

Έκθεση Project για το μάθημα «Προγραμματισμός Διαδικτύου» - Ακαδημαϊκό Έτος 2020 - 2021

Τίτλος Project: «Ιστοσελίδα για αστικές συγκοινωνίες»

Ομάδα

Καρδούτσος Άγγελος – Α.Μ.: 1059372, Έτος: 4^ο

Καλδής Χρήστος - Α.Μ.: 1059364, Έτος: 4^ο

Περιεχόμενα

Περιεχόμενα	1
1. Εισαγωγή – Παρουσίαση Μικρόκοσμου	2
1.1 Εισαγωγή	2
1.2 Περιγραφή Μικρόκοσμου	2
1.3 Εννοιολογικός Σχεδιασμός	2
1.4 Σχεσιακό Μοντέλο	3
2. Σχεδιασμός της Πλευράς του Πελάτη	4
2.1 Τεχνολογίες Υλοποίησης Πελάτη	4
2.2 HTML	4
2.3 CSS	4
2.4 JavaScript	4
3. Σχεδιασμός στην Πλευρά του Εξυπηρετητή	5
3.1 Σχεδιασμός Βάσης Δεδομένων	5
3.2 Τεχνολογίες Υλοποίησης Εξυπηρετητή	6
3.2.1 Express	6
3.2.1 Handlebars	6
3.2.3 Πρόσθετα packages	7
4. Αποτελέσματα	8
4.1 Αξιολόγηση Εφαρμογής	8
4.2 Χάρτες	8
4.3 Χρονοδιάγραμμα	9
5. Παράρτημα	10
5.1 Σύνδεσμοι	10
5.2 Οδηγίες Εγκατάστασης	10
6. Βιβλιογραφία	10

1. Εισαγωγή – Παρουσίαση Μικρόκοσμου

1.1 Εισαγωγή

Στόχος της εργασίας είναι η δημιουργία ενός server, ο οποίος θα φιλοξενεί μια ιστοσελίδα σχετικά με τα αστικά λεωφορεία. Ειδικότερα, σκοπός της ιστοσελίδας είναι να εξυπηρετεί το επιβατικό κοινό, παρέχοντας πληροφορίες για ό,τι χρειαστεί. Μερικές από τις πληροφορίες αυτές μπορούν να αφορούν τις στάσεις μιας λεωφορειακής γραμμής, τις ώρες των δρομολογίων, καθώς και τύποι και τιμές των εισιτηρίων.

Με βάση τις προηγούμενες ανάγκες, κρίνεται απαραίτητη η δημιουργία ενός server, ο οποίος θα φιλοξενεί μία ιστοσελίδα. Ο χρήστης (client) θα πραγματοποιεί αιτήματα στο server, ο οποίος πρέπει να του επιστρέφει τη σελίδα, που περιέχει τις πληροφορίες, που επιθυμεί ο χρήστης. Οι σελίδες δημιουργήθηκαν προσπαθώντας να καλύψουν τις απαιτήσεις του μικρόκοσμου.

Η ανάπτυξη της ιστοσελίδας και των τεχνολογιών έγινε με τη σειρά που εισηγήθηκαν στο μάθημα. Η μοντελοποίηση μίας βάσης δεδομένων σύμφωνα με τις απαιτήσεις ενός μικρόκοσμου ήταν ήδη γνωστός από προηγούμενο μάθημα και αναπτύχθηκε πρώτος. Έπειτα, αναπτύχθηκε η ιστοσελίδα από την πλευρά του χρήστη και στη συνέχεια ξεκίνησε η ανάπτυξη της εφαρμογής στην πλευρά του εξυπηρετητή και η σύνδεση και επικοινωνία με τη βάση δεδομένων που σχεδιάστηκε στην αρχή.

1.2 Περιγραφή Μικρόκοσμου

Οι ανάγκες και οι δυνατότητες του μικρόκοσμου εξετάστηκαν σε σχέση με τις ιστοσελίδες των αστικών λεωφορείων μεγάλων πόλεων της Ελλάδας. Η ιστοσελίδα μας βασίζεται στα δρομολόγια και τις υπηρεσίες που παρέχονται στην Πάτρα. Τα λεωφορεία χωρίζονται σε γραμμές και κάθε μία περνάει από διαφορετικές στάσεις. Ταυτόχρονα, η τιμή του εισιτηρίου διαφοροποιείται ανάλογα με τη ζώνη που βρίσκεται ο τελικός προορισμός του επιβάτη. Ο χρήστης μπορεί να διαθέτει μηνιαία κάρτα ή εισιτήριο. Τέλος τα νέα περιέχουν τις τελευταίες ανακοινώσεις σχετικά με την μετακίνηση στην πόλη. Με βάση τα παραπάνω μπορούμε εύκολα να βρούμε τις οντότητες που χρειάζονται για τον εννοιολογικό σχεδιασμό της βάσης μας.

1.3 Εννοιολογικός Σχεδιασμός

Ο εννοιολογικός σχεδιασμός αποτελεί το πρώτο βήμα δημιουργίας της βάσης δεδομένων μετά την περιγραφή του μικρόκοσμου. Προσπαθεί να αποτυπώσει τα κύρια στοιχεία της βάσης που αποτελούν τις οντότητες και τα χαρακτηριστικά τους που τα περιγράφουν.

Παράλληλα γίνεται ο χαρακτηρισμός των τύπων κάθε γνωρίσματος καθώς και η εύρεση γνωρισμάτων κλειδιών. Στη συνέχεια, μια πρώτη εκτίμηση του πεδίου ορισμού κάθε γνωρίσματος για την ανάθεση σωστού τύπου μεταβλητής στο σχεσιακό μοντέλο καθώς και στη βάση δεδομένων μας. Ένας καλός εννοιολογικός σχεδιασμός επιταχύνει τη δημιουργία, την ανάπτυξη και την συντήρηση μίας βάσης.

2. Σχεδιασμός της Πλευράς του Πελάτη

2.1 Τεχνολογίες Υλοποίησης Πελάτη

Η δημιουργία της εφαρμογής ξεκίνησε από το σχεδιασμό του front-end της ιστοσελίδας. Οι βασικές τεχνολογίες που χρησιμοποιήθηκαν ήταν η HTML5 για να ορίσουμε τη δομή και το περιεχόμενο του κειμένου, η CSS3 για τη μορφοποίηση του κειμένου και τον τρόπο παρουσίασης και η JavaScript για τη διαχείριση των γεγονότων και τη διαδραστικότητα της εφαρμογής. Η σχέση των 3 αυτών προγραμματιστικών γλωσσών είναι άμεση και η δυνατότητα συγγραφής εμφωλευμένου κώδικα μίας από αυτές σε μία άλλη είναι ιδιαίτερως εύκολη και μερικές φορές απαραίτητη. Παρόλα αυτά δεν προτείνεται καθώς ο σκοπός της κάθε γλώσσας είναι ξεχωριστός και δεν πρέπει να συγχέονται οι ρόλοι της κάθε μιας.

Στην εργασία έγινε προσπάθεια διαχωρισμού των αρχείων σύμφωνα με το σκοπό τους.

2.2 HTML

Το πρώτο βήμα ήταν ο σχεδιασμός του περιεχομένου της ιστοσελίδας. Βρέθηκαν οι απαιτήσεις του μικρόκοσμου που θα παρουσιάζονται σε διαφορετική σελίδα και σχεδιάστηκε το περιεχόμενο της αρχικής σελίδας. Η αρχική σελίδα ακολουθεί τη βασική δομή μιας ιστοσελίδας, περιέχει τίτλο, ένα εισαγωγικό σημείωμα και μια μπάρα περιήγησης που μεταφέρει το χρήστη στις υπόλοιπες σελίδες. Επιλέχθηκαν επίσης κομμάτια των υπόλοιπων σελίδων που είναι πιο πιθανό να αναζητούν οι επισκέπτες της ιστοσελίδας και παρουσιάστηκαν στην αρχική σελίδα για διευκόλυνσή τους. Η εφαρμογή δίνει στο χρήστη τις απαραίτητες γνώσεις για τις γραμμές λεωφορείων, τα εισιτήρια, τις νέες ανακοινώσεις της εταιρίας καθώς και τα στοιχεία επικοινωνίας. Αναλυτικότερα οι ενημερώσεις, οι γραμμές των λεωφορείων με τις εκάστοτε στάσεις και τα δρομολόγιά τους, οι τιμές εισιτηρίων, η τοποθεσία και τα στοιχεία επικοινωνίας είναι διαθέσιμα σε διαφορετικές σελίδες και αρχεία HTML.

2.3 CSS

Στη συνέχεια μορφοποιήθηκε ο χαρακτήρας της εφαρμογής. Επιλέχθηκαν τα χρώματα που θα περιέχει καθώς και η παρουσίαση του κειμένου για τα διάφορα μεγέθη του παραθύρου. Μια ιστοσελίδα δεν χρειάζεται να είναι φορτωμένη με πολλά χρώματα ή αναλυτικές οδηγίες, μακροσκελή κείμενα και πολλά widget επειδή υπάρχει κίνδυνος να κουράσει τον επισκέπτη. Η παρουσίαση της σελίδας έμεινε απλή χωρίς να στερείται λειτουργικότητας.

Με τον ίδιο τρόπο που χωρίστηκαν τα αρχεία για τη δομή της σελίδας χωρίστηκαν και για τη μορφοποίησή της και γράφτηκαν σε CSS. Τα αρχεία CSS αποθηκεύτηκαν στο φάκελο “css” και σε κάθε σελίδα αντιστοιχεί ένα δικό της αρχείο CSS. Παράλληλα, οι σελίδες φορτώνουν επιπλέον αρχεία CSS, τα περιέχουν εντολές για την μορφοποίηση στοιχείων, που υπάρχουν σε περισσότερες από μία σελίδες (π.χ. μενού).

2.4 JavaScript

Τελευταίο αλλά εξίσου σημαντικό ήταν η προσθήκη των γεγονότων στα στοιχεία που ήταν απαραίτητο. Κουμπιά και σύνδεσμοι οι οποίοι δημιουργούνται δυναμικά στη σελίδα χρειάζονται επεξεργασία τόσο στο περιεχόμενο όσο και στη λειτουργία τους. Ταυτόχρονα, δυνατότητες για διευκόλυνση του χρήστη όπως κουμπί κύλισης στο πάνω μέρος της

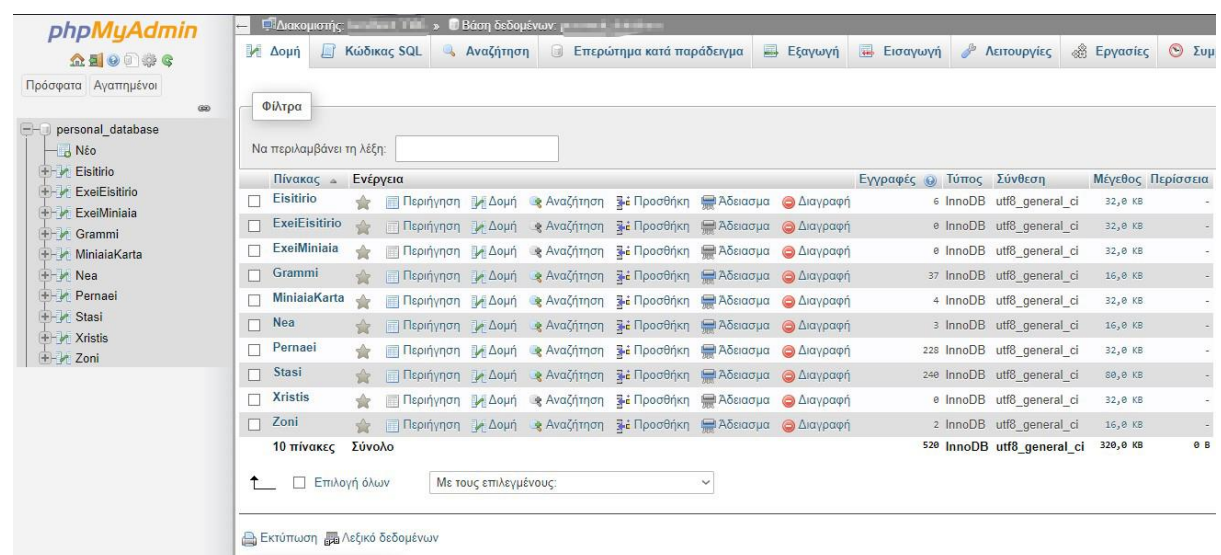
σελίδας ή τροποποίηση της μπάρας περιήγησης ώστε να είναι διαθέσιμη χωρίς όμως να χάνεται το απαραίτητο κενό διάστημα στη σελίδα ώστε να παραμένει ξεκούραστη η περιήγηση του χρήστη.

Τα αρχεία JavaScript αποθηκεύτηκαν στον φάκελο “**scripts**”, ο οποίος περιέχει και τον φάκελο “**json**”, ο οποίος περιέχει json αρχεία με δεδομένα για τη δημιουργία αρχικά, στατικών σελίδων στα πρώτα στάδια της υλοποίησης.

3. Σχεδιασμός στην Πλευρά του Εξυπηρετητή

3.1 Σχεδιασμός Βάσης Δεδομένων

Κατά τον σχεδιασμό και υλοποίηση της διεπαφής του χρήστη παρατηρήθηκε η ανάγκη και η πρακτικότητα της βάσης δεδομένων. Με βάση το σχεσιακό μοντέλο που σχεδιάστηκε παραπάνω δημιουργήθηκε η βάση μας. Τα δεδομένα στηρίχτηκαν στα πραγματικά δεδομένα των αστικών λεωφορείων της Πάτρας. Η βάση υλοποιήθηκε σε MySQL και η εισαγωγή και διαχείριση των δεδομένων πριν τη σύνδεσή της με την εφαρμογή μας έγινε μέσω phpMyAdmin.



Εικόνα 1: Τελική βάση δεδομένων (phpMyAdmin)

Εντοπίζονται ορισμένες διαφορές μεταξύ του σχεσιακού μοντέλου και της τελικής βάσης δεδομένων. Αυτό οφείλεται σε ανάγκες, οι οποίες προέκυψαν κατά τη διάρκεια υλοποίησης του project, οπότε η βάση δεδομένων έπρεπε να προσαρμοστεί ανάλογα των απαιτήσεων.

Θα πραγματοποιηθεί μια σύντομη περιγραφή της βάσης δεδομένων και ορισμένων πινάκων, καθώς δεν αποτελεί βασικό αντικείμενο αυτής της εργασίας. Ορισμένοι από τους κυριότερους πίνακες είναι:

- Grammi: περιέχει τους κωδικούς και τις ονομασίες των λεωφορειακών γραμμών
- Stasi: έχει αποθηκευμένα τις ονομασίες και τις γεωγραφικές θέσεις των στάσεων
- Pernaiei: συνδέει τους πίνακες **Grammi** και **Stasi**, περιγράφοντας τις διαδρομές των γραμμών

- Nea: περιέχει τα απαραίτητα στοιχεία των ανακοινώσεων (τίτλος, ημερομηνία, περιεχόμενο)

3.2 Τεχνολογίες Υλοποίησης Εξυπηρετητή

Ο προγραμματισμός στην πλευρά του εξυπηρετητή έγινε με **JavaScript**. Είναι μία τεχνολογία που συνεχώς αυξάνεται και αποτελεί μία εύχρηστη λύση τόσο για τις δυνατότητες που παρέχει μέσα από βιβλιοθήκες όσο και για τη δομή του προγράμματος καθώς δεν χρειάζεται να αποτελείται από πολλές προγραμματιστικές γλώσσες.

Χρησιμοποιήθηκε η `node.js` και συγκεκριμένα η βιβλιοθήκη **`express.js`** που διευκολύνει τη διεκπεραίωση HTTP και URI αιτημάτων που γίνονται στην ιστοσελίδα μας. Ένα προτέρημα της `express` είναι ότι χρησιμοποιεί το `middleware pattern`. Με τον τρόπο αυτό η επεξεργασία των αιτημάτων γίνεται σε τμήματα. Έτσι, διευκολύνεται ο διαχωρισμός του προγράμματος σε διαφορετικά αρχεία, η αποφυγή μακροσκελών συναρτήσεων και η συγγραφή δυσανάγνωστου κώδικα. Η απάντηση των αιτημάτων γίνεται σε μηχανές για `template` συγκεκριμένα γίνεται χρήση της **`Handlebars.js`**.

3.2.1 Express

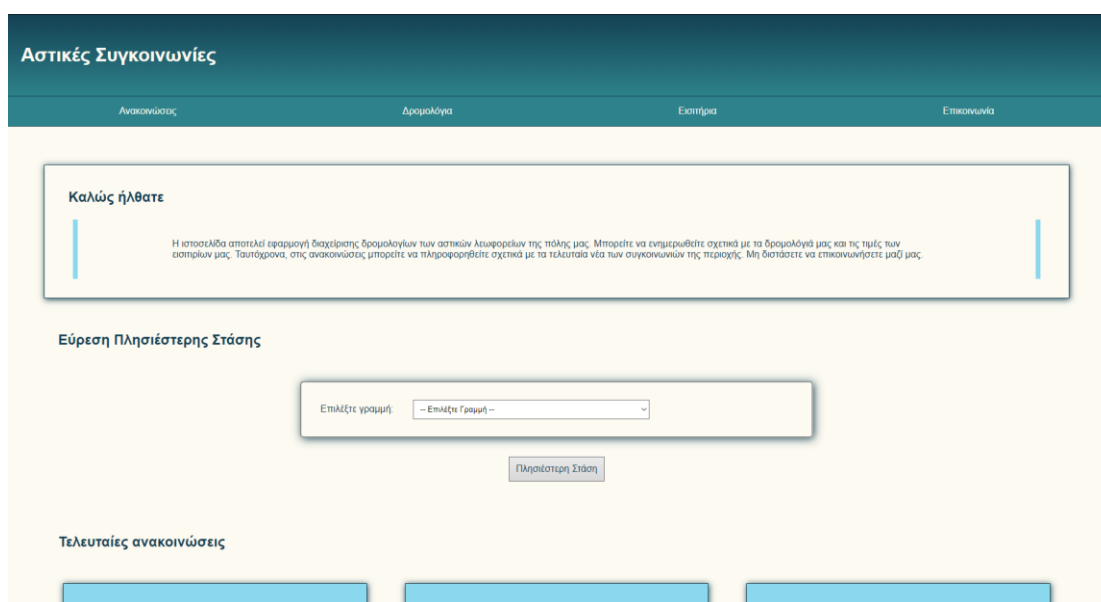
Το πρόγραμμα όπως γνωρίζουμε χωρίζεται σε τμήματα. Αρχικά υπάρχουν τα αρχεία **`index.js`** και **`app.js`** που ορίζουν τη σύνδεση και τα απαραίτητα αρχεία και πακέτα που χρειάζεται η εφαρμογή για την ορθή λειτουργία της. Στο φάκελο **`"server"`** υπάρχουν τα αρχεία για τη διαχείριση των αιτημάτων των πελατών. Το αρχείο **`pt-routes.js`** περιέχει τις διαδρομές που αναγνωρίζει ο `server` μας, ενώ το αρχείο **`pt-model-sql.js`** μας παρέχει τη σύνδεση με τη βάση, καθώς επίσης αντλεί τα απαραίτητα δεδομένα. Τέλος το αρχείο **`pt-controller.js`** επεξεργάζεται τα δεδομένα που αντλούμε από τη βάση και τα προσθέτει στις πληροφορίες, που επιστρέφει στο χρήστη μέσω ενός `template`.

3.2.1 Handlebars

Τα αρχεία `Handlebars` βρίσκονται στο φάκελο **`"views"`**. Αντικαθιστούν τα παλιά αρχεία που γράφτηκαν σε HTML και λύνουν το πρόβλημα των επαναλαμβανόμενων γραμμών και στοιχείων του DOM στις σελίδες της εφαρμογής καθώς εισάγονται σε μορφή json κάθε φορά τα ίδια δεδομένα. Υπάρχει ένα βασικό layout, το αρχείο **`main.hbs`** το οποίο αποτελείται από το `head`, το `header` και το `footer` της σελίδας, δηλαδή τα κοινά στοιχεία όλων των σελίδων. Με βάση αυτό το layout, δημιουργείται η κάθε σελίδα, στην οποία αντιστοιχεί ένα διαφορετικό αρχείο **`hbs`** και το οποίο γίνεται `render` εντός του `<main>` του αρχείου **`main.hbs`**. Ανάλογα με το URI request του πελάτη, γίνεται `render` διαφορετικό αρχείο **`hbs`**, ενώ παράλληλα `server` στέλνει ένα JSON object με τα στοιχεία που πρέπει να προστεθούν και διαφέρουν από τις άλλες σελίδες (π.χ. τίτλος, αρχεία **`css`**). Επιπλέον, αξιοποιούνται οι `Helpers` που παρέχει το **`Handlebars`** για την ευκολότερη προσθήκη πολλαπλών όμοιων στοιχείων. Όλα τα **`hbs`** αρχεία συνδέονται με τα αντίστοιχα αρχεία μορφοποίησης και διαχείρισης γεγονότων που γράφτηκαν σε **`CSS`**, **`JavaScript`** κατά την ανάπτυξη του `front-end`.

Ορισμένες από τις κυριότερες σελίδες της ιστοσελίδας είναι:

- Αρχική σελίδα: Περιέχει ένα μήνυμα καλωσορίσματος, ένα widget εύρεσης πλησιέστερης στάσης (θα περιγραφεί στην ενότητα [4.2 Χάρτες](#)) και ορισμένους χρήσιμους συνδέσμους. (Εικόνα 3)
- Ανακοινώσεις: Περιέχει τις σημαντικότερες ανακοινώσεις, ταξινομημένους ως προς την πιο πρόσφατη.
- Δρομολόγια: Μια λίστα με όλα τα δρομολόγια. Επιλέγοντας ένα συγκεκριμένο δρομολόγιο, ο χρήστης ανακατευθύνεται σε ξεχωριστή σελίδα, όπου παρουσιάζονται αναλυτικά οι ώρες των δρομολογίων και οι στάσεις.
- Εισιτήρια: Περιέχει πίνακες σχετικά με τις τιμές εισιτηρίων και τα εκδοτήρια.
- Επικοινωνία: Βασικές πληροφορίες της εταιρείας και μια φόρμα επικοινωνίας.



Εικόνα 3: Παράδειγμα σελίδας της ιστοσελίδας (Αρχική Σελίδα)

3.2.3 Πρόσθετα packages

Εκτός από τα πακέτα που αναλύθηκαν στις προηγούμενες ενότητες, χρησιμοποιήθηκαν ορισμένα ακόμα για συγκεκριμένες λειτουργίες του server:

- mysql: για την επικοινωνία του server με τη βάση δεδομένων.
- node-fetch: για να στέλνει ο server αιτήματα get σε APIs και να λαμβάνει απαντήσεις.
- nodemailer: για την αποστολή email, τα οποία συντάχθηκαν στην φόρμα επικοινωνίας.
- dotenv (devDependency): για την ανάγνωση του αρχείου **.env**.

4. Αποτελέσματα

4.1 Αξιολόγηση Εφαρμογής

Η αξιολόγηση έγινε εξετάζοντας αν οι πληροφορίες οι οποίες αναζητεί ο επισκέπτης βρίσκονται στην εφαρμογή και αν ναι, πόσο εύκολο είναι να βρεθούν. Οι απαιτήσεις του μικρόκοσμου δομήθηκαν έτσι ώστε να περιέχουν τις απαραίτητες πληροφορίες που παρουσιάζονται σε παρόμοιες ιστοσελίδες και η προβολή τους χωρίστηκε σε διαφορετικές σελίδες της εφαρμογής.

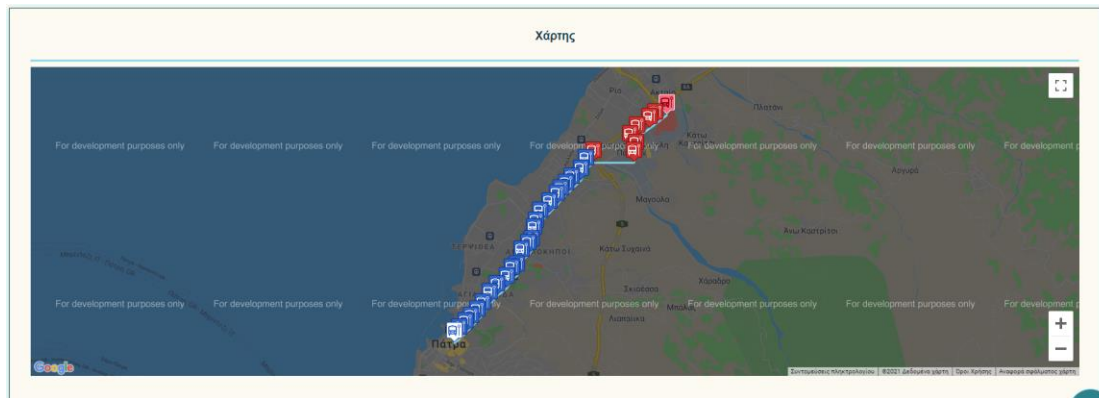
Οι χρήστες του διαδικτύου που επιλέγουν να περιηγηθούν στον παγκόσμιο ιστό χρησιμοποιώντας το κινητό τους τηλέφωνο αυξάνονται συνεχώς. Ωστόσο, κατά τη σχεδίαση μιας ιστοσελίδας ένα συνηθισμένο λάθος είναι να μη λαμβάνεται υπόψη η προβολή και η λειτουργία της εφαρμογής για μικρής κλίμακας οθόνες. Επειδή η εφαρμογή έχει άμεση σχέση με εξωτερικούς χώρους εξετάστηκε η λειτουργικότητα της ιστοσελίδας σε τηλέφωνα.

4.2 Χάρτες

Ο μεγαλύτερος όγκος των δεδομένων, που ήταν αποθηκευμένα στη βάση δεδομένων, αφορούσε πληροφορίες σχετικά με τις στάσεις των λεωφορείων. Συγκεκριμένα, είχαμε στη διάθεση μας το όνομα μίας στάσης, τη ζώνη στην οποία ανήκει, τη γεωγραφική της θέση (σε μορφή διεύθυνσης και συντεταγμένων), καθώς και για μια συγκεκριμένη λεωφορειακή γραμμή, τις στάσεις από τις οποίες περνάει και με ποια σειρά. Επομένως, κρίθηκε απαραίτητη η οπτικοποίηση των δεδομένων αυτών με σκοπό τη διευκόλυνση του χρήστη.

Για την απαίτηση αυτή, αξιοποιήθηκαν τα **Google Maps APIs**, τα οποία προσφέρει η Google. Ειδικότερα, με τη χρήση του Maps JavaScript API, δημιουργείται στην πλευρά του client ένα Google Map χάρτης. Εντός αυτού τοποθετούνται markers με τις στάσεις των στάσεων μιας λεωφορειακής γραμμής. Για την δημιουργία των markers, ο server ανακτά τις συντεταγμένες από τη βάση δεδομένων και τις στέλνει στο client, ώστε να γνωρίζει τη θέση όπου θα τοποθετηθεί κάθε marker, ο οποίος διαφέρει μορφή ανάλογα με τις ιδιαιτερότητες της στάσης, δηλαδή ανάλογα με τη ζώνη στην οποία ανήκει ή αν είναι αφετηρία. (Εικόνα 4)

Επιπλέον, η ιστοσελίδα δίνει τη δυνατότητα στο χρήστη να βρει την πλησιέστερη στάση μιας συγκεκριμένης λεωφορειακής γραμμής. Αφού ο χρήστης επιλέξει την γραμμή, ο server στέλνει ένα GET request στο **Distance Matrix API** της Google, στέλνοντας τις συντεταγμένες των στάσεων, καθώς και αυτή του χρήστη. Το API ένα JSON object με την απόσταση του χρήστη από κάθε στάση, καθώς και ο χρόνος που απαιτείται για να φτάσει σε αυτή περπατώντας. Αφού ο server επεξεργαστεί τα δεδομένα, στέλνει στον client τις πληροφορίες της πλησιέστερης στάσης, τα οποία και προβάλλονται στον χρήστη. (Εικόνα 5)



Εικόνα 4: Χάρτης με τις στάσεις και τη διαδρομή της γραμμής 601

Εύρεση Πλησιέστερης Στάσης

Επιλέξτε γραμμή: 609 – Κέντρο

Πλησιέστερη Στάση

Η πλησιέστερη στάση της γραμμής είναι η στάση "Νοσοκομείο"

Απόσταση: 211 χλμ
Χρόνος Αφίξης: 1 μέρες 19 ώρες
Διεύθυνση: "Λεωφ. Ιπποκράτους, Πάτρα 265 04, Ελλάδα"

[Δείτε τη στάση στο χάρτη](#)

Εικόνα 5: Widget για την εύρεση της πλησιέστερης στάσης

4.3 Χρονοδιάγραμμα

Στο πρώτο μισό του εξαμήνου έγινε ο σχεδιασμός της βάσης και της ιστοσελίδας. Αναλυτικότερα, το μήνα Απρίλη σχεδιάστηκε το εννοιολογικό και το σχεσιακό μοντέλο της βάσης. Έως τις 15 Μαΐου σχεδιάστηκε η δομή και το περιεχόμενο της ιστοσελίδας χωρίς τη σύνδεση με τη βάση δεδομένων και τη διαχείριση της δομής με πίνακες της JavaScript στη πλευρά του πελάτη. Στο δεύτερο μισό του Μαΐου έγινε η προσθήκη δεδομένων στη βάση και ξεκίνησε ο σχεδιασμός του back-end. Τον Ιούνιο έγινε προσπάθεια εντοπισμού σφαλμάτων που τυχόν υπάρχουν στον κώδικα και γράφτηκε η αναφορά και η παρουσίαση της εργασίας.

Ο Χρήστος Καλδής ασχολήθηκε κυρίως με το σχεδιασμό του front-end και τη μεταφορά των απαιτήσεων του μικρόκοσμου στα εκάστοτε μέρη της ιστοσελίδας. Έγραψε επίσης την έκθεση και την παρουσίαση της εργασίας.

Ο Άγγελος Καρδούτσος έδωσε ιδιαίτερη έμφαση στον εννοιολογικό σχεδιασμό της βάσης δεδομένων και στη δομή και σχεδιασμό του back-end. Πρόσθεσε επίσης τις οδηγίες εγκατάστασης στο αποθετήριο του GitHub.

5. Παράρτημα

5.1 Σύνδεσμοι

- Σύνδεσμος στο GitHub: <https://github.com/Angelos-Kard/public-transport-site>
- Σύνδεσμος στο Heroku: <https://public-transport-server.herokuapp.com/index>

5.2 Οδηγίες Εγκατάστασης

Αναλυτικές οδηγίες για την εγκατάσταση μπορείτε να βρείτε στο αρχείο **README.md**, καθώς και στο README section του GitHub.

Για να εγκαταστήσετε την εφαρμογή:

1. Κατεβάζετε το συμπιεσμένο αρχείο που περιέχει τον κώδικα από το GitHub και τον αποσυμπιέζετε σε ένα φάκελο του υπολογιστή σας.
2. Ανοίγετε ένα τερματικό στη διεύθυνση του φακέλου που κάνατε την εγκατάσταση και εκτελείτε την εντολή **npm install -dev**.
3. Μπορείτε να δείτε την ιστοσελίδα στο localhost στο port 3000 μέσω του nodemon εκτελώντας την εντολή **npm run debug**, ώστε να «τρέξει» ο server. Τα απαραίτητα στοιχεία για την λειτουργικότητα ορισμένων τμημάτων του server (επικοινωνία με βάση δεδομένων, κλειδί για αιτήματα σε API κτλ.) είναι αποθηκευμένο σε ένα αρχείο **.env**, το οποίο δεν περιλαμβάνεται στο **.zip** αρχείο για λόγους ασφαλείας. Ωστόσο μπορείτε να δημιουργήσετε το δικό **.env** αρχείο με δικά σας στοιχεία, με βάση το template που περιγράφεται στο **README.md**.

Σε περίπτωση που βρίσκεστε σε λειτουργικό σύστημα Windows, μπορείτε απλά να εκτελέσετε το αρχείο **initProject.bat**, που βρίσκεται στον root φάκελο.

6. Βιβλιογραφία

- Διαφάνειες και σημειώσεις του μαθήματος «Προγραμματισμός Διαδικτύου»
- [Stack Overflow](#)
- [W3Schools](#)
- <https://developers.google.com/maps/documentation>
- Documentation κάθε package