιμοποιείται ακόμη σε διάφορα frameworks

τόσο του frontend side, όπως είναι η React και η Angular, όσο και του backend side όπως το Node.js.

Αναπτύχθηκε από τον Αμερικανό προγραμματιστή Brendan Eich στο Netspace το 1995. Αρχικά ονομάστηκε “Mocha” και “Livescript”, και τον Δεκέμβριο του 1995 πήρε το παροντικό της όνομα κατά την κυκλοφορία της τρίτης beta εκδοχής της [9].

Η JavaScript ονομάζεται και γλώσσα του περιηγητή. Οι προγραμματιστές την προτιμούν γιατί δεν είναι ανάγκη για την εργασίας τους να λαμβάνουν υπόψη τους τα χαρακτηριστικά του συστήματος. Ο κώδικας εκτελείται από τον περιηγητή και όχι από τον εξυπηρετητή ιστού, περιορίζοντας έτσι την υπερφόρτωση του εξυπηρετητή. Η σύνταξή της είναι σχετικά απλή και χαλαρή και δεν είναι μια typed γλώσσα, δηλαδή ο μεταγλωττιστής JavaScript δεν παράγει type errors και μπορεί να διαχειριστεί μεταβλητές οποιουδήποτε τύπου χωρίς να δηλωθούν αρχικά, χαρακτηριστικό που την καθιστά ακόμη πιο ευέλικτη σαν γλώσσα προγραμματισμού. Έχει την δυνατότητα να εκτελεί απλούς υπολογισμούς στην πλευρά του περιηγητή και να λαμβάνει και να στέλνει δεδομένα στον διακομιστή ασύγχρονα μέσω του AJAX. Επίσης χαρακτηριστικό της είναι και η κληρονομικότητα. Ένα αντικείμενο μπορεί να κληρονομήσει ιδιότητες και από άλλα αντικείμενα χωρίς την έννοια των κλάσεων.

### **3.1.1 Full Stack JavaScript Development**

Η JavaScript αν και μετρά πάνω από είκοσι χρόνια ζωής, δεν ήταν πάντα τόσο δημοφιλής λόγω της απόδοσής της και την έλλειψη συμβατότητας με τους περιηγητές της εποχής. Η βελτίωση της γλώσσας κατέστησε τη χρήση της απαραίτητη στη δημιουργία της frontend πλευράς κάθε εφαρμογής ενώ το backend κομμάτι, δηλαδή η πλευρά του διακομιστή αναπτυσσόταν με τεχνολογίες όπως η PHP, η ASP.NET, η JAVA και άλλες. Η χρήση δύο διαφορετικών τεχνολογιών φέρεται να δυσκολεύει την ανάπτυξη ενός web application και για αυτό το λόγο άρχισαν οι προσπάθειες να ενοποιηθούν οι δύο πλευρές και να χρησιμοποιηθεί μία μόνο τεχνολογία [9].

Λύση στο πρόβλημα αυτό καλείται να δώσει η ανάπτυξη του Node.js, το οποίο είναι ένα περιβάλλον JavaScript ανοιχτού κώδικα, και κάνει δυνατή την εκτέλεση της JavaScript και από την πλευρά του διακομιστή. Αναπτύσσοντας όλα τα μέρη μιας εφαρμογής με JavaScript, δηλαδή κοινή γλώσσα για το frontend και το backend κομμάτι, επιτυγχάνεται η καλύτερη αποδοτικότητα και κατανόηση του κώδικα από την ομάδα ανάπτυξης της εφαρμογής, χωρίς να χρειάζονται δύο διαφορετικές ομάδες για κάθε κομμάτι. Επιπλέον, η χρήση του Node.js στηρίζεται σε ένα μοντέλο προσανατολισμένο στη χρήση γεγονότων, το οποίο καθιστά την εκτέλεσή του πιο ελαφριά και γρήγορη σε σχέση με άλλες backend τεχνολογίες.

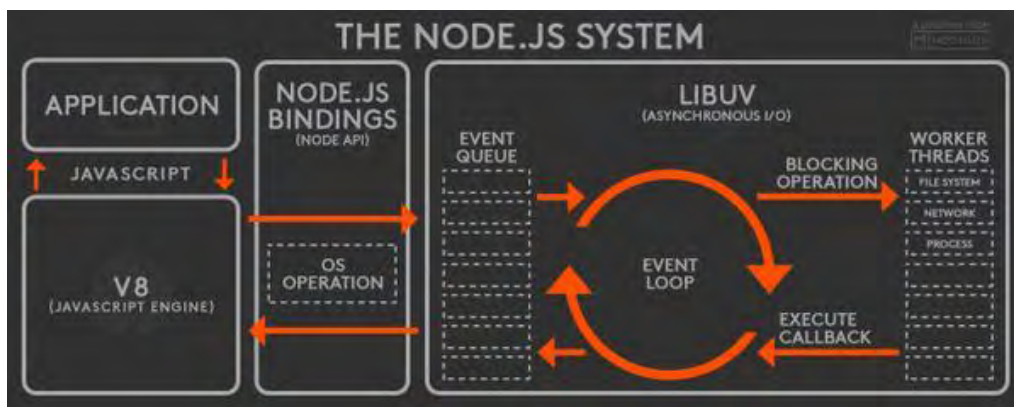
Αξίζει να αναφέρουμε την έκθεση της εταιρείας PayPal, η οποία ερεύνησε την απόδοση μιας εφαρμογής της χτισμένη σε Node.js σε σύγκριση με μια άλλη εφαρμογή της αναπτυγμένη σε Java. Οι δύο εφαρμογές είχαν ακριβώς τις ίδιες λειτουργίες, οπότε και στις δύο περιπτώσεις εκτελέστηκαν οι ίδιες λειτουργικές δοκιμές. Τα αποτελέσματα της έρευνας έδειξαν ότι η εφαρμογή που χρησιμοποιούσε Node.js υλοποιήθηκε σχεδόν δύο φορές πιο γρήγορα με λιγότερα άτομα να συμμετέχουν στην υλοποίηση. Επίσης, γράφτηκε σε 33% λιγότερες γραμμές κώδικα και χρησιμοποιήθηκαν 40% λιγότερα αρχεία σε σχέση με την εφαρμογή της Java. Όσον αφορά την απόδοση των δύο εφαρμογών η διαφορά είναι αξιοσημείωτη. Με την εφαρμογή σε Node.js είχαμε τα διπλάσια αιτήματα (requests) που ολοκληρώθηκαν ανά δευτερόλεπτο σε σχέση με την εφαρμογή της Java και μειώθηκε ο χρόνος ανταπόκρισης των αιτημάτων κατά 35% για την ίδια σελίδα, δηλαδή είχαμε μείωση του χρόνου των προβολών των σελίδων κατά 200ms, κάτι που οι χρήστες μπορούν να παρατηρήσουν [11].

### **3.2 Node.js**

Όπως αναφέρθηκε και παραπάνω, η ανάπτυξη του Node.js άλλαξε τα δεδομένα στην δημιουργία των web applications τα τελευταία χρόνια. Το Node.js δημιουργήθηκε από τον Ryan Dahl το 2009 και είναι μία πλατφόρμα εκτέλεσης ανοιχτού κώδικα JavaScript για την ανάπτυξη της backend πλευράς μιας εφαρμογής. Εκτελείται πάνω στο V8 Javascript engine του Google Chrome, μία μηχανή γραμμένη σε C++ που μετατρέπει κατευθείαν τον κώδικα JavaScript σε γλώσσα μηχανής, γεγονός που καθιστά το Node.js γρήγορο και αποδοτικό.

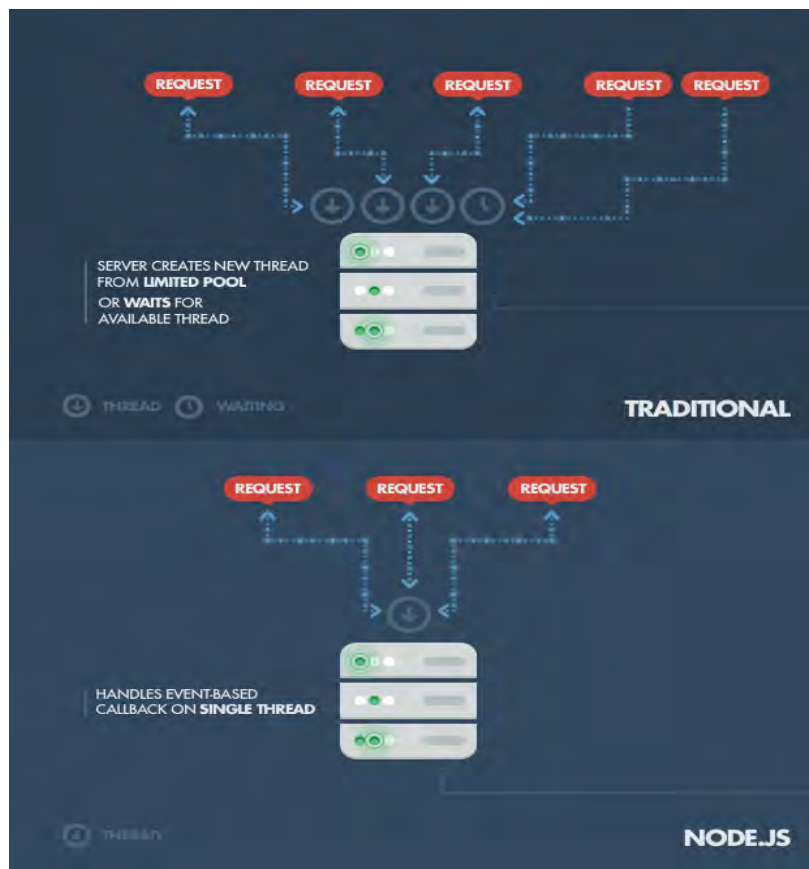
Μια εφαρμογή σε Node.js εκτελείται σε μια ενιαία διεργασία χωρίς να δημιουργείται ένα νέο νήμα για κάθε αίτημα. Στο παραδοσιακό μοντέλο διακομιστή εφαρμογών, συναντούμε πολλαπλά νήματα, δηλαδή ένα νήμα για κάθε σύνδεση. Κάθε νήμα πρέπει να περιμένει για I/O (είσοδο / έξοδο) πριν την επεξεργασία του επόμενου αιτήματος, αφήνοντας ανενεργό τον επεξεργαστή περιμένοντας να εκτελεστεί μια λειτουργία. Σε αντίθεση λοιπόν με το συμβατικό μοντέλο, το Node.js δεν στηρίζεται στην πολυνηματικότητα αλλά στην ασύγχρονη επικοινωνία και στη διαχείριση γεγονότων (events). Όταν μία λειτουργία I/O, όπως η ανάγνωση από το δίκτυο και η πρόσβαση στη βάση δεδομένων, χρειάζεται να εκτελεστεί, το νήμα δεν αποκλείεται με αποτέλεσμα να σπαταλούνται CPU κύκλοι αναμένοντας, αλλά επαναλαμβάνει τις λειτουργίες επεξεργασίας (callbacks) όταν έρθει η απάντηση, όταν δηλαδή τα δεδομένα γίνουν διαθέσιμα μετά από ένα συμβάν (event), όπως το πάτημα ενός κουμπιού [12].

Τα callbacks καλούνται από το βρόχο συμβάντων (event loop). Πιο συγκεκριμένα, ο βρόχος συμβάντων είναι ένα σύνολο συμβάντων που έχουν οριστεί από την εφαρμογή της οποίας το νήμα περιμένει να εκτελεστεί. Όταν μία εφαρμογή στέλνει ένα αίτημα, ένα συμβάν διοχετεύεται και τοποθετείται στην ουρά των συμβάντων. Ο βρόχος συμβάντων λαμβάνει το συμβάν και το επεξεργάζεται. Σε περίπτωση που το συμβάν είναι αποκλεισμός (π.χ. ανάγνωση εικόνων), περνάει την επεξεργασία του σε ένα από τα ενεργά νήματά του. Όταν το ενεργό νήμα ολοκληρώσει την επεξεργασία του συμβάντος, επιστρέφει την απάντηση χρησιμοποιώντας ένα callback που διαβιβάστηκε σε αυτήν με την παραπάνω διαδικασία. Τέλος, το αποτέλεσμα μεταβιβάζεται στην εφαρμογή [12] [13].





Σχήμα 3.2 Το σύστημα Node.js [12]



Σχήμα 3.3 Διαφορά λειτουργίας ενός παραδοσιακού server και ενός server Node.js [13]

### 3.2.1 NPM

Το npm (Node Package Manager) είναι ένας διαχειριστής πακέτων για τη γλώσσα προγραμματισμού Javascript. Είναι ο προεπιλεγμένος διαχειριστής πακέτων για το περιβάλλον εκτέλεσης Node.js. Αποτελείται από μία κονσόλα (command line client), όπου ο χρήστης μπορεί να εκτελεί εντολές, και μία ηλεκτρονική βάση δεδομένων, γνωστή ως μητρώο npm, όπου αποθηκεύονται δημόσια ή ιδιωτικά επί πληρωμή πακέτα. Στο μητρώο υπάρχουν αυτή τη στιγμή παραπάνω από 800.000 πακέτα κώδικα. Οι χρήστες έχουν πρόσβαση στο μητρώο μέσω του περιηγητή και έχουν την δυνατότητα να αναζητήσουν και να κατεβάσουν τα επιθυμητά πακέτα μέσω της σελίδας [www.npmjs.com](http://www.npmjs.com). Το npm εγκαθίσταται ταυτόχρονα με την εγκατάσταση του Node.js στον υπολογιστή [14]. Όλα τα

πακέτα ορίζονται σε αρχεία που ονομάζονται `package.json`, των οποίων το περιεχόμενο πρέπει να είναι γραμμένο σε μορφή JSON.

Η εισαγωγή ενός πακέτου στο μητρώο για αρκετό καιρό γινόταν χωρίς εξέταση της ποιότητας και της ασφάλειας των πακέτων, γεγονός που εγκυμονούσε κινδύνους κακοβουλίας και χαμηλής ποιότητας για τους χρήστες των πακέτων. Για αυτό τον λόγο, στην έκτη έκδοση του npm προστέθηκε έλεγχος για να κάνει πιο εύκολο στους προγραμματιστές τον εντοπισμό και την διόρθωση των προβλημάτων ευπάθειας και ασφάλειας στα εγκατεστημένα πακέτα [14].

### 3.2.2 Express

Η Express είναι ένα λιτό και ευέλικτο framework του Node.js, το οποίο παρέχει λειτουργίες για την υλοποίηση web applications ή εφαρμογών κινητών τηλεφώνων. Αποτελεί πλέον ένα από τα πιο δημοφιλή frameworks για την ανάπτυξη διακομιστών, αλλά είναι επίσης και υποκείμενη βιβλιοθήκη για πολλά ακόμη Node frameworks.

Μερικά από τα χαρακτηριστικά της Express είναι ότι επιτρέπει την ρύθμιση “middleware” για να απαντώνται τα αιτήματα HTTP, επιτρέπει τη δυναμική εκτέλεση των σελίδων HTML περνώντας arguments, χρησιμοποιεί ασύγχρονα APIs, αλληλοεπιδρά με τη βάση δεδομένων, διαχειρίζεται σφάλματα, ορίζει ένα πίνακα δρομολόγησης ο οποίος χρησιμοποιείται για την εκτέλεση διαφορετικών ενεργειών με βάση τη μέθοδο HTTP και τη διεύθυνση URL και πολλά άλλα.

“Middleware” στην Express είναι οι συναρτήσεις της μορφής function (req, res, next). Req είναι το αίτημα HTTP που στέλνεται από τον περιηγητή, res είναι η απάντηση από τον διακομιστή Express και next είναι ένα callback, δηλαδή μία λειτουργία επεξεργασίας που επαναλαμβάνεται. Οι middleware συναρτήσεις αφού εκτελεστούν και στείλουν απάντηση στο αίτημα καλούν την επόμενη middleware συνάρτηση [15].

Κάθε εφαρμογή λαμβάνει αιτήματα HTTP και βάσει του URL και ίσως και άλλων πληροφοριών που περιέχονται στα δεδομένα που αποστέλλονται και καθορίζει ποια

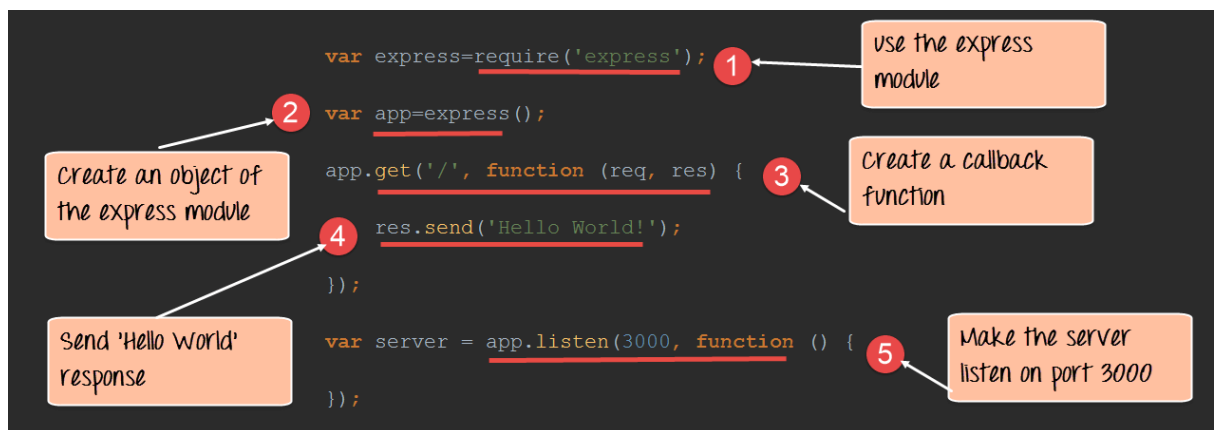
πρέπει να είναι η δράση της. Αναλόγως λοιπόν με τη δράση που απαιτείται καλείται να ικανοποιήσει το αίτημα αλληλοεπιδρώντας με τη βάση δεδομένων ή εκτελώντας άλλες εργασίες που απαιτούνται. Η εφαρμογή θα επιστρέψει το αποτέλεσμα δημιουργώντας μια δυναμική HTML σελίδα ως απάντηση στον περιηγητή.

Η Express καταφέρνει να εκτελεί όλη αυτή την διαδικασία με πολύ απλό τρόπο, δημιουργώντας διαδρομές με πολύ απλή δομή. Συνδυάζει ένα HTTP αίτημα και μία διαδρομή. Το αίτημα είναι μία από τις μεθόδους GET, POST, PUT και DELETE και η διαδρομή είναι η τοποθεσία της πηγής. Μια βασική διαδρομή ακολουθώντας το παρακάτω μοντέλο.

app.METHOD (PATH, HANDLER) όπου:

- η εφαρμογή είναι ένα αντικείμενο της Express
- METHOD είναι μια μέθοδος αιτήματος HTTP
- PATH είναι μια διαδρομή
- HANDLER είναι η λειτουργία που εκτελείται όταν ταιριάζει η διαδρομή [15].

Στην παρακάτω εικόνα μπορούμε να δούμε πως γίνεται χρήση της Express, δημιουργώντας ένα αντικείμενο και πως στέλνουμε απάντηση σε ένα αίτημα GET.



Σχήμα 3.4 Κώδικας σε express, δημιουργία αντικειμένου

### 3.2.3 Socket.io

Το socket.io είναι μια βιβλιοθήκη της JavaScript για web applications πραγματικού χρόνου και αποτελεί μία από τις πιο πολύτιμες βιβλιοθήκες για τους προγραμματιστές που χρησιμοποιούν Node.js. Αποτελείται από δύο κομμάτια, το κομμάτι του διακομιστή (για το Node.js) και το κομμάτι του πελάτη (για τον περιηγητή). Και τα δύο έχουν παρόμοια APIs και στηρίζονται στην κατευθυνόμενη από συμβάντα αρχιτεκτονική. Υποστηρίζει την αμφίδρομη επικοινωνία μεταξύ πελατών και διακομιστών σε πραγματικό χρόνο, επιτρέποντας την δημιουργία και τον έλεγχο μιας υπηρεσίας συνομιλίας σε μια εφαρμογή. Το socket.io κάνει χρήση του πρωτόκολλου WebSocket και έχει κάποια επιπλέον χαρακτηριστικά, όπως η δυνατότητα να αποστέλλει δεδομένα σε ένα μεγάλο αριθμό υποδοχών την ίδια στιγμή και τη δυνατότητα να αποθηκεύει δεδομένα. Μπορεί να εγκατασταθεί με το εργαλείο npm [16].

### 3.3 React.js

Η React είναι μία βιβλιοθήκη της JavaScript και χρησιμοποιείται για τη δημιουργία διεπαφής του χρήστη με μία εφαρμογή, δηλαδή για τη σχεδίαση του frontend μέρους της εφαρμογής. Η ιστορία της React ξεκινάει το 2011, όταν ο Jordan Walke, ένας μηχανικός λογισμικού της εταιρείας Facebook άρχισε να δουλεύει πάνω στη React επηρεασμένος από το XHP, ένα framework για την PHP. Η Facebook χρησιμοποίησε πρώτη τη React και το 2012 την ακολούθησε και η εταιρεία Instagram [17]. Η επέκταση της τα τελευταία χρόνια ήταν τεράστια καθώς χρησιμοποιείται πλέον από τα μεγαλύτερα ονόματα του Internet, όπως είναι το Netflix, το Dropbox, το Whatsapp και οι The New York Times [18].

Η React είναι βασισμένη στην ιδέα ότι αποτελείται από τον συνδυασμό διαφορετικών components, δηλαδή διαφορετικών συστατικών στοιχείων. Η προσέγγιση των components έχει ως μεγάλο πλεονέκτημα ότι μπορούμε να έχουμε τη δημιουργία μεγάλων εφαρμογών, καθώς οι πιθανές αλλαγές που συμβαίνουν σε κάποιο component δεν επηρεάζουν την υπόλοιπη εφαρμογή. Επίσης, τα components μπορούν να ξαναχρησιμοποιηθούν σε διαφορετικά κομμάτια της εφαρμογής, επιτυγχάνοντας με αυτό τον τρόπο λιγότερες γραμμές κώδικα και λιγότερο χρόνο ανάπτυξης. Η React δίνει επίσης

στις εφαρμογές το χαρακτηριστικό του πραγματικού χρόνου. Κάθε φορά που ο χρήστης αλληλοεπιδρά με ένα component και τα δεδομένα αλλάζουν, προστίθενται ή ενημερώνονται, το component ξαναεκτελείται αυτόματα για να εμφανιστεί η νέα αλλαγή, χωρίς να επαναφορτίζεται ολόκληρη η εφαρμογή [19].

Ο κώδικας της React μπορεί να συνδυάσει με απλό και ομαλό τρόπο και τις τρεις γνωστές frontend τεχνολογίες HTML, CSS και Javascript, χωρίς να χρειάζεται η εισαγωγή εξωτερικών αρχείων κώδικα.

Ένα ακόμη χαρακτηριστικό της React είναι ο ασυγχρονισμός. Τα components της εφαρμογής συνεχίζουν να δουλεύουν και να φορτώνουν ακόμη και εάν κάποιο κομμάτι δεν έχει φορτώσει. Δεν παρατηρείται το φαινόμενο του αποκλεισμού λειτουργιών [19].

### **3.3.1 Virtual DOM**

Κάθε φορά που μία σελίδα που δεν χρησιμοποιεί React, φορτώνει, προσθέτοντας ή αφαιρώντας δυναμικά δεδομένα στο backend, δημιουργείται από τον περιηγητή ένα αντικείμενο το οποίο ονομάζεται DOM (Document Object Model). Για κάθε αλλαγή δημιουργείται ένα καινούριο DOM για την ίδια σελίδα, με αποτέλεσμα να έχουμε σπατάλη μνήμης και μειωμένη απόδοση. Σε αυτό το πρόβλημα έρχεται η React να δώσει τη λύση με τη χρήση του Virtual DOM.

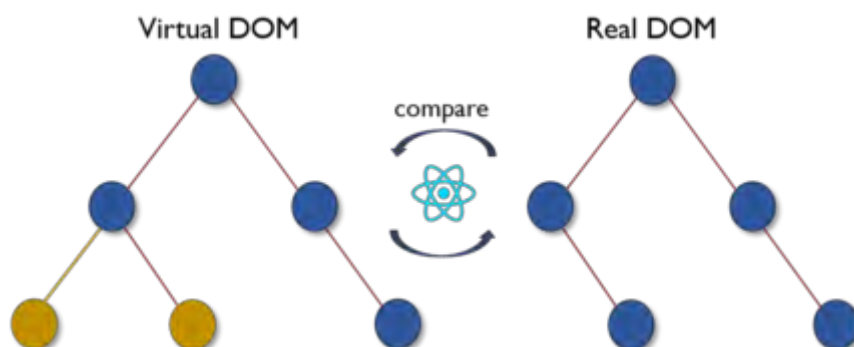
Το Virtual DOM, όπως και το πραγματικό DOM είναι ένα δέντρο που περιέχει τα δεδομένα, τα χαρακτηριστικά, το περιεχόμενο τους ως αντικείμενα και τις ιδιότητές τους. Το δέντρο ενημερώνεται όταν τα δεδομένα αλλάζουν κάθε φορά που ο χρήστης αλληλοεπιδρά με την εφαρμογή. Το Virtual DOM λειτουργεί σε τρία βήματα [19]:

1. Κάθε φορά που έχουμε μεταβολή των δεδομένων, όλη η διεπαφή αναπαρίσταται με ένα καινούριο Virtual DOM.



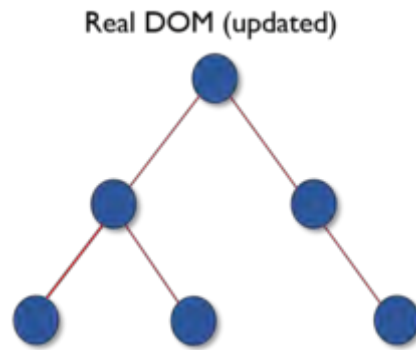
Σχήμα 3.5 Αναπαράσταση διεπαφής σε ένα Virtual DOM [19]

2. Στη συνέχεια υπολογίζεται η διαφορά μεταξύ της προηγούμενης αναπαράστασης DOM και της νέας.



Σχήμα 3.6 Σύγκριση Virtual DOM και πραγματικού DOM [19]

3. Αφού υπολογιστούν οι διαφορές, το πραγματικό DOM θα ενημερωθεί μόνο με τις αλλαγές που έχουν γίνει, εξοικονομώντας χρόνο και μνήμη.



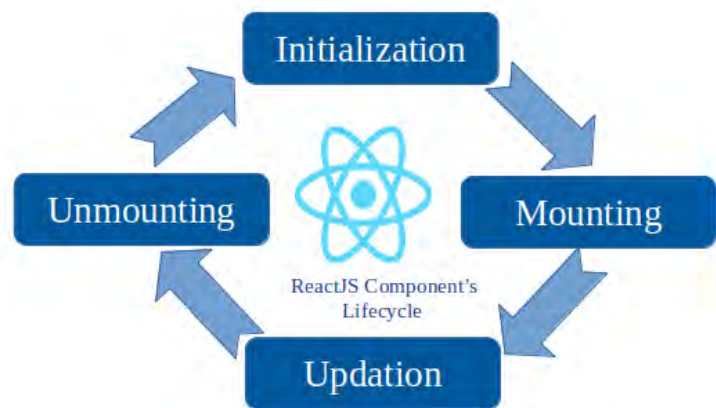
Σχήμα 3.7 Τελική μορφή του πραγματικό DOM [19]

### 3.3.2 Lifecycle Methods

Τα components της React ακολουθούν και αυτά τον κύκλο της ζωής όπως όλα τα πράγματα στον κόσμο μας. Τα components λοιπόν δημιουργούνται, αναπτύσσονται και αναβαθμίζονται και στο τέλος καταστρέφονται - πεθαίνουν. Για τις διαφορετικές φάσεις που περνάει ένα component υπάρχουν και διαφορετικοί μέθοδοι, μέσω των οποίων ελέγχονται και διαχειρίζονται τα components.

Οι φάσεις ζωής των components της React μπορούν να χωριστούν σε κατηγορίες [20]:

- Την αρχικοποίηση, κατά την οποία ορίζονται οι καταστάσεις (states) και τα props μιας κλάσης μέσα στον κατασκευαστή των κλάσεων .
- Την ανάπτυξη, κατά την οποία το component δημιουργείται και εισάγεται στο DOM.
- Την αναβάθμιση, κατά την οποία τα στοιχεία του component, δηλαδή states και props ενημερώνονται έπειτα από τα συμβάντα που δημιουργεί ο χρήστης. Αυτό έχει ως αποτέλεσμα το component να ξαναφτιαχτεί.
- Και την αποσυναρμολόγηση, κατά την οποία το component καταστρέφεται απο το DOM.



Σχήμα 3.8 ReactJS Component's Lifecycle [20]

Η πιο δημοφιλής μέθοδος κύκλου ζωής είναι η **render()**. Είναι η μόνη απαραίτητη μέθοδος για τις κλάσεις των components της React. Ελέγχει τη δημιουργία του component και η χρήση της συμβάλλει στη φάση της ανάπτυξης και της αναβάθμισης του component.

Επίσης, δημοφιλής είναι και η μέθοδος **componentDidMount()**, η οποία όπως είναι κατανοητό και από την ονομασίας της, καλείται αφού το component έχει δημιουργηθεί. Χρησιμοποιείται συνήθως για να καλέσουμε ένα API και να φορτώσουμε δεδομένα από έναν απομακρυσμένο διακομιστή. Στη μέθοδο `componentDidMount()` επιτρέπεται η κλήση της συνάρτησης `setState()`, έτσι ώστε να έχουμε ενημέρωση των states και να προκληθεί αυτόματα νέα απόδοση του component (re-render).

Μετά την αναβάθμιση, η πιο συνηθισμένη μέθοδος που χρησιμοποιείται είναι η **componentDidUpdate()**, η οποία συνηθίζεται να καλείται όταν θέλουμε την ενημέρωση του DOM μετά από αλλαγές των props και state. Και αυτή η μέθοδος επιτρέπει να γίνεται χρήση της `setState()` και να έχουμε αλλαγές στις καταστάσεις του component.

Τέλος, όταν το component αποσυναρμολογείται και καταστρέφεται, συνηθίζεται να κάνουμε χρήση της μεθόδου **componentWillUnmount()**, η οποία βοηθά να πραγματοποιηθούν ενέργειες καθαρισμού, όπως ακύρωση κλήσεων API, εκκαθάριση αποθηκευμένων κρυφών μηνύων και χρονομετρητών. Αυτή η μέθοδος δεν επιτρέπει την χρήση της `setState()` και την αλλαγές στις καταστάσεις του component [20].



### 3.3.3 JSX

Η JSX (JavaScript XML) είναι μία επέκταση της γλώσσας JavaScript, την οποία χρησιμοποιεί η React. Η JSX μοιάζει σε εμφάνιση με την HTML. Φτιάχτηκε έτσι ώστε η XML / HTML να συνυπάρχει με τη JavaScript και να ενσωματώνει τον κώδικα HTML μέσα στον κώδικα της JavaScript. Τα στοιχεία της εφαρμογής React γράφονται συνήθως σε JSX, αλλά υπάρχει η δυνατότητα να γραφτούν και σε καθαρή JavaScript και να ενσωματωθούν. Η React χρησιμοποιεί τον JavaScript μεταγλωττιστή Babel για να μετατρέψει την JSX σε JavaScript.

Είναι πολύ σημαντικό επίσης να σημειώσουμε ότι η JSX έχει αρκετά καλή απόδοση. Σε σύγκριση με την JavaScript, ένας κώδικας JSX τρέχει γρηγορότερα. Σύμφωνα με έρευνα, έχει αποδειχθεί ότι είναι γρηγορότερος κατά 12% όταν τρέχει σε iOS και 29% σε Android [21].

Εκτός από την ταχύτητα του κώδικα JSX, αυτό που βοηθά στην αύξηση της δημοφιλίας του είναι η ασφάλεια του κώδικα, καθώς η JSX προσφέρει δυνατότητες αποσφαλμάτωσης (debugging) κατά την μεταγλώττιση, και η ευκολία των προγραμματιστών να προσαρμοστούν, αφού η JSX ακολουθεί ένα μοντέλο βασισμένο σε κλάσεις όπως και η JAVA και τα χαρακτηριστικά της είναι παρόμοια με την JavaScript.

Ένα παράδειγμα κώδικα JSX βλέπουμε στην παρακάτω εικόνα.

```
function ComponentExample(props) {
  return <h1>{props.text}</h1>;
}

function App() {
  return (
    <div>
      <ComponentExample text="Let's see an example, shall we?" />
    </div>
  );
}

ReactDOM.render(
  <App />,
  document.getElementById('root')
);
```

Σχήμα 3.9 Παράδειγμα κώδικα JSX

### 3.4 PostgreSQL

Η PostgreSQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Είναι ελεύθερο και ανοικτού κώδικα, βασισμένο στην επεκτασιμότητα και αποτελεί ένα από τα πιο δημοφιλέστερα συστήματα διαχείρισης βάσεων δεδομένων παγκοσμίως για την αξιοπιστία και την ακεραιότητα των δεδομένων. Τρέχει σε όλα τα βασικά λειτουργικά συστήματα και είναι προεπιλεγμένη βάση δεδομένων για τον MacOS Server [22].

Η PostgreSQL ήταν εξέλιξη ενός πρότζεκτ της ομάδας Ingres του πανεπιστημίου της Καλιφόρνιας Berkeley. Το 1982 ο αρχηγός της ομάδας Ingres αποχώρησε και το 1985 επέστρεψε για να ξεκινήσει ένα νέο πρότζεκτ, το επόμενο σε σειρά μετά το πρότζεκτ με την ομάδα Ingres (post-ingres project), με θέμα την αντιμετώπιση των προβλημάτων των σύγχρονων βάσεων δεδομένων. Πιο συγκεκριμένα στόχος του νέου πρότζεκτ ήταν να προσθέσει χαρακτηριστικά για να υποστηρίζονται όλοι οι τύποι δεδομένων [23].

Αυτό που ξεχωρίζει την PostgreSQL από τις άλλες σχεσιακές βάσεις δεδομένων είναι ότι περιέχει και κάποιες έννοιες από τον αντικειμενοστραφή προγραμματισμό, συμπεριλαμβανομένων των αντικειμένων που ο ίδιος χρήστης ορίζει και της κληρονομικότητας των πινάκων. Ο χρήστης μπορεί να ορίζει αντικείμενα χωρίς να ακολουθεί τα βασικά αντικείμενα που είναι οι πίνακες, τα ευρετήρια και οι τύποι δεδομένων. Μέσω της κληρονομικότητας δίνεται η δυνατότητα να δημιουργούνται πίνακες παιδιά, οι οποίοι κληρονομούν τις ιδιότητες των γονέων και μπορούν να προσθέτουν και άλλες [24]. Αυτές οι δυνατότητες της PostgreSQL την κάνει κατάλληλη για διαχείριση σύνθετων εργασιών μεγάλου όγκου. Για αυτό το λόγο έχει επιλεγθεί από μεγάλες εταιρείες όπως η Facebook, η Skype, η Apple, η Instagram και η Yahoo.

Η σημερινή μορφή την PostgreSQL υποστηρίζει πλήθος λειτουργιών όπως συναρτήσεις, δείκτες, κανόνες, ορισμένους τύπους δεδομένων, καθώς και λειτουργίες κληρονομικότητας χαρακτηριστικών πινάκων, όψεις, συναλλαγές, λειτουργίες κρυπτογράφησης, αποθήκευσης μεγάλων αντικειμένων και πολλές άλλες [25].

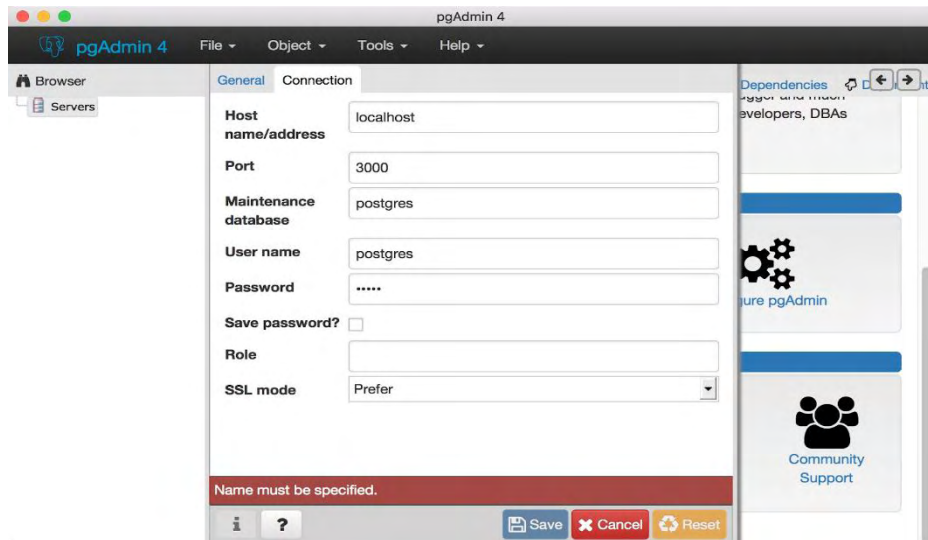
Ένα ακόμη χαρακτηριστικό που κάνει την PostgreSQL ξεχωριστή ανάμεσα στα άλλα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων είναι οι υποστηριζόμενοι τύποι δεδομένων. Σε αντίθεση με τα άλλα συστήματα η PostgreSQL έχει τη δυνατότητα να προσθέτει ή να αλλάζει τους τύπους δεδομένων. Μπορούν δηλαδή οι χρήστες να συνδυάσουν διαφορετικούς βασικούς τύπους δημιουργώντας καινούριους.

Στα βασικά συστήματα διαχείρισης βάσεων δεδομένων είναι αδύνατη η ταυτόχρονη ανάγνωση και γραφή στη βάση δεδομένων και επιβάλλουν την πολιτική κλειδώματος. Στην PostgreSQL δίνει τη δυνατότητα σε πολλαπλούς χρήστες να διαβάσουν και να γράψουν την ίδια στιγμή στη βάση, χρησιμοποιώντας “στιγμιότυπα” (snapshots), τα οποία περιγράφουν την κατάσταση της βάσης τη συγκεκριμένη στιγμή.

#### **3.4.1 pgAdmin**

Το pgAdmin είναι ένα ελεύθερο και ανοικτού κώδικα εργαλείο διαχείρισης για την PostgreSQL. Το pgAdmin υποστηρίζεται από τα περισσότερα βασικά λειτουργικά και εγκαθίσταται στον υπολογιστή του χρήστη ταυτόχρονα με τη βάση δεδομένων. Το pgAdmin III ήταν γραμμένο σε C++ και με τη χρήση του wxWidgets, ενός framework που είναι και αυτό συμβατό με τα βασικά λειτουργικά, αλλά πλέον διαθέσιμο από το 2016 είναι το pgAdmin4, που είναι γραμμένο σε Python, χρησιμοποιώντας Flask και Qt framework [22].

Για την χρήση του pgAdmin χρειάζεται η σύνδεση του διακομιστή μας με την βάση δεδομένων επιλέγοντας χρήστη και κωδικό κατά την εγκατάσταση της βάσης. Μέσω του pgAdmin μπορούμε να εκτελέσουμε όλες τις εντολές που είναι διαθέσιμες στην PostgreSQL, όπως να φτιάξουμε πίνακες, να δηλώσουμε τύπους δεδομένων, να διαγράψουμε δεδομένα και πολλά άλλα.



Σχήμα 3.10 Σύνδεση pgAdmin με τον server

### 3.4.2 Knex.js

Το Knex.js είναι ένα JavaScript εργαλείο δημιουργίας ερωτημάτων SQL για τις σχεσιακές βάσεις δεδομένων PostgreSQL, MSSQL, MySQL, MariaDB, SQLite3, Oracle, and Amazon Redshift [26]. Δημιουργήθηκε για να μειώσει την πολυπλοκότητα που προκύπτει από την συντήρηση και αναζήτηση στη βάση δεδομένων. Τα χαρακτηριστικά που διαθέτει το Knex.js είναι η δυνατότητα δημιουργίας ολοκληρωμένων ερωτημάτων και σχημάτων, η υποστήριξη συναλλαγών, η ύπαρξη συνηθισμένων επανακλήσεων (callbacks) και η δυνατότητα συγκέντρωσης συνδέσεων και συγκεκριμένων απαντήσεων ανάμεσα σε διαφορετικά ερωτήματα των διακομιστών και των περιηγητών [26].

Το Knex.js εγκαθίσταται όπως εγκαθίστανται όλα τα πακέτα npm. Μετά την εγκατάσταση δημιουργείται ένα αρχείο με το όνομα knexfile.js, το οποίο τροποποιούμε βάζοντας τα στοιχεία της βάσης μας για να τη συνδέσουμε με το Knex.js.

```
module.exports = {
  development: {
    client: 'postgresql',
    connection: {
      host: '192.168.1.60',
      database: 'payroll',
      user: 'postgres',
      password: 'abc'
    }
  },
  production: {
    client: 'postgresql',
    connection: {
      database: 'my_db',
      user: 'username',
      password: 'password'
    },
    pool: {
      min: 2,
      max: 10
    },
    migrations: {
      tableName: 'knex_migrations'
    }
  }
};
```

**knex init**  
creates file **knexfile.js**

**Edit knexfile.js to**  
**connect to your**  
**Postgresql database**

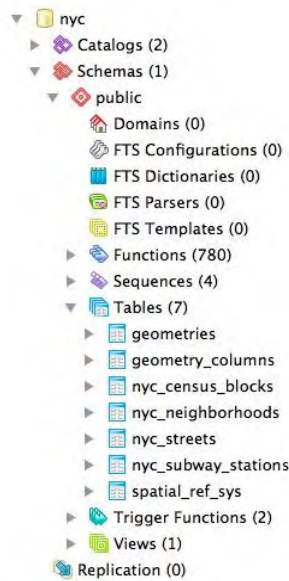
knex migrate will write **public.knex\_migrations**

Σχήμα 3.11 Αρχείο knexfile.js – Διαχείριση σύνδεσης με τη βάση postgresQL

### 3.4.3 Schema

Κάθε βάση δεδομένων είναι απαραίτητο να περιέχει τουλάχιστον ένα σχήμα, το οποίο περιέχει όλα τα αντικείμενα της βάσης, όπως πίνακες, όψεις, αποθηκευμένες διαδικασίες, ενεργοποιητές και άλλα. Μία βάση μπορεί να περιέχει πολλά σχήματα, αλλά τα ονόματα των σχημάτων πρέπει να είναι μοναδικά. Με την δημιουργία μιας καινούριας βάσης δεδομένων, δημιουργείται ένα προεπιλεγμένο σχήμα με το όνομα "public".

Η χρήση σχημάτων είναι απαραίτητη καθώς τα δεδομένα όντας οργανωμένα σε ένα σχήμα κάνει την εφαρμογή μαζικών ενεργειών πολύ πιο εύκολη. Μας δίνει ακόμη την δυνατότητα να δημιουργούμε εύκολα αντίγραφα ασφαλείας δεδομένων, καθώς επίσης η ύπαρξη σχημάτων επιτρέπει σε πολλούς χρήστες να χρησιμοποιούν μία βάση δεδομένων χωρίς να αλληλοεπιδρούν μεταξύ τους.



Σχήμα 3.12 Παράδειγμα σχήματος στο pgAdmin4

### 3.5 Εργαλεία

Για την ανάπτυξη, τον έλεγχο και την δημιουργία αντιγράφων ασφαλείας χρησιμοποιήθηκαν τα εξής εργαλεία :

1. **Sublime Text** : Είναι ένα εξελιγμένο πρόγραμμα για εγγραφή και επεξεργασία κώδικα. Το Sublime Text διαθέτει ένα ισχυρό Python API που επιτρέπει στα plugins να αυξήσουν την ενσωματωμένη λειτουργικότητα. Επίσης παρέχει μεγάλο εύρος πακέτων, που δημιουργήθηκαν από την κοινότητα. Υποστηρίζει εγγενώς πολλές γλώσσες προγραμματισμού και markup languages και οι λειτουργίες μπορούν να προστεθούν από χρήστες με plugins. Παρέχει μια γρήγορη πρόσβαση σε αρχεία , γραμμές ή σύμβολα. Χαρακτηριστικά ακόμα του Sublime Text, είναι αυτόματη αποθήκευση των προγραμμάτων , η οποία προσπαθεί να αποτρέψει τους χρήστες να χάσουν τη δουλειά τους , καθώς και η λειτουργία ορθογραφικού ελέγχου , η οποία διορθώνει καθώς ο χρήστης πληκτρολογεί [27].

Το Sublime Text επιλέχθηκε για την εγγραφή και σύνταξη του κώδικά, καθώς υποστηρίζει όλες τις γλώσσες προγραμματισμού χρειάστηκαν για την εφαρμογή και κάλυψε όλες τις απαιτήσεις και λειτουργίες της εργασίας.

2. **Postman**: Το Postman είναι το μόνο οργανωμένο API περιβάλλον. Μέσω αυτού ο χρήστης μπορεί να σχεδιάζει, να εντοπίζει σφάλματα, να δοκιμάζει και να δημοσιεύει όλα τα API του σε ένα μόνο σημείο. Αυτό το API του δίνει τη δυνατότητα να έχει πρόσβαση στα δεδομένα του, που είναι αποθηκευμένα στον λογαριασμό του στο Postman με ευκολία. Ένα από τα βασικά σημεία που χρειάζεται να γνωρίζει ο χρήστης σχετικά με το Postman API , είναι το API KEY , το οποίο είναι απαραίτητο για την σύνδεση με τον διακομιστή. Ο χρήστης μπορεί να λάβει οποιαδήποτε δεδομένα έχει άδεια πρόσβασης στο Postman με ένα κλειδί API που δημιουργείται από αυτόν [28] [29].

Η χρήση του Postman κατέσπει απαραίτητη για την ανάπτυξη του backend κομματιού. Χρησιμοποιήθηκε στον έλεγχο της λειτουργίας του Node.js server μέσω της εκτέλεσης ερωτημάτων χωρίς να είναι απαραίτητη η αλληλεπίδραση με το frontend κομμάτι.

3. **GitHub** : Είναι μια αμερικάνικη εταιρία, θυγατρική της Microsoft από το 2018, που παρέχει λειτουργίες για τους χρήστες ώστε να αποθηκεύουν, να επεξεργάζονται και να ελέγχουν τον κώδικα τους , χρησιμοποιώντας το Git. Το Git είναι ένα κατακευματισμένο σύστημα ελέγχου ανοικτού κώδικα με σκοπό τον εντοπισμό αλλαγών στον πηγαίο κώδικα ή σε οποιοδήποτε αρχείο. Σκοπός του είναι η ταχύτητα, η ακεραιότητα δεδομένων και η υποστήριξη για κατακευματισμένες , μη γραμμικές ροές εργασίας. Το GitHub λοιπόν, προσφέρει όλες τις λειτουργίες και δυνατότητες του Git για τη διαχείριση κατακευματισμένων εκδόσεων και διαχείρισης πηγαίου κώδικα , καθώς έχει και κάποια δικά του πρόσθετα χαρακτηριστικά. Παρέχει έλεγχο πρόσβασης και πολλές λειτουργίες , όπως παρακολούθηση και έλεγχο σφαλμάτων. Λογαριασμοί στο GitHub δημιουργούνται επίσης για να φιλοξενήσουν έργα ανοικτού κώδικα. Αρκεί έτσι ο χρήστης να δημιουργήσει στην εταιρία έναν λογαριασμό [30] [31].

Πιο συγκεκριμένα, η χρήση του GitHub αρχικά έγινε με σκοπό να υπάρχουν αντίγραφα ασφαλείας των αρχείων της εφαρμογής και για την οργανωμένη και άμεση διαχείριση των αρχείων. Μέσω του GitHub έγινε εύκολος επίσης ο διαμοιρασμός του κώδικα της εφαρμογής με άλλους χρήστες της πλατφόρμας.

## ΚΕΦΑΛΑΙΟ 4

### ΛΕΙΤΟΥΡΓΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

#### 4.1 Λειτουργίες Εφαρμογής

Η εφαρμογή όπως αναφέρθηκε έχει ως στόχο την διευκόλυνση των χρηστών να βρουν μία διαδρομή που επιθυμούν με έναν άλλον χρήστη ή να προσφέρουν μια διαδρομή με το δικό τους όχημα. Η εφαρμογή έχει ονομαστεί Thumbs Up. Ο χρήστης έχει την δυνατότητα να αναζητήσει μία διαδρομή επιλέγοντας τα κριτήρια που επιθυμεί και να δει τις διαθέσιμες διαδρομές σύμφωνα με τα κριτήριά του. Εφόσον βρει μία διαδρομή που τον ενδιαφέρει έχει την δυνατότητα να στείλει μήνυμα στον χρήστη που προσφέρει την διαδρομή και να επικοινωνήσει μαζί του για την περαιτέρω συνεννόηση που απαιτείται. Εάν ο χρήστης δεν ψάχνει για μία διαδρομή, αλλά θέλει να προφέρει το αυτοκίνητό του για μετακίνηση, πρέπει να έχει λογαριασμό στην εφαρμογή. Καλείτε λοιπόν να συνδεθεί στο λογαριασμό της εφαρμογής ή να δημιουργήσει έναν νέο, εάν δεν υπάρχει. Αφού λοιπόν συνδεθεί, μπορεί να συμπληρώσει τα στοιχεία της διαδρομής που προσφέρει και να την δημοσιεύσει. Η εφαρμογή προτείνει στον χρήστη ακόμη μία τιμή για την διαδρομή που προσφέρει σύμφωνα με το κόστος των καυσίμων και το κόστος των διοδίων της διαδρομής. Οι λειτουργίες της εφαρμογής συνοπτικά είναι οι εξής:

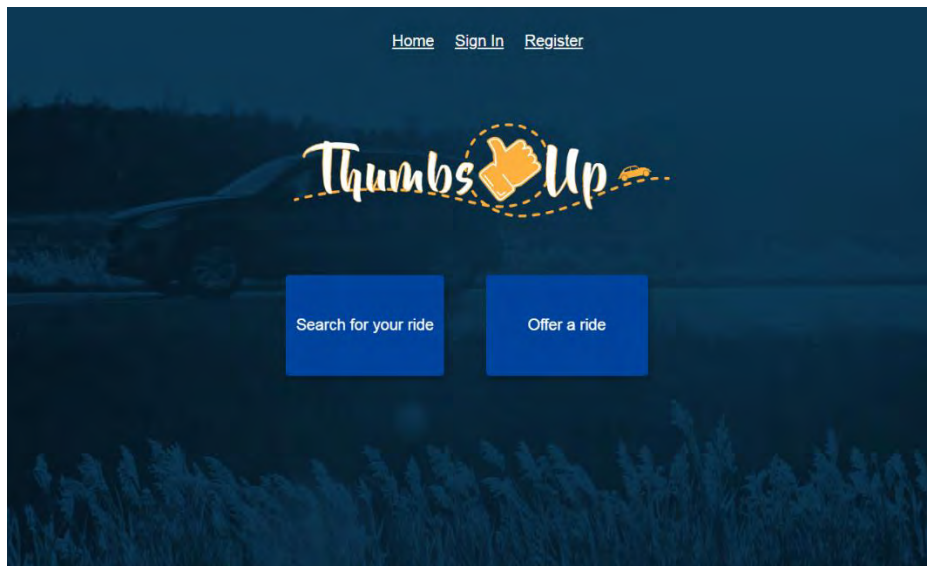
- Σύνδεση σε λογαριασμό της εφαρμογής
- Δημιουργία νέου λογαριασμού
- Επιλογή Αναζήτησης ή Προσφοράς διαδρομής
- Αναζήτηση διαδρομής
- Προσφορά διαδρομής
- Εμφάνιση προσφερόμενων διαδρομών
- Εμφάνιση λεπτομερειών προσφερόμενης διαδρομής
- Επικοινωνία με άλλο χρήστη
- Εμφάνιση προφίλ χρήστη
- Εμφάνιση μηνυμάτων χρήστη



- Διαγραφή προσφερόμενης διαδρομής
- Αποσύνδεση λογαριασμού

#### 4.1.1 Αρχική Σελίδα

Κατά την φόρτωση της εφαρμογής από τον χρήστη εμφανίζεται η κεντρική σελίδα, από την οποία ο χρήστης μπορεί να δει το लोगό και κάποιες πληροφορίες για την εφαρμογή και κυρίως να επιλέξει εάν θέλει να αναζητήσει ή να προσφέρει μία διαδρομή. Επίσης, ο χρήστης μπορεί να επιλέξει να συνδεθεί στο λογαριασμό του, να δημιουργήσει έναν νέο λογαριασμό.



Σχήμα 4.1 Screenshot Αρχικής Σελίδας

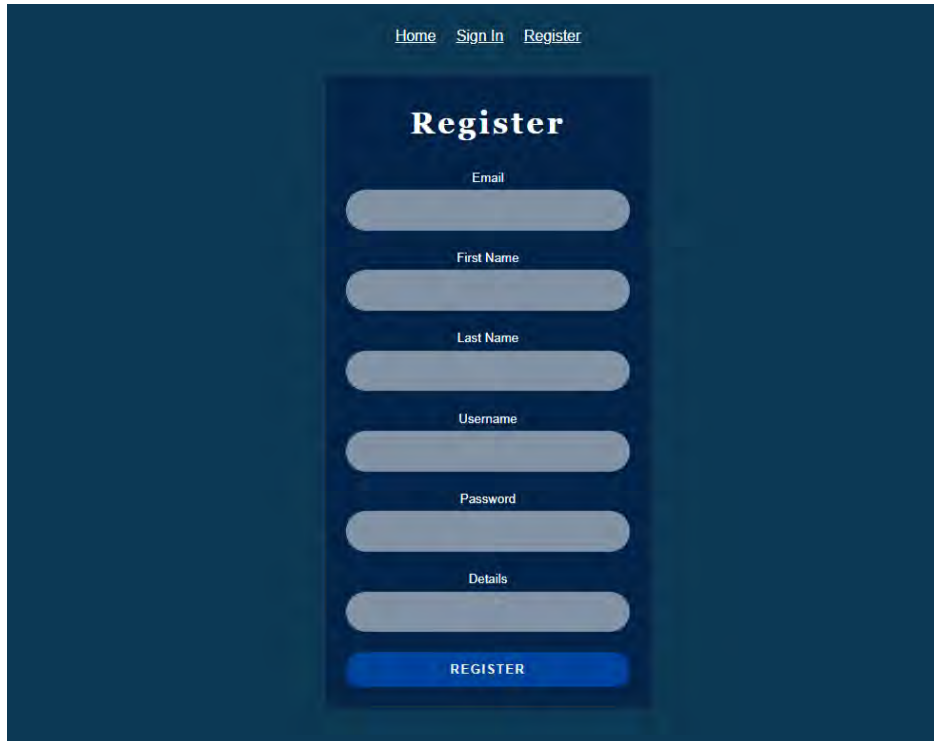
#### 4.1.2 Εγγραφή Χρήστη

Για να προσφέρει κάποιος χρήστης μία διαδρομή με το όχημά του είναι απαραίτητο να έχει δημιουργήσει λογαριασμό στην εφαρμογή και να έχει συνδεθεί. Σύνδεσμοι για να που αποστέλλουν τον χρήστη στη σελίδα εγγραφής βρίσκονται στην κεντρική σελίδα και στην σελίδα της σύνδεσης του χρήστη.

Για την εγγραφή του χρήστη απαιτείται η συμπλήρωση κάποιων προσωπικών στοιχείων του. Αυτά είναι το μέιλ του, το όνομα και το επώνυμό του, το όνομα χρήστη και τον κωδικό με τα οποία θα ήθελε ο χρήστης να συνδέεται και άλλες πληροφορίες που επιθυμεί να εισάγει ο ίδιος.

Μετά την συμπλήρωση των στοιχείων της φόρμας εγγραφής ο χρήστης πατάει το κουμπί “Register” και τα δεδομένα καταχωρούνται στη βάση δεδομένων, εφόσον συμπληρωθούν όλα τα πεδία και δεν υπάρχει στη βάση το όνομα χρήστη που συμπληρώθηκε. Όταν ο χρήστης κάνει εγγραφή, αυτόματα συνδέεται και στην εφαρμογή και εμφανίζεται ξανά η κεντρική σελίδα.

Πρέπει να σημειωθεί ότι οι κωδικοί στη βάση δεδομένων αποθηκεύονται κρυπτογραφημένοι. Για αυτό τον σκοπό χρησιμοποιείται η βιβλιοθήκη του npm, bcrypt, η οποία κωδικοποιεί και συγκρίνει κωδικούς χωρίς να υπάρχει κίνδυνος για διαρροή των κωδικών.



The screenshot shows a registration form on a dark blue background. At the top, there are navigation links: [Home](#), [Sign In](#), and [Register](#). The main heading is "Register". Below it are several input fields: "Email", "First Name", "Last Name", "Username", "Password", and "Details". At the bottom of the form is a blue button labeled "REGISTER".

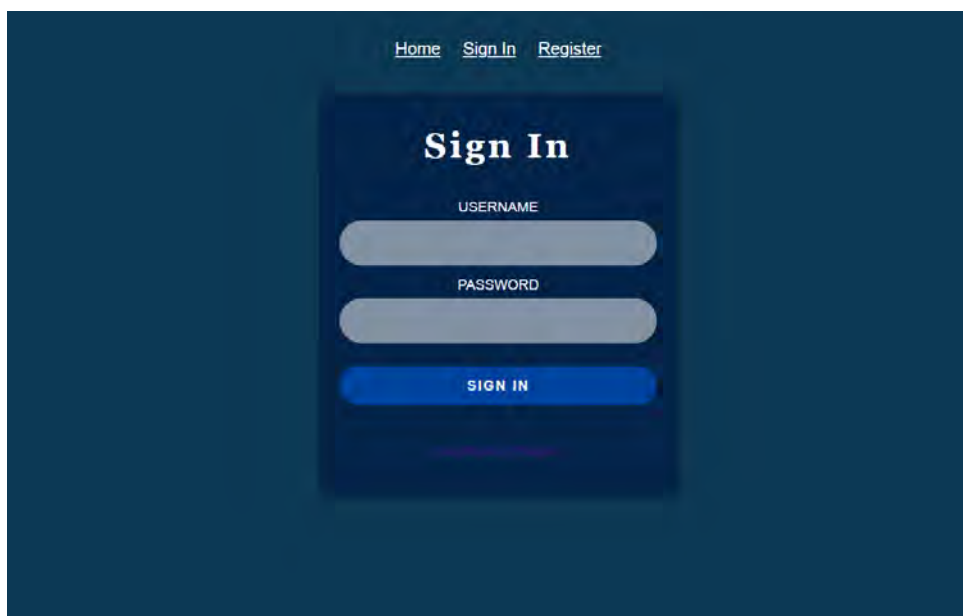
Σχήμα 4.2 Screenshot Εγγραφής Χρήστη

### 4.1.3 Σύνδεση Χρήστη

Όπως αναφέρθηκε και παραπάνω, για να προσφέρει κάποιος χρήστης μία διαδρομή με το όχημά του είναι απαραίτητο να έχει δημιουργήσει λογαριασμό στην εφαρμογή και να έχει συνδεθεί. Για να μεταφερθεί ο χρήστης στη σελίδα σύνδεσης πρέπει να ακολουθήσει τον σύνδεσμο για σύνδεση που υπάρχει στην κεντρική σελίδα.

Για την σύνδεση του χρήστη απαιτείται η συμπλήρωση του ονόματος χρήστη και του κωδικού που έχει επιλέξει ο ίδιος. Αφού συμπληρωθούν και τα δύο πεδία, ελέγχεται εάν υπάρχει στη βάση δεδομένων χρήστης με αυτά τα στοιχεία. Εάν βρεθεί ο χρήστης στη βάση δεδομένων τότε επιτυγχάνεται η σύνδεση και ο χρήστης μεταφέρεται ξανά στην κεντρική σελίδα.

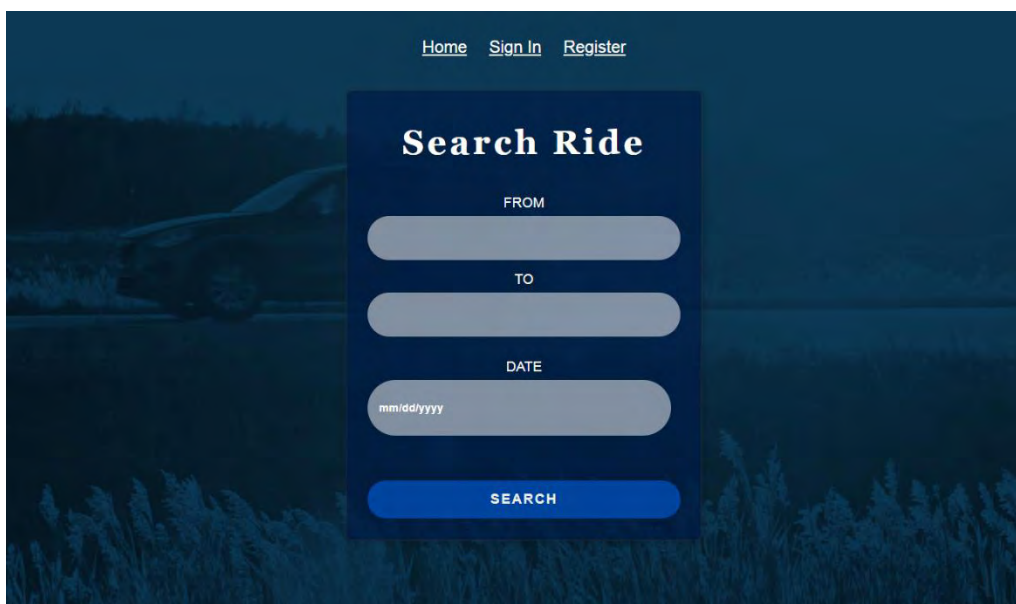
Οι κωδικοί επίσης κωδικοποιούνται με τη χρήση της βιβλιοθήκης bcrypt όπως αναφέρθηκε και στην εγγραφή του χρήστη.



Σχήμα 4.3 Screenshot Σύνδεσης Χρήστη

#### 4.1.4 Αναζήτηση Διαδρομής

Εάν ο χρήστης στην κεντρική σελίδα επιλέξει την αναζήτηση διαδρομής, θα μεταφερθεί στη σελίδα αναζήτησης, όπου καλείται να συμπληρώσει κάποια στοιχεία με βάση τα οποία θα ήθελε να αναζητήσει διαδρομές. Αυτά τα στοιχεία είναι ο χώρος αναχώρησης, ο προορισμός και η επιθυμητή ημερομηνία διαδρομής. Αν συμπληρωθούν όλα τα στοιχεία, πατώντας αναζήτηση γίνεται αναζήτηση στη βάση δεδομένων για να ελεγχθεί εάν υπάρχουν διαδρομές που καλύπτουν τα κριτήρια. Εάν βρεθούν αποτελέσματα, ο χρήστης θα μεταφερθεί στη σελίδα όπου υπάρχουν οι προσφερόμενες διαδρομές.



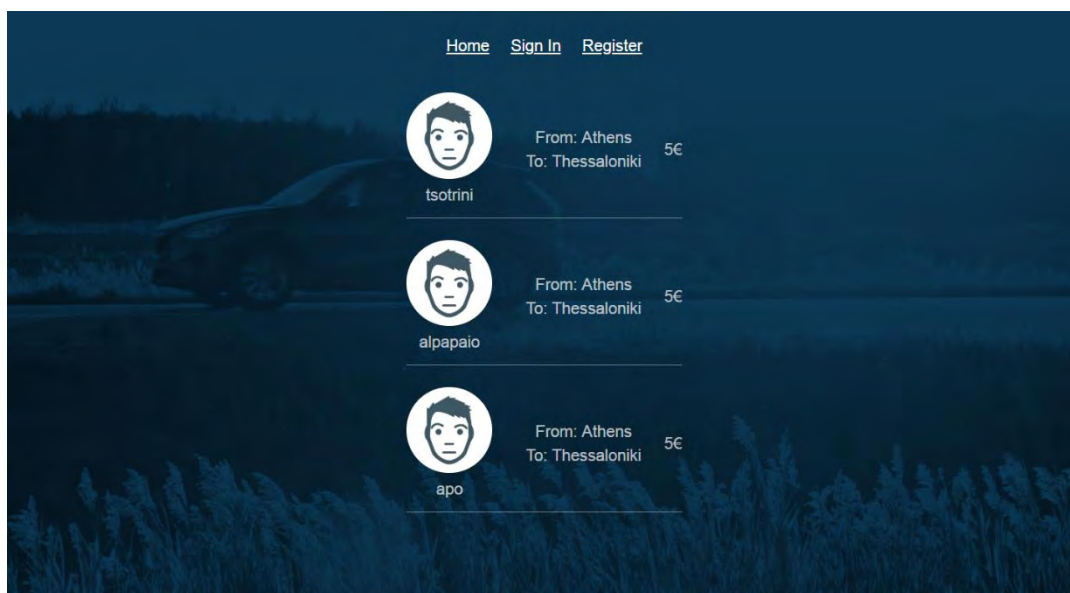
The screenshot shows a search interface for finding rides. At the top, there are navigation links: [Home](#), [Sign In](#), and [Register](#). The main heading is "Search Ride". Below the heading are three input fields: "FROM", "TO", and "DATE". The "DATE" field has a placeholder "mm/dd/yyyy". At the bottom of the form is a blue "SEARCH" button. The background of the page is a dark blue image of a landscape with a body of water and trees.

Σχήμα 4.4 Screenshot Αναζήτησης Διαδρομής

#### 4.1.5 Εμφάνιση Διαδρομών

Η σελίδα όπου γίνεται η εμφάνιση διαδρομών θα εμφανιστεί στον χρήστη μετά την αναζήτηση όπως αναφέρθηκε και παραπάνω. Εμφανίζονται κάρτες με τις διαθέσιμες εφαρμογές, στις οποίες συμπεριλαμβάνονται το όνομα του χρήστη που προσφέρει την διαδρομή, ο χώρος αναχώρησης, ο προορισμός, η ημερομηνία και η ώρα και μία προτεινόμενη τιμή. Ο χρήστης μπορεί να ελέγξει τις διαδρομές και εάν επιθυμεί να δει

περισσότερες πληροφορίες να πατήσει κάποιο κουμπί για πληροφορίες κάποιας διαδρομής και θα μεταφερθεί σε μία άλλη σελίδα.



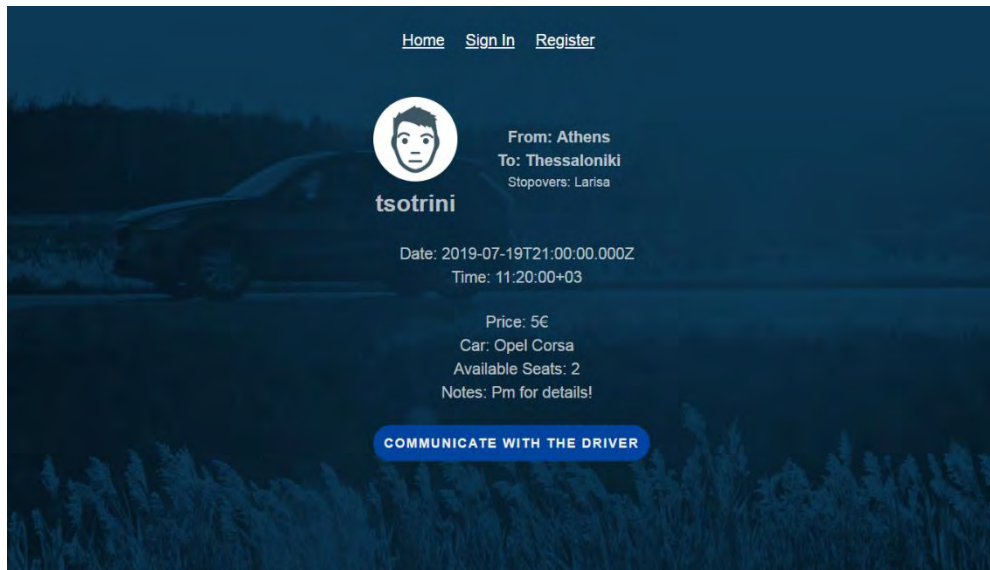
Σχήμα 4.5 Screenshot Εμφάνισης Διαδρομών

#### 4.1.6 Λεπτομέρειες Διαδρομής

Ο χρήστης μεταφέρεται στις λεπτομέρειες μιας διαδρομής μόνο μέσω του κουμπιού για περισσότερες πληροφορίες που υπάρχει στις καρτέλες των διαδρομών κατά την εμφάνιση όλων των διαθέσιμων αποτελεσμάτων. Στις περισσότερες πληροφορίες αναγράφονται κάποια περισσότερα στοιχεία από εκείνα που αναφερόντουσαν στην προηγούμενη σελίδα. Τα παραπάνω στοιχεία είναι πληροφορίες όπως ο τύπος τους αμαξιού της προσφερόμενης διαδρομής, ο αριθμός των θέσεων που προσφέρει ο οδηγός, οι στάσεις που έχει επιλέξει να κάνει και κάποιες σημειώσεις που ίσως έχει γράψει ο οδηγός για την διαδρομή.

Εφόσον η συγκεκριμένη διαδρομή ενδιαφέρει τον χρήστη, υπάρχει η δυνατότητα να πατήσει το κουμπί για επικοινωνία με τον οδηγό. Για να επικοινωνήσει όμως με τον οδηγό απαιτείται ο χρήστης να έχει λογαριασμό στην εφαρμογή και να είναι συνδεδεμένος σε αυτόν. Οπότε ο χρήστης πατώντας το κουμπί για επικοινωνία με τον οδηγό εάν έχει συνδεθεί ήδη θα μεταφερθεί στη σελίδα επικοινωνίας και ανταλλαγής μηνυμάτων,

αλλιώς θα μεταφερθεί στη σελίδα σύνδεσης χρήστη, έτσι ώστε να γίνει η σύνδεση και να συνεχίσει την περιήγησή του στην εφαρμογή.



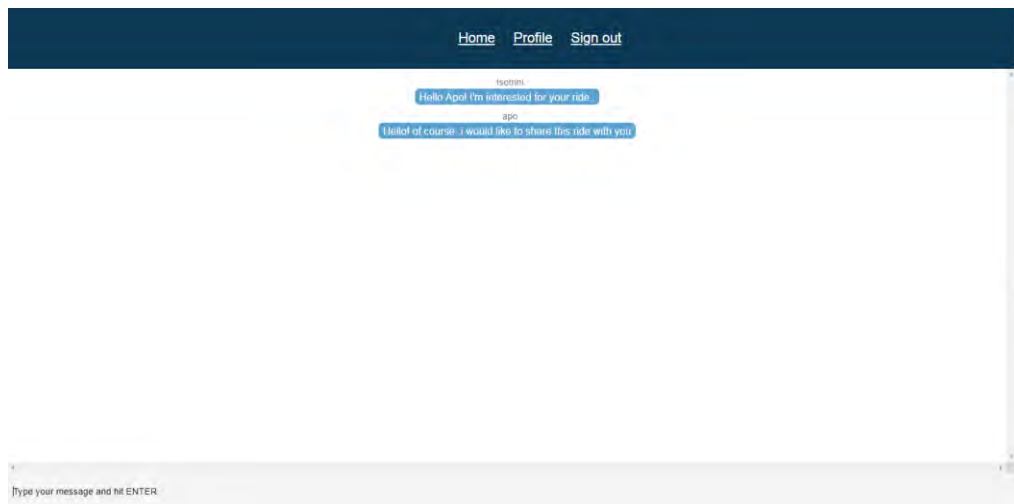
Σχήμα 4.6 Screenshot Λεπτομερειών Διαδρομής

#### 4.1.7 Επικοινωνία με άλλον χρήστη

Στη σελίδα επικοινωνίας με άλλο χρήστη ή σελίδα ανταλλαγής μηνυμάτων, κάποιος έχει πρόσβαση είτε από τη σελίδα εμφάνισης λεπτομερειών για κάποια διαδρομή, είτε από τη σελίδα του προφίλ του χρήστη, έχοντας πρόσβαση στις συνομιλίες που ήδη έχουν αρχίσει από άλλους χρήστες.

Η ανταλλαγή των μηνυμάτων γίνεται σε πραγματικό χρόνο, χρησιμοποιώντας τη βιβλιοθήκη socket.io για την οποία μιλήσαμε στο προηγούμενο κεφάλαιο. Όλα τα μηνύματα αποθηκεύονται στη βάση δεδομένων και έτσι ο κάθε συνομιλητής έχει την δυνατότητα να βλέπει το ιστορικό των συνομιλιών του όταν συνδεθεί στο στην εφαρμογή, αλλά και να ανταλλάξει μηνύματα ζωντανά εκείνη την στιγμή εφόσον και οι δύο χρήστες που συμμετέχουν στην συνομιλία βρίσκονται συνδεδεμένοι στη συνομιλία εκείνη την στιγμή.

Ο χρήστης έχει την δυνατότητα να φύγει από την συνομιλία και να επιστρέψει στην αρχική σελίδα ή στο προφίλ του, μέσω των αντίστοιχων συνδέσμων που υπάρχουν.

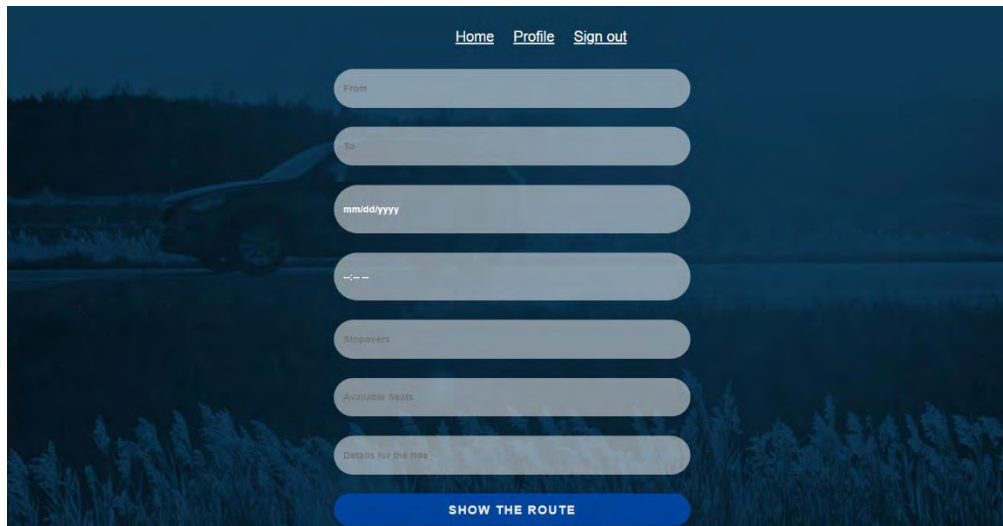


Σχήμα 4.7 Screenshot Επικοινωνία με άλλο χρήστη

#### 4.1.8 Προσφορά Διαδρομής

Ο χρήστης όταν βρίσκεται στην κεντρική σελίδα έχει την δυνατότητα να επιλέξει να προσφέρει μία διαδρομή με το όχημά του. Κατά την επιλογή του μεταφέρεται σε μία φόρμα, την οποία καλείται να συμπληρώσει με βάση τα χαρακτηριστικά που επιθυμεί να έχει η διαδρομή που προσφέρει. Τα πεδία προς συμπλήρωση είναι ο τόπος αναχώρησης, ο προορισμός, η ημερομηνία, η ώρα, οι διαθέσιμες θέσεις που προσφέρει και κάποιες σημειώσεις ή παραπάνω πληροφορίες που εκείνος κρίνει απαραίτητο να προστεθούν.

Αφού συμπληρωθούν όλα τα πεδία, τότε ο χρήστης μπορεί να πατήσει το κουμπί για το επόμενο βήμα καταχώρησης της δημοσίευσης της διαδρομής. Τα δεδομένα που ο χρήστης έχει εκχωρήσει περιμένουν την συμπλήρωση του επόμενου βήματος για να καταχωρηθούν στη βάση δεδομένων. Το επόμενο βήμα στο οποίο θα μεταφερθεί ο χρήστης είναι η σελίδα εμφάνισης χάρτη και προτεινόμενης τιμής διαδρομής.



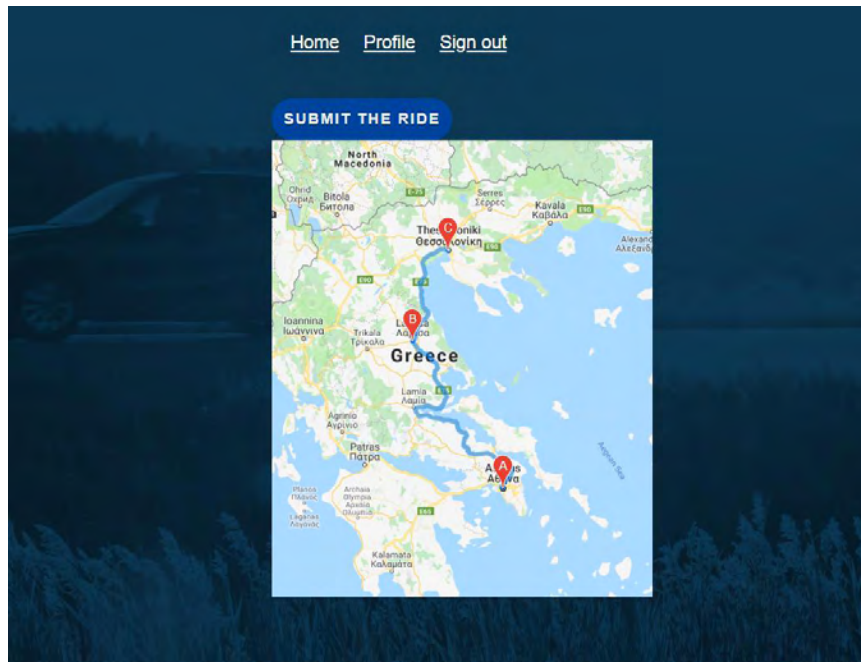
Σχήμα 4.8 Screenshot Προσφοράς Διαδρομής

#### 4.1.9 Εμφάνιση Χάρτη και προτεινόμενης διαδρομής

Η εμφάνιση του χάρτη και της προτεινόμενης τιμής είναι το δεύτερο βήμα για την καταχώρηση μιας διαδρομής. Ο χρήστης αφού καταχωρήσει τα στοιχεία που του ζητούνται και πατήσει επόμενο θα μεταφερθεί σε μία σελίδα όπου θα εμφανιστεί ένας χάρτης της Google με σχεδιασμένη την διαδρομή σύμφωνα με τα δεδομένα που έχουν δοθεί στο προηγούμενο βήμα. Η εμφάνιση του χάρτη και της σχεδιασμένης εφαρμογής γίνεται έπειτα από το αίτημα της εφαρμογής στο Google Maps API.

Εφόσον ο χρήστης θέλει να προχωρήσει στη δημοσίευση της διαδρομής του, πατά το κουμπί για τη δημοσίευση, τα στοιχεία καταχωρούνται στη βάση δεδομένων και εμφανίζεται μήνυμα ενημέρωσης για την επιτυχή καταχώρηση της διαδρομής. Σε άλλη περίπτωση που ο χρήστης δεν θέλει να προχωρήσει στη δημοσίευση πατάει ακύρωση και μεταφέρεται στην κεντρική σελίδα.



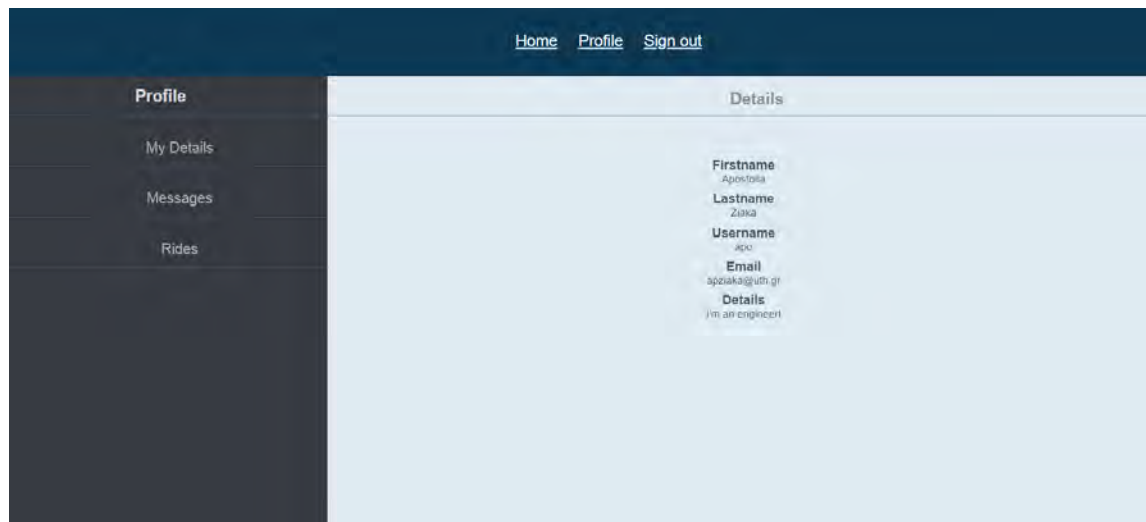


Σχήμα 4.9 Screenshot Εμφάνιση Χάρτη και Προτεινόμενης Τιμής

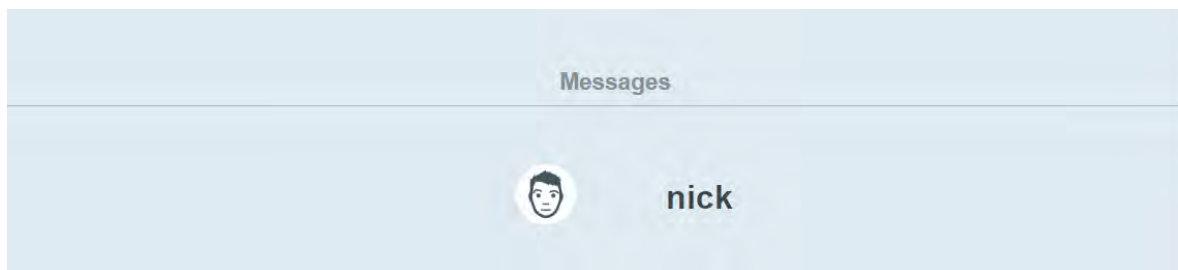
#### 4.1.10 Προφίλ Χρήστη

Για την εμφάνιση του προφίλ του χρήστη, ο χρήστης θα πρέπει να έχει λογαριασμό στην εφαρμογή και να είναι συνδεδεμένος σε αυτόν. Μόνο τότε μπορεί να δει τον σύνδεσμο στην κεντρική σελίδα που θα τον μεταφέρει στη σελίδα του προφίλ.

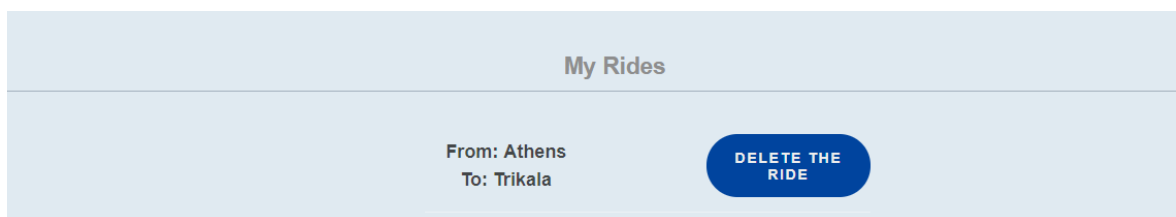
Η σελίδα του προφίλ περιέχει τις πληροφορίες που έχει δώσει ο χρήστης κατά τις καταχώρησή του και μπορεί να έχει πρόσβαση επίσης στα μηνύματά του και στις ήδη προσφερόμενες από τον ίδιο διαδρομές. Μπορεί να δει τις υπάρχουσες συνομιλίες και να ανοίξει κάποια συνομιλία μεταφερόμενος στη σελίδα της επικοινωνίας με άλλο χρήστη (chat). Έχοντας πρόσβαση στις προσφερόμενες από τον ίδιο διαδρομές, ο χρήστης έχει την δυνατότητα να σβήσει από τη βάση δεδομένων μία διαδρομή και να μην προσφέρεται πλέον. Υπάρχουν ακόμη σύνδεσμος που μπορεί να μεταφερθεί στην κεντρική σελίδα και να συνεχίσει την περιήγησή του.



4.10 Screenshot Προφίλ Χρήστη



4.10 Screenshot Εμφάνισης Μηνυμάτων Χρήστη



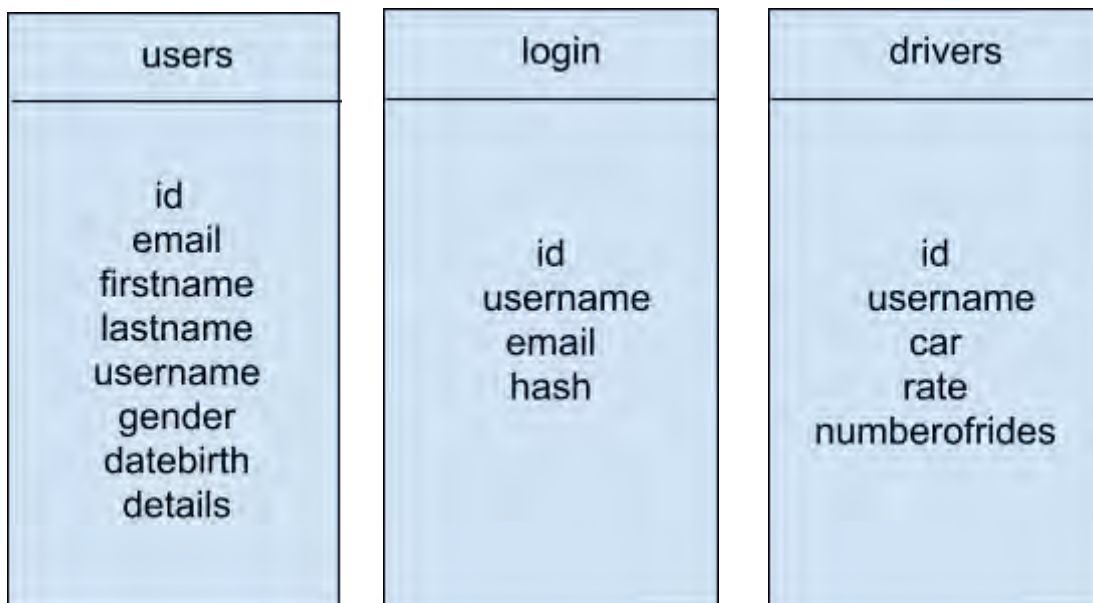
4.11 Screenshot Εμφάνισης Προσφερόμενων Διαδρομών Χρήστη

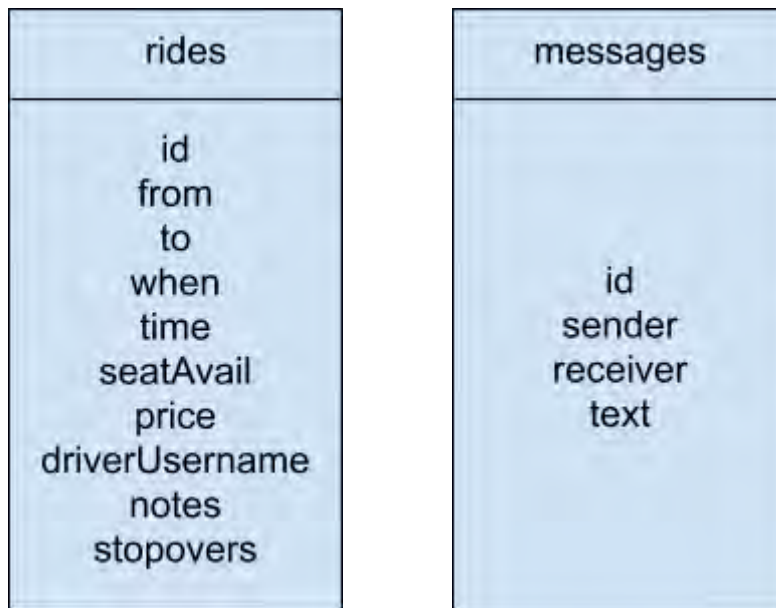
#### 4.1.11 Αποσύνδεση Χρήστη

Υπάρχει ακόμη και η λειτουργία της αποσύνδεσης του χρήστη από την εφαρμογή. Είναι πάντα διαθέσιμος σύνδεσμος για αποσύνδεση στη μπάρα πλοήγησης. Όταν ο χρήστης θελήσει να αποσυνδεθεί και να δεν θέλει άλλη πρόσβαση σε συγκεκριμένες σελίδες πατά τον σύνδεσμο και μεταφέρεται ξανά στην κεντρική σελίδα για περιήγηση ως επισκέπτης, χωρίς να έχει δυνατότητα να έχει πρόσβαση σε όλες τις σελίδες της εφαρμογής.

#### 4.2 Βάση Δεδομένων

Για την ανάπτυξη της εφαρμογής αναπτύχθηκε μία βάση δεδομένων PostgreSQL με τη βοήθεια του εργαλείου διαχείρισης pgAdmin4. Η βάση δεδομένων αποτελείται από πέντε πίνακες, στους οποίους αποθηκεύονται τα όλα τα δεδομένα που διαχειρίζεται η εφαρμογή. Οι πέντε πίνακες είναι οι παρακάτω και έχουν τα εξής χαρακτηριστικά:





4.12 Σχήματα πινάκων βάσης δεδομένων

**users:** Ο πίνακας users χρησιμοποιείται για να αποθηκεύσει τα δεδομένα των χρηστών που καταχωρούνται κατά την εγγραφή στην εφαρμογή. Ο πίνακας αυτός χρησιμοποιείται επίσης για να ανακληθούν λεπτομέρειες για τον χρήστη στην εμφάνιση περισσότερων λεπτομερειών για κάποια διαδρομή.

**login:** Στον πίνακα login αποθηκεύονται τα στοιχεία του χρηστών που συνδέονται στην εφαρμογή. Στο γνώρισμα hash του πίνακα βρίσκονται αποθηκευμένοι οι κωδικοποιημένοι κωδικοί. Ο πίνακας login χρησιμοποιείται κατά τον έλεγχο εγκυρότητας των στοιχείων που καταχωρεί ο χρήστης στη φόρμα σύνδεσης.

**drivers:** Στον πίνακα drivers αποθηκεύονται τα στοιχεία του χρήστη εάν προσφέρει κάποια διαδρομή. Χρήση αυτού του πίνακα γίνεται κατά την εμφάνιση περισσότερων πληροφοριών κάποιας διαδρομής.

**rides:** Ο πίνακας rides χρησιμοποιείται για την αποθήκευση των στοιχείων που εκχωρεί ο χρήστης όταν προσφέρει μία διαδρομή. Τα στοιχεία αυτά ανακαλούνται κατά την αναζήτηση μιας διαδρομής με συγκεκριμένα χαρακτηριστικά και κατά την εμφάνιση των διαδρομών που ταιριάζουν στα στοιχεία αναζήτησης.

**messages:** Τα μηνύματα που ανταλλάσσουν οι χρήστες μεταξύ τους για να επικοινωνήσουν αποθηκεύονται στον πίνακα messages. Επίσης, χρήση του πίνακα αυτού γίνεται όταν ο χρήστης ανοίγει τα μηνύματά του από την σελίδα του προφίλ του και φορτώνει το ιστορικό κάποιας συνομιλίας.

## ΚΕΦΑΛΑΙΟ 5

### ΣΥΝΟΨΗ

#### 5.1 Συμπεράσματα

Η εργασία ασχολήθηκε με τις νέες τεχνολογίες στο χώρο του προγραμματισμού διαδικτυακών εφαρμογών και συγκεκριμένα με τεχνολογίες που έχουν ως βάση τη JavaScript. Στόχος της ήταν η μελέτη των τεχνολογιών με τις οποίες αναπτύχθηκε η εφαρμογή και η αναγνώριση των πλεονεκτημάτων και μειονεκτημάτων τους, αλλά και ανάπτυξη της εφαρμογής αυτής καθαυτής.

Με την ολοκλήρωση τόσο του θεωρητικού όσο και του πρακτικού κομματιού της διπλωματικής αυτής εργασίας μπορούμε να συμπεράνουμε ότι οι νέες τεχνολογίες δίνουν στον προγραμματιστή τεράστιες δυνατότητες και εργαλεία ανάπτυξης εφαρμογών που μπορούν να καλύψουν όλες τις απαιτήσεις και να κάνουν τη διαδικασία πιο εύκολη.

Επίσης, η χρήση της JavaScript και στο frontend αλλά και στο backend κομμάτι κάνει την ανάπτυξη της εφαρμογής πιο κατανοητή, συνδράμει στην εύκολη σύνδεση των frontend και backend πλευρών και κάνει την full stack ανάπτυξη μιας εφαρμογής πιο εύκολο εγχείρημα για έναν και μόνο προγραμματιστή.

Πρέπει να σημειωθεί ακόμη, ότι η ύπαρξη πληθώρας τεχνολογιών και δυνατοτήτων απαιτεί από τον προγραμματιστή περισσότερο χρόνο να μάθει και να αποκτήσει εμπειρία έτσι ώστε να ολοκληρώσει ένα άρτιο και ποιοτικό έργο. Στην περίπτωση της εφαρμογής που αναπτύχθηκε για το σκοπό της εργασίας σίγουρα χρειάζεται επιπλέον χρόνος για την τελειοποίηση της ανάπτυξης.

## 5.2 Μελλοντικές Επεκτάσεις της εφαρμογής

Η εφαρμογή είναι λειτουργική και μπορεί να αποτελέσει εργαλείο της καθημερινότητας ενός χρήστη, αλλά η ανάπτυξη της έχει περιθώρια βελτίωσης και επέκτασης.

Κάποια από τις ενέργειες που θα μπορούσαν να γίνουν μελλοντικά με σκοπό η εφαρμογή να γίνει πιο χρηστική και αποτελεσματική είναι οι εξής:

- Εγγραφή και σύνδεση του χρήστη χρησιμοποιώντας ήδη υπάρχοντες λογαριασμούς στα μέσα κοινωνικής δικτύωσης, όπως λογαριασμός Facebook ή Google.
- Ύπαρξη δυνατότητας βαθμολόγησης και σχολίων κριτικής από του χρήστες προς τους οδηγούς που προσέφεραν τις διαδρομές, δημιουργώντας έτσι ένα σύστημα έμπιστης αξιολόγησης των χρηστών.
- Ένταξη τραπεζικής πλατφόρμας με σκοπό ο χρήστης να μπορεί να πληρώνει για την διαδρομή την τιμή διαδρομής που επιβάλλει ο οδηγός, αποφεύγοντας έτσι τις χρηματικές δοσοληψίες μεταξύ των χρηστών.
- Βελτίωση του design της εφαρμογής.

## BIBΛΙΟΓΡΑΦΙΑ

- [1] Ride.guru. (2019). *RideGuru - What is Carpooling? The difference between Carpools and Rideshares*. [online] Available at: <https://ride.guru/content/newsroom/what-is-carpooling-the-difference-between-carpools-and-rideshares> [Accessed 2 Jul. 2019].
- [2] WTOP. (2019). *A history of carpooling: The early 1900s to now | WTOP*. [online] Available at: <https://wtop.com/council-of-government/2018/02/history-of-carpooling/slide/1/> [Accessed 2 Jul. 2019].
- [3] Erik Ferguson. The rise and fall of the American carpool: 1970–1990. *Transportation* 24.4. (1997)
- [4] Wauters, R. (2019). *Ride-sharing in Europe: the race between BlaBlaCar and Carpooling.com*. [online] Tech.eu. Available at: <https://tech.eu/features/481/ride-sharing-europe-carpooling-blablacar/> [Accessed 2 Jul. 2019].
- [5] Vardalos, S. Full Stack Javascript Development using MEAN stack. [online] Dione.lib.unipi.gr. Available at: [http://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/9352/Vardalos\\_Stefanos.pdf?sequence=1&isAllowed=y](http://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/9352/Vardalos_Stefanos.pdf?sequence=1&isAllowed=y) [Accessed 2 Jul. 2019].
- [6] Definitions, W. and Hope, C. (2019). *What is a Web Application?*. [online] Computerhope.com. Available at: <https://www.computerhope.com/jargon/w/web-application.htm> [Accessed 2 Jul. 2019].
- [7] Medium. (2019). *Web Application Architecture Part-1 (Guide to become Full Stack Developer)*. [online] Available at: <https://medium.com/codiumclub/web-application-architecture-part-1-guide-to-become-full-stack-developer-cc9526a3519b> [Accessed 2 Jul. 2019].
- [8] Medium. (2019). *Web Application Architecture - EXISTEK - Medium*. [online] Available at: <https://medium.com/existek/web-application-architecture-da77ea0cb520> [Accessed 2 Jul. 2019].
- [9] Adhikari, A. Full Stack JavaScript: Web Application Development with MEAN (2016). [online] Available at:



<https://www.theseus.fi/bitstream/handle/10024/116597/ThesisAA.pdf?sequence=1&isAllowed=y> [Accessed 2 Jul. 2019].

- [10] Stack Overflow. (2019). *Stack Overflow Developer Survey 2019*. [online] Available at: <https://insights.stackoverflow.com/survey/2019> [Accessed 2 Jul. 2019].
- [11] Medium. (2019). *Node.js at PayPal - PayPal Engineering - Medium*. [online] Available at: <https://medium.com/paypal-engineering/node-js-at-paypal-4e2d1d08ce4f> [Accessed 2 Jul. 2019].
- [12] Mikhailova, D. GPS pinned messaging application. [online] Available at: <chrome-extension://oemmdcbldboiebfnladdacbdm/https://dspace.cvut.cz/bitstream/handle/10467/65099/F8-DP-2016-Mikhailova-Daria-thesis.pdf?sequence=-1&isAllowed=y> [Accessed 2 Jul. 2019].
- [13] Toptal Engineering Blog. (2019). *Why The Hell Would I Use Node.js? A Case-by-Case Tutorial*. [online] Available at: <https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js> [Accessed 2 Jul. 2019].
- [14] En.wikipedia.org. (2019). *Npm (software)*. [online] Available at: [https://en.wikipedia.org/wiki/Npm\\_\(software\)](https://en.wikipedia.org/wiki/Npm_(software)) [Accessed 2 Jul. 2019].
- [15] Expressjs.com. (2019). *Express basic routing*. [online] Available at: <https://expressjs.com/en/starter/basic-routing.html> [Accessed 2 Jul. 2019].
- [16] Merixstudio.com. (2019). *Websockets: how it works and how to use it*. [online] Available at: <https://www.merixstudio.com/blog/websockets-how-it-works-and-how-use-it/> [Accessed 2 Jul. 2019].
- [17] Krill, P. (2019). *React: Making faster, smoother UIs for data-driven Web apps*. [online] InfoWorld. Available at: <https://www.infoworld.com/article/2608181/react--making-faster--smoother-uis-for-data-driven-web-apps.html> [Accessed 2 Jul. 2019].
- [18] Reactjs.org. (2019). *React – A JavaScript library for building user interfaces*. [online] Available at: <https://reactjs.org/> [Accessed 2 Jul. 2019].
- [19] Chand, S. (2019). *What Is React | JavaScript Library For Building User Interfaces | Edureka*. [online] Edureka. Available at: <https://www.edureka.co/blog/what-is-react/> [Accessed 2 Jul. 2019].

- [20] Developer News. (2019). *How to understand a component's lifecycle methods in ReactJS*. [online] Available at: <https://www.freecodecamp.org/news/how-to-understand-a-components-lifecycle-methods-in-reactjs-e1a609840630/> [Accessed 2 Jul. 2019].
- [21] Stefanov Stoyan, editor. *React: Up and Running: Building web Applications*. First Edition; 2016. Accessed April 4 2017
- [22] Apple Support. (2019). *Lion Server: MySQL not included*. [online] Available at: <https://support.apple.com/en-us/HT4828> [Accessed 2 Jul. 2019].
- [23] Web.archive.org. (2019). *PostgreSQL: History*. [online] Available at: <https://web.archive.org/web/20170326020245/https://www.postgresql.org/about/history/#> [Accessed 2 Jul. 2019].
- [24] PostgreSQL: What You Need to Know. [online] Available at: <https://www.upwork.com/hiring/data/postgresql-basics/> [Accessed 2 Jul. 2019].
- [25] Repository.ellak.gr. (2019). [online] Available at: [https://repository.ellak.gr/ellak/bitstream/11087/1441/1/postgis\\_qgis\\_gr\\_2.0.pdf](https://repository.ellak.gr/ellak/bitstream/11087/1441/1/postgis_qgis_gr_2.0.pdf) [Accessed 2 Jul. 2019].
- [26] Knexjs.org. (2019). *Knex.js - A SQL Query Builder for Javascript*. [online] Available at: <https://knexjs.org/> [Accessed 2 Jul. 2019].
- [27] Sublimetext.com. (2019). *Sublime Text - A sophisticated text editor for code, markup and prose*. [online] Available at: <https://www.sublimetext.com/> [Accessed 2 Jul. 2019].
- [28] Postman Learning Center. (2019). *Intro to the Postman API | Postman Learning Center*. [online] Available at: [https://learning.getpostman.com/docs/postman/postman\\_api/intro\\_api/](https://learning.getpostman.com/docs/postman/postman_api/intro_api/) [Accessed 2 Jul. 2019].
- [29] Postman. (2019). *About Postman | Meet Our Founders, Learn About Our Values, & More*. [online] Available at: <https://www.getpostman.com/about-postman> [Accessed 2 Jul. 2019].
- [30] How-To Geek. (2019). *What Is GitHub, and What Is It Used For?*. [online] Available at: <https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/> [Accessed 2 Jul. 2019].

[31] Kinsta Managed WordPress Hosting. (2019). *What Is GitHub? A Beginner's Introduction to GitHub*. [online] Available at: <https://kinsta.com/knowledgebase/what-is-github/> [Accessed 2 Jul. 2019].



## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

<b>API</b>	Application Programming Interface
<b>HTML</b>	Hypertext Markup Language
<b>CSS</b>	Cascading Style Sheet
<b>PHP</b>	Hypertext Preprocessor
<b>AJAX</b>	Asynchronous JavaScript and XML
<b>XML</b>	Extensible Markup Language
<b>JSON</b>	JavaScript Object Notation
<b>NPM</b>	Node Package Manager
<b>URL</b>	Uniform Resource Locator
<b>HTTP</b>	Hypertext Transfer Protocol
<b>DOM</b>	Document Object Model
<b>JSX</b>	JavaScript XML

