



Faculty of Engineering and Information Technology
Computer Engineering Department
An-Najah National University

Software Graduation Project

Wassalni Ma'ak

Supervised By:

Dr. Anas Toma

Written By:

Aya Walid Mohammad Awad
Malak Mohammad Jawabreh

Submitted in partial fulfillment of the requirements for a Bachelor's degree
in Computer Engineering

January 28, 2025

Acknowledgment

To our parents, your sacrifices, love, and unwavering support have been our greatest source of strength and motivation. Your belief in us has fueled our determination and perseverance.

To our friends and family, your encouragement and belief in us were invaluable. Thank you for being the driving force behind our success.” ”*We would like to express our heartfelt gratitude to Dr. Anas Toma for his continuous guidance and unwavering support throughout our project. Special thanks to the Computer Engineering Department at An Najah National University for providing a conducive learning environment and to all the academics who generously shared their knowledge.*

To our parents, your sacrifices, love, and unwavering support have been our greatest source of strength and motivation. Your belief in us has fueled our determination and perseverance.

To our friends and family, your encouragement and belief in us were invaluable. Thank you for being the driving force behind our success.”

Disclaimer statement

This report was written by Aya Walid Awwad and Malak Mohammed Jawabreh at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. The information presented in this report is intended for informational purposes only. The authors and the organization are not liable for any actions taken or decisions made based on the information provided. Readers are advised to consult with relevant experts or professionals before making any decisions or taking any actions. The authors do not endorse or guarantee the accuracy or completeness of the information in this report.

Abstract

"Wasalni Ma'ak" is an innovative application designed to facilitate transportation within Palestine in an economical and efficient manner based on the principle of (Share A Car). The application aims to connect drivers who plan to take certain trips with passengers who are heading to the same destination. The driver can create an account on the application, and record his trips by specifying all the necessary information, such as the starting point, destination, trip time, and number of available seats, in addition to specifying a competitive price for the trip that is lower than the cost of traditional taxi services.

On the other hand, passengers can browse the available trips and easily book a seat if the trip suits their needs. The application allows sharing the car between several passengers heading to the same destination, which reduces transportation costs for each passenger and provides an additional source of income for the driver.

The application seeks to promote a culture of cooperation between individuals by providing an innovative transportation solution that contributes to alleviating economic burdens, reducing fuel consumption, and limiting traffic congestion. "Wasalni Ma'ak" is not just a means of transportation, but rather a means of achieving effective communication between members of the Palestinian community within a safe and reliable environment.

Contents

<i>Acknowledgment</i>	1
<i>Disclaimer statement</i>	2
<i>Abstract</i>	3
1 Introduction	9
1.1 Problem Statement	9
1.2 Scope	9
1.3 Outcomes/Objectives	11
2 Constraints and Earlier Coursework	13
2.1 Constraints and Limitations	13
2.2 Earlier coursework	14
2.3 Standards	15
2.3.1 MVC Pattern	15
2.3.2 Agile Model	16
2.4 Earlier work	16
3 Literature Review	17
4 Methodology	19
4.1 Introduction	19
4.2 Non-Functional Requirements	20
4.3 Tools, Methods and Programming Languages	21
4.3.1 Client Side	21
4.3.2 Server Side	24
4.3.3 Integrated Development Environments (IDEs):	25
4.3.4 Version Control:	25
4.3.5 Testing of the Server and API:	25
4.3.6 DataBase Desgin	26

4.4	<i>System Features and Implementation</i>	26
4.4.1	<i>Onboarding Screen</i>	26
4.4.2	<i>Login and Registration Feature</i>	30
4.4.3	<i>Passenger User Interface (UI)</i>	34
4.4.4	<i>Browse the menu at the top of the page.</i>	47
4.4.5	<i>Driver User Interface (UI)</i>	64
4.4.6	<i>Admin permissions</i>	71
4.5	<i>Website</i>	80
5	<i>Result And Discussion</i>	84
6	<i>Conclusion and Recommendation</i>	85
6.1	<i>SUMMARY</i>	85
6.2	<i>RECOMMENDATION</i>	85
6.3	<i>WHAT WE HAVE LEARNED</i>	85
6.4	<i>FUTURE WORK</i>	86
7	<i>References</i>	87

List of Figures

2.1	<i>MVC pattern Figure explanation</i>	15
2.2	<i>Agile methodology Figure</i>	16
4.1	<i>Client-Server Architecture</i>	24
4.2	<i>MongoDB with Node.js server</i>	26
4.3	<i>Onboarding screen for All Users</i>	27
4.4	<i>Onboarding2 screen for All Users</i>	28
4.5	<i>Onboarding3 screen for All Users</i>	29
4.6	<i>Login page for all types</i>	30
4.7	<i>password is wrong!</i>	31
4.8	<i>Email is wrong!</i>	31
4.9	<i>Sign Up!</i>	32
4.10	<i>Sign Up If without information</i>	32
4.11	<i>Step1</i>	33
4.12	<i>Step2</i>	33
4.13	<i>Step3</i>	33
4.14	<i>Passenger Home Page1</i>	35
4.15	<i>Passenger Home Page</i>	36
4.16	<i>Browse all available trips</i>	37
4.17	<i>Browse all available Nearby trips</i>	38
4.18	<i>Booking in Trip!</i>	39
4.19	<i>search page for trips</i>	40
4.20	<i>Filters you can choose to search for a trip</i>	41
4.21	<i>Search for specific trips</i>	41
4.22	<i>Active Booking</i>	42
4.23	<i>booking details</i>	42
4.24	<i>Driver profile</i>	43
4.25	<i>Driver profile-2</i>	43

4.26	<i>cancellation</i>	44
4.27	<i>Edit booking</i>	44
4.28	<i>Payment by Visa</i>	45
4.29	<i>All Drivers In App</i>	46
4.30	<i>Menu</i>	48
4.31	<i>My Profile</i>	49
4.32	<i>Account</i>	49
4.33	<i>Change Password</i>	50
4.34	<i>personal details</i>	50
4.35	<i>Status Profile</i>	51
4.36	<i>Settings</i>	52
4.37	<i>Select Language</i>	53
4.38	<i>Application in Arabic</i>	53
4.39	<i>Change Theme</i>	54
4.40	<i>Email Address</i>	55
4.41	<i>Two-Step Verification</i>	55
4.42	<i>Privacy Page</i>	56
4.43	<i>Block Contact</i>	57
4.44	<i>Complaints Step1</i>	58
4.45	<i>Complaints Step2</i>	58
4.46	<i>Complaints Step3</i>	58
4.47	<i>Support</i>	59
4.48	<i>Open WhatsApp to contact support</i>	60
4.49	<i>Conversation from sender side - app user</i>	60
4.50	<i>Conversation from the receiver side-the support side</i>	60
4.51	<i>Open gmail to contact support</i>	61
4.52	<i>Mail sent by app user</i>	61
4.53	<i>Mail received from the application support</i>	61
4.54	<i>FAQs</i>	62
4.55	<i>Show answers to FAQs</i>	62
4.56	<i>Search for FAQs</i>	62
4.57	<i>Support Form-1</i>	63
4.58	<i>Support Form-2</i>	63
4.59	<i>Driver Dashboard</i>	65
4.60	<i>Passenger List</i>	65
4.61	<i>Passenger List</i>	66
4.62	<i>Passenger Profile</i>	66
4.63	<i>Completed Trip</i>	67
4.64	<i>Create Trip</i>	68

4.65	<i>Visibility Trip</i>	68
4.66	<i>Main Menu</i>	69
4.67	<i>Feedbacks</i>	69
4.68	<i>Chat List</i>	70
4.69	<i>Chatting Page</i>	70
4.70	<i>Admin Dashboard View</i>	71
4.71	<i>Admin Dashboard View</i>	72
4.72	<i>User View</i>	72
4.73	<i>Edit User</i>	73
4.74	<i>Add User</i>	73
4.75	<i>Trip Management</i>	74
4.76	<i>Booking Management</i>	75
4.77	<i>add Booking</i>	75
4.78	<i>Select Trip</i>	76
4.79	<i>Add</i>	76
4.80	<i>General Setting</i>	78
4.81	<i>Add Admin</i>	78
4.82	<i>Show Admin And Delete</i>	79
4.83	<i>first webpage</i>	80
4.84	<i>login webpage</i>	80
4.85	<i>Driver Dashboard webpage 1</i>	81
4.86	<i>Driver Dashboard webpage 2</i>	81
4.87	<i>Driver Profile</i>	82
4.88	<i>Income OverView</i>	82
4.89	<i>Passenger Feedback Trip</i>	83
4.90	<i>Passenger Active Booking</i>	83

Chapter 1

Introduction

1.1 Problem Statement

In recent times, moving between cities and regions within Palestine has been a major challenge for many people, especially in light of the high costs of transportation via taxis due to several factors, the most important of which are the obstacles and barriers placed by the occupier, the lack of reliable options, and the security conditions that impose restrictions on freedom of movement. These challenges lead to an economic burden on passengers and limit the ease of access to the desired places.

On the other hand, drivers face difficulty in compensating for fuel costs and operating their cars due to the lack of passengers or reliance on single trips. In the absence of an effective means of organizing shared transportation between individuals who share the same destination, transportation remains expensive and inefficient, which negatively affects everyone.

Accordingly, the need arose for an innovative solution that provides an effective and safe way to organize shared transportation trips, so that drivers and passengers benefit from sharing resources in a fair and appropriate manner for all.

1.2 Scope

The project will encompass the following:

- *Developing a mobile and web platform called "Waslni Ma'ak" for connecting drivers and passengers for shared trips within Palestine.*
- *Allowing drivers to create trips by specifying the destination, time, and available seats.*

- *Enabling passengers to search for available trips, select the most suitable options, and book seats.*
- *Integrating a secure communication system for drivers and passengers to coordinate trip details.*
- *Implementing a rating and review system to ensure the quality of service and build trust between users.*
- *Providing real-time notifications about trip status, including confirmations, cancellations, and updates.*
- *Using modern technologies such as Flutter for mobile app development, and Node.js with MongoDB for backend and database management.*
- *Ensuring a user-friendly interface for both drivers and passengers, making the platform accessible to a wide range of users.*
- *Incorporating a map feature for both drivers and passengers to view the trip route and estimated time.*
- *Allowing drivers to upload their personal license, car license, and insurance image during account creation, with verification for validity to ensure safety.*
- *Enabling Visa payments for trip bookings.*
- *Automatically displaying nearby trips for passengers based on their location.*
- *Providing a filtering system to customize trip searches according to specific criteria.*
- *Allowing drivers to choose whether their trip is visible to everyone or only to selected individuals.*
- *Supporting profile pictures for both drivers and passengers.*
- *Implementing a complaint system for both drivers and passengers to report issues with specific users.*
- *Supporting both English and Arabic languages.*
- *Driver Notes feature A space for users to leave a specific note for the driver when they book a ride.*
- *Notification system Alerts the passenger and the driver as well when something new happens.*

1.3 Outcomes/Objectives

The expected outcomes and objectives the following:

- **Develop a user-friendly interface:** Build a mobile and web platform that allows ordinary people to use their personal cars for shared trips with passengers within Palestine, providing an alternative to taxis and cabs.
- **Reduce transportation costs:** Reduce transportation costs for passengers by offering lower fares compared to taxis, while enabling drivers to earn additional income using their personal cars.
- **Empowering drivers:** Enable informal drivers to use their private cars to earn money, by providing a simple and easy platform to manage trips, set prices, and communicate effectively with passengers.
- **Provide a safe transportation option:** Ensure safety by verifying driver information, including driver's licenses, vehicle registration, and insurance, which enhances confidence and security for passengers.
- **Improve user convenience:** Provide a smooth and intuitive user experience for both passengers and drivers, with live trip information, secure payment, and an efficient booking system.
- **Encourage a culture of collaborative travel:** Promote the use of ride-sharing between individuals to reduce the number of vehicles on the road, reduce traffic congestion, and minimize environmental impact.
- **Ensuring reliability and transparency:** Including features such as a rating system, complaint management, and giving drivers the option to decide who can see their trips, to ensure a transparent and responsible experience.
- **Increasing accessibility:** Supporting both Arabic and English languages to meet the needs of a wide audience, making the platform accessible to different communities.
- **Increasing economic opportunities:** Providing individuals with the opportunity to earn additional income through their cars in a flexible and cost-effective manner, contributing to the local economy by enabling more people to participate in the transportation sector.

- *Saving time and effort: Especially during peak times and for university and school students and employees who suffer from a lack of taxis in the complexes during peak times*

Chapter 2

Constraints and Earlier Coursework

2.1 Constraints and Limitations

When we started this project, we faced many problems and challenges. One of the main issues was Building a strong foundation and really understanding the basic concepts, we also faced difficulty with new programming languages that we were not familiar with as we learned them during the period of working on this application, we also did not have experience as a team and we lacked practical experience in the beginning in developing smartphone applications before, in addition, we faced difficulty in dealing with large computer programs that were difficult to install on devices and this was at the beginning of the stage, We also faced many limitations in technology, the first of which was the Flutter framework, although Flutter have many and varied features in terms of mobile and web, there were some limitations such as compatibility with the native components required effort, which increased the development time until we came out with the best result. Another challenge in MongoDB, despite its known ease and simplicity of use, when the data is huge and complex, it becomes more difficult in MongoDB. Integration challenges in several things such as the payment process via Visa, because it requires a high security guarantee. Also, real time features such as applying the chat feature and notification in a live manner and sharing the location in real time require the use of specialized libraries and APIs. We also faced regulatory challenges, restrictions of local laws related to the use of private cars as a means of public transportation that may affect the full adoption of the application. The time challenges we faced, which is working under the pressure of the project schedule, significantly affected our ability to apply and implement all

the expected features with high accuracy.

2.2 Earlier coursework

The computer engineering courses we took provided us with valuable knowledge that we applied in developing this application. The database, web programming principles, object-oriented programming, and advanced software, which we learned in our courses, were the building blocks of this project. Additionally, we enhanced our skills by enrolling in online courses on Node.js, MongoDB, and Flutter, which were instrumental in the successful implementation and development of our application.

2.3 Standards

2.3.1 MVC Pattern

The MVC - Model View Controller - pattern was employed in our system. We could divide the entire project into different layers to make it easier to follow the flow of work. These layers are as follows:

- *Model Layer:* - Manages data related to car rentals, dealerships, garages, and user profiles. - Contains business logic for rentals, car listings, and services. - Interacts with databases and APIs to fetch and store data.
- *View Layer:* - Represents the user interface. - Displays data to users and updates based on changes from the Model. - Ensures a responsive design for various devices.
- *Controller Layer:* - Handles user inputs and interactions. - Fetches data from the Model and updates the View. - Manages the workflow between the Model and the View.

By using the MVC pattern, Wassalni Ma'ak structure becomes clearer and more manageable, with each layer having a distinct role.

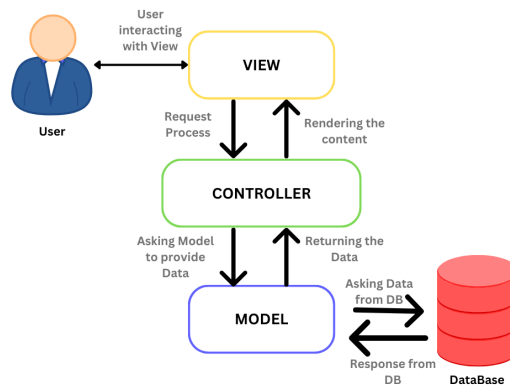


Figure 2.1: MVC pattern Figure explanation

2.3.2 Agile Model

The Agile methodology is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement. Teams follow a cycle of planning, executing, and evaluating. We classified the tasks we had to do into the following five groups:

- *Brainstorming: Gather all needed requirements and features required.*
- *Design: Create the UML diagram to clarify the functionality of new features and their interaction with the system.*
- *Development: Begin writing the code for the front end, followed by the back end for each screen in the app.*
- *Testing: Connect the back end with the front end and ensure that every page works correctly.*
- *Feedback: Continuously get feedback from the supervisor and colleagues to ensure the app meets the requirements.*



Figure 2.2: Agile methodology Figure

2.4 Earlier work

There are similar applications, but they all deal with private offices. However, the idea of our application is that each person can share his car when he goes on a specific trip in a specific area, and through that he can achieve his own profit in addition to going to the trip he wants.

Chapter 3

Literature Review

Wasalni Ma'ak app is designed to facilitate finding trips between cities and connect drivers with passengers in a safe, smooth and efficient way. Research also indicates that car-sharing transportation services reduce transportation costs by 30% making them an economical and convenient option for users in general. Wasalni Ma'ak app relies on advanced mapping technologies, which helps drivers and passengers easily track the trip route, expected arrival time, and nearby services. Studies show that applications that rely on mapping technologies meet user needs and achieve 40

The app also uses a comprehensive verification system for drivers' personal information, including checking the driver's license, car insurance, and vehicle validity. This system helps enhance trust among users and ensure their safety, as all information is verified and the dates are valid to ensure safety and security. Many studies have also shown that applications that care about user safety and security experience increase the rate of repeated use and demand for them in a comfortable manner.

One of the main features of Wasalni Ma'ak is the instant chat feature between drivers and passengers, which enhances the user experience and provides an effective means of instant communication, in addition to the feature of leaving notes to the driver through the reservation and the notifications feature. Research indicates that systems that allow instant communication and enable notifications increase user satisfaction and retention by up to 50

In addition, the application provides an advanced system for filtering trips based on user requirements, such as trips close to the current location, or specifying trips for a specific category of users only. These features help improve efficiency and meet the needs of users accurately. The application also supports both Arabic and English languages to provide a convenient experience for a wide range of users.

In short, the Wasalni Ma'ak application stands out from others by pro-

viding a safe and efficient environment for shared mobility, along with a comprehensive verification system, advanced maps, and real-time chat, which enhances the user experience and promotes the culture of sharing transportation within the community.

Chapter 4

Methodology

4.1 Introduction

This chapter outlines the methodology adopted for the design and implementation of the Wassalni Ma'ak Application , with a specific emphasis on the design process, data collection, analysis, and processing.,tools and technologies utilized and details of usage .

4.2 Non-Functional Requirements

- ***Adaptive User Interface Design*** *adapting to different screens is essential to ensure a seamless experience across different devices. Using responsive design, the app adjusts fonts, buttons, and layouts to fit different screen sizes. Whether accessed from smartphones, tablets, or desktops, Wassalni Ma'ak maintains a consistent and user-friendly interface. Extensive testing ensures that users can navigate effortlessly, regardless of their device, enhancing accessibility for a wide range of users.*
- ***Ease-of-use*** *Wassalni Ma'ak is designed to be simple and easy to use whether users are looking for a ride for their work, their universities or just regular trips, the categories are clear and the features are well organized so that the rider can search for what suits them and also add driver friends. The process of creating an account and using the app is straightforward and smooth, ensuring that users can access the ride search services and see what's new with ease.*
- ***Usability*** *The Wassalni Ma'ak app is easy to use, with both customers and service providers finding it easy to navigate and navigate. The interface is intuitive, with clear templates for creating an account and booking a service. The Wassalni Ma'ak app aims to make the process of finding and accessing a trip easy for all users, ensuring a smooth and efficient experience.*

4.3 Tools, Methods and Programming Languages

In order to develop the Wassalni Ma'ak application, we utilized a variety of tools, methods, and programming languages. For the front-end development, we used Flutter and Dart programming languages, which allowed us to create a modern and intuitive user interface. For the backend development, we utilized Node.js and Express.js to create a RESTful API for handling the app's data and logic. We also designed a non-relational database schema using MongoDB to store and manage the app's data. Throughout the development process, We also used Postman so we could check the back-end we wrote and verify that it was done correctly and with high reliability. we followed the Agile methodology .

4.3.1 Client Side

- **Design**

The design of the client side of the Wassalni Ma'ak app began with choosing the idea, then conducting extensive research to come up with a creative and innovative concept for the app design.

Our decision to pursue this idea was based on the suffering people face while waiting for taxis, especially during peak times, as well as the high prices of taxis and transportation, especially in the current times due to the difficulty of moving between cities through detours, and the desire to create a more effective and easy-to-use app.

Instead of using a pre-designed UI template for the app, we devoted a great deal of effort to designing a unique UI for the app.

The UI design was developed internally based on the research conducted, taking into account the user requirements and the most important ideas that the passenger is looking for and requests to be located near him.

This allowed us to create a custom design that is easy to use, aesthetically pleasing, and consistent with the overall theme and branding of the app. We carefully selected a color theme that would enhance the user experience and enable them to feel comfortable while using the app.

The design of the majority of the app pages and website has been kept simple, easy to use, and basic to ensure ease of use. Additionally, we have incorporated modern design elements to keep the app fresh and in line with current trends in the world.

- **Framework**

We decided to leverage the power of Google’s cross-platform UI framework for client-side development. Flutter is a popular open-source mobile application development framework created by Google that utilizes the Dart programming language.

It offers a wide range of features, including hot-reload, widget-based development, and platform-specific widgets, among others.

Through Flutter, we can create mobile apps for iOS and Android with a single code base that renders everything on its own, eliminating the need for previous native programming for these two platforms and saving a significant amount of time and effort in developing cross-platform apps.

Therefore, Flutter is a viable and efficient tool for developing mobile applications, and we chose to use it for our app for its reliability, speed, and cross-platform capabilities.

Flutter’s popularity has grown steadily over the years, with Google’s support and the community’s contributions. As of May 2023, Flutter has been downloaded more than 290 million times worldwide, and there are over 120,000 Flutter packages available for developers to use, demonstrating its extensive ecosystem.

- **Programming Languages:**

The programming language used in Flutter for client-side development is Dart. Developed by Google, Dart is an object-oriented language with similarities to C++, Java, and JavaScript.

It is a compiled language, type-safe, and has scripting capabilities similar to Python and JavaScript. Our familiarity with these languages from university made Dart easy to learn and work with.

Despite its relative youth, Dart has a wealth of libraries that simplify and speed up the script-writing process, and its popularity is rapidly increasing.

- **Website:** *The website is built using the same client-side technology as the mobile app, leveraging the cross-platform UI framework of Flutter.*

This approach ensures a consistent design and functionality across all platforms, delivering a seamless user experience.

Additionally, Flutter's capability to render high-performance web applications with the same codebase used for mobile apps offers a significant advantage, reducing development time and effort while maintaining code maintainability.

4.3.2 Server Side

- **Architecture of the Server-Side:**

We employed a client-server architecture for our application , allowing both the website and mobile application to use the same API. Our servers are designed to support a RESTful API, facilitating efficient communication between clients and the server.

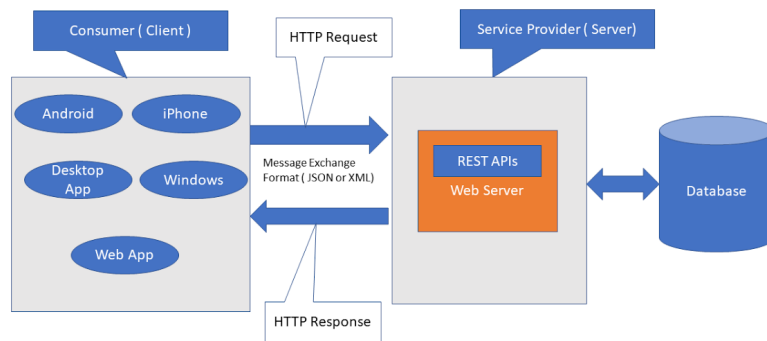


Figure 4.1: Client-Server Architecture

- **Server-Side Framework:**

We chose to work with Node.js and Express.js for the server-side environment.

Node.js is known for its ease of learning, extensive documentation, scalability, and performance, making it an ideal choice for our needs.

Express.js, a widely-regarded back-end framework for Node.js, complements Node.js perfectly, providing a reliable and efficient server-side structure for our application.

4.3.3 Integrated Development Environments (IDEs):

For writing JavaScript code for the REST API, we primarily used Visual Studio Code.

This free, open-source, cross-platform code editor is renowned for its powerful extensions, which allow developers to customize their development environment to meet specific needs.

We found Visual Studio Code to be robust and user-friendly, with advanced features like IntelliSense, debugging, and code navigation that helped us write clean, efficient, and error-free code.

Given that our project required developing for both Android and iOS platforms, we needed an IDE that could handle both.

We chose Android Studio, which allowed us to develop, debug, and test our code for both platforms in one place.

4.3.4 Version Control:

To manage the complexity and size of our Wassalni Ma'ak project, we utilized Two separate GitHub repositories. One repository stored the app code, another held the server-side RESTful API code.

Version control was vital for our project, enabling us to track code changes over time, identify and fix issues, and collaborate effectively as a team.

It also provided a secure backup of all the code, allowing us to revert to previous versions if necessary.

This streamlined our development process, reduced errors and conflicts, and ensured the project's stability and scalability.

4.3.5 Testing of the Server and API:

Testing the server and API was crucial to ensure the smooth operation of our Wassalni Ma'ak project , We used Postman for our testing.

Postman, a popular tool for API testing, allowed us to automate the testing process, saving time and effort. It enabled us to simulate various requests and responses, test edge cases, and validate the API's functionality and performance and reliability.

4.3.6 DataBase Desgin

Database design is a crucial aspect of any system development project, involving the definition of the structure, organization, and relationships of data within a database. For the Wassalni Ma'ak application, the database design aims to capture and manage the relevant information required to effectively handle car-related services such as garages, dealerships, rental offices, and parts shops.

Our choice for the database management system is MongoDB. MongoDB is an open-source, NoSQL database system that allows for flexible data storage and manipulation. Its schema-less nature makes it suitable for handling diverse data types and evolving application requirements. The objective of this design is to accommodate a wide range of database applications and provide robust functionalities to handle the storage, access, and management of data.

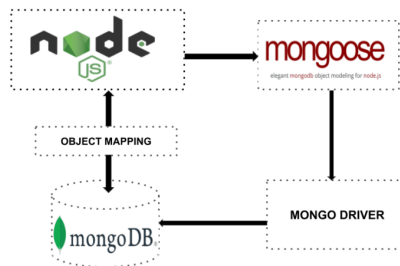


Figure 4.2: MongoDB with Node.js server

Overall, the database design for the Wassalni Ma'ak application is a vital component of the system's architecture. It provides a structured way to organize, store, and retrieve data, enabling efficient and effective management of Trips-related services.

4.4 System Features and Implementation

4.4.1 Onboarding Screen

These pages are the first pages that appear to the user when he opens the application to welcome him and give him a nice and positive impression. They also simplify the application for the user and how to deal with it and the benefit from it, and also inform him of the features that the application offers.

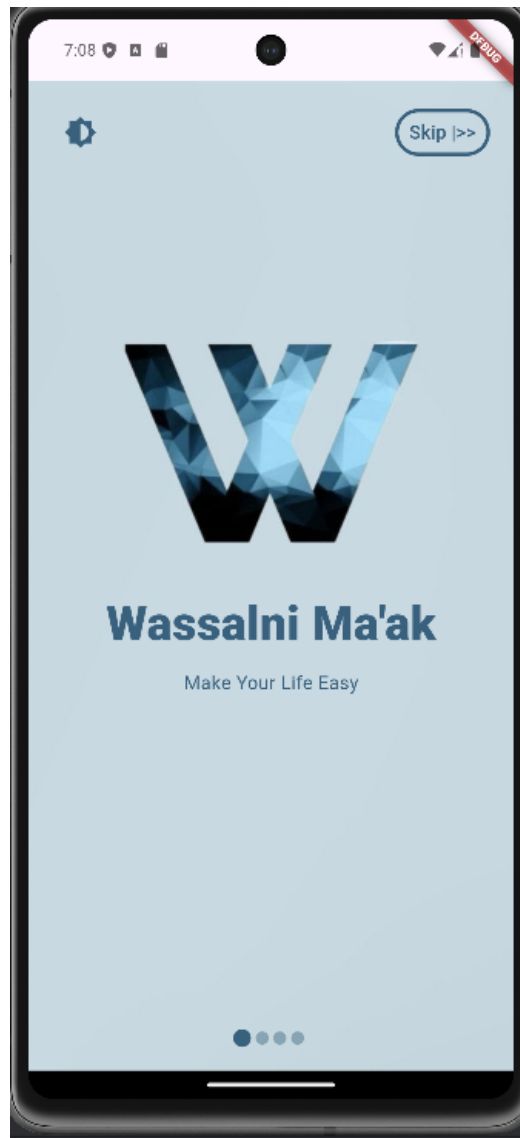


Figure 4.3: Onboarding screen for All Users

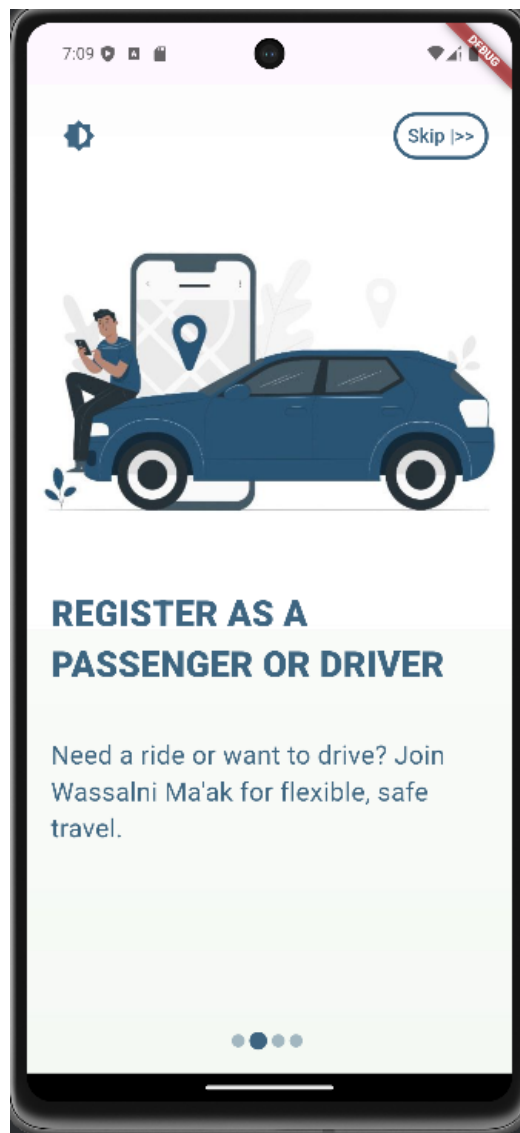


Figure 4.4: Onboarding2 screen for All Users

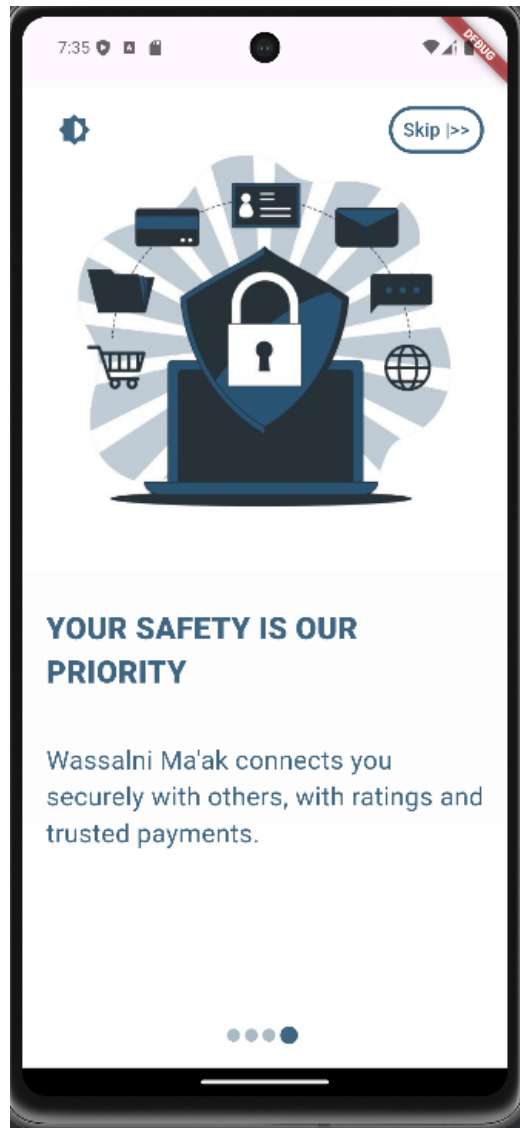


Figure 4.5: Onboarding3 screen for All Users

4.4.2 Login and Registration Feature

This is the login page that appears to all users after the onboarding page, and through it they can login to the application if they are registered and already have an account by entering the correct email and password. If they do not have an account, they register and enter the necessary information depending on whether they are a passenger or a driver.

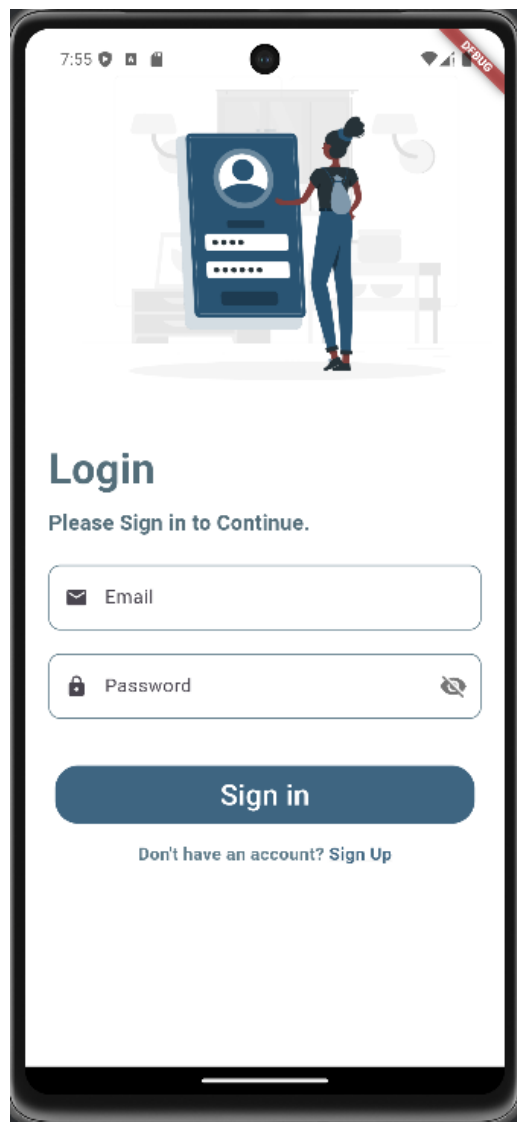


Figure 4.6: Login page for all types

The following images show attempts to login with an **incorrect password or email** and show that the application gives a notification that the operation is incorrect.

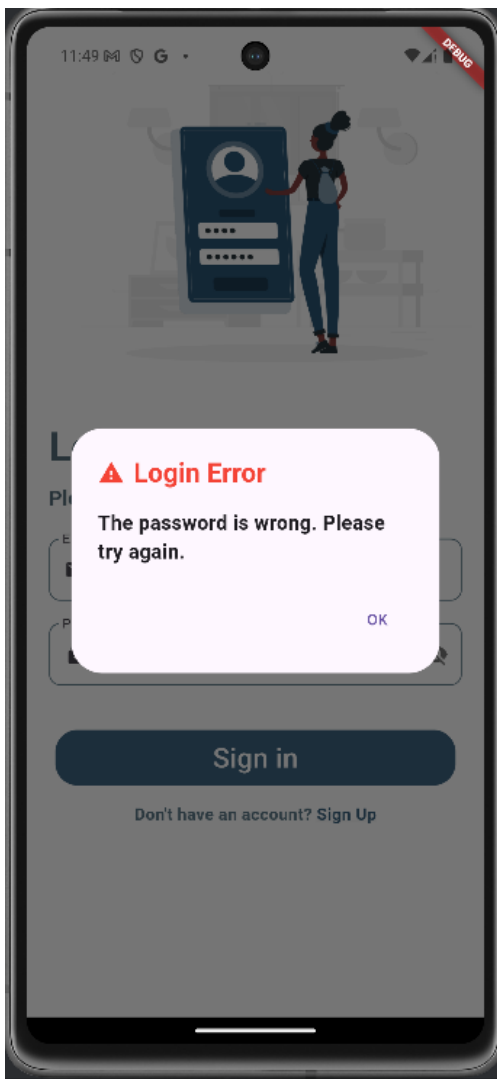


Figure 4.7: password is wrong!

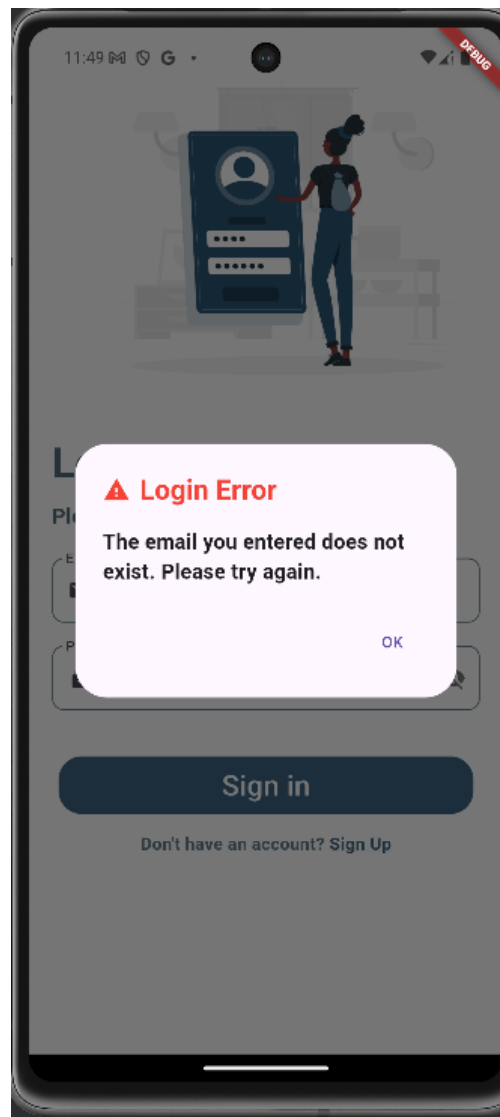


Figure 4.8: Email is wrong!

Now here an account is created in the application through the sign-up. Here appears the screen through which the user creates an account, but if the user is a driver, there are other pages that he must fill out so that he can create an account to verify and ensure high credibility by entering a picture of his personal license, car license and insurance. In the other picture here I put it to verify that all fields must be filled in with the correct information.

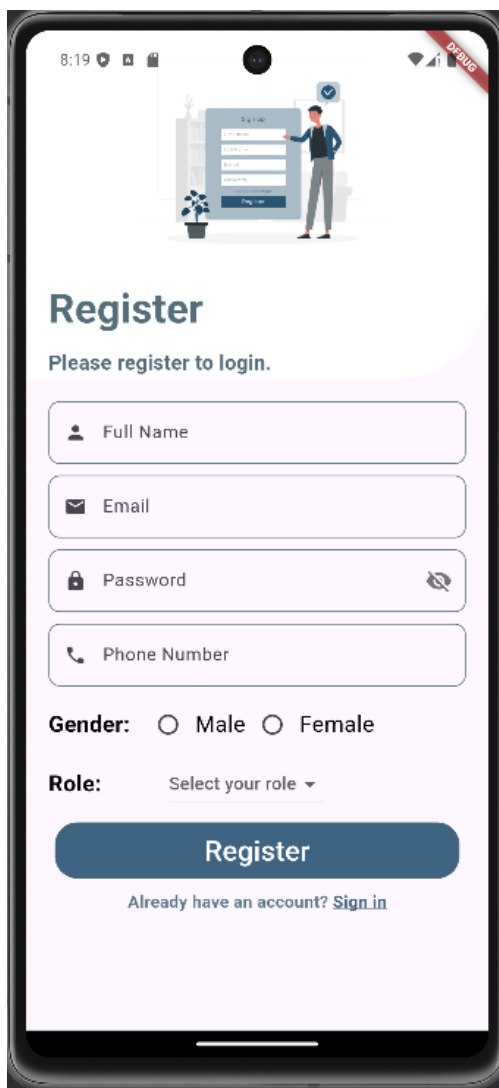


Figure 4.9: Sign Up!

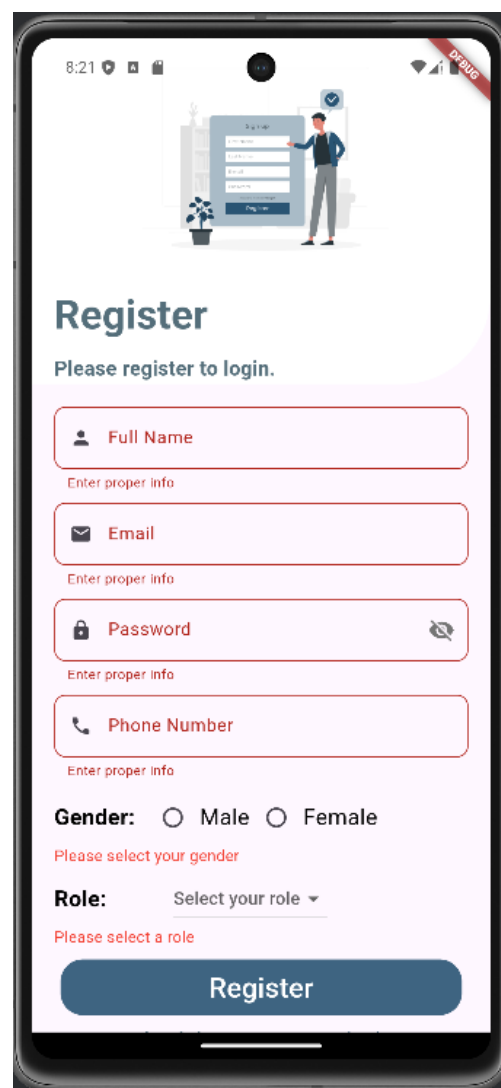


Figure 4.10: Sign Up If without information

These are the **three steps** that a driver goes through when creating his own account on the Wassalni app, to ensure his credibility, verify his identity, and maintain the safety of the passengers who will go on trips with him.

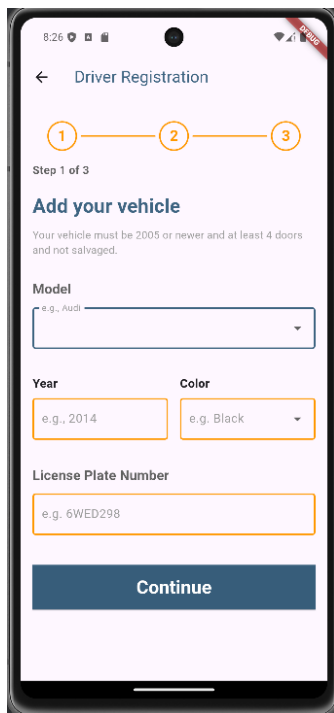


Figure 4.11: Step1

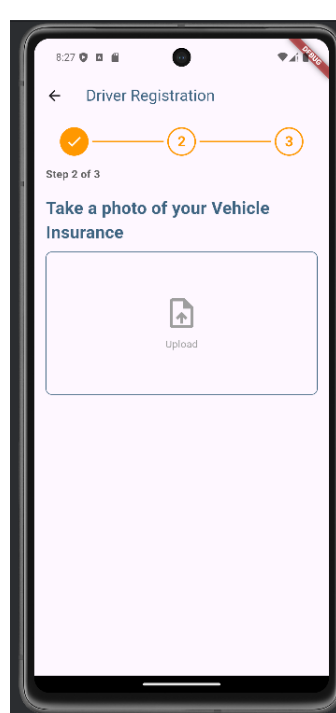


Figure 4.12: Step2

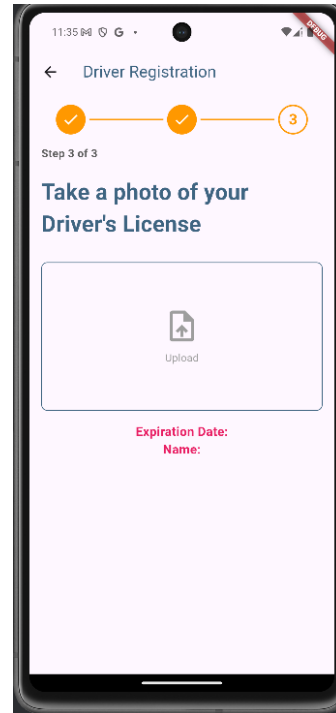


Figure 4.13: Step3

4.4.3 Passenger User Interface (UI)

Now let's go through the passenger interface that the user will interact with in the application. These interfaces represent the passenger side of the Wassalni app.

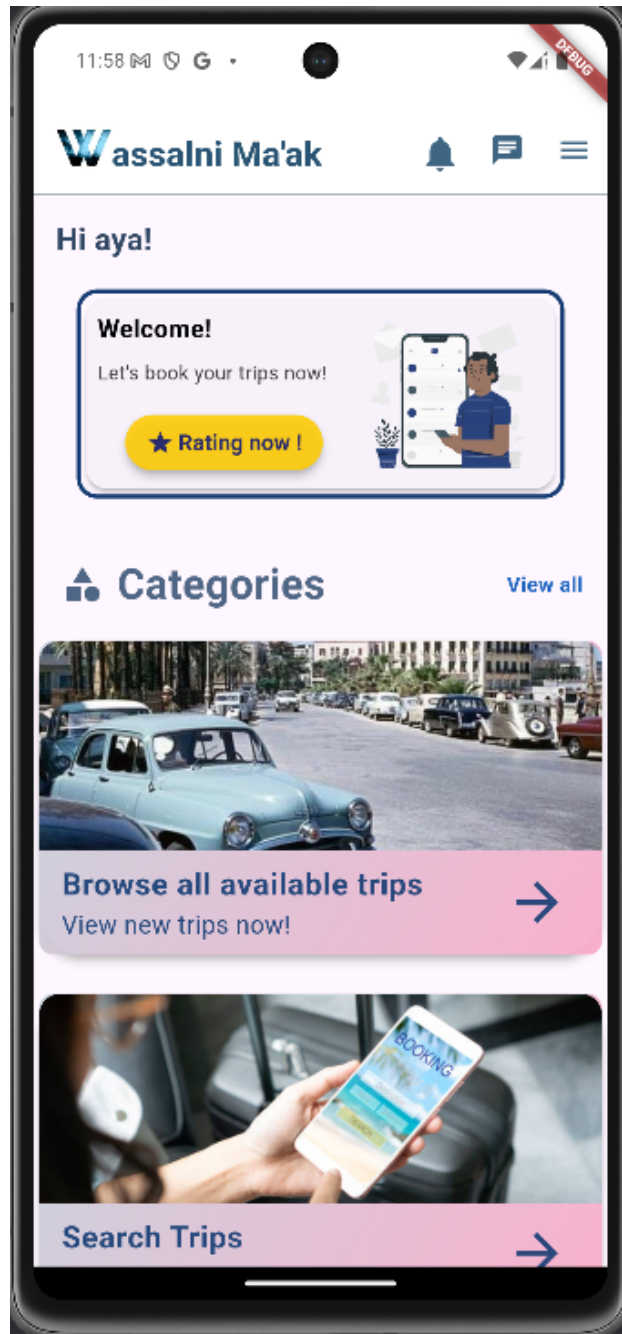


Figure 4.14: Passenger Home Page1

On the **main page**, when the passenger enters the application, there are several branched **categories**, and the passenger can navigate between them and choose what he wants from them. At the top, something appears in a striking way through which he can evaluate the flights he booked and the trip ended.

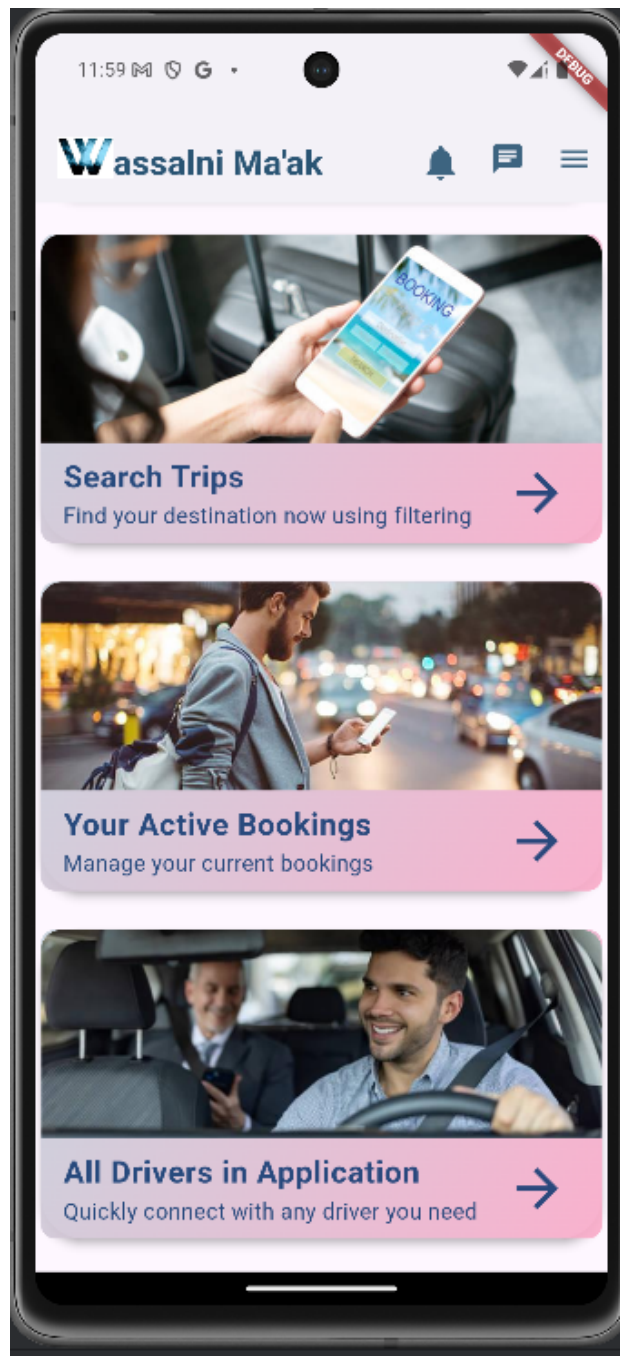


Figure 4.15: Passenger Home Page

From this **category** that displays the trips, the user can choose when entering the Page whether he wants to see **all the trips** or see **the trips close to him** by specifying his location. He can browse both sides as he wants.

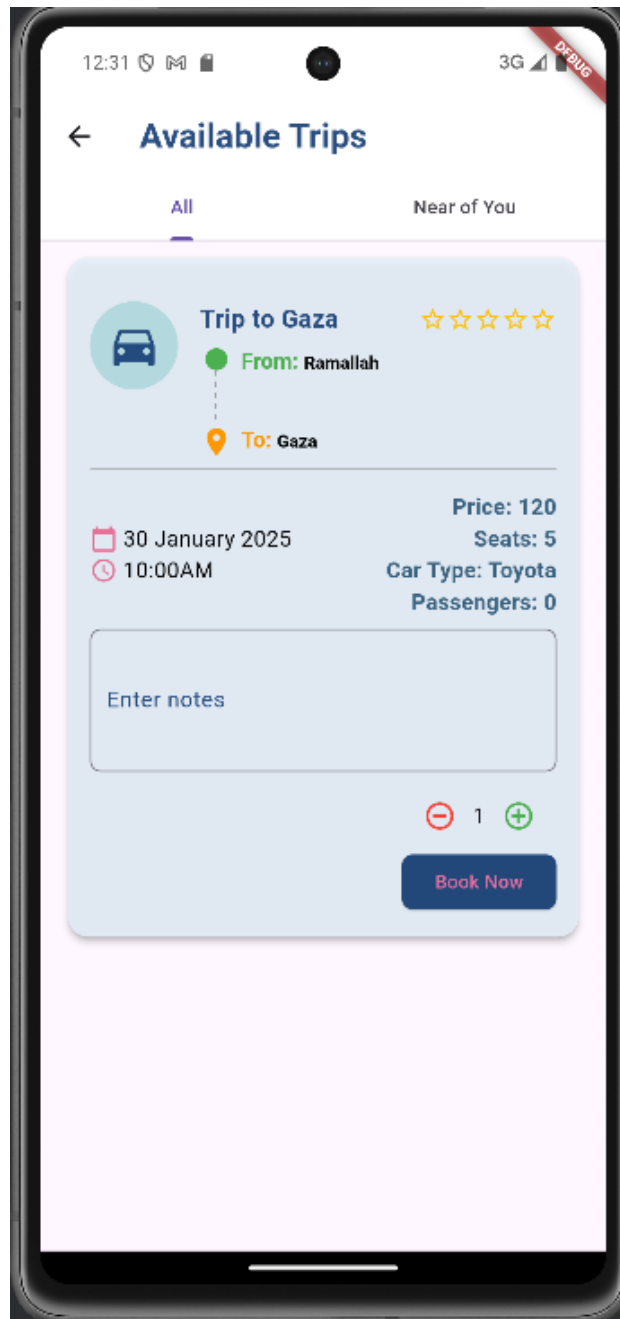


Figure 4.16: Browse all available trips
all the trips advertised by the drivers, and the passenger can book them.

Trips close to the user appear on this page. The application determines his location and then displays all trips surrounding or close to his location.

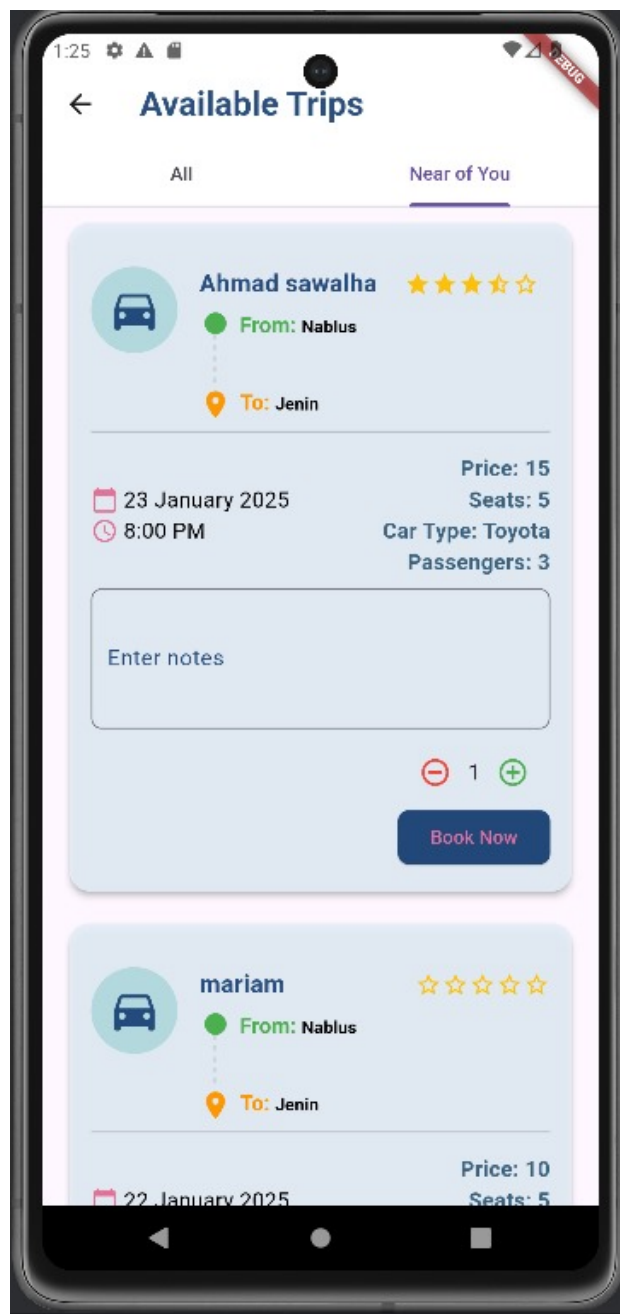


Figure 4.17: Browse all available Nearby trips

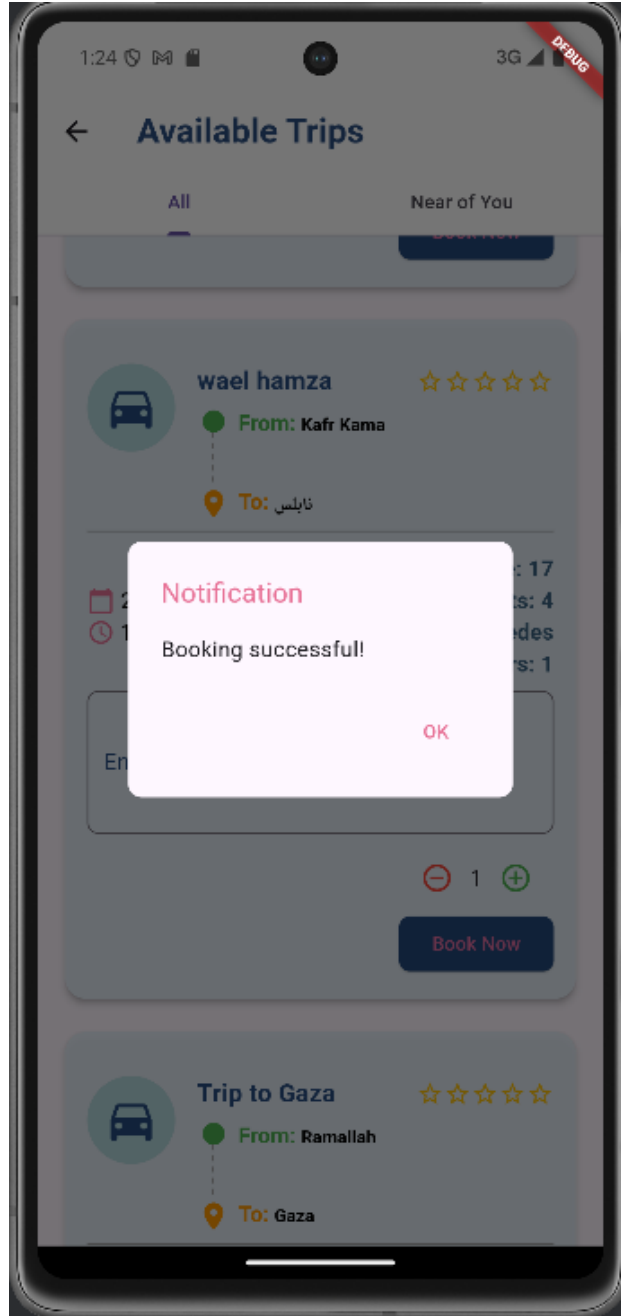


Figure 4.18: Booking in Trip!
The message that appears after booking a specific Trips.

By selecting the search category for a trip, the user can search for a specific trip using any type of filter he wants from the existing filters and apply it, and the results will appear based on the filter that was selected.

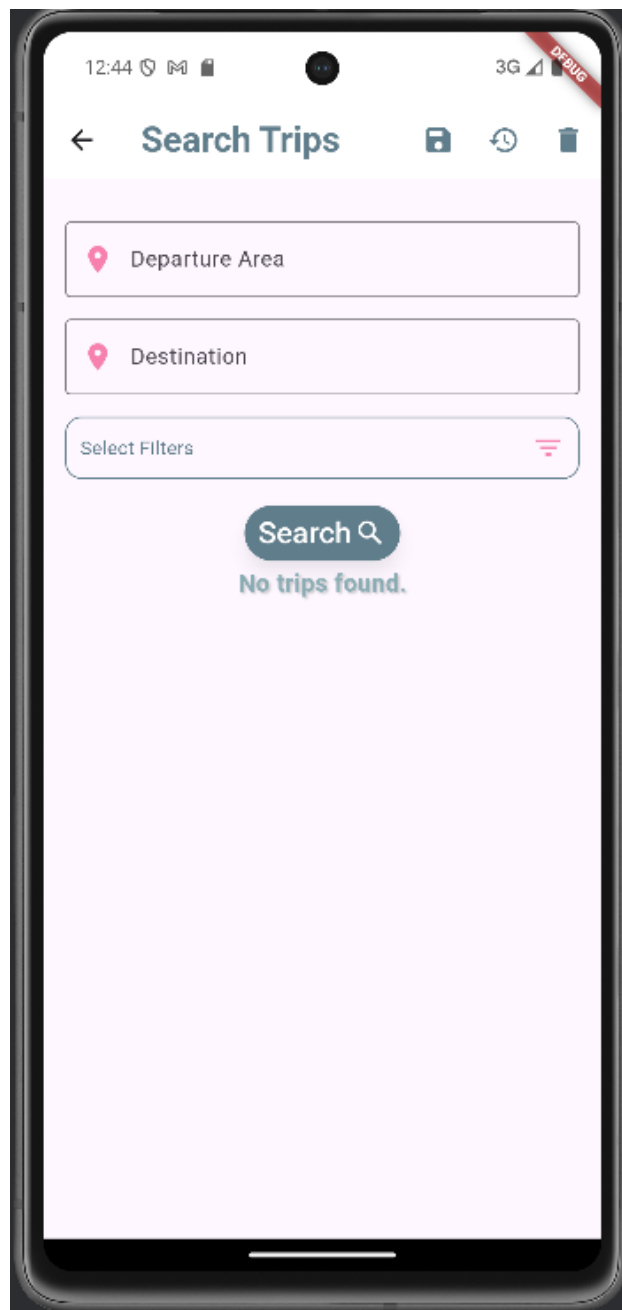


Figure 4.19: search page for trips

The following pages show the filters that the user can choose, and the following image shows the results of the filtering according to the price filter. Note that I used the price filter here as an example, and the rest of the filters work in exactly the same way.

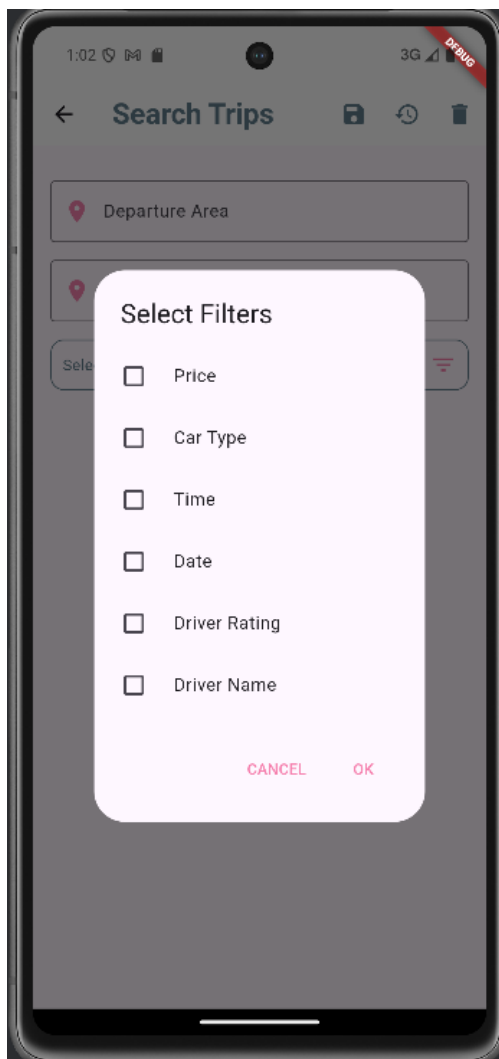


Figure 4.20: Filters you can choose to search for a trip

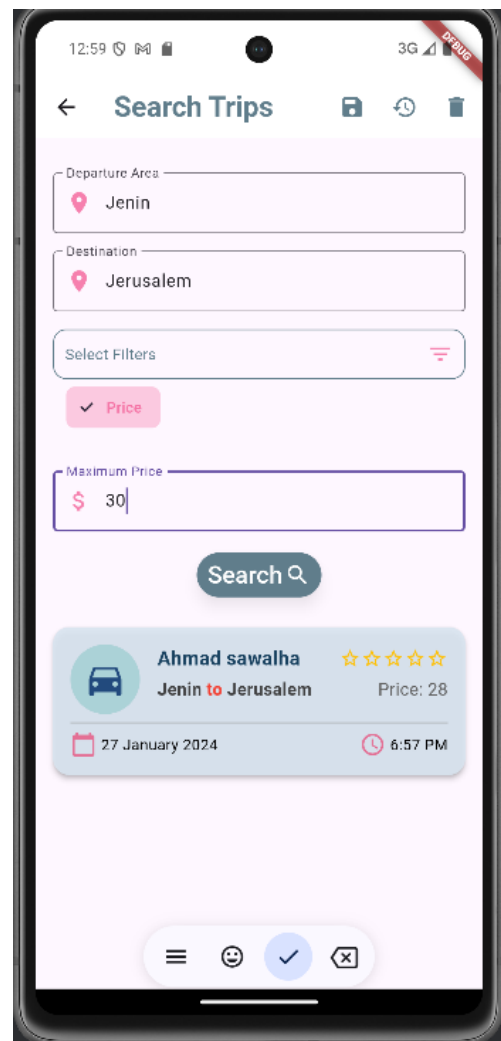


Figure 4.21: Search for specific trips

By entering the reservations category, the user can view all the reservations he has made, see his future trips, see the reservation details and information, and enter the page of the driver he booked with and chat with him as well.

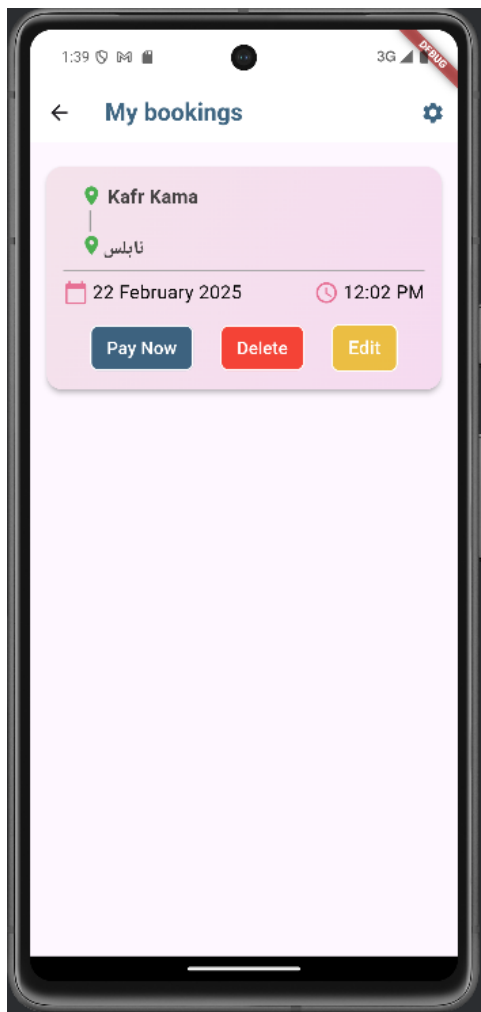


Figure 4.22: Active Booking

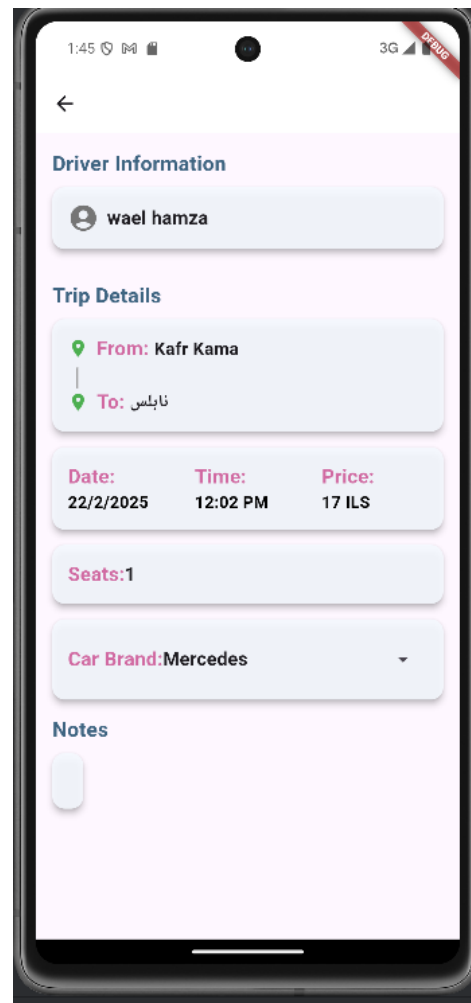


Figure 4.23: booking details

The card that displays the reservation information made by the passenger also contains the driver's name and his profile icon. When the passenger enters it, he can see the driver's personal page, communicate with him, chat with him, and do other things.

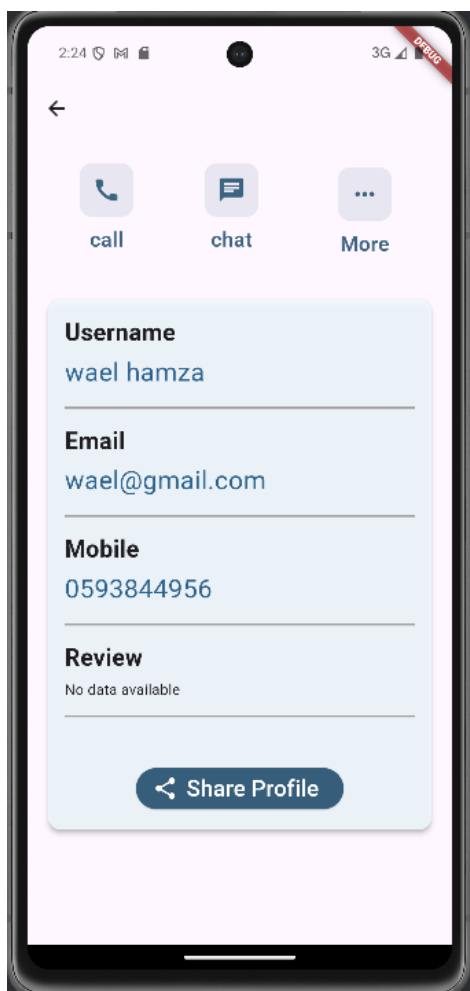


Figure 4.24: Driver profile.

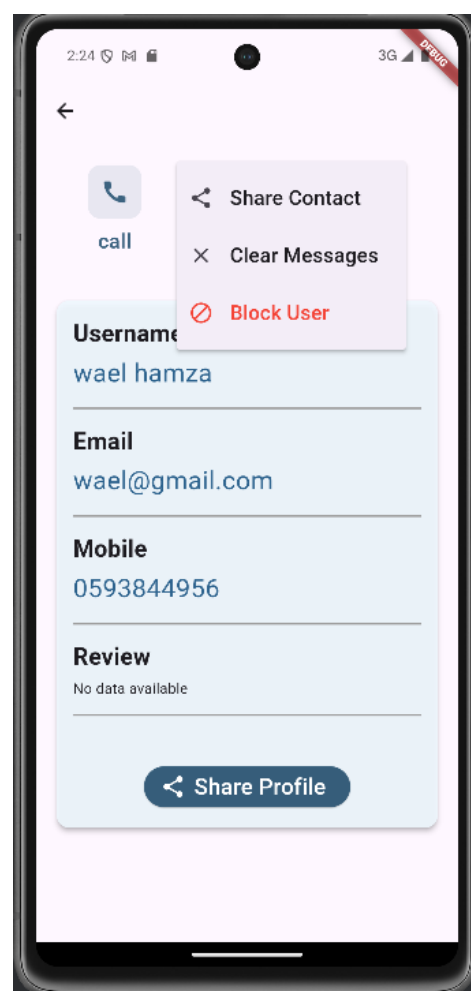


Figure 4.25: Driver profile-2

There are other things that the passenger can do regarding the reservation he made. He can cancel the reservation or modify certain details in the reservation, such as the number of seats he wants to reserve and add a note.

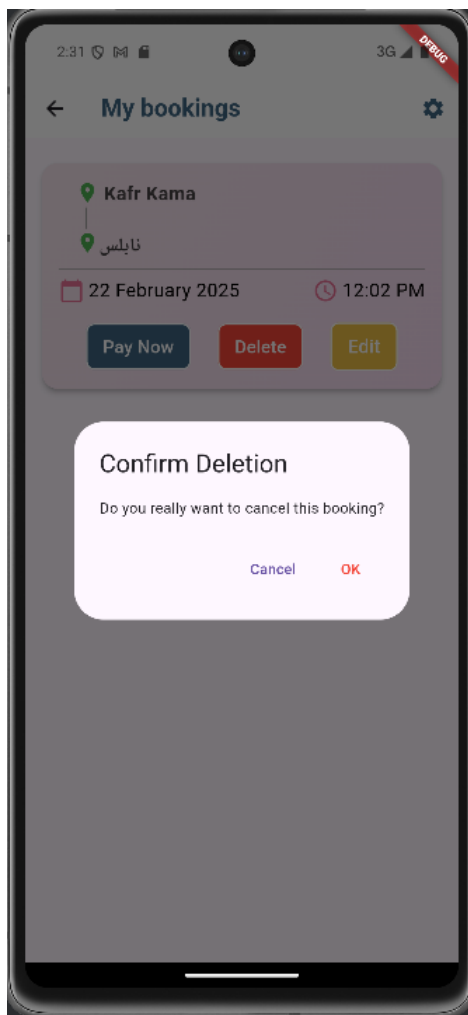


Figure 4.26: cancellation

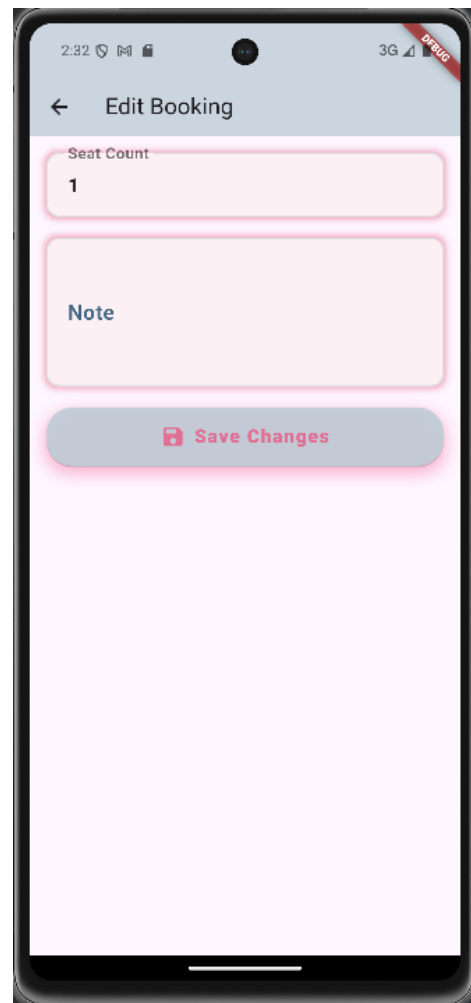


Figure 4.27: Edit booking

The passenger can pay when booking a trip using Visa and entering the necessary information. When the payment is made correctly, a notification is sent to the driver that the payment process was completed correctly.

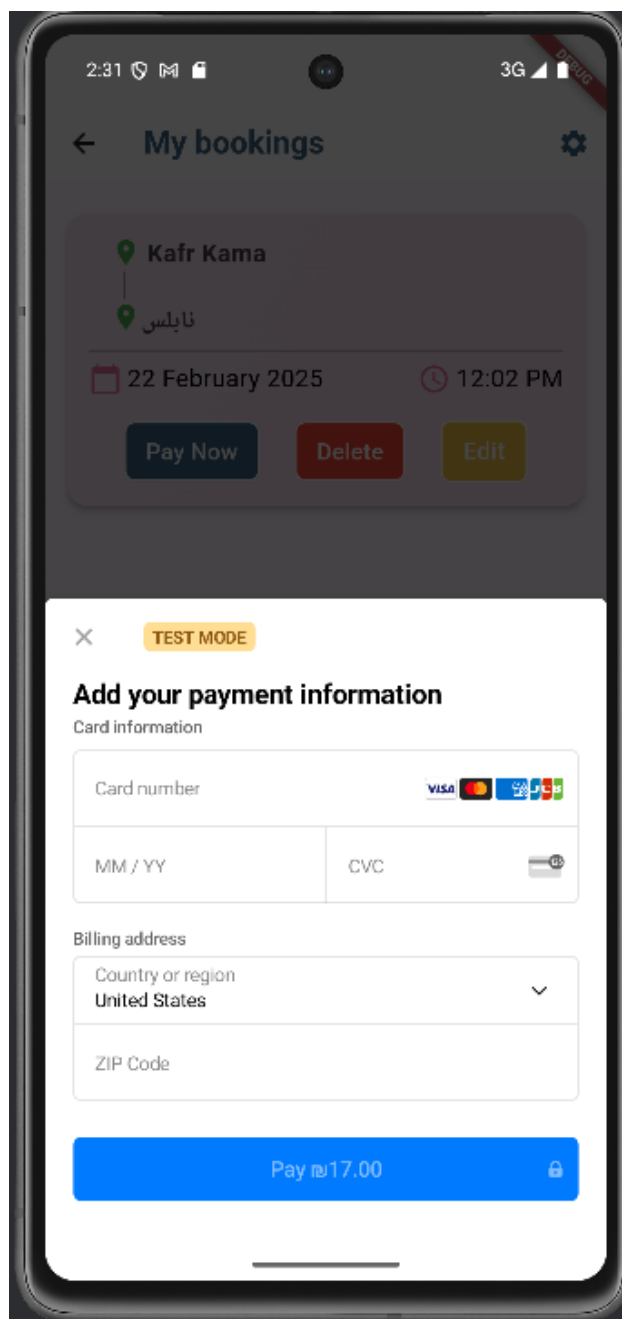


Figure 4.28: Payment by Visa

Category to display all drivers on the application. You can access their pages, exchange messages with them, add them, and see their trips.

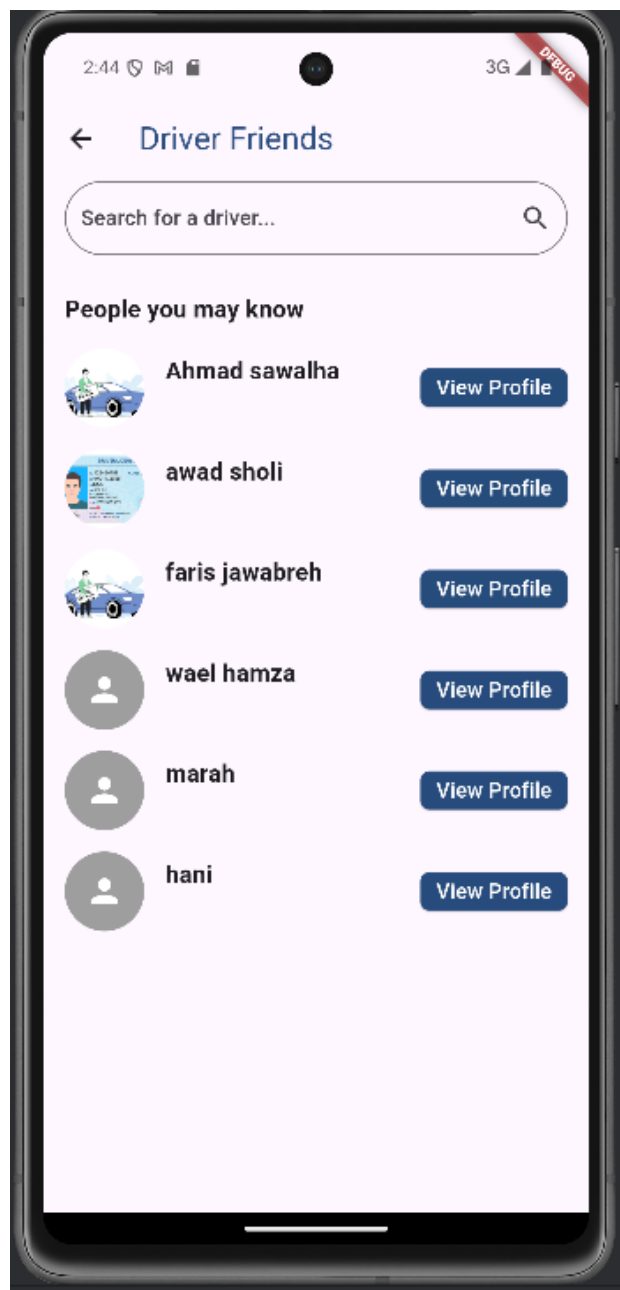


Figure 4.29: All Drivers In App

4.4.4 Browse the menu at the top of the page.

By clicking the three lines at the top of the page, the user can access everything related to their personal account, as shown in the figure below.

Through the menu, the user can view and manage their profile, settings, privacy options, and other features.

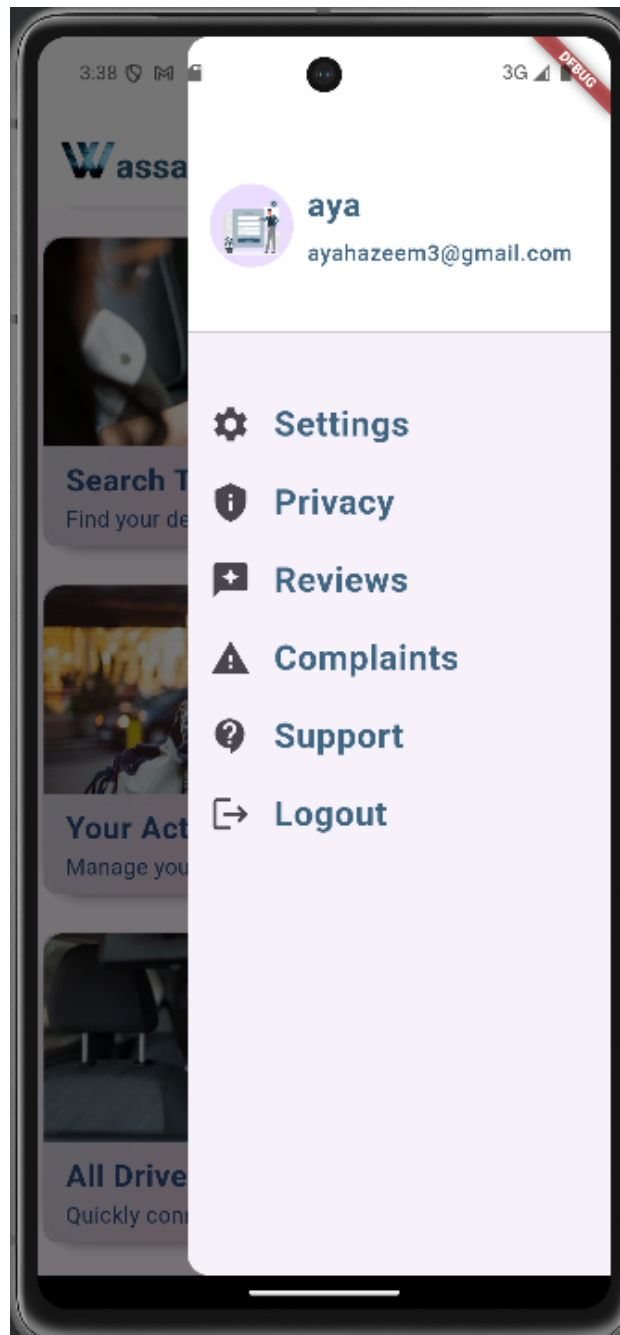


Figure 4.30: Menu

If I enter the profile, I will find this page, through which I can do several things, such as setting a profile picture, controlling the status of my profile, and also controlling some chat features. By entering the account as shown in the second image can do several things such as change the password and browse the personal details.

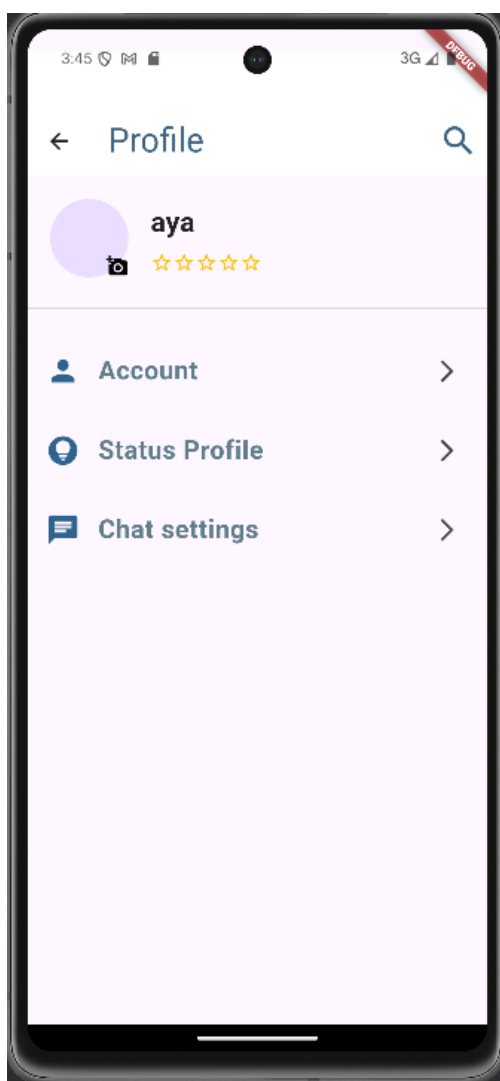


Figure 4.31: My Profile

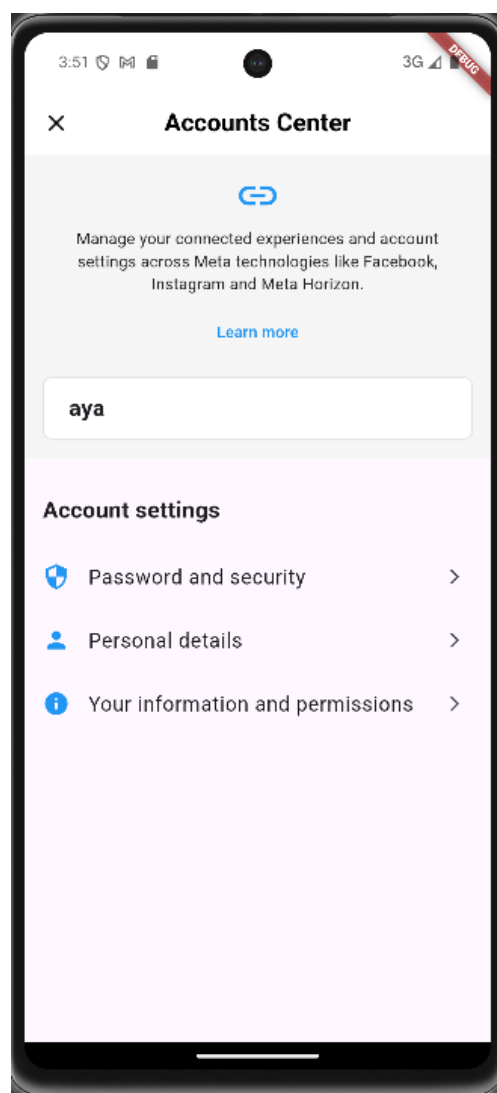


Figure 4.32: Account

When I entered the **Account**, then to change the password, the data details page shows all of these images.

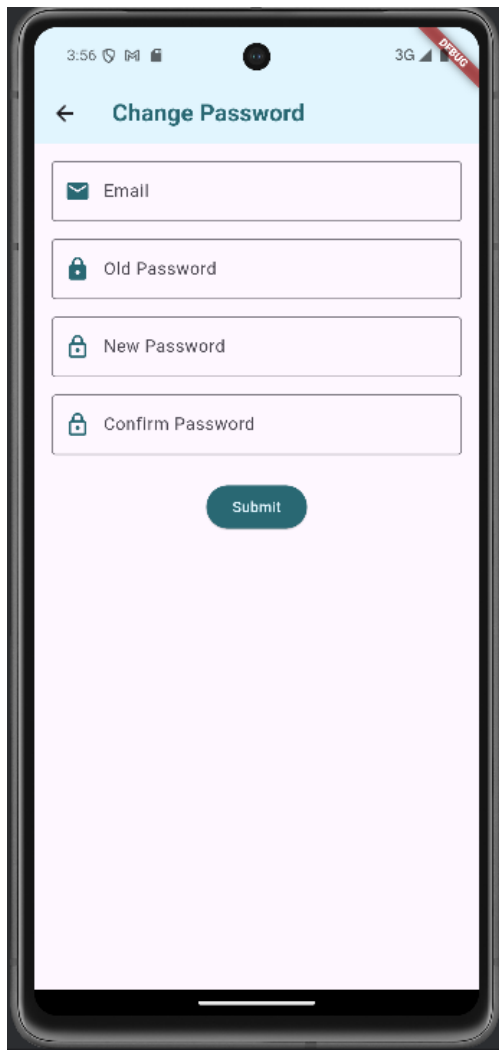


Figure 4.33: Change Password
By entering this page, the user can change his password.

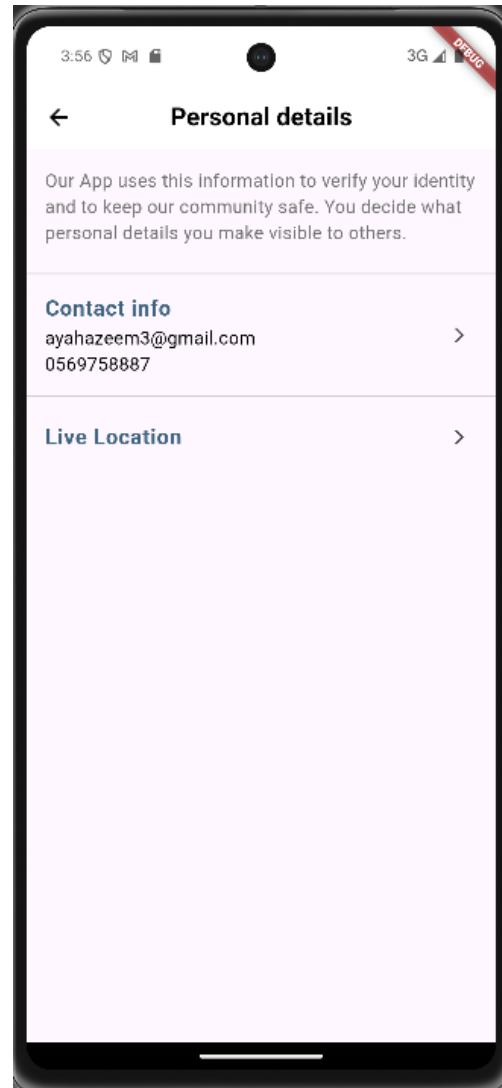


Figure 4.34: personal details
By entering this page, the user can see his private information and also allow his location to be determined.

When I entered the **Status Profile**, from Profile Page, This page will appear, through which I can specify the status I want my profile to be in.

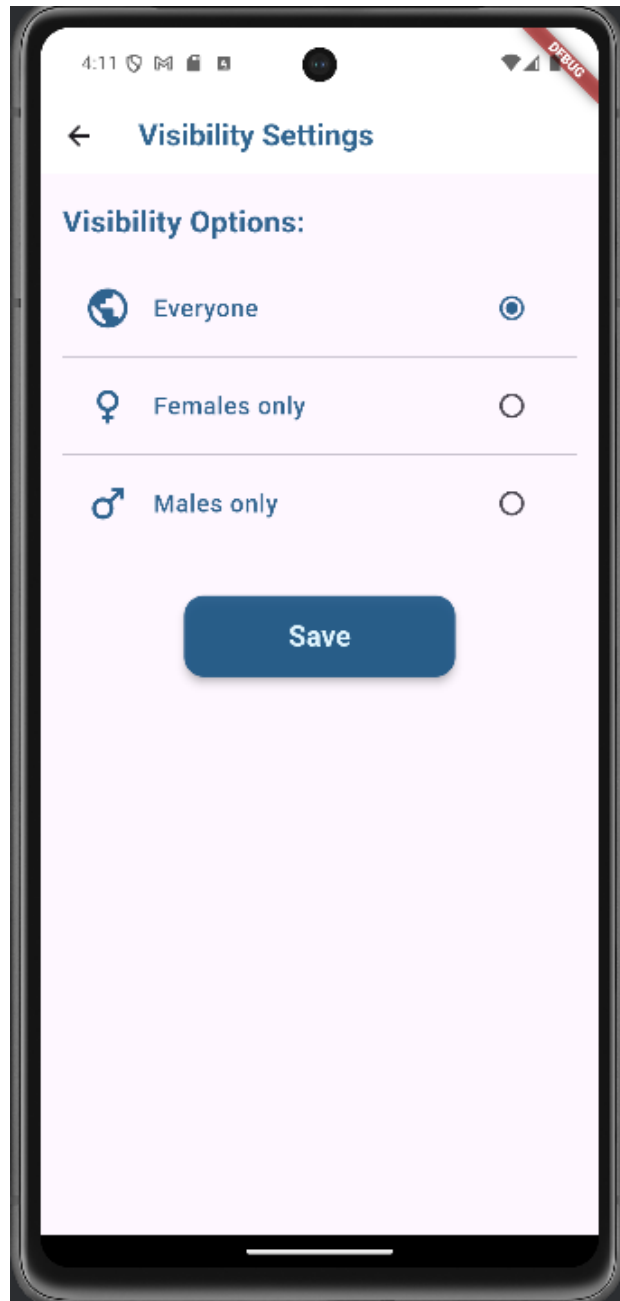


Figure 4.35: Status Profile

By specifying the profile status, a person can specify who they want to view their profile.

From the main **menu**, when I enter the **settings page**, this page will appear, which has several options that I can do, and all of them appear in the image, such as choosing the language I want for the application or theme, and many other things.

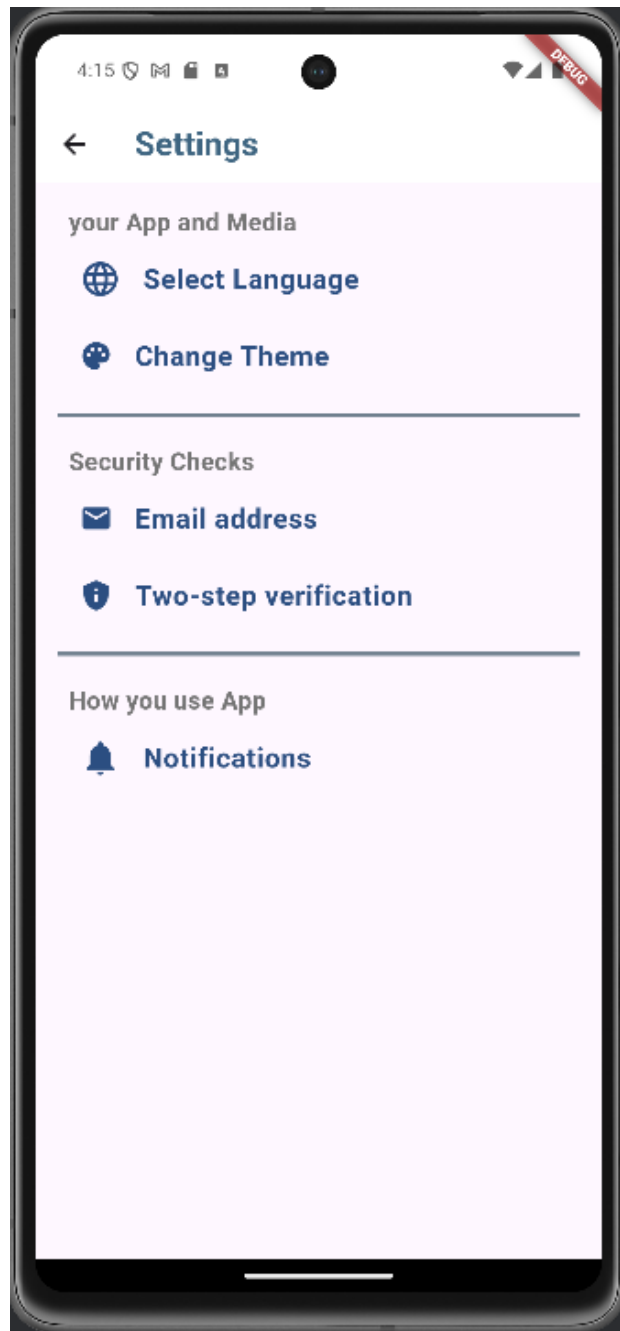


Figure 4.36: Settings

Now when I click on **select language** from the **settings page**, this pop-up screen will appear from which I can choose Arabic or English.

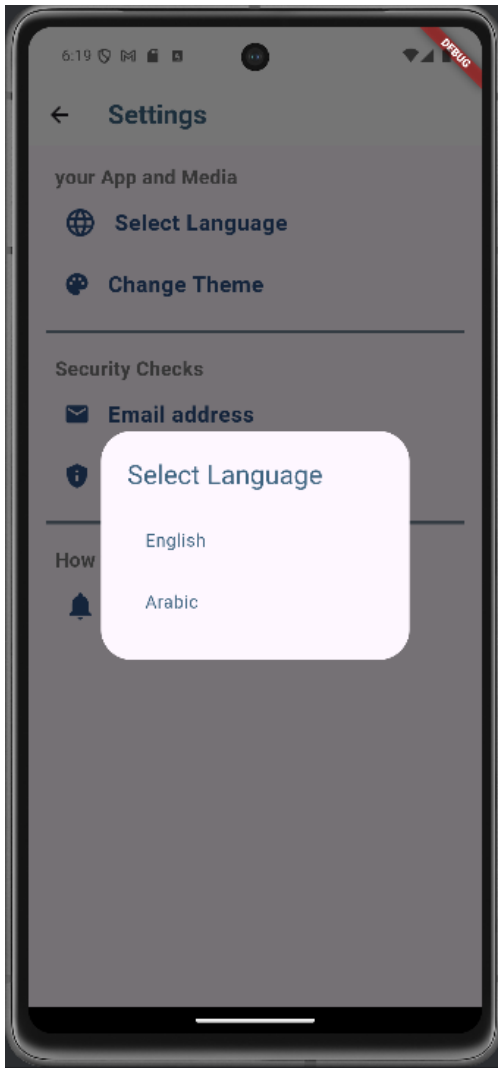


Figure 4.37: Select Language



Figure 4.38: Application in Arabic

Now when I click on **Change Theme** from the **settings page**,The application colors will change to black theme.

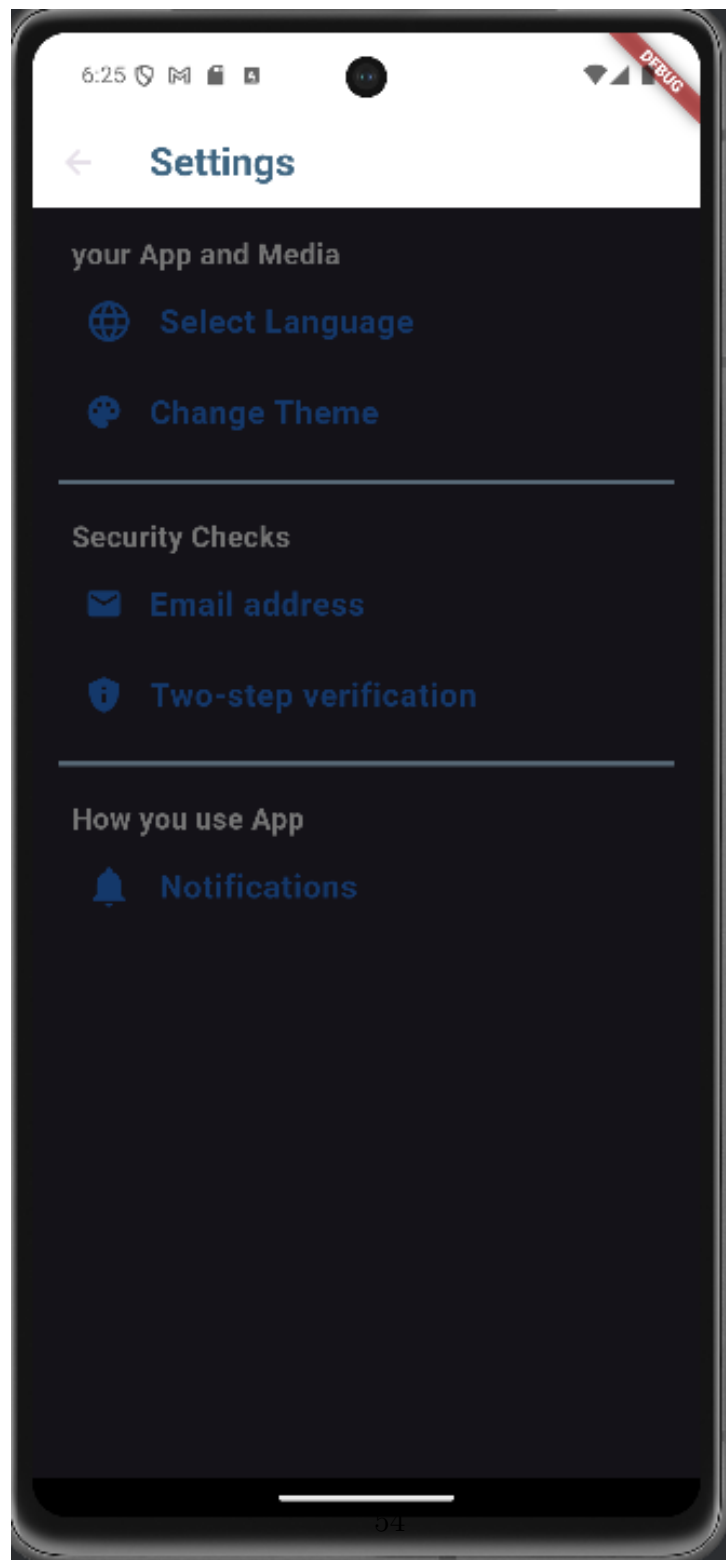


Figure 4.39: Change Theme

Now when I click on **Email Address** from the **settings page**,A new page will appear for the user, through which he can add his email and verify it so that his account remains protected from hacking,When I click on **two-step verification**, another page appears through which I can do all the necessary things for verification.

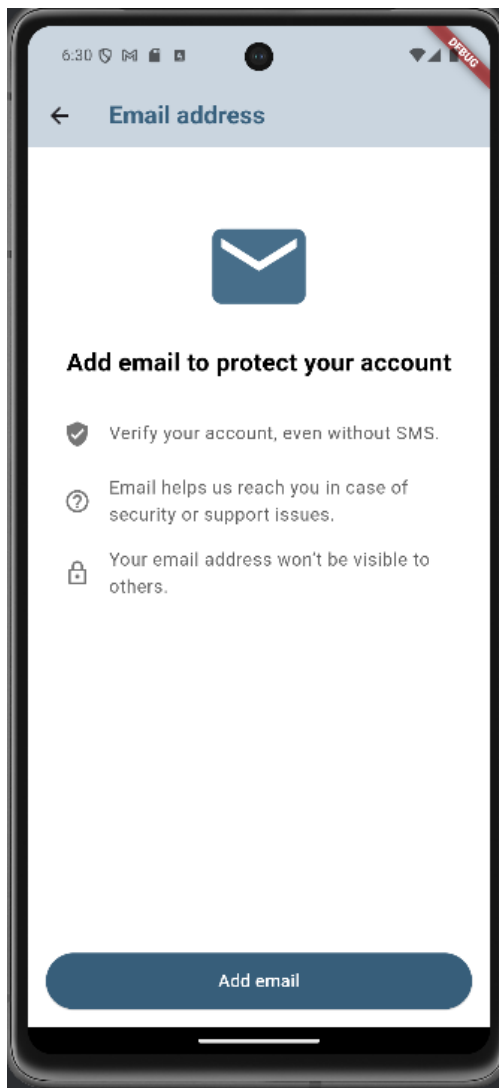


Figure 4.40: Email Address

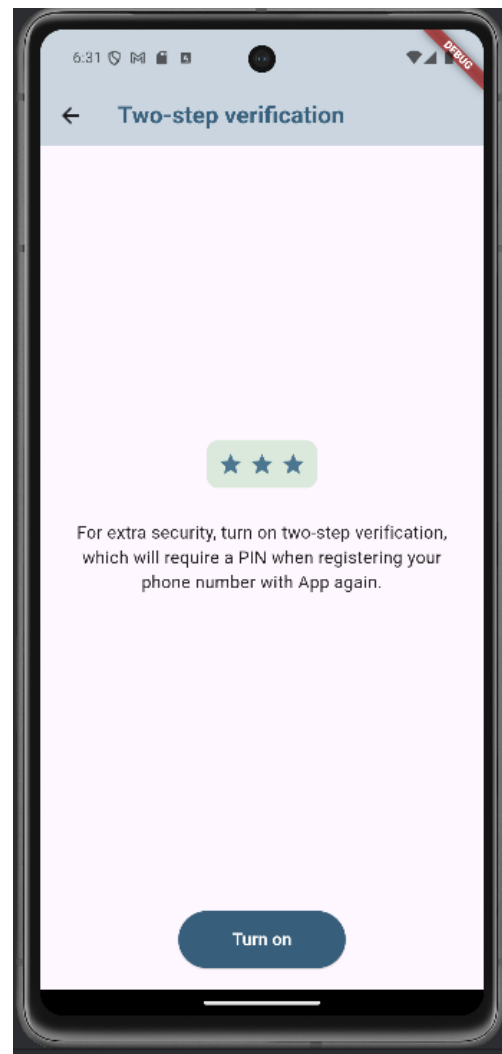


Figure 4.41: Two-Step Verification

From the main **menu**, when I enter the **Privacy page**, this page will appear, which has several options that I can do, and all of them appear in the image, And control all account privacy.

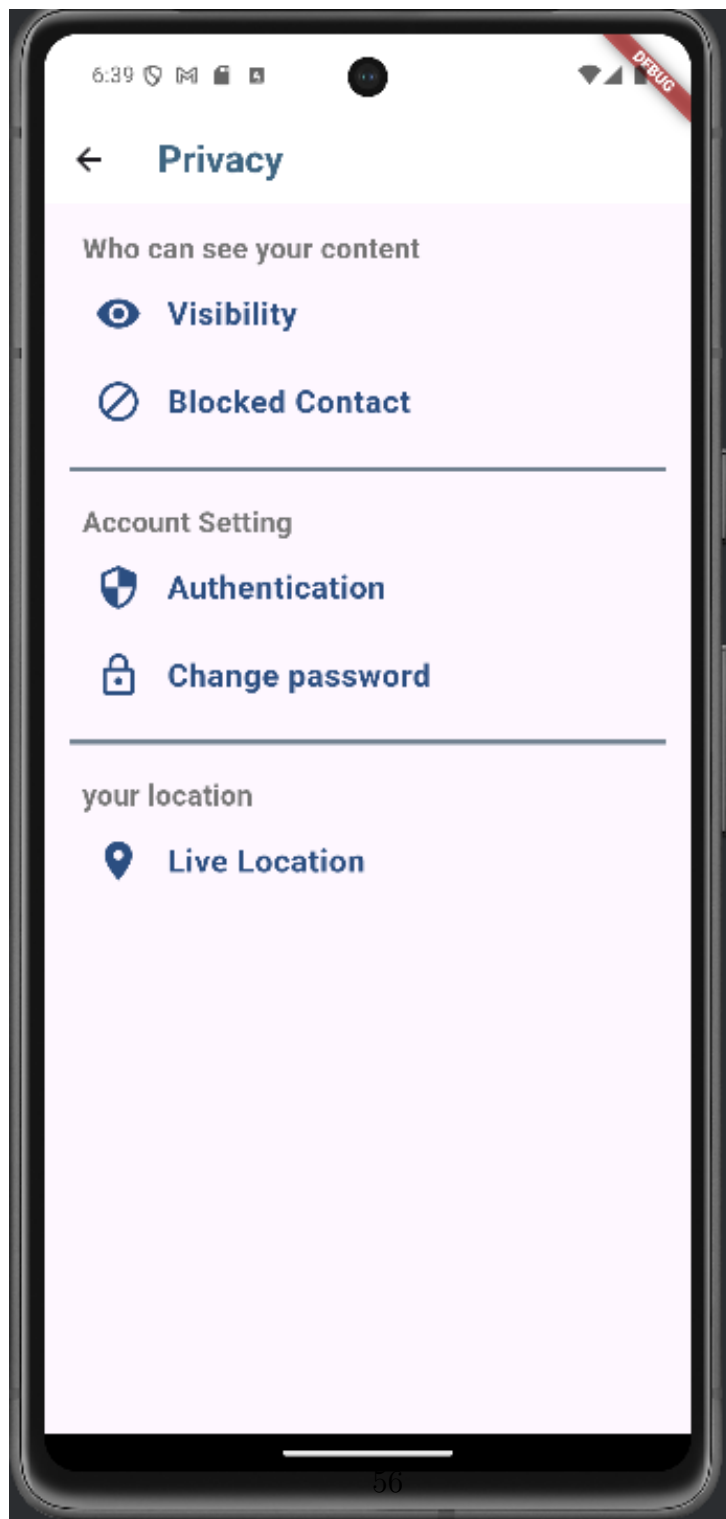


Figure 4.42: Privacy Page

When I enter the **block Contact** from the **privacy section**, I can see all the people I have blocked, and I can also add anyone to the block list by entering their name or entering their mobile number.

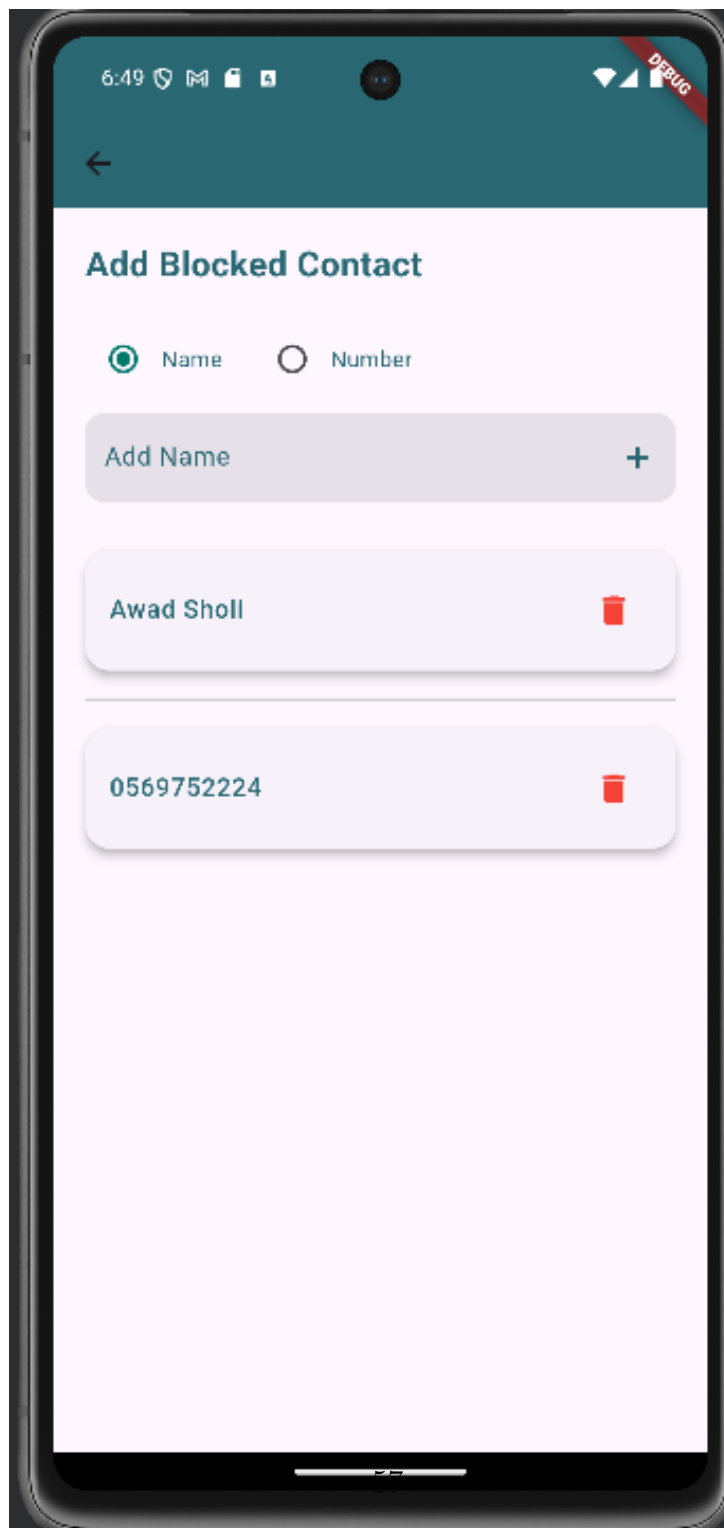


Figure 4.43: Block Contact

From the main **menu**, when I enter the **Complaints page**, Through which the user can file a complaint against another person, whether a passenger or a driver.

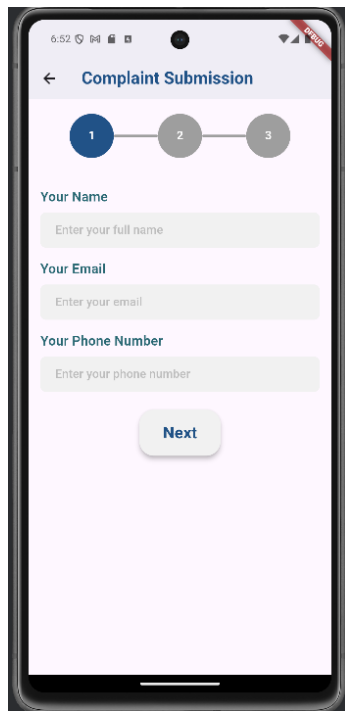


Figure 4.44: Complaints Step1

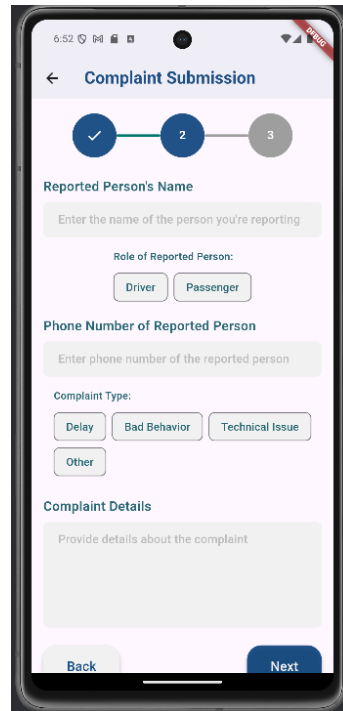


Figure 4.45: Complaints Step2

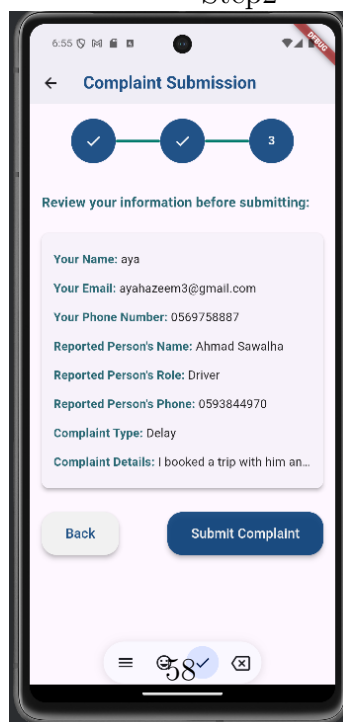


Figure 4.46: Complaints Step3

From the main **menu**, when I enter the **Support page**, You can see all the ways and means that help the user solve a problem he has or request an inquiry from the application officials, whether by sending an email or communicating directly via WhatsApp and other ways to answer all inquiries.

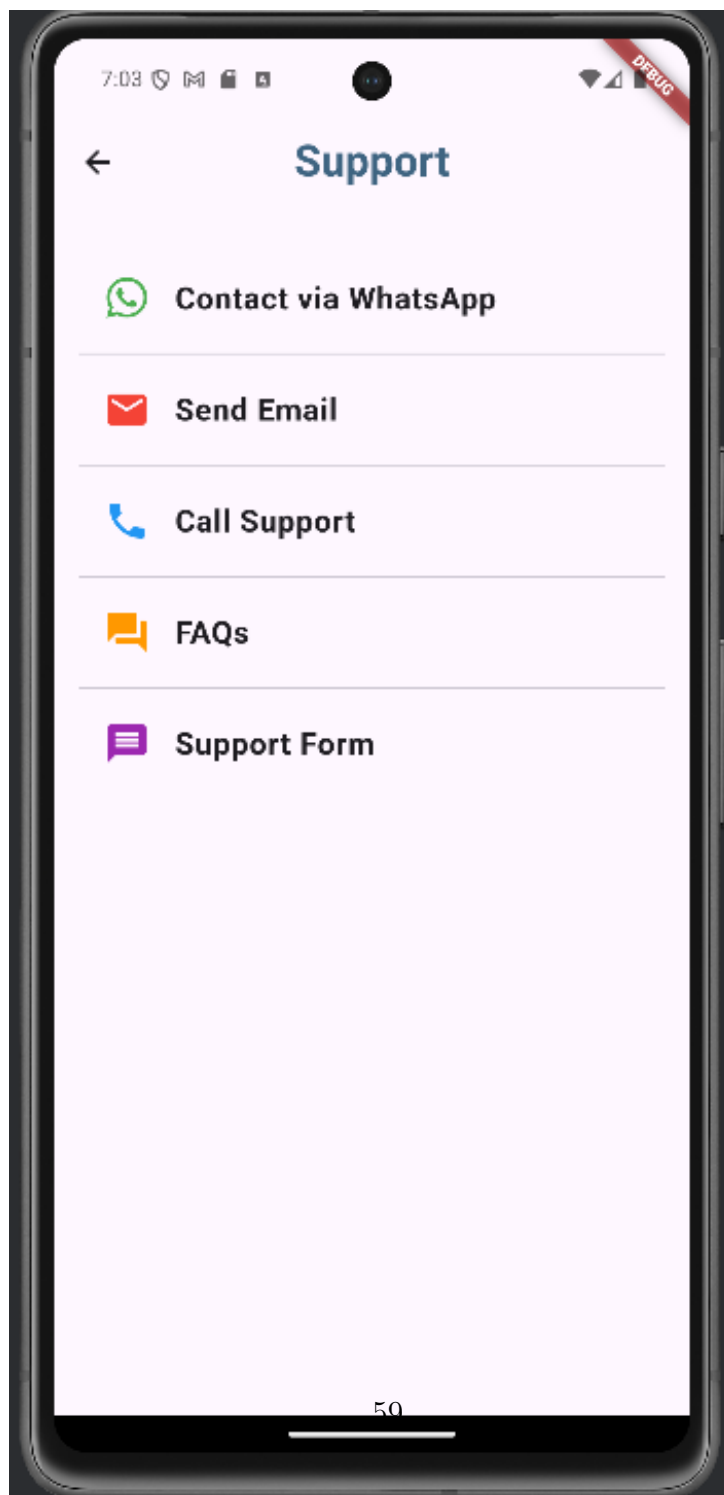


Figure 4.47: Support

From the **Support**, when I enter the **Contact Via WhatsApp** ,It takes me directly to the number responsible for support that you contact on WhatsApp



Figure 4.48: Open WhatsApp to contact support

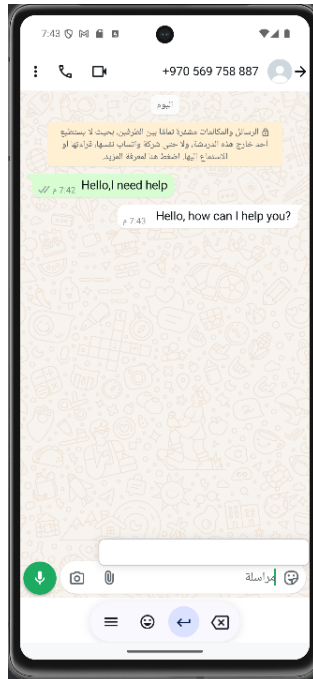


Figure 4.49: Conversation from sender side - app user

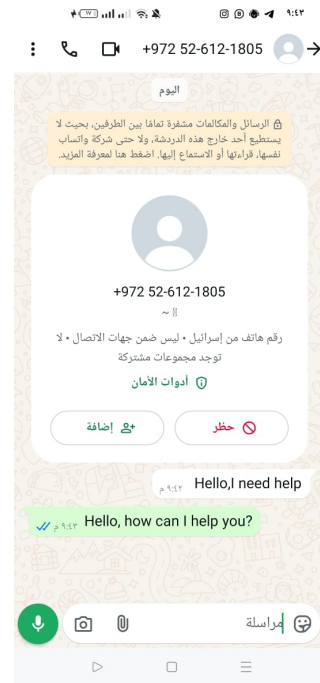


Figure 4.50: Conversation from the receiver side-the support side

From the **Support**, when I enter the **Send Email**, Who sends me messages directly to the Gmail application linked to the application's email

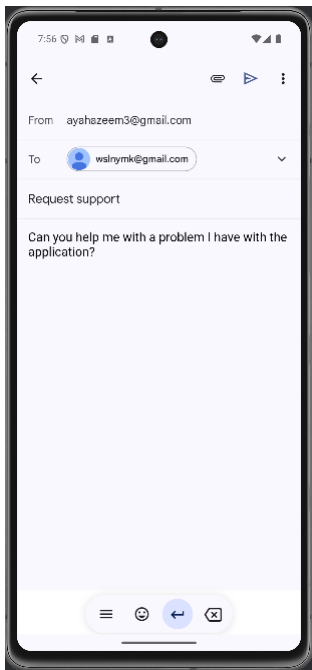


Figure 4.51: Open gmail to contact support

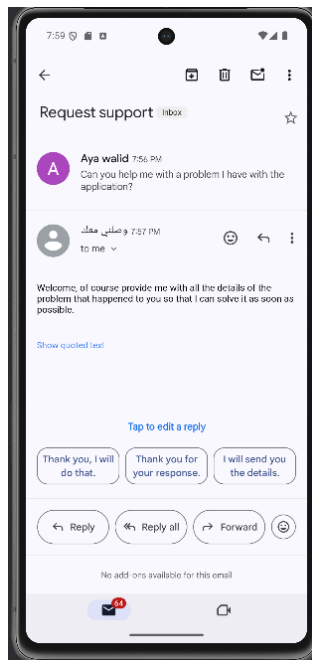


Figure 4.52: Mail sent by app user

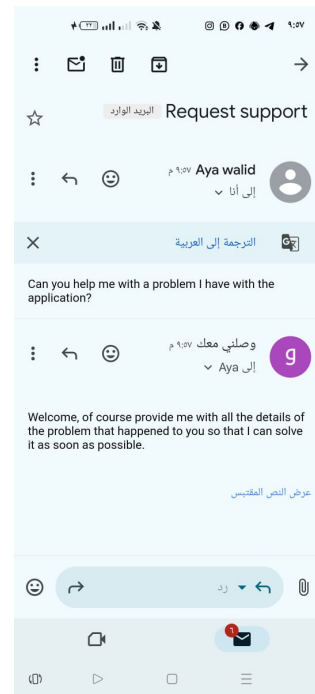


Figure 4.53: Mail received from the application support

From the **Support**, when I enter the **FAQs** ,Which allows me to see all the common questions that the application users are thinking about and looking for answers to.

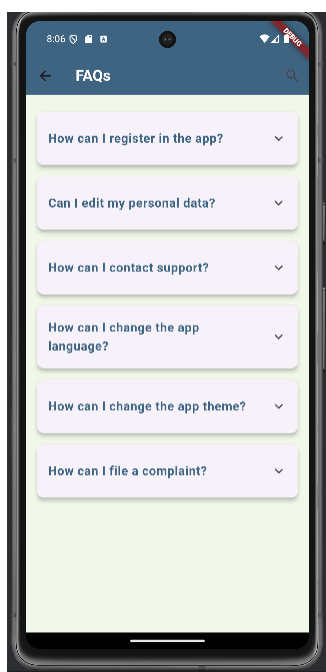


Figure 4.54: FAQs

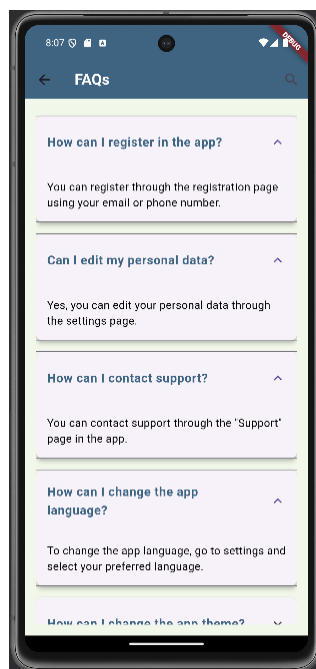


Figure 4.55: Show answers to FAQs

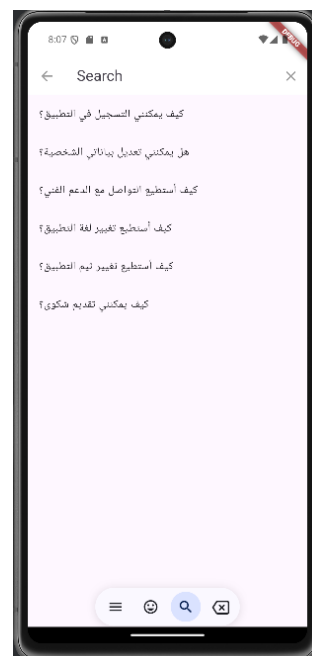
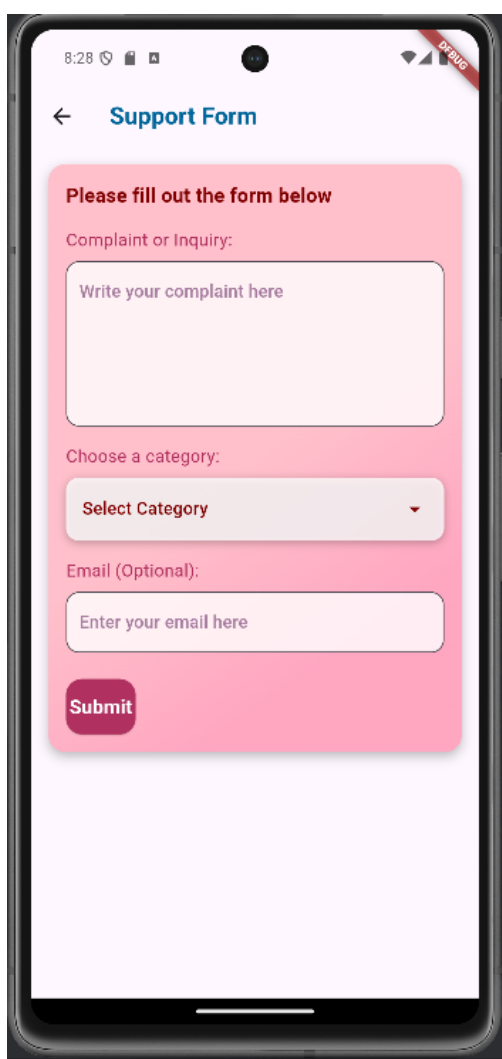


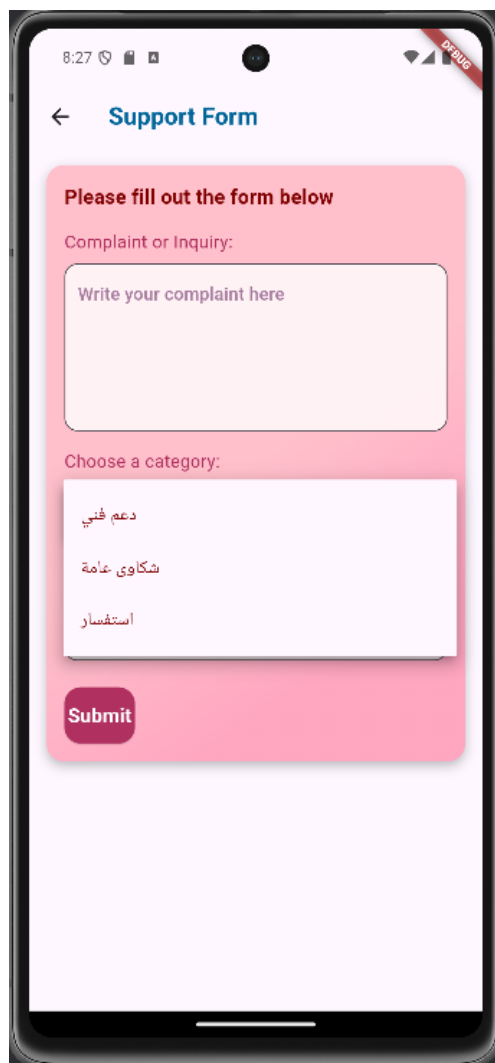
Figure 4.56: Search for FAQs

From the **Support**, when I enter the **Support Form** ,Through which the user can submit a complaint or send a specific inquiry regarding a topic he is looking for to the admin.



The screenshot shows a mobile app interface for a 'Support Form'. At the top, there is a back arrow and the title 'Support Form'. Below this, a pink header contains the text 'Please fill out the form below'. The form consists of three main sections: 1. 'Complaint or Inquiry:' with a large text input field containing the placeholder 'Write your complaint here'. 2. 'Choose a category:' with a dropdown menu currently showing 'Select Category'. 3. 'Email (Optional):' with a text input field containing the placeholder 'Enter your email here'. At the bottom of the form is a red 'Submit' button. The status bar at the top shows the time as 8:28 and various system icons.

Figure 4.57: Support Form-1



The screenshot shows the same mobile app interface as Figure 4.57, but with the category dropdown menu expanded. The dropdown menu is open, showing three options in Arabic: 'دعم فني' (Technical Support), 'شكاوى عامة' (General Complaints), and 'استفسار' (Inquiry). The 'Submit' button remains at the bottom. The status bar at the top shows the time as 8:27.

Figure 4.58: Support Form-2

4.4.5 Driver User Interface (UI)

Now let's go through the Driver interface that the user will interact with in the application. These interfaces represent the Driver side of the Wassalni app.

Here on the driver's home page there is his name and current location and there is also a section for trips so that the trip consists of 3 categories.

trips that are considered upcoming and their date has not yet arrived and the trip card contains several things and details such as the place from which the trip starts and the destination it is going to, it also shows the time and date to remind the driver of that, and shows the number of passengers booked for this trip and this variable changes dynamically according to the passengers' reservations. The driver can also look at the map of the trip route with an estimate of the time for the distance.

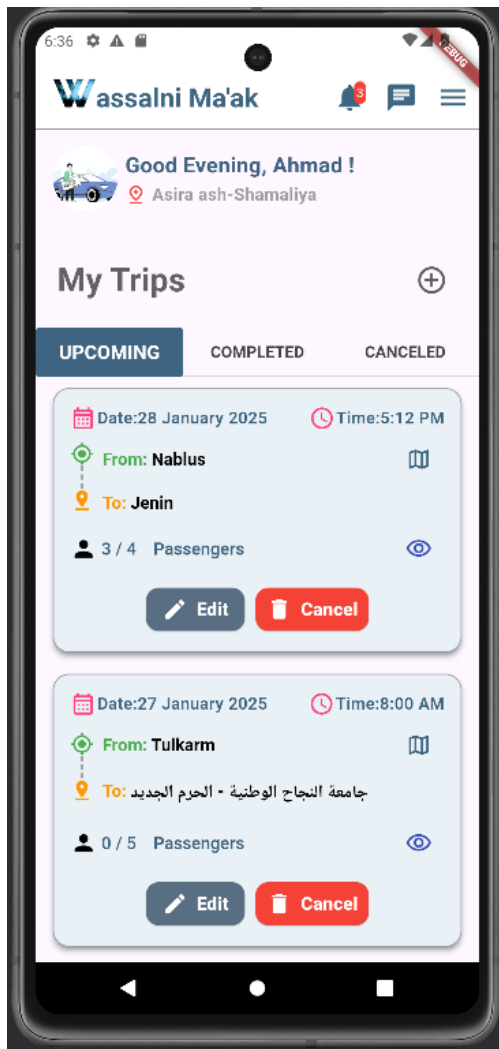


Figure 4.59: Driver Dashboard

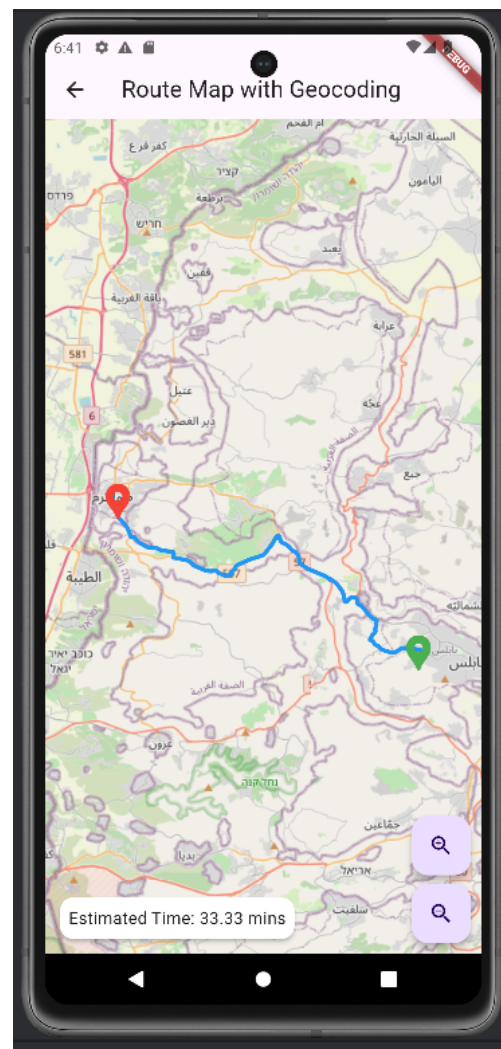


Figure 4.60: Passenger List

and he can also view the passengers booked for this trip from the display icon.

The list for the passengers shows some details if there is a note from the passenger or if he paid by Visa card so that the driver can know these details through this window, and if he enters the name of any person from the Passenger List, he enters his personal page through which he can know how to communicate with him either by email, chat or call.

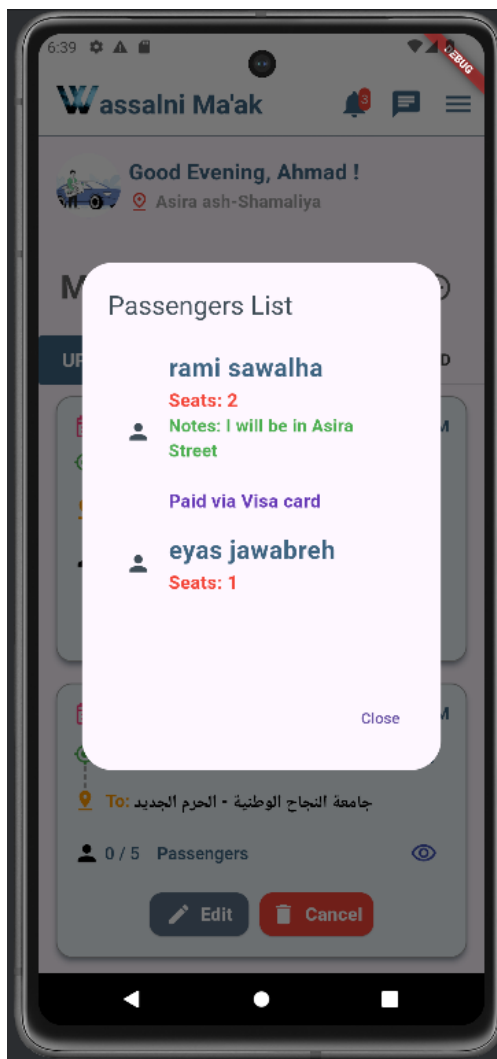


Figure 4.61: Passenger List

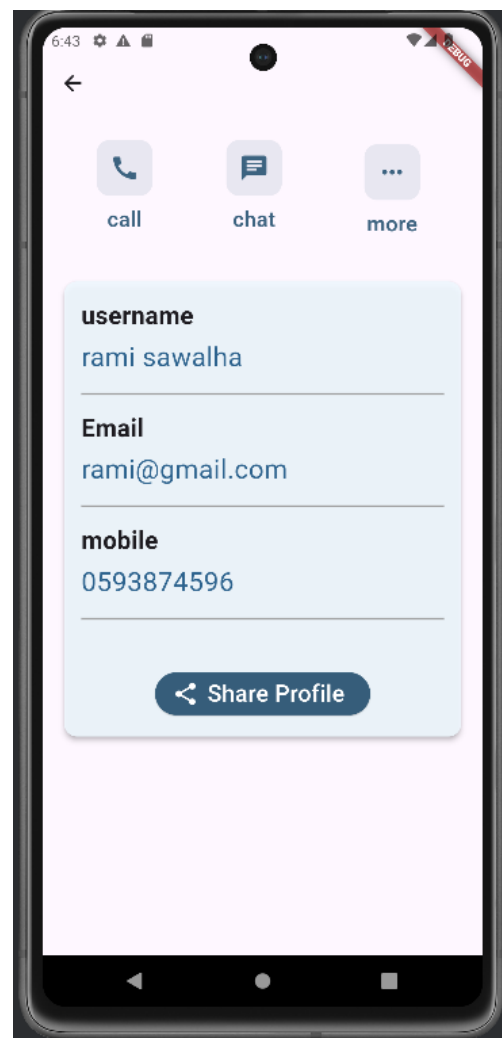


Figure 4.62: Passenger Profile

When the trip ends, it automatically moves to the completed trip section, and the final income earned from this trip appears on each trip.

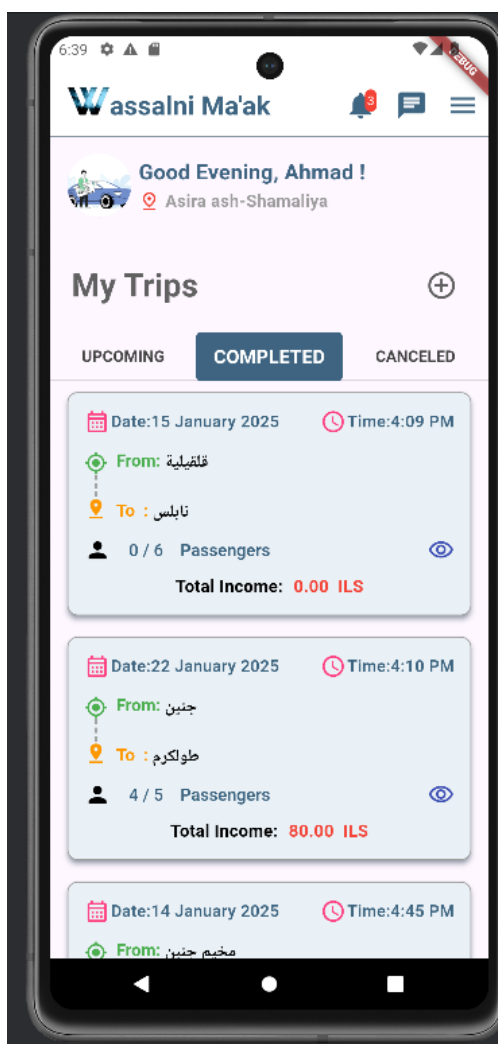


Figure 4.63: Completed Trip

When the driver clicks on the plus Icon on his home page, he is transferred to the **trip creation page**. On this page, the driver creates his trip, adding the trip details from the departure point and destination. He can also capture the departure point relative to his current location via the red location capture icon, the date, time, number of passengers available, and the price of each passenger. He can also specify the trip status to show to everyone, females, or males only.

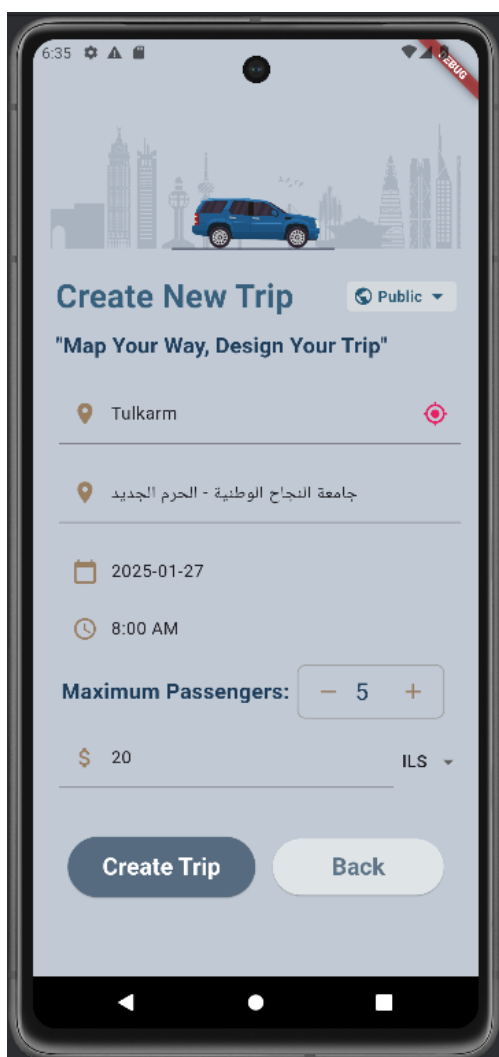


Figure 4.64: Create Trip

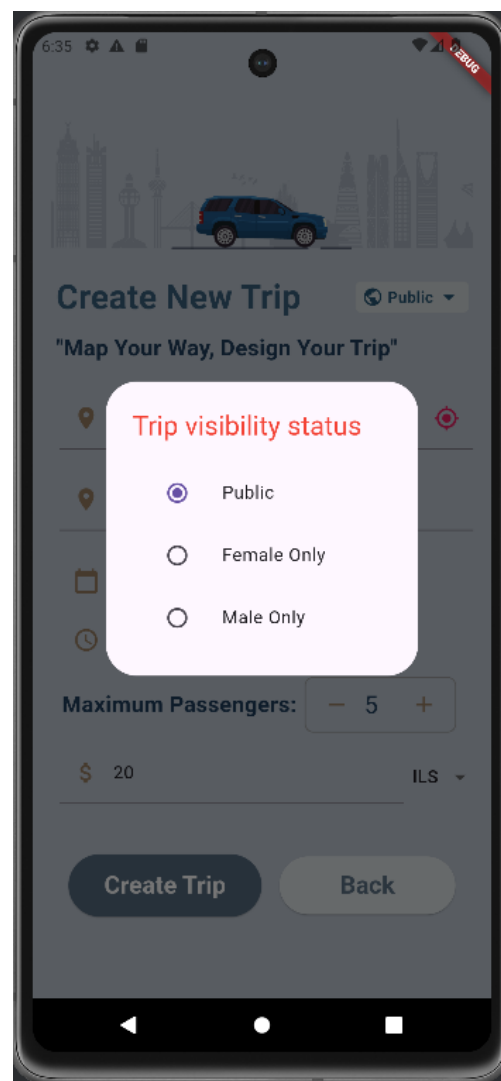


Figure 4.65: Visibility Trip

From the main **menu**, when I enter the **Reviews** page, the feedback page appears, which contains all the feedback from passengers about him and the evaluation of each person with comments, if any. This page aims to develop and improve performance.

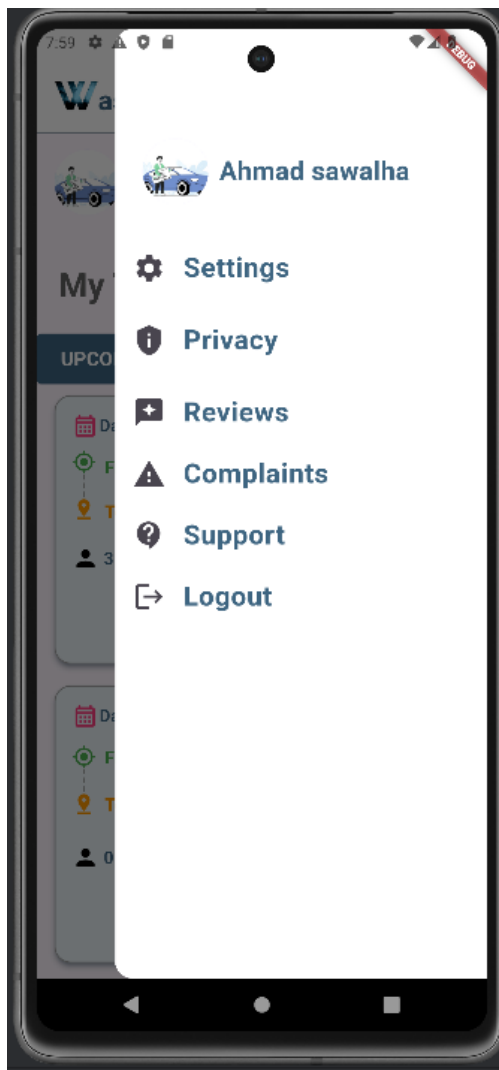


Figure 4.66: Main Menu

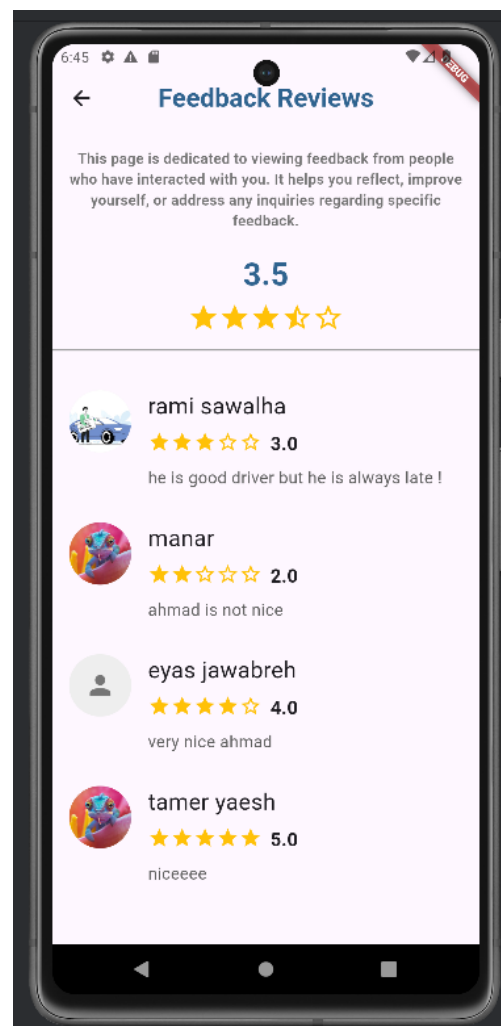


Figure 4.67: Feedbacks

As for the rest of the details of the account itself, such as setting, personal information, etc., we mentioned them in the explanation of Passenger Features. There are also general things in the project itself, such as the **chat** and **notifications** feature. We used **Firestore** to do this. There is a chat list and a chat page for each user in this application.

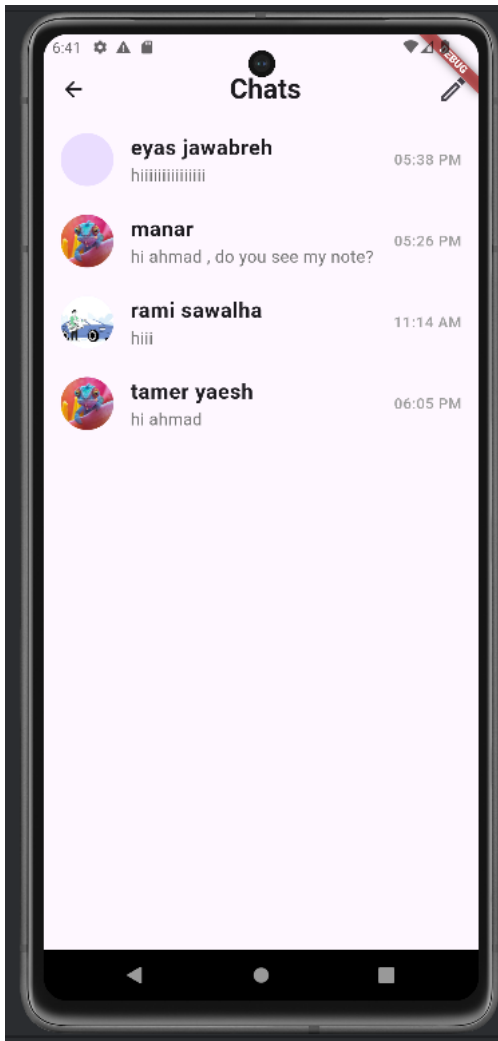


Figure 4.68: Chat List



Figure 4.69: Chatting Page

4.4.6 Admin permissions

Admin Dashboard

On its home page, the user will see some important statistics in the application, such as the number of active users, the total number of upcoming trips, the number of currently existing reservations, and the number of complaints that need to be resolved. There is also a bar chart for male and female user statistics, and there is also a chart for dividing trips.

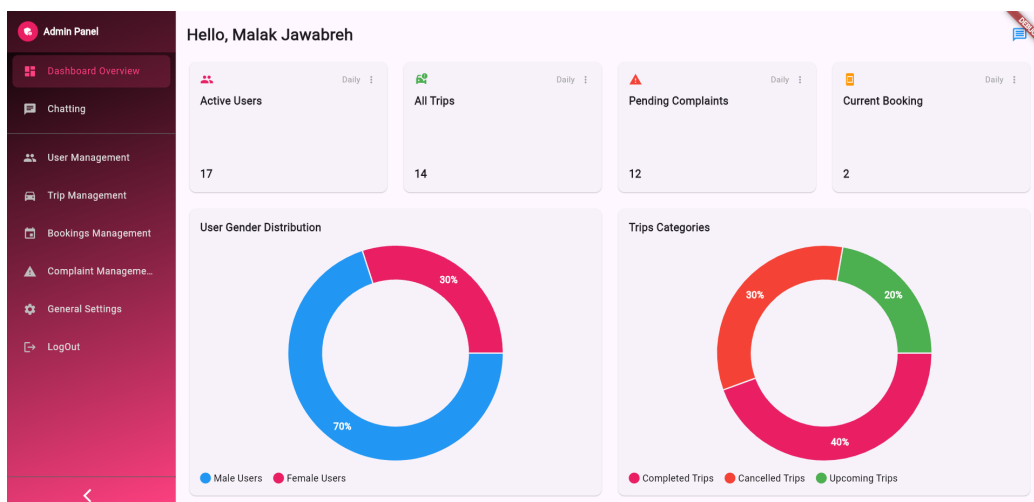


Figure 4.70: Admin Dashboard View

User Management

As for the permissions related to the user, the user's classification appears at the beginning: everyone, passengers, or drivers.

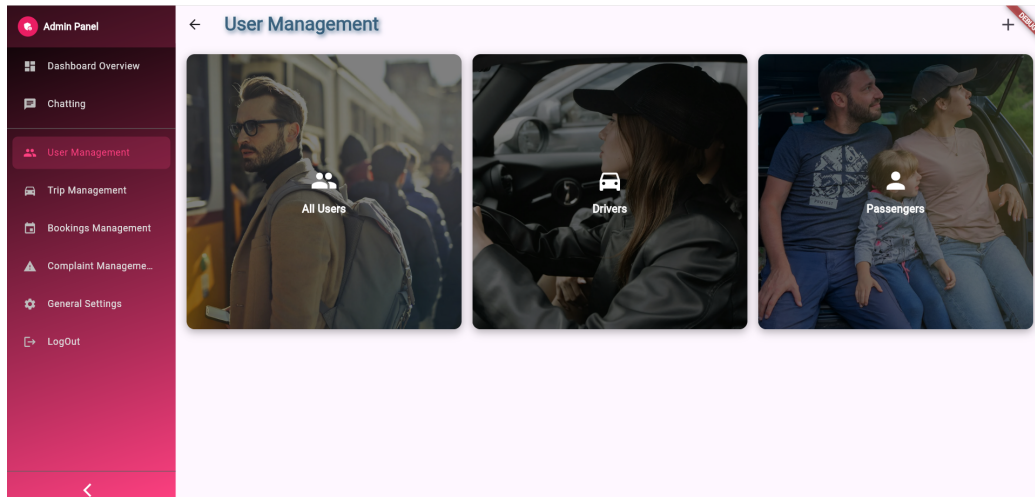


Figure 4.71: Admin Dashboard View

When the admin enters the user, all users appear to him, and when he enters the passengers, all passengers and drivers appear to him as well. He has the authority to delete any user he wants or modify his information or communicate with him via chatting.

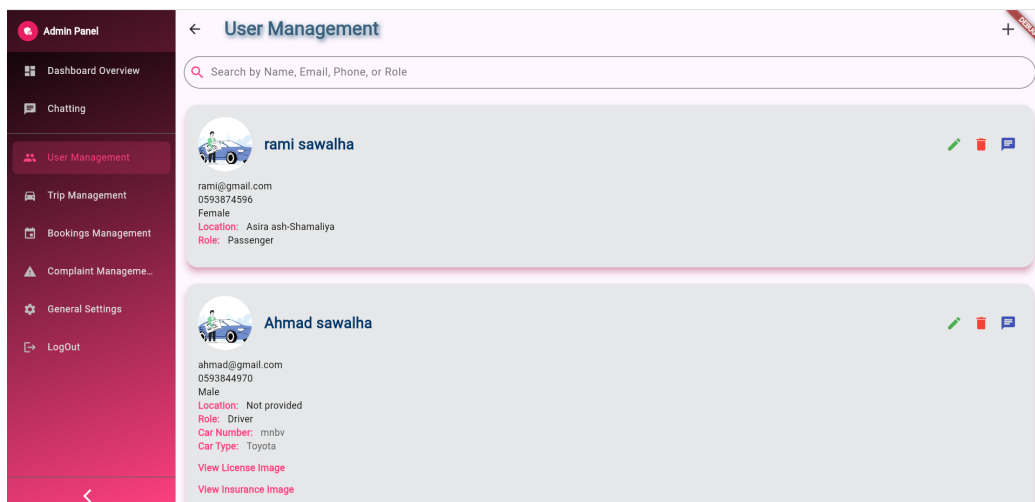


Figure 4.72: User View

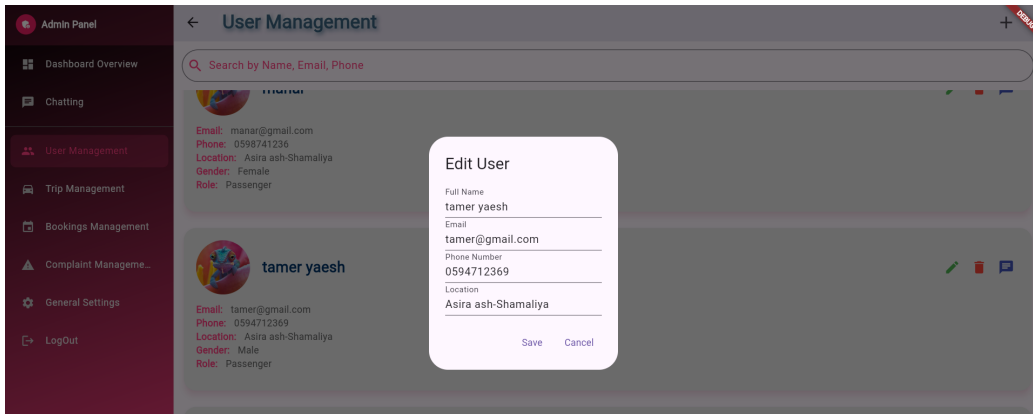


Figure 4.73: Edit User

When he clicks on the add icon on the user management page, a page opens for him to add a new User to the application, so that he enters his official information such as his name, mobile number, email, and the type of this user, whether he is a driver or a passenger.

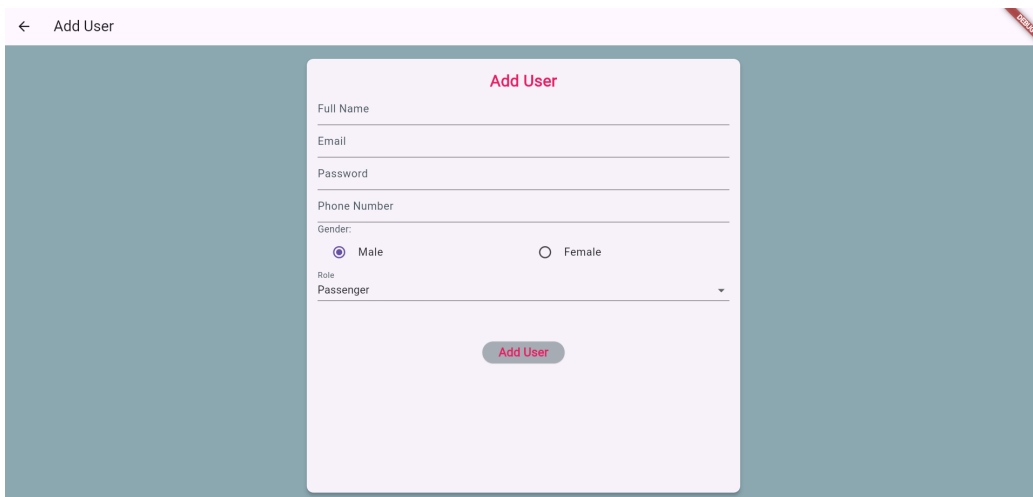


Figure 4.74: Add User

Trip Management

As for trip management, the admin can view all trips and know the details of each trip and whether it is completed or upcoming. If it is upcoming, he can modify or delete any trip he wants and communicate with the driver for this trip. He can also view the passengers registered for this trip.

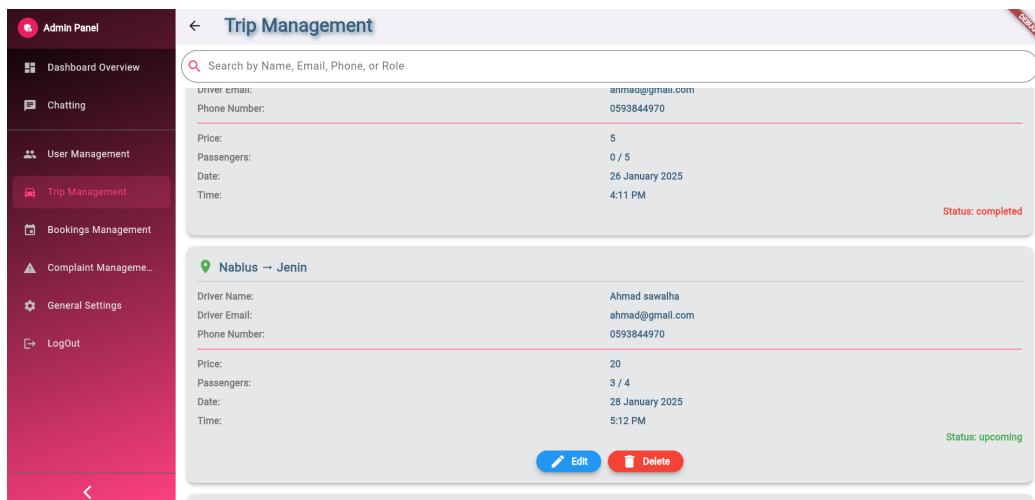


Figure 4.75: Trip Management

Booking Management

As for the reservations management, it shows him all the reservations and the details of each reservation, including the name of the passenger, the driver, the trip, and everything else.

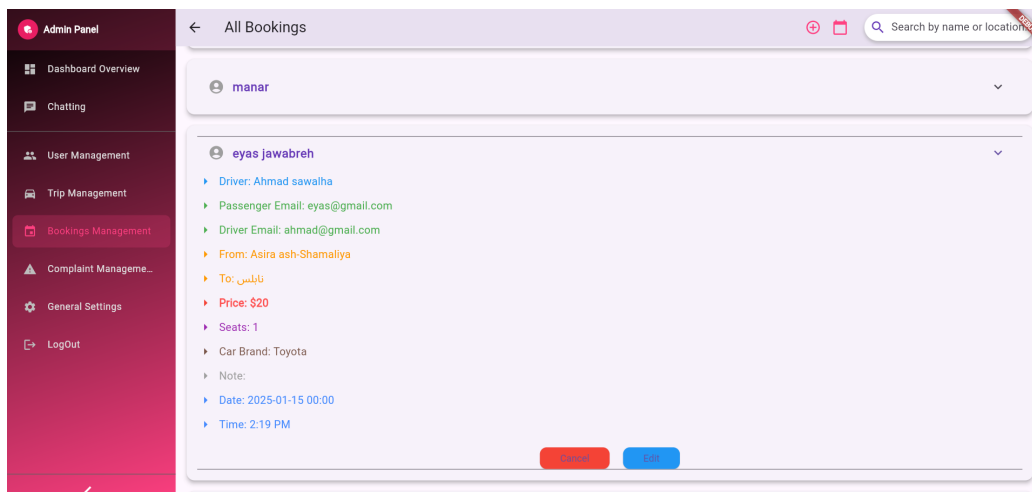


Figure 4.76: Booking Management

The admin can also add any reservation he wants if someone contacts him. He can add the reservation by adding the passenger's information from name, email and mobile number, and then writing the driver's email. A list of all the trips for this driver will appear. When choosing a specific trip, the rest of its information is filled in and then the reservation is confirmed.

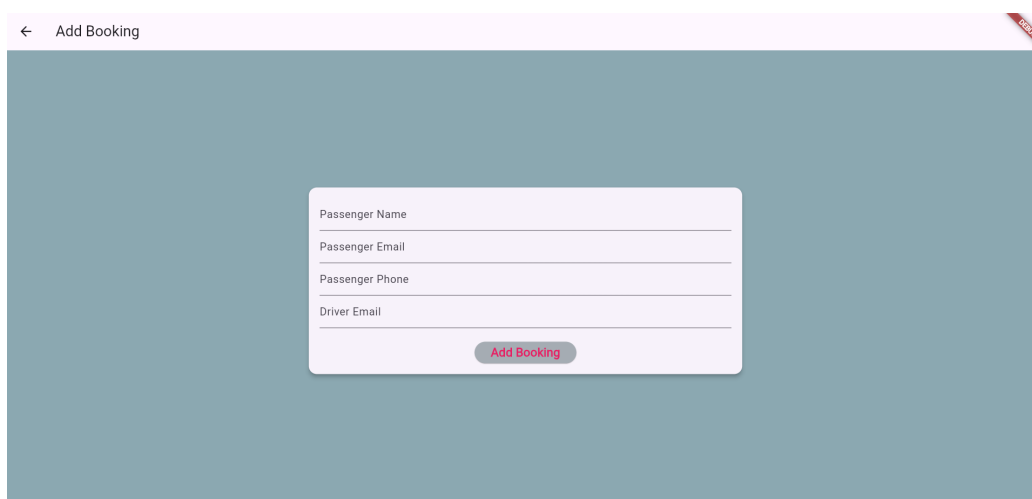


Figure 4.77: add Booking

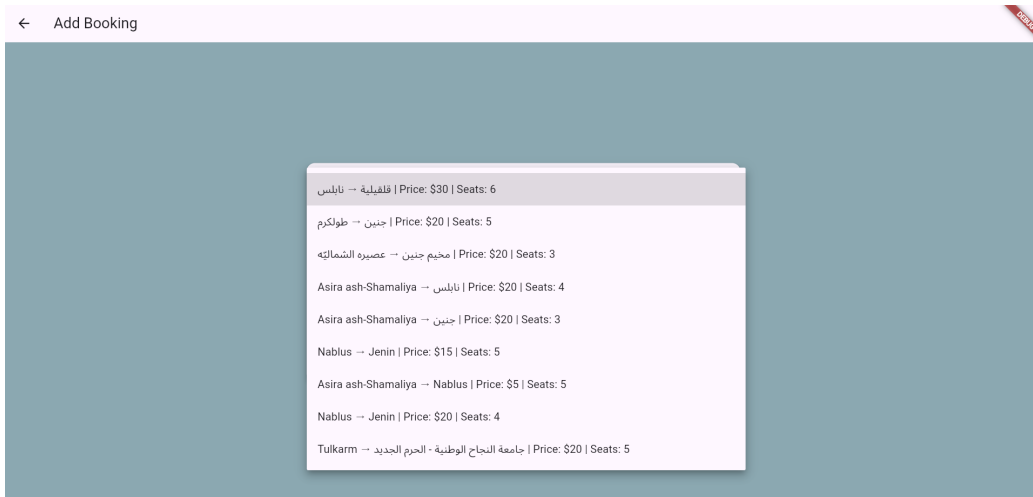


Figure 4.78: Select Trip



Figure 4.79: Add

General Permission

Here in this he has general permissions such as adding a new admin or modifying and deleting a specific admin.

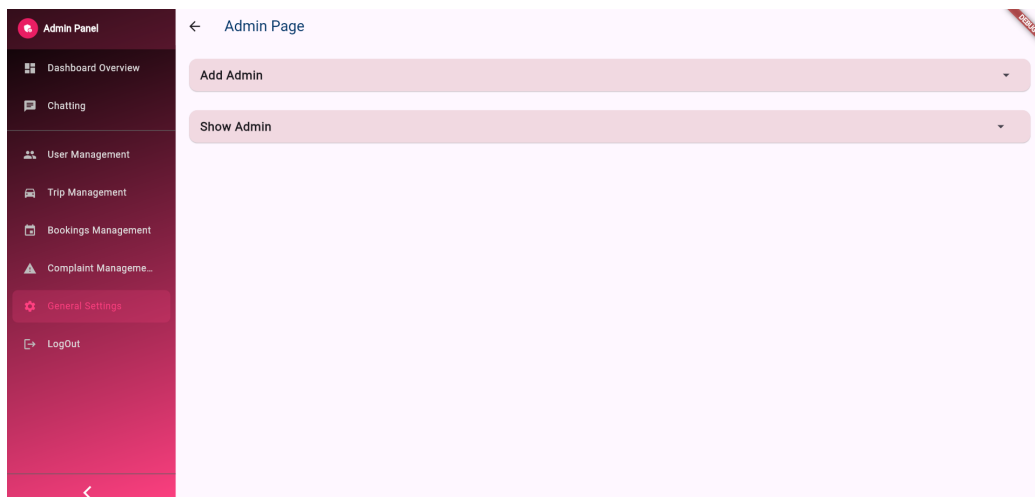


Figure 4.80: General Setting

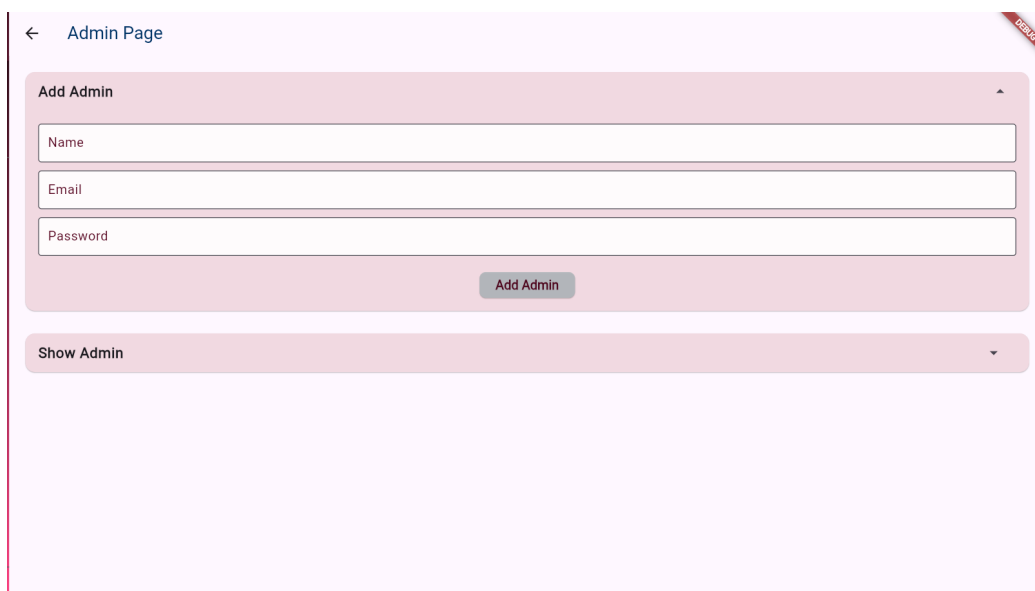


Figure 4.81: Add Admin

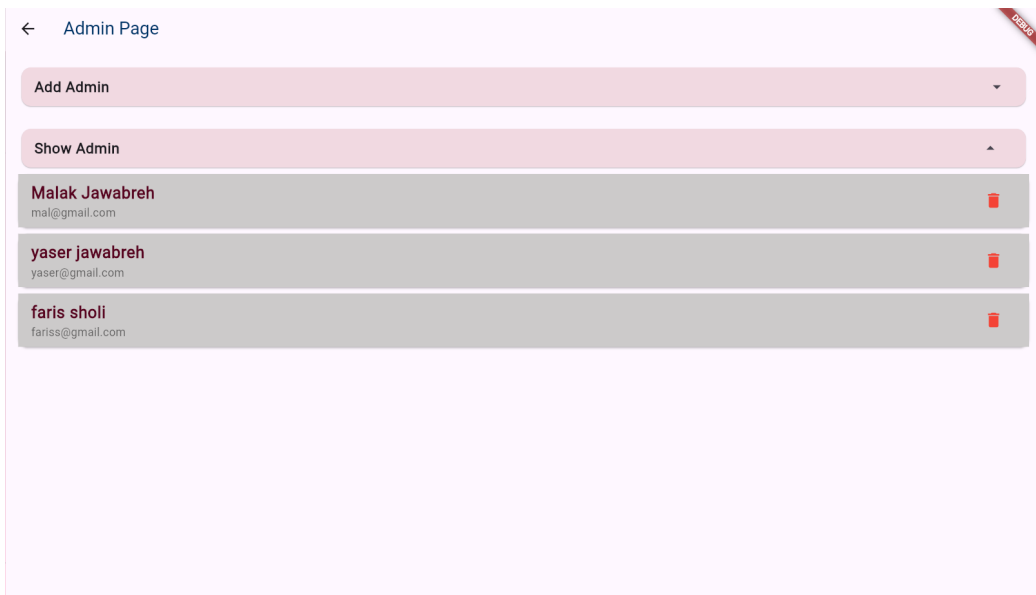


Figure 4.82: Show Admin And Delete

4.5 Website

All the features in the mobile application are available on our application's website, except for the details that are not useful for a website, such as contacts or obtaining the current website. I want to display some web pages that work on the same principle as the mobile.



Figure 4.83: first webpage

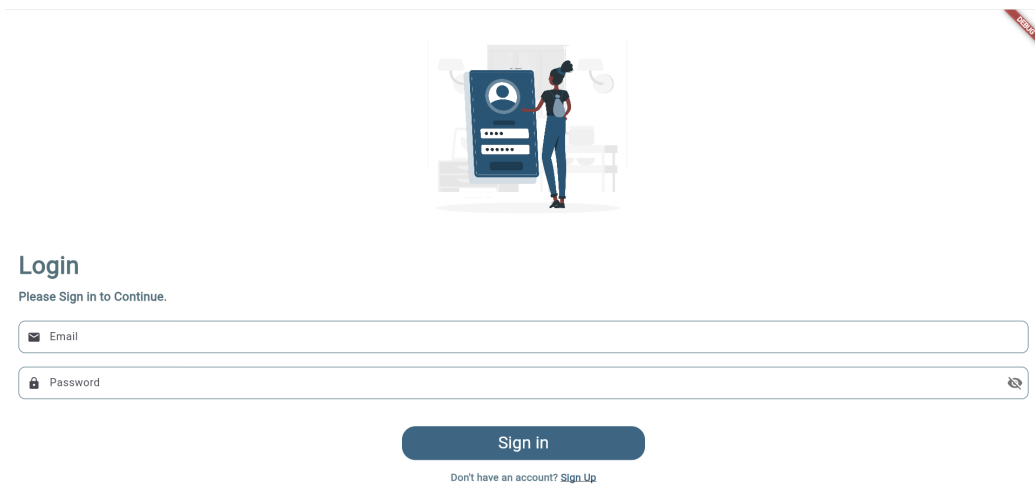


Figure 4.84: login webpage

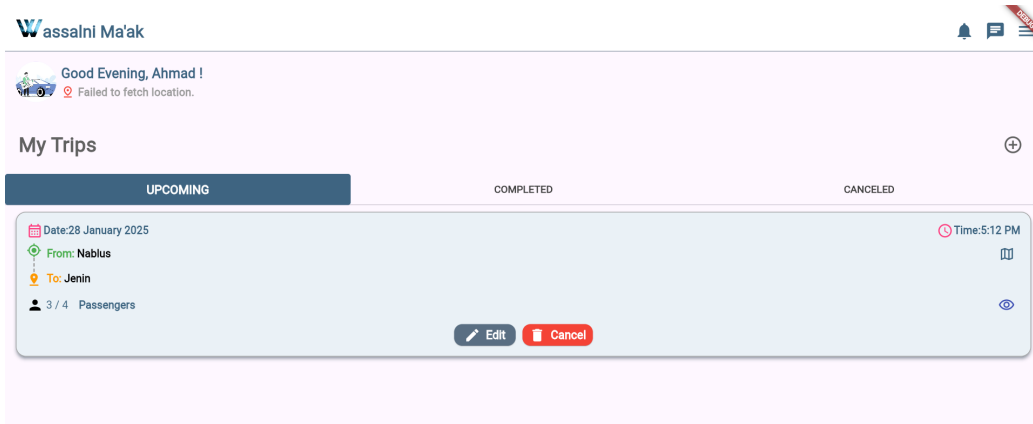


Figure 4.85: Driver Dashboard webpage 1

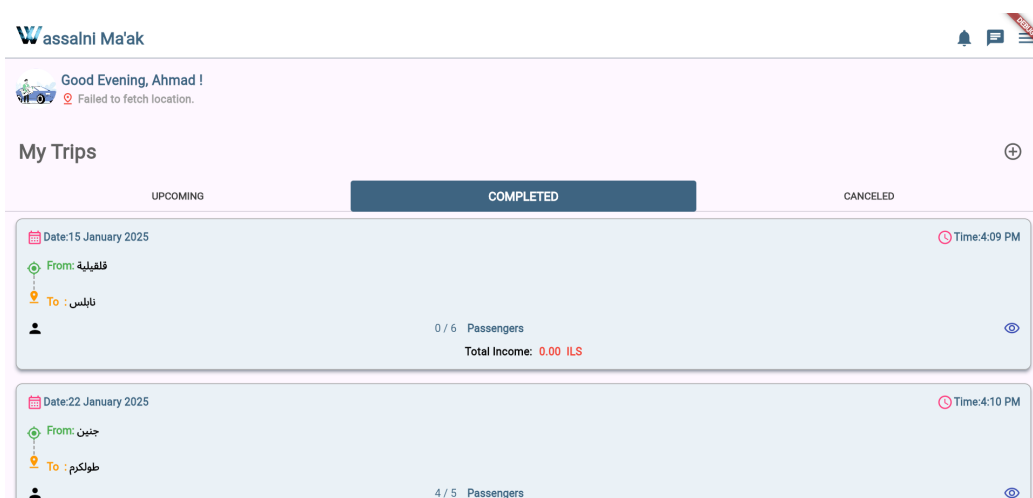


Figure 4.86: Driver Dashboard webpage 2

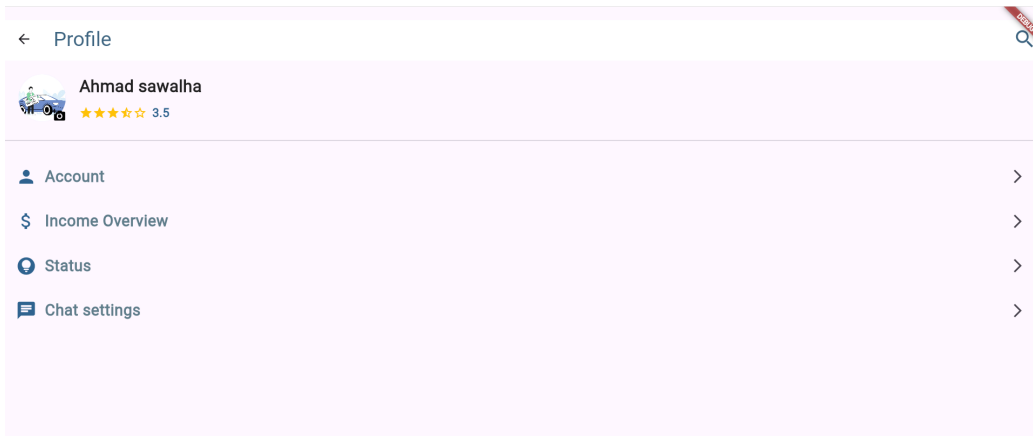


Figure 4.87: Driver Profile

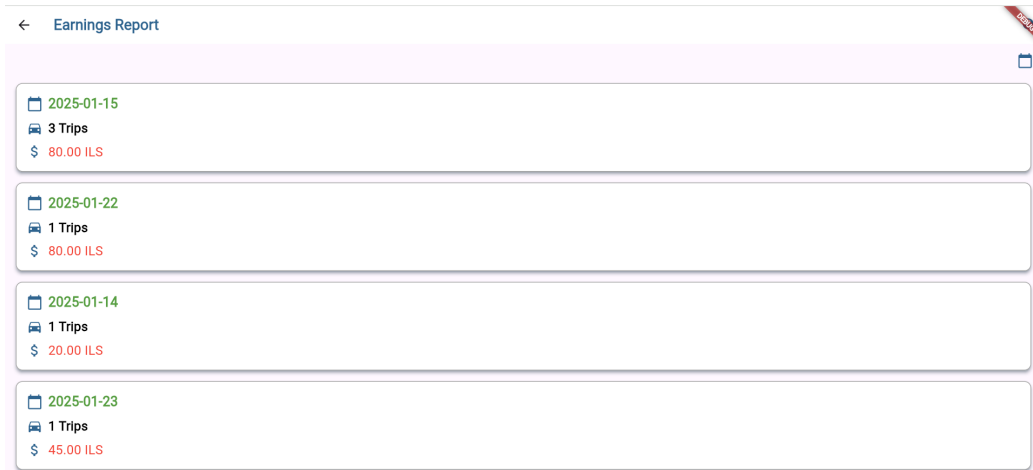


Figure 4.88: Income OverView

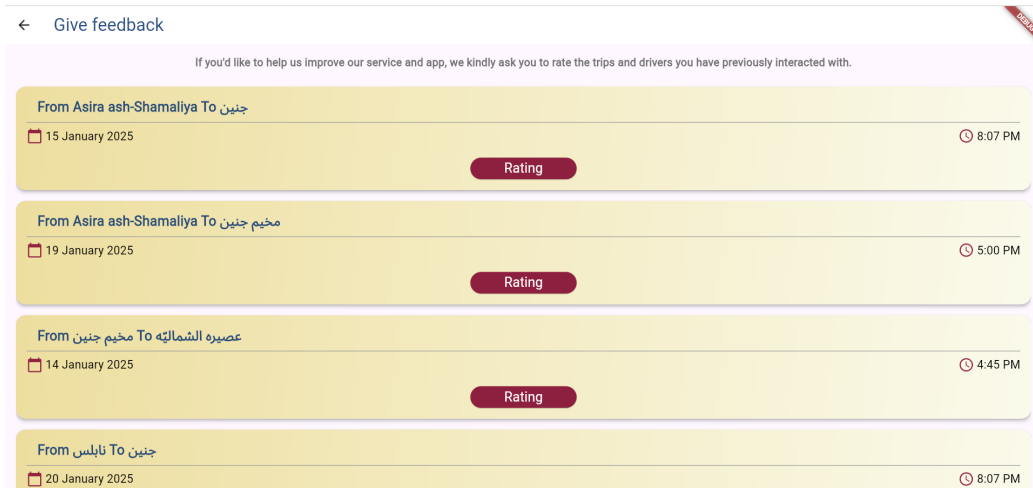


Figure 4.89: Passenger Feedback Trip

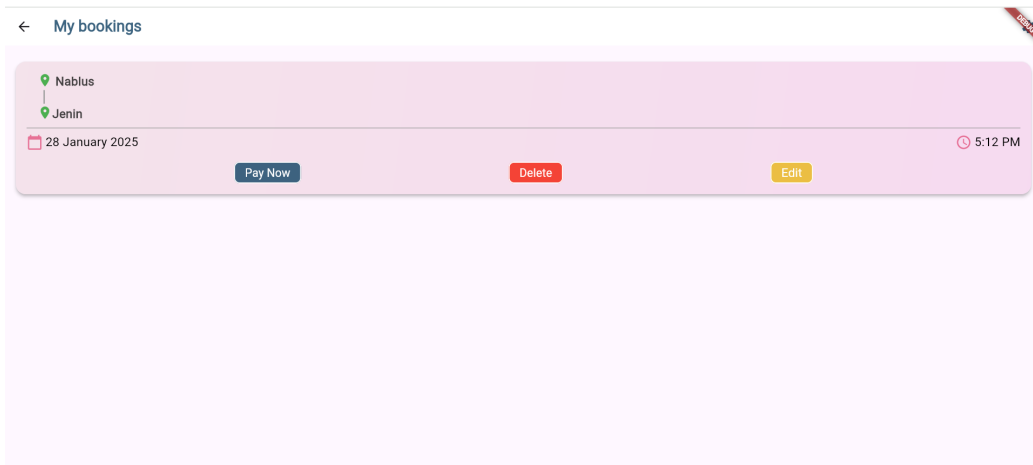


Figure 4.90: Passenger Active Booking

Chapter 5

Result And Discussion

Thanks to God, we have succeeded in developing our application "Wasalni Ma'ak", which is a platform to facilitate the movement of citizens and a job opportunity for drivers.

Users can use the "Wasalni Ma'ak" application either through the mobile application or the web, which gives users the ability to benefit from effective and easy-to-use features, with a focus on making the design easy to use, simple and clear to ensure ease of use.

With "Wasalni Ma'ak", you do not have to worry about choosing a trip to go to the place you want at any time you want and with complete ease and flexibility for reservations with the ability to give any comments or ratings to drivers in the application, and provide job opportunities for drivers so that they can get their income by creating trips at any time they want and under any circumstances. The application facilitates planning travel and accessing specific information and benefiting from the recommendations of others. In addition, the presence of an administrator in the "Wasalni Ma'ak" application helps in continuous follow-up.

Chapter 6

Conclusion and Recommendation

6.1 SUMMARY

The "Wasalni Ma'ak" application was developed and designed to save time and effort in transportation at any time and place, by providing passengers with the trips available from each driver, the time of each trip, and other details to enable them to book this trip.

The development process included solving many limitations, including coding errors, designing a suitable database, saving information in the best way without overloading the database, and dealing with map features.

The "Wasalni Ma'ak" application provides a solution to a problem imposed on us, using a mobile or web application.

6.2 RECOMMENDATION

- 1. Make a comprehensive plan before starting the project and determine the architecture with which you will build the project to suit the application, and plan significantly how the database will be built.*
- 2. Choose the technologies you want to use in the project based on what is appropriate for the project sent.*
- 3. Make sure to test each part separately first, and combine the parts in an integrated manner.*

6.3 WHAT WE HAVE LEARNED

- 1. We learned how to use Flutter and Node-js to build a complete application.*

- 2. Correct use of MongoDB database*
- 3. We learned how to send notifications and chat using Firebase.*

6.4 FUTURE WORK

- 1. To develop our application to support helping people search for any services they want and meet their needs easily and quickly.*
- 2. To add the ability to track the car when it sets off to the intended destination.*
- 3. To add the feature of tracking road conditions so that the driver and passenger can see the status of barriers and take the best route.*
- 4. Organize trips through the application to tourist places.*
- 5. Employ artificial intelligence to determine the best path according to traffic conditions.*
- 6. Use features that help the driver on the road such as Google Map or Waze.*
- 7. Provide offers for distinguished passengers.*
- 8. Cooperate with car rental companies from within the application.*

Chapter 7

References

- *Open Sreet Map*: <https://www.openstreetmap.org>
- *Firestore Documentation*: <https://firebase.google.com/docs>
- *Flutter*: <https://docs.flutter.dev>
- *Overview - GitHub Docs*: docs.github.com