DALHOUSIE
UNIVERSITY

Mobile Computing
Project Milestone - 2

B00991820
ns910276@dal.ca

App Name: Take Me There

Gitlab

https://git.cs.dal.ca/courses/2025-winter/csci-4176_5708/project-milestone-2/njose/-/tree/main

App Summary

"Take Me There" is an application used to request cab rides. In India when a user tries to book a cab it takes a lot of time as the drivers usually do not accept the rides because of certain reasons. So they have to request rides from multiple apps and take the one that is confirmed first. The purpose of this app is to automate the process and enhance the user experience.

People who depend a lot on cabs for their daily commute are my app's target audience. Think of someone having to open 4 to 5 fives and spending 5 to 10 minutes app switching trying to book a cab, frustrating right? And there are many people who have to do this more than twice a day. These people are my major focus as my app could save them from the frustration of booking a cab.

Progress Report

I have developed the **Login** and **Signup** screen which is fully functional now. This is achieved using **Firebase authentication.** The user is asked to enter mail id, name, mobile number, password and re-enter password to confirm it matches while signing up. There are also basic validation which is done in the sign up screen like password should be at least 6 characters long, mobile number should be 10 digits, and email and name cannot be empty. If some other error with authentication happens, the error message from firebase is displayed using a **snackbar**.

Once the user is signed up an **api** call is made to the backend to store the user details. The backend apis are written using **Node js** and hosted in **Render.** For **complex data storage** I am using **MongoDB** which is running remotely using **Mongo Atlas**. The user details are stored in the Mongo Atlas.

For the navigation of the application I have used drawer navigation. Using the drawer navigation we can navigate to **Ride Booking** Screen, **Trip History** Screen, and **Profile** Screen. The **Logout** option is also given at the end of the drawer. The drawer navigation is also fully functional now.

The Ride Booking Screen is the **root screen** of the app and this is where the user lands once they login. This screen has 2 input fields to enter pick up and drop location. Once the pickup and drop location is given, the option to select cab type is displayed. The user can then select the cab type and proceed. Even though the UI is ready, the functionality is not ready yet as it requires the Geocoding api integration to do some backend calculations before proceeding to the next screen (Trip confirmation).

The next screen is Profile Screen. The data displayed in this screen is fetched from **MongoDB** and is stored in **AsyncStorage.** This screen allows the user to edit their name, mobile and display pictures. For now the changes made to name and mobile number are stored locally in simple storage. But there will be api changes done which will update the data in DB. The photo functionality which is still not implemented will allow users to upload or click a picture and set it as the display picture. I have planned to convert the images into base 64 and store it in MongoDB. Say the user is editing the name and mobile number and feels like they don't want to, they can click on cancel which will revert the changes.

The last screen in the Trip History Screen which will display the overview of trips taken by the users before. This page just has placeholders now and will be implemented.

The logout option in the drawer is fully functional. When a user clicks on the logout option they get redirected to the login screen and they lose access to the apps other functionalities.

A major problem I encountered during the development process is while doing an npm install there was a version mismatch and while trying to fix it I made some changes which caused a lot of version mismatches. I tried to resolve it for a longer time and couldn't succeed so I have deleted the package.json, package-lock.json, and node_modules then from a previous commit I took the proper package.json and used it to reinstall packages. This helped me fix the issue.

A Common Use Case

A current user wanting to update their mobile number:
One of the app’s users has changed their mobile number and they want the drivers to see their new mobile when the drivers are trying to contact the user.

Once the user has logged in they can open the drawer navigator and click on Profile to navigate to Profile Screen. Then they can click on the Edit button in the profile screen which will let them edit their name and mobile number. They can then update their number and click on the save button to save the changes.