CSE 546 — Project 2 Report

Manasa Pola , Spoorthi Karnati , Riya Saxena

1.Problem statement

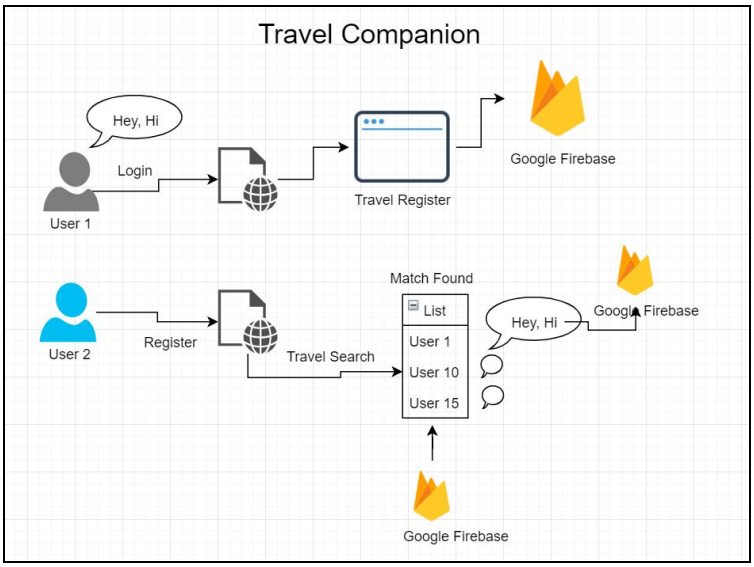
People travel across the world on different purposes but sometimes they might need someone to travel with. For example, when a senior citizen is travelling to a place he will have so many questions in mind like Can I catch a connecting flight on time? Is there anyone who can accompany me? Can someone help me with the wheel-chair? Will there be any language barrier? What if there is any medical emergency? So, before starting guessing the answers they can visit our application which helps them to find potential travel partners or seek advice. This can be useful in other scenarios also like children who need an escort, student travelling abroad for the first time. As we have experienced a similar situation, so we came up with this idea of Travel Companion to make everyone's travel easy, interesting and better.

Travel Companion allow you to register your travel itinerary and matches travelers with similar itineraries based on source , destination and dates of travel. This platform helps to communicate by sending a message with the relevant details.

2.Design and implementation

We developed this application using Firebase Realtime Database, Firebase Authentication, Angular Framework.

2.1 Architecture

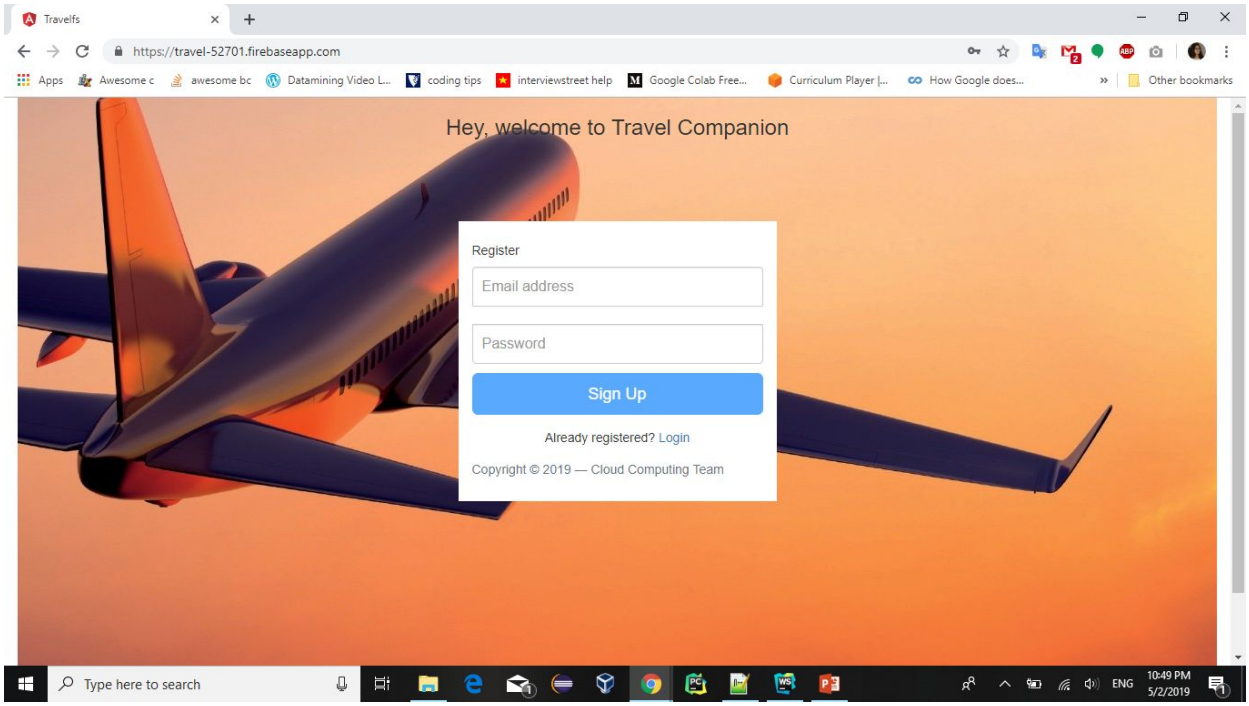


2.2: Component Description

The major components in this application are

2.2.1) Register Screen Component

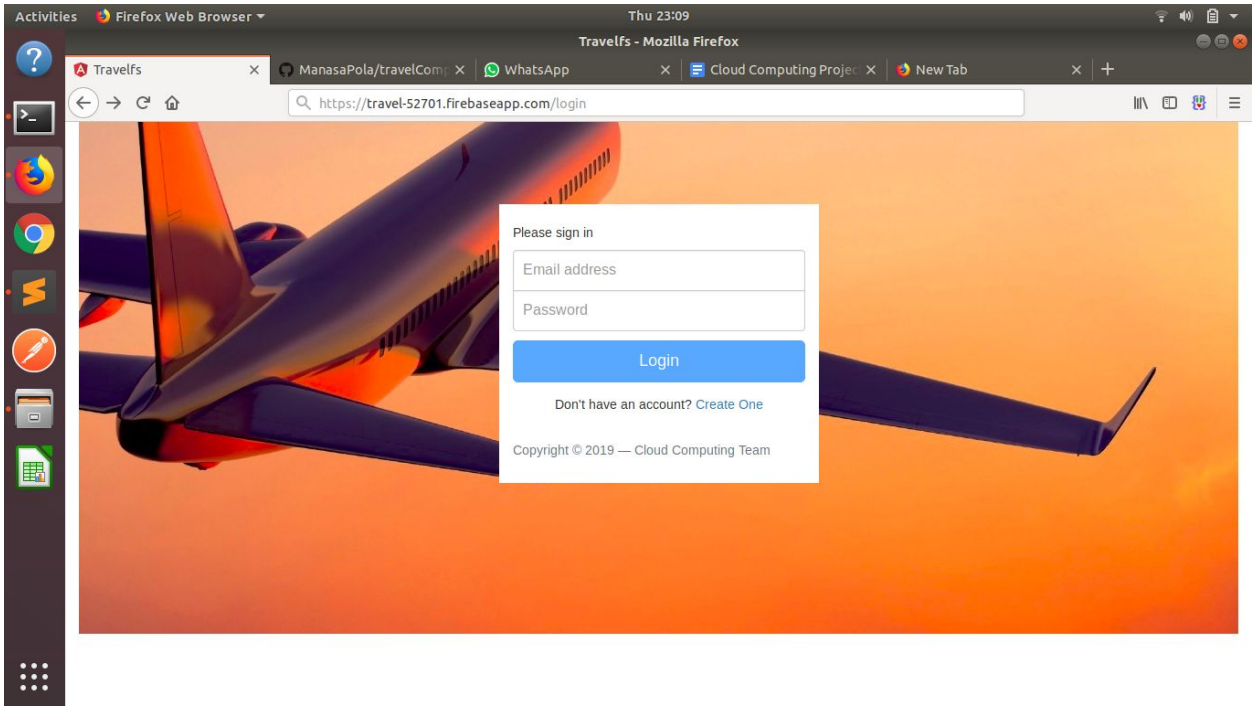
Register components allows new user to register into Travel Companion. The user registers on the website by entering email and password. The email id and passwords are stored in firebase database and each time, a new user registers, a separate key i.e. User Id(UID) is created. This UID is later on used for authentication of users. Below is the example of webpage showing signup functionality.



<firebase auth database>

2.2.2) Login Screen Component

If the user is already registered then he can skip the registration and click the link to login. The login will take user's email id and password that he has given while registering. The email id and password will be compared with what is stored in the firebase and if it doesn't match then user won't be able to login. On successful login user will be redirected to the home page.



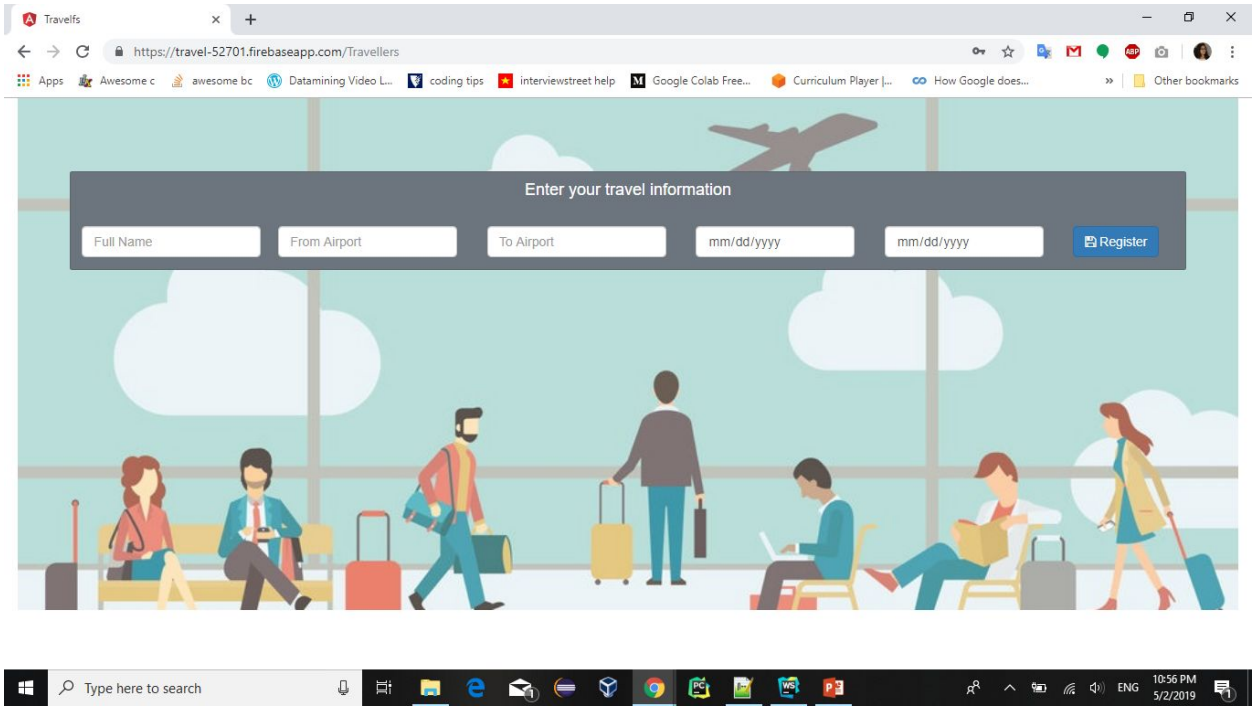
2.2.3) Home Component

The home page has 3 buttons and a chat icon. On clicking button named “Register Travel Dates”, it will redirect the user to register his travel information. Similarly, on clicking “Find Travel Companion”, the user will be redirected to the page where user can search for the companions and chat with them. Chat icon on the bottom right corner of the home page is the feature to get all the messages sent by other users. On clicking this icon user will see all the messages sent to him by other users.

<screenshot of home page>
<screenshot of home page when chat icon is clicked>

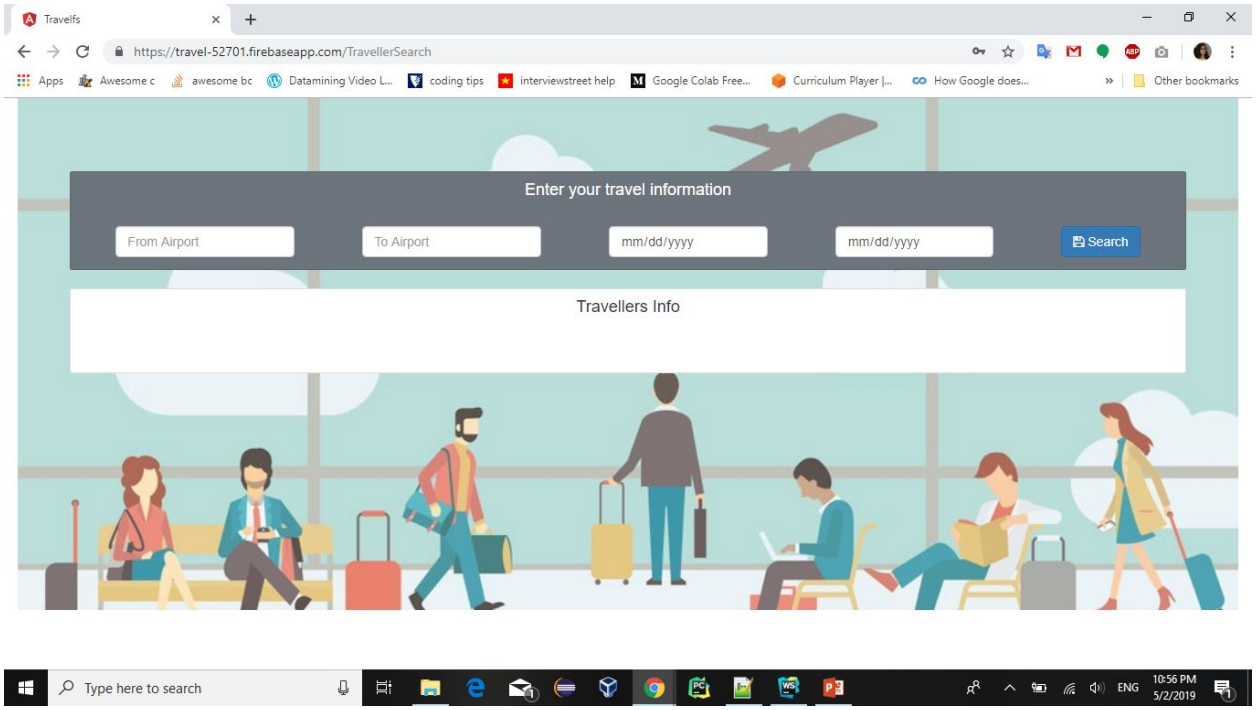
2.2.4) Travel Register Component

The user can enter here his name that he wants to display while chatting with the fellow companion, source and destination airport and the dates between which he is planning to travel. On clicking submit button, the user details will be stored in the firebase.



2.2.5) Travel Search Component

This component lets the user to search for his travel companions and send messages to them. The user can provide the dates between which he is planning to travel as well as source and destination airports. On clicking Search button, the data will be fetched from the firebase showing the details of the other users travelling on similar dates. Also, there will be a chat button to send a message to the user. Through this feature user can connect with the fellow traveller and discuss the travel plans.



Provide an architecture diagram including all the major components, and explain the design and implementation of each component in detail.

2.3 Autoscaling

We have used Firebase database which is Firestore for user authentication. Firestore automatically scales up the number of users based on account creation. We haven't implemented account deletion as we didn't want out users to leave. If it was implemented, scale down of users would have been taken care by Firestore again.
We also used Firebase - Realtime Database for storing user travel records. We created two database tables. Firebase handles scaling in and out for the records in these databases and also allows simultaneous access to multiple databases.
We hosted our website in Firebase which allows greater

3.Testing and evaluation

3.1 Testing:

We have developed an Angular application with multiple components. Each component implements plays a key role in the application functionality. We have implemented several components based on the modules. Below are the set of modules we came up in our project.

a. Components
b. Models
c. Screens
d. Services

We have done Unit testing, Integration testing while developing and merging the project. Performed thorough end user testing to make our application error free. Below are the steps involved in the testing process:

1. Tested the functionality of each standalone component.
2. Merged few components into a big single component and performed regression testing of the functionality.
3. Combined merged components together and performed rigorous unit and integration testing.
4. These components are inter dependent on services module. Tested these modules effectively and eliminated redundant code.
5. Tested the whole application end to end with all promised features.

3.2 Evaluation:

Following are the evaluation results for the operations performed on the website:

Working functionalities are:

1. Sign up if the user is new to the application
2. Login if the user has already signed up
3. Register the travel information of the user to database
4. Date picker for date input fields in Register and Search components
5. Auto complete property for Cities field in Register and Search components.
6. Finding the companions by searching with user tentative travel information(dates, cities)
7. User send a secure message to the user with matching travel requirements \
8. Both way communication between users with matched travel dates using chat box.
9. Logging out from the application

Flow of the application:

1. Login/Signup
2. Register your travel dates or search for travellers with same travel itinerary.
3. If registered , the information is stored in database created in Realtime Database of Firestore. This information is used to display for other users whose preferences matches.
4. Search the companions with your wish of dates of travel and cities.
5. Send a message to the traveller whose travel itinerary matches exactly or approximately near to your requirements.
6. The message sent is only visible to destined user. (Message is only visible in the chat box after login)
7. Sending multiple messages to multiple users

4.Code

Zipped file contains travelCompanion folder. Inside the path src/app we have basically 6 modules containing ->

1) Components - All the 5 components discussed in section 2
2) Material - Angular material designs
3) Models - User and message models
   User Model Structure:
   {$key: string; name: string;fromAirport: string; toAirport: string; fromDate: string; toDate: string;}

   Message Model structure:
   { $key: string; receiverId: string; senderId: string; messageText: string;}

4) Screens- Login Screen and Signup Screen

5). Services- Services corresponding to each of the 5 components

6). Assets- Containing the images for the UI and json file for autpopulating airports data

7). Environments- Firebase Configuration file environment.ts

Steps to Install and Run the project

1. To create Angular Project - *'ng new <ProjectName>'*
2. Create Firebase Project in Firebase Console
3. To Install Firebase and AngularFirebase CLI - *'npm install firebase angularfire --save'*
4. To create component - *'ng g c <ComponentName>'*
5. To create service - *"ng g s <ServiceName>"*
6. To build the project - *"ng build"*
7. To run the project - *"ng serve"*

Steps to deploy the project in Firebase:

1. *"cd dist"*
2. *"firebase init"*
3. *copy all files except firebase related into public folder*
4. *"Firebase deploy"*
5. *"Firebase open hosting:site"*

5.Individual contributions (optional)

Riya Saxena (ASU ID: 1215077747)

My contribution to this project includes:

1). Searching the travel companions

2). Sending and receiving messages

Design:

The whole application is built using Angular-CLI and Firebase. The design phase involved how the whole architecture will look like and what google cloud resources to be used to build this application.

We decided to use firebase as a data store and firebase cloud messaging(fcm) for push notifications. Also, we decided to use Angular because it has many features which can support the functionality of our application. I came up with the design of the flow of the application and to use Angular and firebase to build the application.

Implementation:

I mainly created the Travel Search and Chat components in the application.

For implementing it, I first created the travellerService which will have functions to get all the users filtered by dates and airport name from the travellers collection in the firebase. The structure of the travellers collection in firebase is kept as:

{$key: string; name: string;fromAirport: string; toAirport: string; fromDate: string; toDate: string;}

Similarly, I developed traveller component which will communicate with html component to get the form data. The form data will contain the fields on the basis of which user can filter and search for travel companions. On clicking the search button, the user will get a list of travellers travelling on similar dates. For this I used firebase queries to extract the registered travellers data.

Now against each list item there is a chat button to chat with the travellers. On clicking it, a bootstrap modal will popup which takes the message text as a form data and send it to firebase messages collection along with the email-id of the sender. Now when the receiver will login to his account, he can see the sender's message on clicking the chat icon present in the bottom right corner of the home page. This, I achieved by executing the query on firebase to get the list

of messages filtered by email-id of the receiver. The structure of the document in message collection is kept as :

{ $key: string; receiverId: string; senderId: string; messageText: string;} where senderId and receiverId are sender and receiver's email-id respectively.

Testing:

I have done unit testing for Traveller Search and Chat components. I also fixed the cross browser issues for the webpages of the above said components. I helped my teammates in integration testing of the whole project. I also did end to end testing by checking the messages are being sent to and received by the correct person and validation testing by checking that when entering any data in the form, the user is getting relevant search results.

Spoorthi Karnati (ASU ID: 1214227624)

1. Design

Designing is an important and crucial step in the development of the project. I have contributed to the design of the website and its main modules. I have made some ground work along with my teammates to come up with a set of technologies required to develop our project. Finally we have decided to go with Angular, Angular-Cli, Firebase and FireStore.

I came up with the design part of all the web pages. I did research several websites and decided to make one which looks clear and easy so that any user can use our website without hurdles or help from others.

2. Implementation

I implemented the web pages of our application as per my design approach. The Sign Up page shows welcome message to users and takes email and password as input fields for creating an account. Login page also has the same inputs of Email Id and Password. I have provided an option of Login in Sign-up screen and Sign-up in Login screen. After logging in, I have provided two buttons for users to choose an option to either register their travel dates or search for a travel companion. I have placed a chat icon which enables users to see their received messages. 'Register and Find Companion' pages UI are very responsive and intuitive. On click of 'Find Companion' button, user is driven to land in 'Find Companion' page. Input fields for cities/airports have auto complete feature which shows suggestions of cities airports for each keystroke in input field. Date picker is implemented to make easy selection of dates instead of entering values. Validation of dates is handled in the front end.

I have also implemented the functionality for Login and Sign up screens. Validation of this sign-in form is handled in front end in such a way that it doesn't allow user to sign up with same email id. Validation of Login is done which doesn't allow user to login if they are not signed up.

Another addition to my implementation is chat module. I have created a chat window which allows user to see received messages. Helped my teammates in implementing functionalities related to reading or writing information from database. Handled hosting of project into Firebase.

3. Testing

I have done unit testing for Sign up and Login modules. Tested website view in all browsers and fixed resolution issues. Did integration testing for Login, Signup modules along with chat module. Helped my teammate during integration testing for whole project. Did end user testing along with regression testing. Performed smoke testing.