



IONIAN UNIVERSITY

DEPARTMENT OF INFORMATICS



THESIS

WEB APPLICATION FOR CARPOOLING

by

ZAMPOUNIS CHRISTOS

**Supervisor:** Stefanidakis Michail, Associate professor

**Committee Members:** Tsoumakos Dimitrios, Andronikos Theodoros,  
Associate professors

Corfu, February 2021

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ WEB ΓΙΑ ΣΥΝΕΠΙΒΑΤΙΣΜΟ  
(CARPOOLING)

ΤΟΥ  
ΖΑΜΠΟΥΝΗ ΧΡΗΣΤΟΥ

**Επιβλέπων:** Στεφανιδάκης Μιχαήλ, Αναπληρωτής καθηγητής  
**Μέλη Επιτροπής:** Τσουμάκος Δημήτριος, Ανδρόνικος Θεόδωρος,  
Αναπληρωτές καθηγητές

Κέρκυρα, Φεβρουάριος 2021

## ΠΕΡΙΕΧΟΜΕΝΑ

|      |                                         |    |
|------|-----------------------------------------|----|
| 1.   | ΠΡΟΛΟΓΟΣ .....                          | 4  |
| 2.   | PROLOGUE.....                           | 5  |
| 3.   | ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.....      | 6  |
| 4.   | ΣΥΝΔΕΣΗ ΜΕ ΤΟ GOOGLE MAPS API.....      | 8  |
| 5.   | ΣΧΕΔΙΑΣΜΟΣ ΔΙΕΠΑΦΗΣ ΤΟΥ ΧΡΗΣΤΗ.....     | 10 |
| 6.   | ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΟΔΗΓΟΥ.....             | 11 |
| 7.   | ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΕΠΙΒΑΤΗ.....            | 14 |
| 8.   | ΠΑΡΑΔΕΙΓΜΑΤΑ ΕΠΙΛΕΓΜΕΝΟΥ ΚΩΔΙΚΑ.....    | 16 |
| 8.1. | ΑΠΟΘΗΚΕΥΣΗ ΔΕΔΟΜΕΝΩΝ ΣΤΟ FIREBASE.....  | 16 |
| 8.2. | ΕΜΦΑΝΙΣΗ ΔΕΔΟΜΕΝΩΝ ΣΤΟ FIREBASE.....    | 19 |
| 8.3. | ΔΙΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ ΣΤΟ FIREBASE.....    | 22 |
| 8.4. | ΑΝΑΝΕΩΣΗ ΔΕΔΟΜΕΝΩΝ ΣΤΟ FIREBASE.....    | 23 |
| 9.   | ΣΥΝΔΕΣΗ ΤΩΝ KEYS.....                   | 25 |
| 10.  | HOSTING.....                            | 26 |
| 11.  | ΕΓΚΑΤΑΣΤΑΣΗ ΕΦΑΡΜΟΓΗΣ (DEPLOYMENT)..... | 28 |
| 12.  | ΣΥΜΠΕΡΑΣΜΑΤΑ.....                       | 39 |
| 13.  | ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΕΚΤΑΣΕΙΣ.....            | 30 |
| 14.  | ΒΙΒΛΙΟΓΡΑΦΙΑ.....                       | 35 |

# 1. ΠΡΟΛΟΓΟΣ

Σε αυτή τη διατριβή, ασχολούμαστε με τη δημιουργία μιας web εφαρμογής συνεπιβατισμού (carpooling) με το όνομα Share the Ride. Στόχος της είναι να δώσει τη δυνατότητα στους χρήστες να μοιραστούν τα έξοδα της διαδρομής τους. Η συγκεκριμένη υπηρεσία μπορεί να μειώσει σημαντικά τα κόστη ενός ταξιδιού, καθώς επίσης, αν έχει ευρεία κοινωνική αποδοχή, αναμένεται να έχει σημαντικά θετικό αντίκτυπο στην μόλυνση του περιβάλλοντος καθώς και την κυκλοφοριακή συμφόρηση.

Δίνεται η δυνατότητα σε οδηγούς να δημοσιεύουν πληροφορίες σε σχέση με το ταξίδι που πρόκειται να πραγματοποιήσουν και μ' αυτό τον τρόπο ταξιδιώτες που ενδιαφέρονται να κάνουν την ίδια διαδρομή να έρθουν σε επικοινωνία μαζί τους, έτσι ώστε να συνταξιδέψουν και να μοιραστούν τα έξοδα της. Δυνατότητα δημοσίευσης δίνεται μόνο στους οδηγούς και οι υποψήφιοι συνταξιδιώτες μπορούν απλά να αναζητήσουν ενεργά ταξίδια και να έρθουν σε επικοινωνία με τους οδηγούς μέσω WhatsApp. Επίσης η συγκεκριμένη πτυχιακή αφορά μόνο διαδρομές εντός Ελλάδος. Παρόμοιες εφαρμογές υπάρχουν εδώ και πολλά χρόνια και έχουν καταφέρει να εδραιωθούν, σε ένα ευρύ κοινό σε χώρες του εξωτερικού, κάνοντας μ' αυτό τον τρόπο τα ταξίδια προσβάσιμα σε μεγαλύτερο ποσοστό ανθρώπων. Μερικά παραδείγματα αυτών είναι το BlaBlaCar<sup>1</sup> που καλύπτει αρκετές Ευρωπαϊκές χώρες, καθώς επίσης και το Scoop<sup>2</sup> που αφορά κυρίως την Αμερική. Οι συγκεκριμένες υπηρεσίες δραστηριοποιούνται εδώ και χρόνια οπότε έχουν καταφέρει να καλύπτουν διαδρομές ακόμα και μεταξύ διαφορετικών χωρών (BlaBlaCar) ή για διαδρομές ρουτίνας από/ προς τη δουλειά (Scoop).

Επειδή στην Ελλάδα δεν υπήρξε ποτέ κάποια αντίστοιχη υπηρεσία, καλύφθηκε αυτή η ανάγκη μέσω ομάδων στο Facebook, οι οποίες χρησιμοποιούνται από μεγάλο αριθμό ανθρώπων. Το μειονέκτημα αυτών είναι πως απαιτείται η ύπαρξη λογαριασμού στο Facebook και το ότι εκτός της μιας πιο γνωστής ομάδας<sup>3</sup> υπάρχουν και άλλες που απευθύνονται σε συγκεκριμένες διαδρομές από νομό σε νομό. Παραδείγματα: Κέρκυρας<sup>4</sup>, Κρήτης<sup>5</sup>, Ξάνθης<sup>6</sup>, Πάτρας<sup>7</sup>.

---

<sup>1</sup> <https://www.blablacar.com/>

<sup>2</sup> <https://play.google.com/store/apps/details?id=com.takescoop.android.scoopandroid&hl=en&gl=US>

<sup>3</sup> <https://www.facebook.com/groups/sharetheride/>

<sup>4</sup> <https://www.facebook.com/groups/330672663711375/>

<sup>5</sup> <https://www.facebook.com/groups/1705738889675469/>

<sup>6</sup> <https://www.facebook.com/groups/ShareTheRideXanthi/>

<sup>7</sup> <https://www.facebook.com/groups/329194634541753/>

## 2. PROLOGUE

This thesis concerns creating a carpooling web application named Share the Ride. The main goal is giving the opportunity to travelers sharing the expenses of their journey and also decreasing the environmental footprint, improving pollution and traffic jams.

Share the Ride allows drivers to post their future trip's information so that passengers who travel at the same destination, share the expenses of their journey. Posting a journey is available only for drivers; passengers can only search for available routes and communicate with drivers through WhatsApp. Also this thesis is available only for traveling inside Greece.

There are similar applications abroad that have a huge impact on users. As a result, traveling is more affordable for more people. For example BlaBlaCar<sup>8</sup> works for many European countries and Scoop<sup>9</sup> is for the USA. Both of them have existed for some time so they have succeeded in supporting routes from different countries (BlaBlaCar) or daily routine journeys from/ to work (Scoop).

It is also good to notice that in Greece there isn't any similar application, but this necessity is covered from Facebook groups that are used by many people. The disadvantages of those are that a Facebook account is necessary and there are separate groups without an aggregation service handling the common need to share-the-ride groups from/ to specific places except for the most active group<sup>10</sup>. Examples: Corfu<sup>11</sup>, Crete<sup>12</sup>, Xanthi<sup>13</sup>, Patras<sup>14</sup>.

---

<sup>8</sup> <https://www.blablacar.com/>

<sup>9</sup> <https://play.google.com/store/apps/details?id=com.takescoop.android.scoopandroid&hl=en&gl=US>

<sup>10</sup> <https://www.facebook.com/groups/sharetheride/>

<sup>11</sup> <https://www.facebook.com/groups/330672663711375/>

<sup>12</sup> <https://www.facebook.com/groups/1705738889675469/>

<sup>13</sup> <https://www.facebook.com/groups/ShareTheRideXanthi/>

<sup>14</sup> <https://www.facebook.com/groups/329194634541753/>

### 3. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Για το **backend** χρησιμοποιήθηκε το Firebase<sup>15</sup> της Google και πιο συγκεκριμένα το Cloud Firestore. Είναι μια ευέλικτη, επεκτάσιμη βάση δεδομένων που αναλαμβάνει την υποστήριξη του backend για Android/ iOS / Web εφαρμογές. Επίσης είναι υπεύθυνο για την αυθεντικοποίηση των χρηστών, τη φιλοξενία και την ανακατεύθυνση της ιστοσελίδας σε εξωτερικό domain του papaki<sup>16</sup> με όνομα sharetheride.gr<sup>17</sup>. Επιπλέον, το Firebase παρέχει διάφορα στατιστικά που δείχνουν την επισκεψιμότητα της σελίδας, καθώς και άλλες χρήσιμες πληροφορίες για την ταχύτητα περιήγησης και ενδεχόμενα warnings τα οποία προκύπτουν.

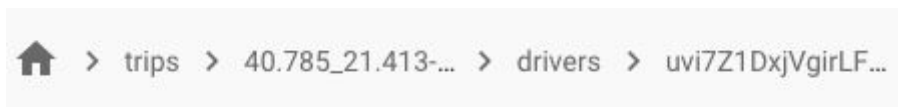
Για τον σχεδιασμό της βάσης στο Cloud Firestore μπορούν να χρησιμοποιηθούν τα απλά

Documents, τα πολλαπλά Collections και τα Subcollections μέσα σε Documents.

Συγκεκριμένα για την αρχιτεκτονική της βάσης χρειάστηκαν δύο Collections, trips και users όπου εξηγούνται παρακάτω.

- **trips:**

αποθηκεύονται σαν Documents οι διαδρομές και εμφωλευμένα σε αυτές σαν Documents οι drivers:



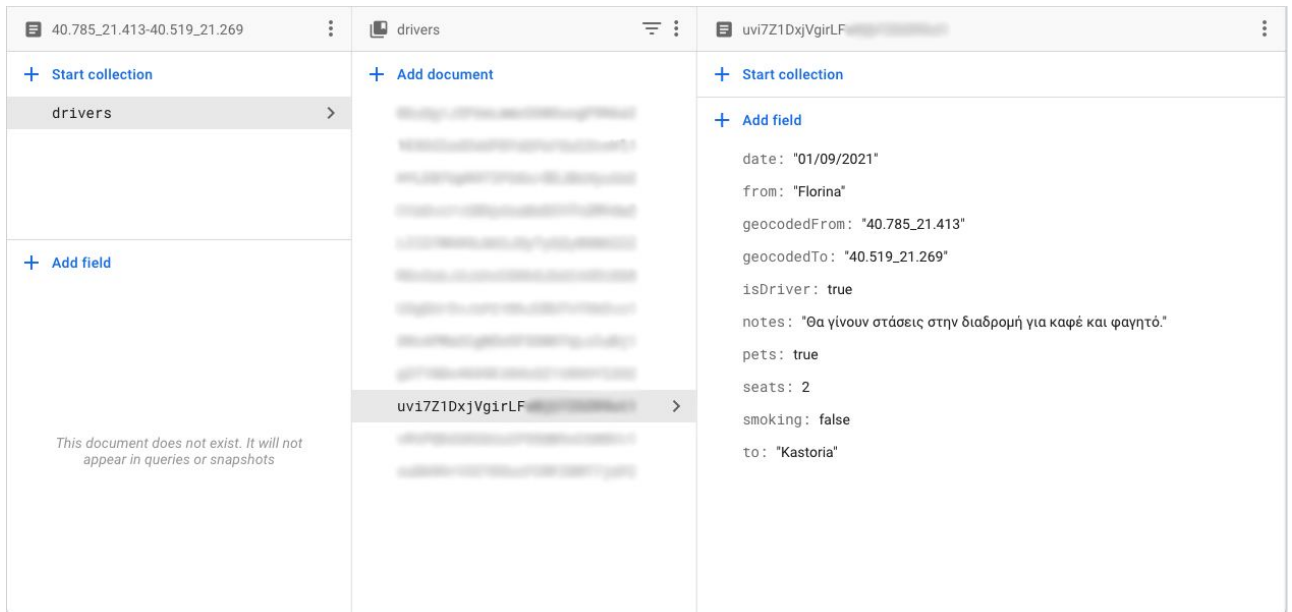
Είναι εμφανές πως οι διαδρομές έχουν την μορφή .../latFrom\_lngFrom-latTo\_lngTo/... για να διευκολύνεται η αναζήτηση των διαδρομών με βάση τα γεωγραφικά μήκη και πλάτη, επίσης εσωτερικά στους drivers εμφανίζονται τα user IDs των οδηγών που πρόκειται να κάνουν τη συγκεκριμένη διαδρομή (αν υπάρχουν), όπου και εκεί αποθηκεύονται όλες οι απαραίτητες πληροφορίες του κάθε ταξιδιού. Οι πληροφορίες αυτές είναι: η ημερομηνία, ο αρχικός και τελικός προορισμός, το αν είναι οδηγός ή επιβάτης, τα διαθέσιμα καθίσματα, αν επιτρέπονται τα κατοικίδια ή το κάπνισμα και τέλος οι σημειώσεις όπου ο οδηγός σημειώνει επιπλέον λεπτομέρειες για το ταξίδι που θα πραγματοποιήσει.

---

<sup>15</sup> <https://firebase.google.com/>

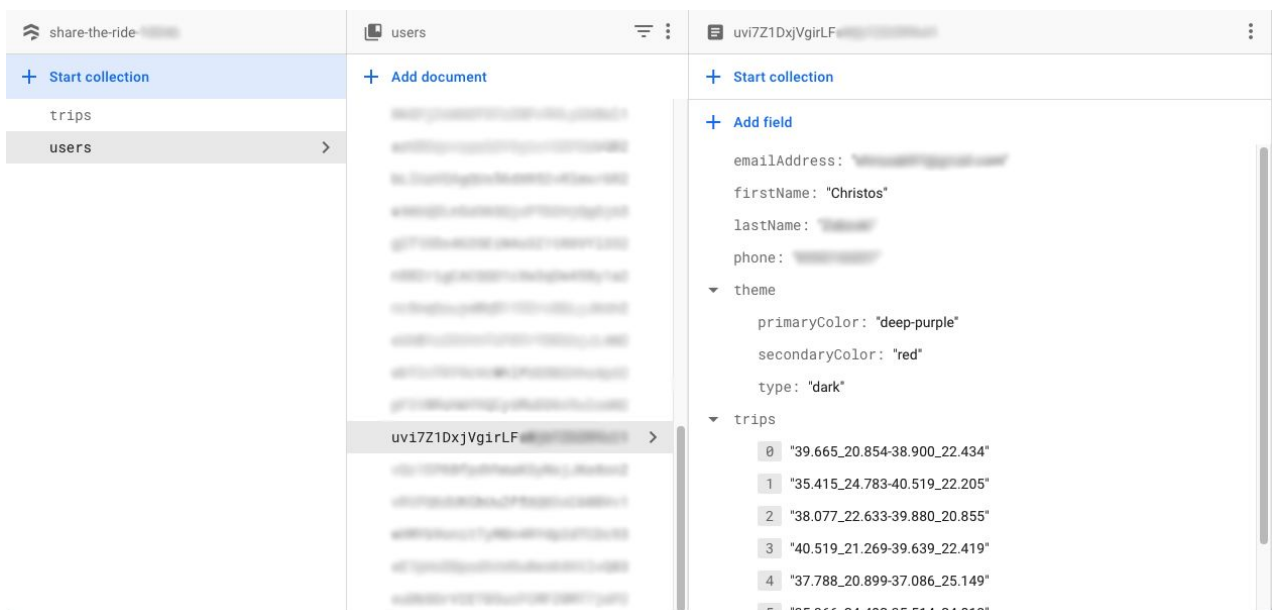
<sup>16</sup> <https://www.papaki.com/el>

<sup>17</sup> <https://www.sharetheride.gr/>



### • Users:

Αποθηκεύονται σαν Documents οι χρήστες με ξεχωριστό user ID ο καθένας και μέσα στον κάθε user αποθηκεύονται πληροφορίες όπως το email, το όνοματεπώνυμο, το τηλέφωνο, το username, οι ρυθμίσεις του κάθε χρήστη για τα χρώματα που θέλει στον λογαριασμό του και τέλος, ένας πίνακας με τα ενεργά ταξίδια του χρήστη. Αυτό συμβαίνει όταν δημιουργείται ένας νέος χρήστης, κάνοντας εγγραφή μέσω της εφαρμογής.



## 4. ΣΥΝΔΕΣΗ ΜΕ ΤΟ GOOGLE MAPS API

Για τη χρήση του Google Maps στα Autocomplete fields του αρχικού και τελικού προορισμού χρειάζεται η σύνδεση του Google Cloud Platform με τον υπάρχοντα λογαριασμό στο Firebase καθώς επίσης και το billing του. Στη συνέχεια η ενεργοποίηση των API που χρησιμοποιούνται, τα οποία είναι το **Maps JavaScript API**<sup>18</sup>, **Places API**<sup>19</sup>, **Geocoding API**<sup>20</sup> και η χρήση του Google Maps Places Autocomplete<sup>21</sup>.

**Share-the-Ride/src/components/MapsFields/MapsFields.js**



```
fetch(
  { input: inputValue, componentRestrictions: { country: "gr" } },
  (results) => {
    if (active) {
      let newOptions = [];

      if (value) {
        newOptions = [value];
      }

      if (results) {
        newOptions = [...newOptions, ...results];
      }

      setOptions(newOptions);
    }
  }
);
```

Με την παραπάνω μέθοδο ορίζεται σαν παράμετρος, πως επιτρέπονται οι αναζητήσεις μόνο για προορισμούς εντός Ελλάδας.

---

<sup>18</sup> <https://developers.google.com/maps/documentation/javascript/overview>

<sup>19</sup> <https://developers.google.com/places/web-service/overview>

<sup>20</sup> <https://developers.google.com/maps/documentation/geocoding/overview>

<sup>21</sup> <https://material-ui.com/components/autocomplete/>

## Share-the-Ride/src/components/MapsFields/MapsFields.js

```
<Autocomplete
  id={props.id}
  getOptionLabel={(option) =>
    typeof option === "string" ? option : option.description
  }
  filterOptions={(x) => x}
  options={options}
  autoComplete
  autoSelect
  autoHighlight
  includeInputInList
  filterSelectedOptions
  value={value ? value : history[props.id]}
  defaultValue={history[props.id]}
  onChange={(event, newValue) => {
    setOptions(newValue ? [newValue, ...options] : options);
    setValue(newValue);
    props.handleChange(newValue, `${props.id}`);

    if (!geocoder.current && window.google) {
      geocoder.current = new window.google.maps.Geocoder();
    }
    if (!geocoder.current) {
      return undefined;
    }

    newValue &&
    newValue.place_id &&
    geocoder.current.geocode(
      { placeId: newValue.place_id },
      (place, status) => {
        if (status === window.google.maps.places.PlacesServiceStatus.OK) {
          let lat = place[0].geometry.location.lat().toFixed(3);
          let lng = place[0].geometry.location.lng().toFixed(3);

          let location = place[0].address_components[0].long_name;

          props.handleChange(`${lat}_${lng}`, `geocoded${props.label}`);
          props.handleChange(location, `${props.id}`);
        }
      }
    );
  }}
  onInputChange={(event, newInputValue) => {
    setInputValue(newInputValue);
  }}
  renderInput={(params) => (
    <TextField
      {...params}
      label={props.label}
      variant="outlined"
      fullWidth
      required
      fullWidth
      value={history[props.id]}
    />
  )}
```

Όταν ο χρήστης αναζητά κάποιο προορισμό μέσω του Autocomplete τότε κάθε entry έχει ένα μοναδικό placeID, με βάση το οποίο ορίζονται τα lat, lng τα οποία είναι το γεωγραφικό μήκος και πλάτος του entry. Παρακάτω υπάρχουν όλα τα props της κάθε τοποθεσίας, όπως φαίνονται από debugger:

```
> newValue
< {description: "Florina, Greece", matched_substrings: Array(1), place_id: "ChIJnbon6P1AVxMR2RheIRaMi-g", reference: "ChIJnbon6P1AVxMR2RheIRaMi-g", structured_formatting: {_, _}, ...}
  description: "Florina, Greece"
  matched_substrings: [{_}]
  place_id: "ChIJnbon6P1AVxMR2RheIRaMi-g"
  reference: "ChIJnbon6P1AVxMR2RheIRaMi-g"
  structured_formatting: {main_text: "Florina", main_text_matched_substrings: Array(1), secondary_text: "Greece"}
  terms: (2) [{_}, {_}]
  types: (3) ["locality", "political", "geocode"]
  __proto__: Object
```

Από αυτά κρατούνται μόνο τα τρία πρώτα ψηφία μετά την υποδιαστολή και αποθηκεύονται με την μορφή lat\_lng για κάθε τοποθεσία. Επειδή οι καταχωρήσεις στη βάση δεδομένων γίνονται με βάση τα lat / lng είναι πολύ σημαντικό ότι ο χρήστης να μπορεί να πραγματοποιεί αναζητήσεις με Ελληνικούς αλλά και Αγγλικούς χαρακτήρες.

## 5. ΣΧΕΔΙΑΣΜΟΣ ΔΙΕΠΑΦΗΣ ΤΟΥ ΧΡΗΣΤΗ

Το **frontend** κατασκευάστηκε με τη χρήση της Javascript<sup>22</sup> και πιο συγκεκριμένα με την βοήθεια της βιβλιοθήκης ReactJS<sup>23</sup> και του Material-UI<sup>24</sup> framework, το οποίο παρέχει κάποια έτοιμα themes, και βοηθά να δημιουργηθούν γρήγορα και ευέλικτα React components.

Συγκεκριμένα για την αρχική σελίδα χρησιμοποιήθηκε έτοιμο theme<sup>25</sup> του Material-UI (προσαρμόζοντάς το στις ανάγκες της ιστοσελίδας). Εκεί παρέχονται γενικές πληροφορίες σε σχέση με τη σελίδα, τα link των κοινωνικών δικτύων της υπηρεσίας και το email επικοινωνίας με το support της. Επίσης ο σκελετός της υπηρεσίας και η αυθεντικοποίηση των χρηστών είναι από εδώ<sup>26</sup>, όπου δίνεται και η δυνατότητα δημιουργίας λογαριασμού.

Αρχικά ελέγχεται αν ο χρήστης είναι εγγεγραμμένος ή όχι. Σε περίπτωση που δεν είναι πρέπει να δημιουργηθεί λογαριασμός. Στη συνέχεια, για την αναζήτηση ή δημιουργία μιας διαδρομής συμπληρώνονται τα πεδία Από, Προς, Ημερομηνία και αν είναι οδηγός ή επιβάτης. Για την καλύτερη περιγραφή της λειτουργικότητας

<sup>22</sup> <https://www.javascript.com/>

<sup>23</sup> <https://reactjs.org/>

<sup>24</sup> <https://material-ui.com/>

<sup>25</sup> <https://material-ui.com/store/items/onepirate/>

<sup>26</sup> <https://github.com/rmuif/web>

παρατίθεται το παράδειγμα: έστω πως ο αρχικός προορισμός είναι η Φλώρινα και ο τελικός προορισμός είναι η Καστοριά.

The screenshot shows a mobile app interface titled "Share the Ride". At the top, there's a progress bar with three steps: "1 Plan your trip" (active), "2 Details", and "3 Review". Below the progress bar, the heading "Fill the forms" is centered. The form contains three main sections: "From \*" with a dropdown menu showing "Φλώρινα, Greece", "To \*" with a dropdown menu showing "Καστοριά, Greece", and "Date picker \*" with a date field showing "01/09/2021" and a calendar icon. Below these, there's a toggle switch for "Passenger" (selected) and "Driver". At the bottom, there's a checkbox labeled "Select this if you want to post a trip" and a red "NEXT" button.

#### Material-UI used components:

```
<Stepper />
<Grid />
<TextField />
<KeyboardDatePicker />
<FormControlLabel />
<Button />
```

Τα επόμενα πεδία που συμπληρώνει ο χρήστης, καθώς και η λειτουργικότητα στη συνέχεια της υπηρεσίας, είναι διαφορετική και εξαρτάται από το αν ο χρήστης δήλωσε οδηγός ή επιβάτης.

## 6. ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΟΔΗΓΟΥ

Σε αυτή την περίπτωση τα επόμενα πεδία είναι οι Διαθέσιμες Θέσεις, οι Σημειώσεις (περαιτέρω πληροφορίες του ταξιδιού), αν επιτρέπονται ή αν θα υπάρχουν στο όχημα κατοικίδια και αν επιτρέπεται ή όχι το κάπνισμα κτλ.

### Material-UI used components:

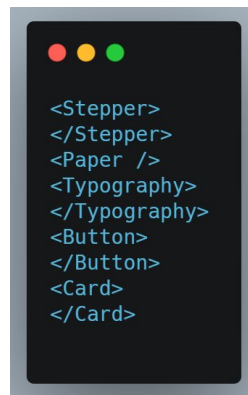
```

<Stepper>
</Stepper>
<TextField/>
<FormControlLabel/>
<Button>
</Button>

```

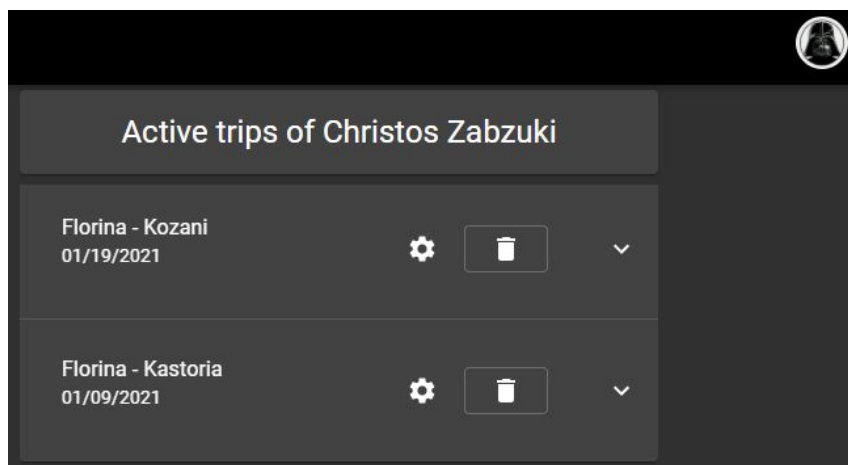
Στη συνέχεια γίνεται ένας τελικός έλεγχος των πεδίων που συμπληρώθηκαν.

### Material-UI used components:

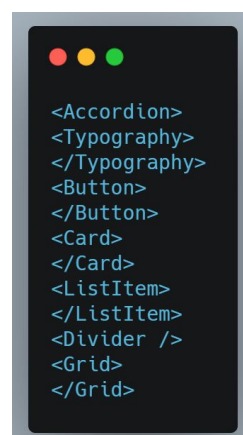


Αν όλα είναι εντάξει και ο χρήστης συνεχίσει, εμφανίζεται μήνυμα επιτυχούς διαδικασίας.

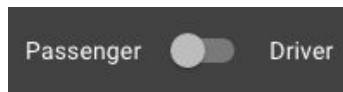
Τέλος, ένας οδηγός έχει τη δυνατότητα τροποποίησης ή διαγραφής των ενεργών ταξιδιών του, πηγαίνοντας στο Activity μέσω του μενού.



### Material-UI used components:



## 7. ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΕΠΙΒΑΤΗ



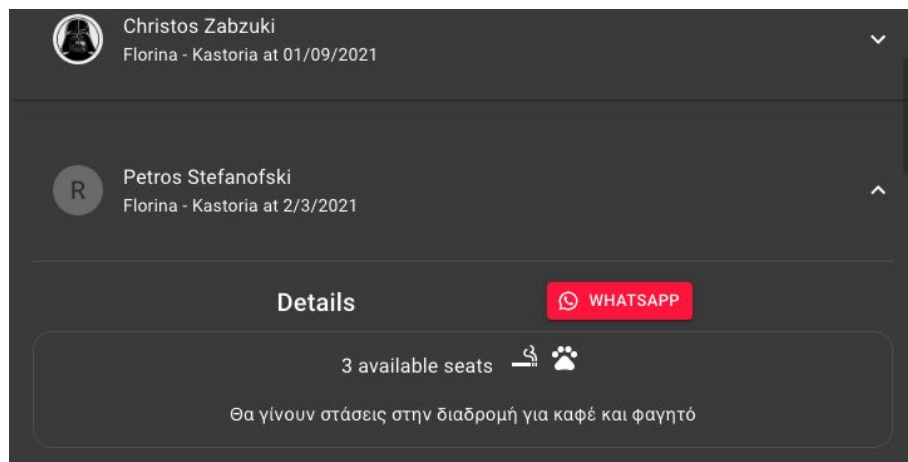
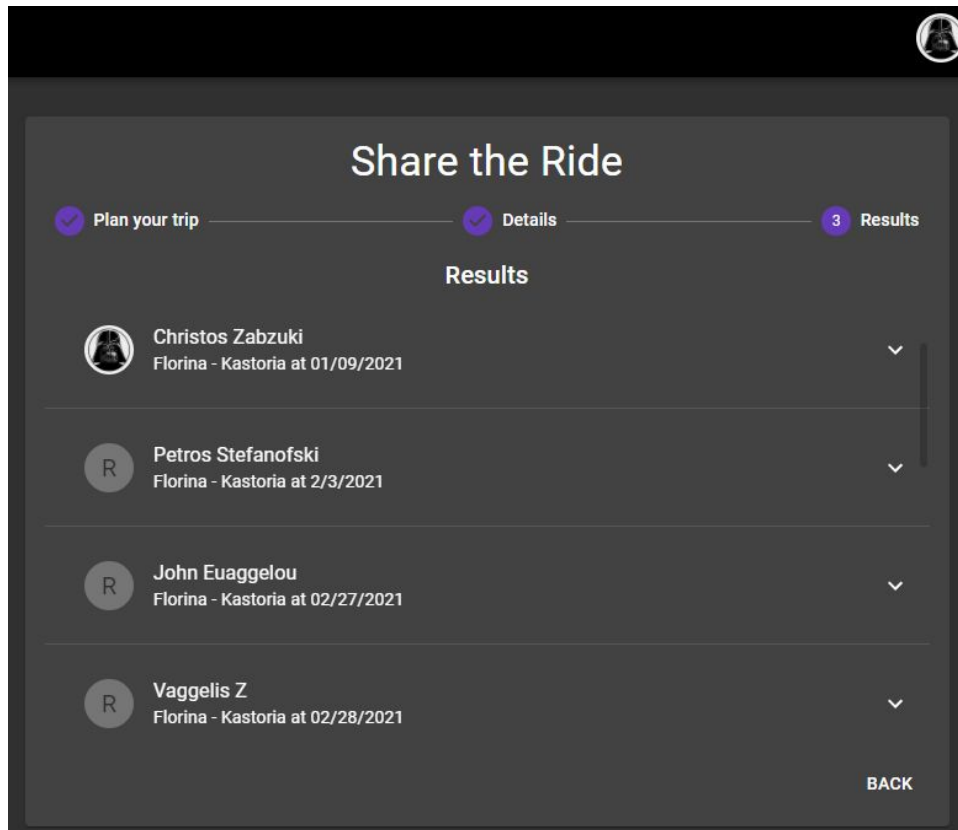
Εφόσον ο χρήστης στην αρχική σελίδα όπου επιλέγεται αρχική / τελική διαδρομή και η ημερομηνία έχει επιλέξει το Passenger τότε εμφανίζονται τα επόμενα πεδία. Αυτά είναι: ο αριθμός των Επιβατών και των Αποσκευών.

A screenshot of a mobile app interface titled 'Share the Ride'. At the top right is a circular profile icon. Below the title is a progress bar with three steps: '1 Plan your trip' (checked), '2 Details' (active), and '3 Results'. Under the 'Details' step, there is a section titled 'Passenger'. It contains two dropdown menus: 'Travelers' with the value '1' and 'Luggages' with the value '2'. At the bottom right of the form are two buttons: 'BACK' and 'NEXT' (highlighted in red).

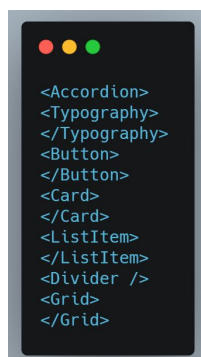
**Material-UI used components:**



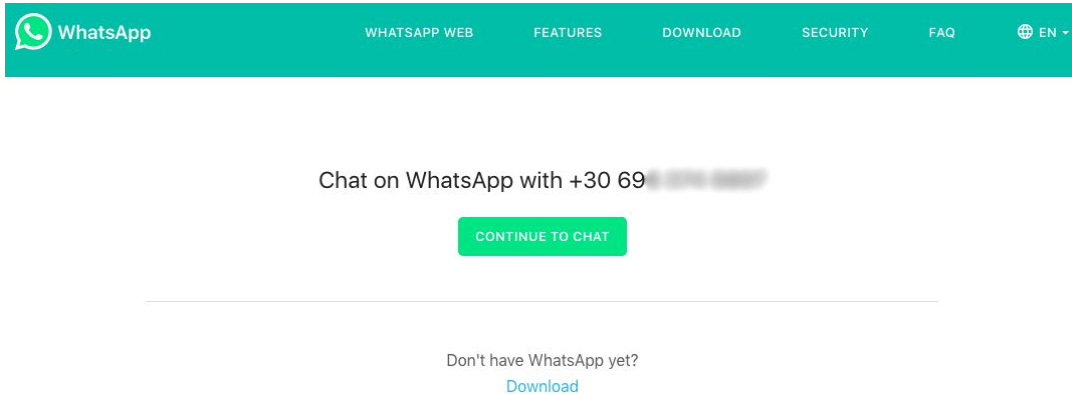
Συμπληρώνοντας τα παραπάνω δεδομένα εμφανίζονται σαν αποτελέσματα τα ταξίδια που έχουν ίδιο αρχικό και τελικό προορισμό, καθώς επίσης και **ίδια ή μεταγενέστερη ημερομηνία από αυτή που αναζητεί ο χρήστης.**



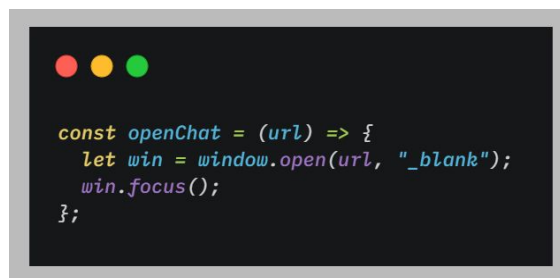
### Material-UI used components:



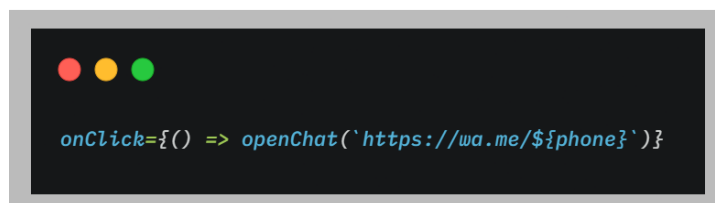
Τέλος πατώντας το κουμπί WhatsApp ανοίγει το WhatsApp web ώστε να γίνει η απαραίτητη συνεννόηση για το ταξίδι.



Για να επιτευχθεί αυτό χρειάστηκε να δημιουργηθεί το function



Το οποίο καλείται βάζοντας την παρακάτω εντολή μέσα στον onClick event handler του WhatsApp κουμπιού



## 8. ΠΑΡΑΔΕΙΓΜΑΤΑ ΕΠΙΛΕΓΜΕΝΟΥ ΚΩΔΙΚΑ

Για την ορθή και ομαλή λειτουργία της υπηρεσίας χρειάστηκε να κατασκευαστούν ξεχωριστές μέθοδοι, οι οποίες καλύπτουν ανάγκες όπως για παράδειγμα την αποθήκευση, την εμφάνιση, την διαγραφή ή και την ανανέωση δεδομένων. Οι συγκεκριμένες μέθοδοι θεωρούνται αρκετά σημαντικές, γι' αυτό και αναλύονται παρακάτω.

## 8.1. ΑΠΟΘΗΚΕΥΣΗ ΔΕΔΟΜΕΝΩΝ ΣΤΟ FIREBASE

Η διαδικασία της αποθήκευσης δεδομένων απαιτεί την υλοποίηση μιας μεθόδου στο authentication service του project καθώς επίσης και την κλήση της στο component που αποθηκεύονται τα δεδομένα.

### Share-the-Ride/src/services/authentication.js

```
authentication.storeDataToDB = (
  from,
  to,
  date,
  isDriver,
  seats,
  notes,
  smoking,
  pets,
  geocodedFrom,
  geocodedTo
) => {
  return new Promise((resolve, reject) => {
    const currentUser = auth.currentUser;

    if (!currentUser) {
      reject();

      return;
    }

    const uid = currentUser.uid;

    if (!uid) {
      reject();

      return;
    }

    let fromTo = authentication.manipulateFromTo(geocodedFrom, geocodedTo);

    const referenceUsers = firestore.collection("users").doc(uid);

    if (!referenceUsers) {
      reject();

      return;
    }

    referenceUsers
      .update({
        trips: firebase.firestore.FieldValue.arrayUnion(fromTo),
      })
      .then(() => {
        analytics.logEvent("update_trip");

        const reference = firestore
          .collection("trips")
          .doc(fromTo)
          .collection("drivers")
          .doc(uid);

        if (!reference) {
          reject();

          return;
        }
      })
  });
}
```

```

reference
.set({
  from: from,
  to: to,
  date: date,
  isDriver: isDriver,
  seats: seats,
  notes: notes,
  smoking: smoking,
  pets: pets,
  geocodedFrom: geocodedFrom,
  geocodedTo: geocodedTo,
})
.then((value) => {
  analytics.logEvent("update_trip");

  resolve(value);
})
.catch((reason) => {
  reject(reason);
});
})
.catch((reason) => {
  reject(reason);
});
});
};

```

### Share-the-Ride/src/components/LoggedIn/LoggedIn.js

```

storeDataToDB = () => {
  const {
    from,
    to,
    date,
    isDriver,
    seats,
    notes,
    smoking,
    pets,
    geocodedFrom,
    geocodedTo,
  } = this.state;
  authentication.storeDataToDB(
    from,
    to,
    date,
    isDriver,
    seats,
    notes,
    smoking,
    pets,
    geocodedFrom,
    geocodedTo
  );
};

```

Η μέθοδος `manipulateFromTo()`; επίσης βρίσκεται στο ίδιο αρχείο και στοχεύει στη λήψη του αρχικού και τελικού προορισμού και στη δημιουργία ενός καινούργιου,

λαμβάνοντας υπόψη και τους δύο προορισμούς, έχοντας την μορφή latFrom\_lngFrom-latTo\_lngTo.

```
authentication.manipulateFromTo = (from, to) => {  
  let fromTo = `${from}-${to}`;  
  return fromTo;  
};
```

Επιπλέον χρειάζεται η κλήση της μεθόδου, δηλαδή το `this.storeDataToDB()` που καλείται όταν έχουν ολοκληρωθεί τα βήματα.

## 8.2. ΕΜΦΑΝΙΣΗ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΤΟ FIREBASE

Η διαδικασία της εμφάνισης δεδομένων απαιτεί επίσης την ύπαρξη μιας μεθόδου στο authentication service του project, καθώς και την κλήση της στο component που θέλουμε να εμφανιστούν τα δεδομένα.

### Share-the-Ride/src/services/authentication.js

```
authentication.getDataFromDB = (from, to) => {  
  return new Promise((resolve, reject) => {  
    const currentUser = auth.currentUser;  
  
    if (!currentUser) {  
      reject();  
      return;  
    }  
  
    const uid = currentUser.uid;  
  
    if (!uid) {  
      reject();  
      return;  
    }  
  
    let fromTo = authentication.manipulateFromTo(from, to);  
    const reference = firestore  
      .collection("trips")  
      .doc(fromTo)  
      .collection("drivers");  
  
    if (!reference) {  
      reject();  
      return;  
    }  
  
    reference.get().then((querySnapshot) => {  
      const tripWithUsersProm = (id, doc) => {  
        return new Promise((resolve, reject) => {  
          const usersRef = firestore.collection("users").doc(id);  
  
          if (!usersRef) {  
            reject();  
            return;  
          }  
        });  
      };  
    });  
  });  
};
```

```

usersRef
  .get()
  .then((user) => {
    doc.firstName = user.data().firstName;
    doc.lastName = user.data().lastName;
    doc.phone = user.data().phone;
    authentication
      .getImagesFromDB(id, doc)
      .then(() => {
        resolve(doc);
      })
      .catch((_error) => {
        console.log(
          `didn't find user's avatar in database, can't load it at results`
        );
        resolve(doc);
      });
  })
  .catch((error) => {
    reject(error);
  });
});

let trips = [];

querySnapshot.forEach((doc) => {
  trips.push(tripWithUsersProm(doc.id, doc.data()));
});

Promise.all(trips)
  .then((data) => {
    resolve(data);
  })
  .catch((error) => {
    reject(error);
  });
});
});
};

```

Η μέθοδος `getImagesFromDB()`, επίσης βρίσκεται στο ίδιο αρχείο και αποσκοπεί να πάρει την φωτογραφία του χρήστη από τη βάση δεδομένων.

### Share-the-Ride/src/services/authentication.js

```

authentication.getImagesFromDB = (id, doc) => {
  return new Promise((resolve, reject) => {
    const imagesRef = firebase
      .storage()
      .ref()
      .child("images")
      .child("avatars")
      .child(id);
    imagesRef.getDownloadURL().then((userAvatar) => {
      doc.avatar = userAvatar;
      resolve(doc);
    }).catch((_error) => {
      console.log(`didn't find user's avatar in database`);
      reject();
    });
  });
};

```

Η παρακάτω μέθοδος είναι μέρος μιας άλλης (renderRow) της οποίας επιστρέφονται τα αποτελέσματα (αν υπάρχουν), όλα σε μια συγκεκριμένη μορφή.

### Share-the-Ride/src/components/Results/Results.js

```
getDataFromDB = () => {
  const { geocodedFrom, geocodedTo, seats, date } = this.props.history;
  authentication
    .getDataFromDB(geocodedFrom, geocodedTo)
    .then((results) => {
      let activeTrips = results
        .filter((trip) => {
          const tripDate = new Date(trip.date);
          const currentDate = new Date(date);
          return (
            tripDate.getTime() >= currentDate.getTime() && trip.seats >= seats
          );
        })
        .sort((t1, t2) => {
          return new Date(t1.date) - new Date(t2.date);
        });

      this.setState({
        results: activeTrips,
        lookingForTrips: false,
      });
    })
    .catch((error) => console.log(error));
};

componentDidMount() {
  this.getDataFromDB();
}
```

Επιπλέον τα `results` & `lookingForTrips` δηλώνονται στον `constructor` όπως φαίνεται παρακάτω.

```
constructor(props) {
  super(props);
  this.state = {
    results: [],
    lookingForTrips: true,
  };
}
```

## 8.3. ΔΙΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ ΣΤΟ FIREBASE

Η διαγραφή μιας διαδρομής χρειάζεται μια μέθοδο στο authentication service και η κλήση της στο component που θέλουμε να διαγραφούν τα δεδομένα. Καλώντας την

διαγράφεται ο user στο trips/FromTo/drivers/**UserID** καθώς επίσης και από το **Array** των διαδρομών του χρήστη στο users/userID

### Share-the-Ride/src/services/authentication.js

```
authentication.deleteTrip = (geocodedFrom, geocodedTo) => {
  return new Promise((resolve, reject) => {
    const currentUser = auth.currentUser;

    if (!currentUser) {
      reject();

      return;
    }

    const uid = currentUser.uid;

    if (!uid) {
      reject();

      return;
    }

    let fromTo = authentication.manipulateFromTo(geocodedFrom, geocodedTo);
    const referenceUsers = firestore.collection("users").doc(uid);

    if (!referenceUsers) {
      reject();

      return;
    }

    referenceUsers
      .update({
        trips: firebase.firestore.FieldValue.arrayRemove(fromTo),
      })
      .then(() => {
        analytics.logEvent("update_trip");

        const reference = firestore
          .collection("trips")
          .doc(fromTo)
          .collection("drivers")
          .doc(uid);

        if (!reference) {
          reject();

          return;
        }

        reference
          .delete()
          .then((value) => {
            analytics.logEvent("delete_account");

            resolve(value);
          })
          .catch((reason) => {
            reject(reason);
          });
      })
      .catch((reason) => {
        reject(reason);
      });
  });
};
```

## Share-the-Ride/src/components/DeleteTripDialog/DeleteTripDialog.js

```
handleDeleteTrip = () => {
  const { routeItemDetails, handleIsDeleted } = this.props;
  const { geocodedFrom, geocodedTo } = routeItemDetails;
  authentication
    .deleteTrip(geocodedFrom, geocodedTo)
    .then(() => {
      handleIsDeleted();
    })
    .catch((error) => {
      console.log(` failed to delete trip with error ${error}`);
    });
};
```

## 8.4. ΑΝΑΝΕΩΣΗ ΔΕΔΟΜΕΝΩΝ ΣΤΟ FIREBASE

Η ανανέωση των δεδομένων καλείται όταν ο χρήστης προσπαθεί να τροποποιήσει ένα ενεργό του ταξίδι και απαιτεί επίσης την ύπαρξη μιας μεθόδου στο authentication service με την διαφορά ότι στην συγκεκριμένη γίνεται η χρήση δύο άλλων μεθόδων, της `deleteTrip()` και `storeDataToDB()`. Επιπλέον και εδώ η κλήση της γίνεται στο component που θέλουμε να εμφανιστούν τα δεδομένα.

## Share-the-Ride/src/services/authentication.js

```
authentication.updateDataToDB = (
  from,
  to,
  date,
  isDriver,
  seats,
  notes,
  smoking,
  pets,
  geocodedFrom,
  geocodedTo
) => {
  return authentication
    .deleteTrip(geocodedFrom, geocodedTo)
    .then(() => {
      authentication
        .storeDataToDB(
          from,
          to,
          date,
          isDriver,
          seats,
          notes,
          smoking,
          pets,
          geocodedFrom,
          geocodedTo
        )
        .catch((error) => {
          console.error(error);
        });
    })
    .catch((error) => {
      console.error(error);
    });
};
```

## Share-the-Ride/src/components/TripSettingsDialog/TripSettingsDialog.js

```
handleUpdateData = () => {
  const {
    from,
    to,
    date,
    seats,
    notes,
    smoking,
    pets,
    geocodedFrom,
    geocodedTo,
  } = this.state;
  authentication
    .updateDataToDB(
      from,
      to,
      date,
      true,
      seats,
      notes,
      smoking,
      pets,
      geocodedFrom,
      geocodedTo
    )
    .then(() => {
      this.props.updateIsUpdated({
        from,
        to,
        date,
        isDriver: true,
        seats,
        notes,
        smoking,
        pets,
        geocodedFrom,
        geocodedTo,
      });
    });
};
```

## 10. ΣΥΝΔΕΣΗ ΤΩΝ KEYS

Αρχικά όταν δημιουργείται ένα καινούργιο project στο firebase ορίζεται ένα όνομα, αν είναι επιθυμητή η χρήση των Google Analytics και όταν ολοκληρωθεί δηλώνεται η σύνδεση του με κάποιο web application. Δηλώνεται τίτλος και στη συνέχεια εμφανίζεται το Firebase SDK Snippet, το οποίο και πρέπει να τοποθετηθεί στο project στα αρχεία package.json και package-lock.json. Το object που χρειάζεται να αντιγραφεί είναι μιας συγκεκριμένης μορφής και είναι μοναδικό για κάθε εφαρμογή, όπως φαίνονται και παρακάτω.

```
// For Firebase JavaScript SDK v7.20.0 and later, `measurementId` is an optional field
var firebaseConfig = {
  apiKey: "API_KEY",
  authDomain: "PROJECT_ID.firebaseapp.com",
  databaseURL: "https://PROJECT_ID.firebaseio.com",
  projectId: "PROJECT_ID",
  storageBucket: "PROJECT_ID.appspot.com",
  messagingSenderId: "SENDER_ID",
  appId: "APP_ID",
  measurementId: "G-MEASUREMENT_ID",
};
```

## 11. HOSTING

Το Hosting όπως αναφέρθηκε και προηγουμένως, παρέχεται από τις υπηρεσίες του Firebase και οι εντολές που χρειάστηκαν για να ενωθεί το Firebase Hosting με το project, είναι οι παρακάτω.

### Set up Firebase Hosting

#### 1 Install Firebase CLI

To host your site with Firebase Hosting, you need the Firebase CLI (a command line tool).

Run the following [npm](#) command to install the CLI or update to the latest CLI version.

```
$ npm install -g firebase-tools
```

Doesn't work? Take a look at the [Firebase CLI reference](#) or change your [npm permissions](#)

- ☐ Also show me the steps to add the Firebase JavaScript SDK to my web app  
The SDK includes Cloud Firestore, Authentication, Performance Monitoring and more. It can be added now or later.

Next

#### ✓ Install Firebase CLI

#### 2 Initialise your project

Open a terminal window and navigate to or create a root directory for your web app

Sign in to Google

```
$ firebase login
```

Initiate your project

Run this command from your app's root directory:

```
$ firebase init
```

Next

Στη συνέχεια ακολουθούν μια σειρά από ερωτήσεις και επιλογές που πρέπει να απαντηθούν προσεκτικά, όπως φαίνεται στα screenshots.

```
#####
##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
#####
##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##      ##      ##      ##
#####

You're about to initialize a Firebase project in this directory:
/home/.../firebase-hosting-demo

? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm
your choices.
  ○ Database: Deploy Firebase Realtime Database Rules
  ○ Firestore: Deploy rules and create indexes for Firestore
  ○ Functions: Configure and deploy Cloud Functions
  > ● Hosting: Configure and deploy Firebase Hosting sites
  ○ Storage: Deploy Cloud Storage security rules
```

```
? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm
your choices. Hosting: Configure and deploy Firebase Hosting sites

=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option:
> Use an existing project
  Create a new project
  Add Firebase to an existing Google Cloud Platform project
  Don't set up a default project
```

```
=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? build
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? File build/index.html already exists. Overwrite? (y/N) N
```

Επιπλέον στην ερώτηση για το αν το Firebase πρέπει να διαμορφωθεί ως εφαρμογή μιας σελίδας εισάγεται Y (Ναι).

Η τελευταία επιλογή είναι αν το Firebase πρέπει να αντικαταστήσει το υπάρχον build/index.html. Σε περίπτωση που δεν έχει δημιουργηθεί ακόμα μια έκδοση για την εφαρμογή, εισάγεται N (Όχι) για αυτήν την επιλογή.

Ο σύνδεσμος που παρέχεται από το firebase είναι της αντίστοιχης μορφής appname.firebaseio.com<sup>27</sup>, επειδή όμως χρησιμοποιήθηκε εξωτερικός domain, το hosting του Firebase κάνει Redirect στον εξωτερικό domain του paraki, που είναι sharetheride.gr<sup>28</sup>.

<sup>27</sup> <https://appname.firebaseio.com/>

<sup>28</sup> <https://www.sharetheride.gr/>

## 12. ΕΓΚΑΤΑΣΤΑΣΗ ΕΦΑΡΜΟΓΗΣ (DEPLOYMENT)

Ό,τι αλλαγή πραγματοποιείται ελέγχεται τοπικά μέσω localhost και μόνο όταν αυτό που υλοποιείται έχει ολοκληρωθεί σε ικανοποιητικό βαθμό, τότε γίνεται το deployment, έτσι ώστε να ανανεωθεί και η σελίδα που υπάρχει online. Γι' αυτή τη διαδικασία απαιτείται η χρήση του Node Package Manager (npm<sup>29</sup>) και υλοποιείται πρώτα το build και στην συνέχεια το deploy της εφαρμογής. Οι εντολές που χρειάζονται είναι οι παρακάτω και μόλις εκτελεστούν η online σελίδα είναι πλήρως ανανεωμένη.

```
→ share-the-ride git:(develop) git clone https://github.com/Zabzuki/
```

```
→ share-the-ride git:(develop) npm install
```

```
→ share-the-ride git:(develop) x npm build
```

```
→ share-the-ride git:(develop) x firebase deploy
```

---

<sup>29</sup> <https://www.npmjs.com/>

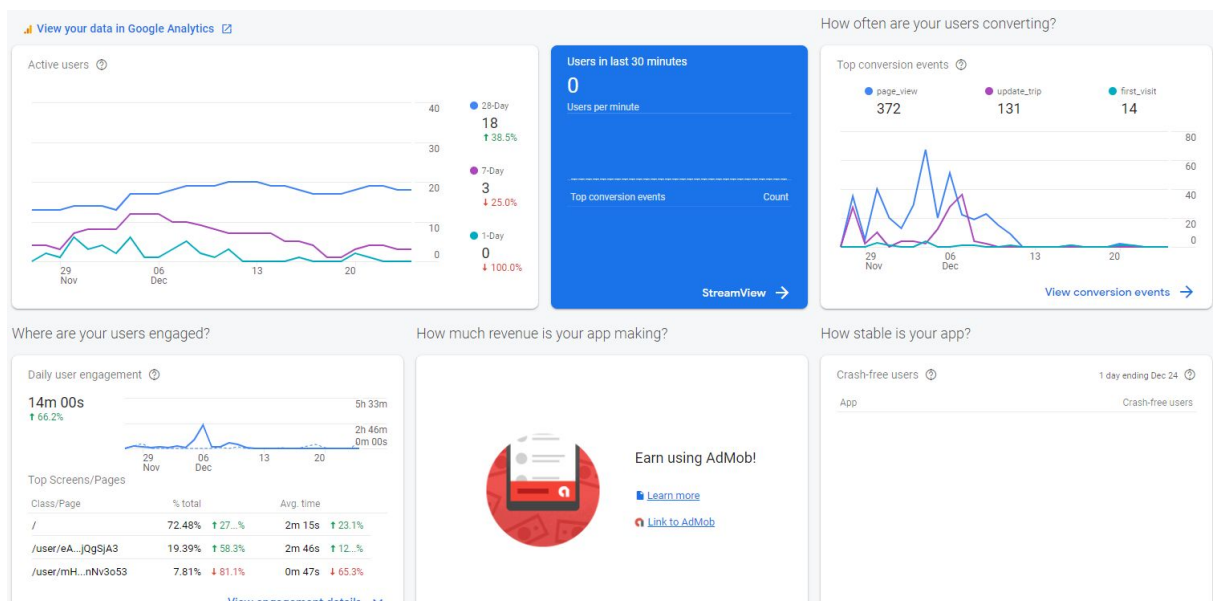
### 13. ΣΥΜΠΕΡΑΣΜΑΤΑ

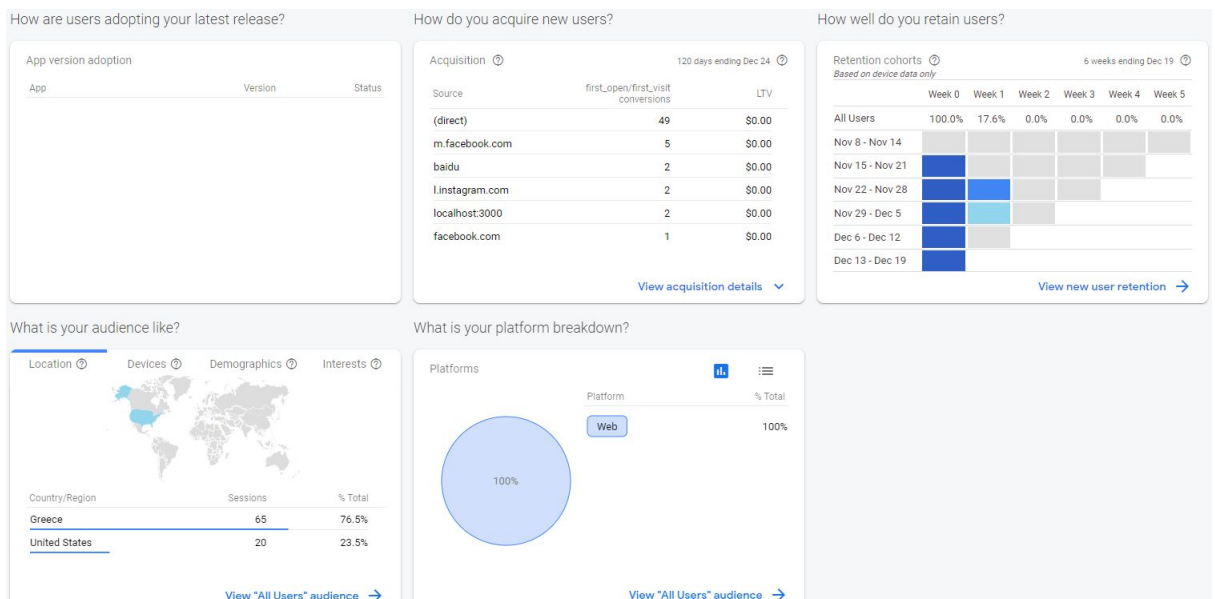
Σε αυτό το άρθρο αναλύθηκε εν συντομία η ιδέα καθώς και η ως ένα βαθμό υλοποίηση της υπηρεσίας συνεπιβατισμού, με την χρήση διαφόρων web τεχνολογιών. Στη συνέχεια, περιγράφηκε σε βάθος το πλάνο υλοποίησης, καθώς και η αρχιτεκτονική της για backend, frontend, με συγκεκριμένα παραδείγματα. Συμπερασματικά, από τη δουλειά που χρειάστηκε συνολικά η ιστοσελίδα, προκύπτει πως είναι ένα έργο πολύπλευρο με πολλές προεκτάσεις, το οποίο συνδυάζει πολλές και διαφορετικές τεχνολογίες που προσφέρουν μια συνολικότερη και πιο σφαιρική εμπειρία στην ανάπτυξη εφαρμογών και λογισμικού, κάνοντας χρήση στο μέγιστο των γνώσεων που προέκυψαν από τη φοίτηση στο τμήμα Πληροφορικής. Επιπροσθέτως θεωρείται κρίσιμο το γεγονός της ύπαρξης μιας άκρως ενεργούς κοινότητας ανθρώπων ικανής να υποστηρίξει τον λόγο ύπαρξης της συγκεκριμένης σελίδας, κάτι που μας αποδεικνύει ότι οι στόχοι μας είναι ρεαλιστικοί και έγκυροι.

## 14. ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΕΚΤΑΣΕΙΣ

Οι μελλοντικές προεκτάσεις της συγκεκριμένης ερευνητικής εργασίας που θα βελτίωναν αισθητά τη χρήση αλλά και τη συνολική λειτουργικότητα της είναι οι παρακάτω:

- Δημιουργία chat για την άμεση επικοινωνία των χρηστών μέσω της εφαρμογής, χωρίς τη χρήση κάποιας τρίτης εφαρμογής.
- Δημιουργία εξ'ολοκλήρου καινούργιας Βάσης Δεδομένων χωρίς την χρήση του Firebase, για την ύπαρξη περισσότερων δυνατοτήτων από την πλευρά του developer.
- Υποστήριξη συστήματος βαθμολόγησης καθώς και σχολιασμού των χρηστών, έτσι ώστε να περιορίζονται οι ενδεχόμενες ανάρμοστες συμπεριφορές που μπορεί να προκύψουν.
- Αυτόματη διαγραφή μιας διαδρομής σε περίπτωση που υπάρξει πληρότητα.
- Δυνατότητα ανεβάσματος μιας διαδρομής και από τους επιβάτες και όχι μόνο από τους οδηγούς.
- Καλύτερος έλεγχος και διαχείριση των Google Analytics καθώς και του usage tracking. Η τωρινή κατάσταση φαίνεται παρακάτω:





- Δυνατότητα εμφάνισης ταξιδιών σε ταξιδιώτες που βρίσκονται σε ενδιαμέσους προορισμούς. Για το συγκεκριμένο ζήτημα πραγματοποιήθηκε έρευνα όπου προκύπτουν τα βήματα που πρέπει να ακολουθηθούν για την υλοποίηση αυτού.

Απαιτείται η ενεργοποίηση του Directions API καθώς επίσης και αρκετές αλλαγές στο UI της εφαρμογής για την καλύτερη πλοήγηση των χρηστών. Συγκεκριμένα θα ήταν πιο εύχρηστο αν μετά την επιλογή αρχικού / τελικού προορισμού και ημερομηνίας, οι οδηγοί να επιλέγουν τη διαδρομή που θα ακολουθήσουν απ' όλες τις διαθέσιμες επιλογές μέσω ενός χάρτη (σαν την αναζήτηση δύο προορισμών στο Google Maps). Στη συνέχεια ιδανικά θα εμφανίζεται το ταξίδι του οδηγού και στα αποτελέσματα αναζήτησης από κάποια ενδιαμέση τοποθεσία.

Με την χρήση του Postman<sup>30</sup> και με βάση την τεκμηρίωση του Directions API<sup>31</sup>, αν τεθεί σαν αρχικός προορισμός η Αθήνα και τελικός η Θεσσαλονίκη, προκύπτει ένα HTTP GET request της παρακάτω μορφής, όπου αποτυπώνονται δεδομένα όπως το ID του κάθε προορισμού, τα lat,lng τους καθώς επίσης και τα lat,lng του πιο βόρειου και νότιου σημείου.

<sup>30</sup> <https://www.postman.com/>

<sup>31</sup> [https://developers.google.com/maps/documentation/directions/overview?\\_ga=2.75077586.199989889.4.1610317668-1993556290.1610317668#ExampleRequests](https://developers.google.com/maps/documentation/directions/overview?_ga=2.75077586.199989889.4.1610317668-1993556290.1610317668#ExampleRequests)

```
"geocoded_waypoints": [
  {
    "geocoder_status": "OK",
    "place_id": "ChIJ8UNwBh-9oRQR3Y1mdkU1Nic",
    "types": [
      "locality",
      "political"
    ]
  },
  {
    "geocoder_status": "OK",
    "place_id": "ChIJ7eAoFPQ4qBQRqXTVuBXnugk",
    "types": [
      "locality",
      "political"
    ]
  }
],
"routes": [
  {
    "bounds": {
      "northeast": {
        "lat": 40.6550659,
        "lng": 23.8555195
      },
      "southwest": {
        "lat": 37.983814,
        "lng": 22.4597657
      }
    },
    "copyrights": "Map data ©2021",
    "legs": [
      {
        "distance": {
          "text": "502 km",
          "value": 501989
        },
        "duration": {
          "text": "5 hours 7 mins",
          "value": 18437
        },
        "end_address": "Thessaloniki, Greece",
        "end_location": {
          "lat": 40.6400672,
          "lng": 22.9444164
        },
        "start_address": "Athens, Greece",
        "start_location": {
          "lat": 37.983814,
          "lng": 23.7275412
        }
      },
    ]
  }
]
```

Στη συνέχεια στο ίδιο JSON αρχείο εμφανίζονται όλες οι πληροφορίες σε σχέση με την διαδρομή μεταξύ των δύο πόλεων, απ' όλες τις εναλλακτικές διαδρομές που μπορούν να ακολουθηθούν, καθώς και σχετικές πληροφορίες σε σχέση με την κάθε τοποθεσία που διασχίζεται.

```

"steps": [
  {
    "distance": {
      "text": "31 m",
      "value": 31
    },
    "duration": {
      "text": "1 min",
      "value": 6
    },
    "end_location": {
      "lat": 37.9837749,
      "lng": 23.7278856
    },
    "html_instructions": "Head <b>east</b> on <b>Pl. Omonias</b> toward <b>Athinas</b>",
    "polyline": {
      "points": "yuyfFchyoCAQBYDY"
    },
    "start_location": {
      "lat": 37.983814,
      "lng": 23.7275412
    },
    "travel_mode": "DRIVING"
  },
  {
    "distance": {
      "text": "0.1 km",
      "value": 124
    },
    "duration": {
      "text": "1 min",
      "value": 26
    },
    "end_location": {
      "lat": 37.9831703,
      "lng": 23.7289558
    },
    "html_instructions": "Continue onto <b>Stadiou</b>",
    "polyline": {
      "points": "quyfFijyoC@_@S?MBW@IDIBEDIDG@?ZYZY\\\\"
    },
    "start_location": {
      "lat": 37.9837749,
      "lng": 23.7278856
    },
    "travel_mode": "DRIVING"
  },
  {
    "distance": {
      "text": "0.1 km",
      "value": 99
    },
    "duration": {
      "text": "1 min",
      "value": 36
    },
    "end_location": {
      "lat": 37.9839027,
      "lng": 23.7293523
    },
    "html_instructions": "Turn <b>left</b> onto <b>Aiolou</b>",
    "maneuver": "turn-left",
    "polyline": {
      "points": "yqyfF_qyoC?E?G?KEIGIa@Ia@Gc@G{E"
    },
    "start_location": {
      "lat": 37.9831703,
      "lng": 23.7289558
    },
    "travel_mode": "DRIVING"
  },

```

Από το παραπάνω JSON προκύπτει πως για κάθε τοποθεσία που διασχίζει ο οδηγός εμφανίζονται σαν `steps` οι οδηγίες και η περιγραφή της κάθε διαδρομής και το κάθε `step` περιλαμβάνει πληροφορίες όπως τα `lat`, `lng` της κάθε τοποθεσίας όπου και μπορούν να συγκριθούν με μια αναζήτηση στην βάση και να ταυτοποιούνται οι προορισμοί με ίδιες συντεταγμένες. Όπως εξηγείται και παραπάνω το `key` της κάθε διαδρομής είναι της μορφής `latFrom_lngFrom-latTo_lngTo` οπότε η σύγκριση γίνεται με βάση τις συντεταγμένες.

Με τη χρήση μιας `For loop` θα είναι εφικτή η σύγκριση της κάθε τοποθεσίας που διασχίζεται με τη βάση και θα μπορεί (παρόλο που ο αρχικός και τελικός προορισμός είναι άλλοι), να εμφανίζεται σαν επιλογή σε περίπτωση που η αναζήτηση που πραγματοποιείται είναι κάποια ενδιάμεση τοποθεσία.

Τέλος, μια μελλοντική επέκταση της εφαρμογής θα μπορούσε να είναι η εμφάνιση διαδρομών όχι μόνο με ίδιες συντεταγμένες αλλά και με συντεταγμένες που βρίσκονται σε μια  $\chi$  ακτίνα μακριά από την διαδρομή που ακολουθείται από τον οδηγό.

## 15. ΒΙΒΛΙΟΓΡΑΦΙΑ - ΔΙΚΤΥΟΓΡΑΦΙΑ

- [1] Credits to Linus Långberg for his great repository. <https://github.com/rmuif/web> [Online accessed 10 Apr. 2020].
- [2] RMUF. Documentation. <https://docs.rmuif.com/getting-started> [Online accessed 10 Apr. 2020].
- [3] Javascript.com/. JavaScript. <https://www.javascript.com/> [Online accessed 15 Apr. 2020].
- [4] developer.mozilla.org/. JavaScript. <https://developer.mozilla.org/en-US/docs/Web/JavaScript> [Online accessed 15 Apr. 2020].
- [5] Reactjs.org/. Documentation. <https://reactjs.org/docs/getting-started.html> [Online accessed 20 Apr. 2020].
- [6] Node Package Manager (npm). Documentation. <https://docs.npmjs.com/> [Online accessed 22 Apr. 2020].
- [7] Material-UI. Documentation. <https://material-ui.com/getting-started/installation/> [Online accessed 25 Apr. 2020].
- [8] Cloud Firestore (Firebase). Documentation <https://firebase.google.com/docs/firestore> [Online accessed 30 Apr. 2020].
- [9] Papaki.com/. Domain. <https://www.papaki.com/el> [Online accessed 20 May 2020]
- [10] WhatsApp.com/. Click to chat Documentation. <https://faq.whatsapp.com/general/chats/how-to-use-click-to-chat> [Online accessed 15 Jun. 2020].
- [11] Google Cloud Platform (GCP). Documentation. <https://cloud.google.com/docs> [Online accessed 16 Sep. 2020].
- [12] Directions API. Documentation. [https://developers.google.com/maps/documentation/directions/overview?hl=en\\_US](https://developers.google.com/maps/documentation/directions/overview?hl=en_US) [Online accessed 5 Dec. 2020].
- [13] Maps JavaScript API. Documentation. <https://developers.google.com/maps/documentation/javascript/overview> [Online accessed 5 Dec. 2020].
- [14] Places API. Documentation. <https://developers.google.com/places/web-service/overview> [Online accessed 5 Dec. 2020].

[15] Geocoding API. Documentation.

<https://developers.google.com/maps/documentation/geocoding/overview> [Online accessed 5 Dec. 2020].

[16] Carbon. Create an image of your source code. <https://carbon.now.sh/> [Online accessed 20 Dec. 2020].

[17] Postman.com/. Collaborator Platform for API Development. <https://www.postman.com/> [Online accessed 8 Jan. 2021].

