



Al-Najah National University

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

Computer Engineering Department

Software Graduation Project

TaxiGo



Students:

Adham Yaqoub 12028277

Yazan Edaily 12011512

Supervised by:

Dr. Suleiman Abu Kharmah

Dr. Sufyan Samara

Acknowledgements

First and foremost, we extend our deepest gratitude to Almighty God, whose divine guidance, strength, and mercy have been our source of perseverance throughout this journey. Without His blessings, this achievement would not have been possible.

To our beloved families, especially our parents — thank you for your endless love, sacrifices, and unwavering belief in us. Your encouragement was the light that led us through every obstacle.

With heartfelt admiration, we offer special thanks to Dr. Sulaiman Abu Khurma and Dr. Sufian Samara, two pillars of inspiration and excellence within our academic journey. Your profound knowledge, passionate teaching, and unwavering dedication left an indelible mark on us. You were not just educators, but mentors who guided us toward excellence and critical thinking — and for that, we are forever grateful.

To all the professors at the Computer Engineering Department at An-Najah National University, thank you for empowering us with knowledge and values that go far beyond the classroom. Your dedication has shaped the engineers we have become.

Last but never least, to our dear friends and classmates: your support, shared struggles, and joyful moments made these five years unforgettable. This journey was never walked alone — you made it memorable, meaningful, and worth every challenge.

Together, we reached the summit. Thank you.

Disclaimer Statement

This report was written by students at the Yazan Edaily & Adham Yaqoub Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Table of Contents

Table of Contents	4
List of Figures	7
1 Chapter 1	10
1.1 Introduction	10
1.1.1 Problem	10
1.1.2 Objective	10
1.1.3 Scope of the Work	11
1.1.4 Importance	11
1.1.5 Report Organization	11
2 Chapter 2	12
2.1 Theoretical Background and Previous Work	12
2.1.1 Practical Challenges and Limitations	12
2.1.2 Adopted Development Standards	12
2.1.2.1 MVC Architecture	12
3 Chapter 3	14
3.1 Literature review	14
4 Chapter 4	15
4.1.1 Methodology	15
4.1.2 Tools, Programming Languages, APIs Technologies	15
4.1.3 client Side (User & Driver Interfaces):	15
4.1.4 Website Side for admin:	16
4.1.5 Server Side:	16
4.1.6 IDEs and Code Editors	17
4.1.7 Database Structure	18
4.1.8 Database - NoSQL:	18
4.1.9 Features of The Application	19
4.1.10 Implementation	19
4.1.11 Mobile Application	19
1. Welcome Screens:	19
2. Home Page – Visitor Interface:	20
3. Side Menu – Dashboard Overview:	20
5. Trip Price Estimation Feature:	21
6. View Taxi Offices on the Map:	22
7. Map Page:	23
8. About Page:	24

.9	User Registration Page:	25
10.	driver Registration Page:	26
11. Sign in:	
	27
12.	Forget Password:	28
.1 User:	
	30
	User Home Page Functionality:	30
	Scheduled Trips Page:	32
	Scheduled Trip Booking via Telegram Bot:	33
.2	Trip History Page:	34
.3	Driver List Page:	35
4.	Settings Page:	36
	• Change Theme & Language:	36
	• Change Password:	36
	• Edit Profile:	36
5.	Support Page:	38
6.	Notifications:	39
.1	Driver Page:	40
.2	Trip Requests page:	41
3.	My Trips:	43
4.	Earnings Page:	44
5.	Settings Page (Driver Panel):	45
	• Change Theme and Language:	45
	• Change Password:	45
	• Edit Profile:	45
	• Logout:	45
	Chat Page (Driver Panel):	46
1.	Manager Panel – Home Page Overview:	47
.2	Driver List Page – Manager Panel:	48
	View Driver Details:	50
	Add New Driver:	50
	Chatting:	51
3.	Settings Page – Manager Panel:	52
	• Theme Mode (Night Mode):	52
	• Language Selection:	52

•	Change Password:	52
•	Logout:	52
1.	Admin Page:	53
2.	Drivers Page:	54
3.	Users Page:	55
4.	Trip Management:	56
5.	Settings page:	57
6.	Taxi Offices page:	58
4.1.12	Website:	60
4.1.13	Database and API Server	85
5	Chapter 5	86
5.1	Result and Analysis	86
6	Chapter 6	87
6.1	Discussion	87
6.1.1	Learning Outcomes	87
6.1.2	Key Challenges	87
7	Chapter 7	88
7.1	Conclusion and Recommendation	88
7.1.1	Summary 88	
7.1.2	Things We Learned	88
7.1.3	Recommendations	88
7.1.4	Future Work	89
8	Chapter 8	90
8.1	References	90

List of Figures

Figure1 Welcome	19
Figure2 Home Page	20
Figure3 Trip Price Estimation	21
Figure 4 View Taxi Offices.....	22
Figure5 Map Page	23
Figure6 Roads Page	24
Figure 7 About Page	24
Figure 8 User Registration Page	25
Figure 9 Driver Registration Page	26
Figure 10 Sign in	27
Figure 11 Forget Password	28
Figure 12 User	31
Figure13 Scheduled Trips Page	32
Figure 14 Scheduled Trip Booking via Telegram Bot	33
Figure 15 Trip History Page.....	34
Figure 16 Driver List Page.....	35
Figure 17 Settings Page.....	37
Figure 18 Support Page.....	38
Figure 19 Notifications	39
Figure 20 Driver Page	40
Figure21 Trip Requests page.....	41
Figure 22 Accepted Trips.....	42
Figure 23 History Trip	43
Figure 24 Earnings Page	44
Figure 25 Settings Page.....	45
Figure 26 Chat Page	46
Figure 27 Manager Panel – Home Page Overview	47
Figure 28 Driver List Page – Manager Panel.....	48
Figure29 View Driver Details	50
Figure30 Add new Driver	50
Figure 31 Chatting.....	51
Figure 32 Settings Page – Manager Panel.....	52
Figure 33 Admin Page.....	53
Figure 34 Drivers Page in Admin	54
Figure 35 Users Page in Admin	55
Figure 36 Trip Management	56
Figure 37 Settings page.....	57
Figure 38 Taxi Offices page	58
Figure 39 Home Page in Web.....	60
Figure 40 Notification in web.....	68
Figure 41 Trip in Web for Driver	74
Figure 42 Admin Page in Web.....	79
Figure 43 Trip Management in Web	81
Figure 44 Payments Table in Web	82
Figure 45 Taxi Offices in Web.....	83

Abstract

This project presents TaxiGo, a comprehensive and intelligent taxi booking and management system developed as a cross-platform application using Flutter for both mobile and web interfaces. The backend is built with Node.js (Express), hosted on Render, and connected to a MongoDB database. Cloudinary is used for efficient and secure image/media management, while Firebase Cloud Messaging (FCM) delivers real-time push notifications.

TaxiGo adopts a role-based access control system supporting four user types: Users (Passengers), Drivers, Managers, and Admins, each with tailored dashboards and permissions.

Users interact with a dynamic map interface to estimate trip fares based on selected start and end locations, date, and time. Additional features include multilingual support (Arabic/English), light/dark themes, and a secure registration process with email verification and password strength validation. Once registered, users can book immediate or scheduled rides, track trips in real time, and receive push notifications at each ride stage — request, approval, start, and completion. Emergency buttons for police, fire, and ambulance add a layer of safety.

TaxiGo also supports scheduled trips, where users can define a future date and time for the ride. Once the scheduled time arrives, the system automatically activates and sends the trip request to nearby drivers.

Additionally, a Telegram bot integration allows users to book trips directly through Telegram. After a one-time login, the system saves the user's session for future requests, providing a lightweight and convenient alternative to the mobile app.

To enhance situational awareness, the "Roads" page was developed using Gemini AI, which analyzes real-time traffic messages from a public Telegram channel. This AI-powered feature offers dynamic road condition updates across major Palestinian cities, helping users plan safer and more efficient routes.

Drivers receive categorized ride requests, follow live navigation, and communicate with managers via real-time WebSocket chat. An automated rating system evaluates driver performance based on responsiveness and punctuality. Drivers can view their earnings, trip history, and customize settings.

Managers oversee their assigned drivers through a dedicated dashboard showing real-time summaries of trips, activity status, and earnings. They can add or deactivate drivers and initiate real-time conversations for smoother coordination.

Admins have full access and control. They manage all users, trips, financials, and can register new taxi offices via an interactive map. Admins assign managers, send credentials via email, and monitor system-wide analytics, including trip volume, user activity, and revenue.

The web version maintains all mobile functionalities with optimized interfaces for desktop screens, such as using data tables instead of cards for managing users and trips.

With its rich feature set — including real-time technologies, AI-driven road analysis, Telegram-based trip booking, scheduled ride automation, and secure role-based control — TaxiGo delivers a robust, scalable, and user-centric platform tailored for modern urban transportation in Palestine.

1 Chapter 1

1.1 Introduction

1.1.1 Problem

In Palestine, the taxi and transportation sector face numerous challenges that impact both service providers and users. Traditional taxi systems lack digital features such as real-time trip tracking, efficient trip management, and instant communication between users, drivers, and managers. Users often experience difficulty in ordering rides, tracking trip status, or estimating trip costs. Meanwhile, drivers and managers struggle to manage ride requests, driver availability, and customer communication through a unified platform.

These issues are further complicated by the unique geographical and political situation in Palestine, where sudden roadblocks, military checkpoints, and closures due to occupation significantly affect mobility and travel planning. Commuters and drivers often lack timely access to information about road conditions and checkpoint status, which can lead to delays, cancellations, and security risks.

Additionally, there is no comprehensive digital solution that offers secure user management (e.g., email verification, password reset), real-time trip updates, intelligent driver assignment, and role-based access for administrators and managers. The absence of emergency support features and real-time communication channels limits the safety and responsiveness of existing services.

TaxiGo aims to solve these problems by introducing a smart, role-based taxi management and booking platform that integrates real-time tracking, instant notifications, and emergency response tools. Uniquely, it leverages local data — including Telegram-based alerts — to inform users and drivers of current road conditions, traffic disruptions, and checkpoint statuses. This feature gives TaxiGo a distinctive advantage in the Palestinian context by enhancing route safety and trip efficiency in a dynamic and often unpredictable environment.

1.1.2 Objective

The main objective of TaxiGo is to modernize and streamline the taxi service experience in Palestine by providing a real-time, role-based platform that connects users, drivers, managers, and administrators. The platform enables users to easily request rides, track their trips, and receive instant updates, while allowing drivers to manage and accept nearby trips efficiently. TaxiGo also supports platform management through powerful tools for admins and managers to oversee operations and driver activity.

In addition, TaxiGo addresses the unique challenges of transportation in Palestine by integrating live updates about road conditions and military checkpoints, helping users and drivers make safer travel decisions. The platform promotes safety through an emergency contact feature and encourages reliability through an automated driver

rating system — all within a user-friendly interface designed for both web and mobile use.

1.1.3 Scope of the Work

TaxiGo is a role-based taxi management and booking system designed for the Palestinian context. The project covers the development of core functionalities including user and driver registration with email verification, trip creation and tracking, real-time notifications, and emergency support. It also includes admin and manager control panels, a public homepage for visitors, and live road condition updates through Telegram bot integration. The system supports real-time chat and tracking using WebSocket technology and is accessible via both web and mobile platforms.

1.1.4 Importance

The TaxiGo application is important as it offers a modern, reliable platform that improves the transportation experience for all users in Palestine. It empowers passengers, drivers, and transportation managers by providing tools for real-time trip tracking, secure registration, emergency support, and efficient ride management. For users, TaxiGo ensures convenience, safety, and transparency throughout the ride process. For drivers and managers, it simplifies communication, trip handling, and driver oversight.

Moreover, the platform addresses unique local challenges such as road closures and military checkpoints by integrating live road condition updates through Telegram bot data, helping users and drivers make safer travel decisions. TaxiGo not only enhances daily transportation but also contributes to building a smarter, more responsive, and community-centered mobility system.

1.1.5 Report Organization

This report is structured in a logical sequence to reflect the development process of the TaxiGo project. It begins with Chapter 1, which introduces the project by outlining the core problem, objectives, scope of work, and the importance of building a smart taxi management system. Chapter 2 covers the theoretical background and previous work, in addition to highlighting the main practical constraints and the adopted development standards, such as the MVC architecture. Following that, Chapter 3 provides a literature review that explores similar systems and relevant research. Chapter 4 represents the core of the report, explaining the methodology used, the tools and technologies adopted (such as Flutter, Node.js, and MongoDB), and presents the full implementation of the application across all roles: users, drivers, managers, and admins. This chapter also details the mobile and web interfaces, including UI features, user flows, and server-side logic. Chapter 5 presents the results and performance analysis of the application based on testing and feedback. Chapter 6 offers a conclusion that summarizes the project outcomes, outlines the key lessons learned, and provides recommendations for future improvements. Finally, Chapter 7 includes all references used during the research and development phases of the project.

2 Chapter 2

2.1 Theoretical Background and Previous Work

2.1.1 Practical Challenges and Limitations

Throughout the development of the TaxiGo project, we encountered several significant challenges:

1. **Time Constraints and Learning Curve:** Since this was our first experience developing a full mobile and web-based application, we had to learn new technologies from scratch. We worked with unfamiliar tools such as the Dart programming language and the Flutter framework for mobile development, as well as Node.js and MongoDB for backend services. Additionally, we integrated Firebase for real-time features like chat and notifications. Balancing this steep learning curve with the demands of other university courses and projects made time management a major challenge. Despite these obstacles, we invested additional effort to understand the technologies and build the app as close to our vision as possible.
2. **Internet and Hosting Issues:** The application relies on a stable internet connection to interact with the backend server and APIs. During development and testing, we used Render for hosting the backend, but we faced performance issues such as slow response times and occasional downtimes, which delayed our progress and required troubleshooting.
3. **Mapping Services Limitations:** Initially, we planned to use Google Maps due to its reliability and rich features. However, the service's high cost made it unfeasible for a student project. As a result, we switched to OpenStreetMap and manually implemented essential features such as route drawing, office locations, and live tracking. This transition required considerable development time and effort, as we had to build many functionalities from scratch.

These challenges not only tested our technical and problem-solving abilities but also strengthened our collaboration and adaptability as a team.

2.1.2 Adopted Development Standards

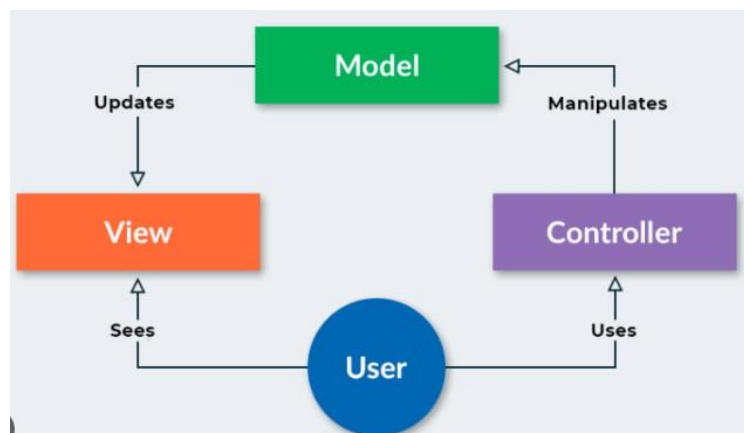
2.1.2.1 *MVC Architecture*

Our system follows the MVC architectural pattern, extended with a Router component to better organize the application's flow and scalability. This structure helped us separate concerns, manage complexity, and speed up the development process. The main components are:

- ❖ **Model:** Represents the data layer, primarily managed through MongoDB. It handles all data operations, including storing and retrieving user, driver, trip, and admin information. The model responds to requests from both the controller and the view to ensure data consistency.

- ❖ **View:** Refers to the user interface (UI) for both mobile and web platforms, developed using Flutter. It allows users, drivers, managers, and admins to interact with the system visually and perform actions such as booking trips, managing accounts, and tracking rides.
- ❖ **Controller:** The logic layer implemented using Node.js. It manages the application's core functionality, processes client requests, applies business logic, and communicates between the view and the model.

This architecture helped maintain a clean separation of concerns and allowed us to develop and test each component independently, leading to a more maintainable and scalable codebase.



3 Chapter 3

3.1 Literature review

The development of the TaxiGo system relies on a combination of modern technologies and methodologies drawn from various open-source tools, commercial platforms, and real-time communication standards. This review highlights the relevant literature and technical resources that influenced the architecture and feature design of the project.

To manage source code efficiently and ensure collaboration among developers, GitHub was used as the primary version control and code hosting platform [1]. For the frontend development, Flutter provided a powerful cross-platform framework that supports building responsive user interfaces for both mobile and web applications [2]. This choice was influenced by Flutter's flexibility in UI design and its ability to maintain a single codebase across multiple platforms.

One of the inspirations for the system was Uber, which offers a mature model for ride-sharing platforms. Its documentation and public insights helped inform the structural and interaction design of the user and driver flows in the TaxiGo app [3]. For geospatial services, OpenStreetMap [4] was selected as a free and open-source alternative to commercial mapping solutions. Additionally, the OpenRouteService API provided advanced routing and geocoding capabilities necessary for dynamic trip management and driver tracking [5].

Although Google Maps was reviewed for potential use due to its comprehensive APIs [6], it was ultimately excluded due to licensing and cost considerations. Instead, the application adopted Cloudinary for managing image uploads and driver profile pictures, offering reliable cloud-based asset handling [7]. Hosting and backend deployment were managed through Render, which streamlined the CI/CD process and supported Node.js environments seamlessly [8].

Firebase played a crucial role in real-time data synchronization and notification delivery [9]. It was especially useful for managing push notifications to users and drivers regarding trip updates. On the backend, Node.js was chosen for its non-blocking event-driven architecture, which suits high-concurrency environments like ride-sharing platforms [10]. The Socket.IO library further enhanced the platform by enabling real-time chat functionality between drivers and their assigned managers [11].

Together, these technologies and platforms provided the technical foundation for building a scalable, reliable, and feature-rich taxi booking system that meets modern user expectations for speed, interactivity, and reliability.

4 Chapter 4

4.1.1 Methodology

4.1.2 Tools, Programming Languages, APIs Technologies

In addition to the fundamental and indispensable components of creating websites and mobile applications, such as frameworks, programming languages, and tools, we have a plethora of possibilities for application creation after much reading and research. All of these specifics will be mentioned and discussed in more detail. We choose to construct them using the subsequent framework.

4.1.3 client Side (User & Driver Interfaces):

1. **Design:** To ensure a smooth and intuitive user experience, we designed the TaxiGo mobile application to be simple, user-friendly, and visually appealing. The interface allows users to easily navigate between features such as trip booking, real-time tracking, emergency calls, and receiving notifications. We chose a color scheme that is both professional and comfortable on the eyes. Before finalizing our design, we studied the interfaces of popular transportation apps such as Uber, inDrive, and apps on Google Play and the App Store. We drew inspiration from these references and adapted the UI to fit our specific goals and user base in Palestine. Special attention was given to accessibility, local context (e.g., road conditions and checkpoints), and responsive layouts to support both Android and iOS platforms using Flutter.
2. **Frameworks:** We choose the flutter framework to build our application which Flutter is a Google open-source user interface 9 toolkit that allows you to create mobile (iOS and Android), web, and desktop apps (Linux, Mac, and Windows) from a single code base. By creating an application in this language, consumers can be given a refined, rich, and immersive user experience. The single code rule states that you write the code once and run it everywhere! This way, you can plan and target an app release across all platforms at the same time. Thus, by quickly releasing the application to the market, time and money can be saved.

Top 10 Motivators for Flutter App Development Teams to Consider:

- (a) Simple to learn.
- (b) Excellent Performance.
- (c) Reusable Devices.
- (d) Quick time to market.
- (e) Quick reload.
- (f) Perfect for creating an MVP.
- (g) Capability to produce excellent design.
- (h) Reduced development expenses.

- (i) Excellent community for documentation.
- (j) Excellent work.

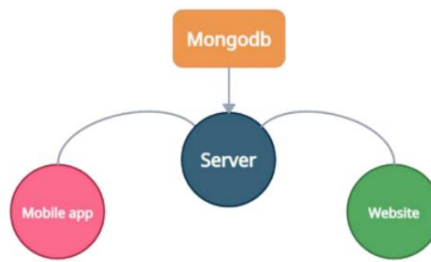
3. **Programming languages:** Flutter makes use of Google created the programming language Dart with online and mobile application developers in mind. It is intended for the development of iOS, Android, and web applications. Being compatible with all cutting-edge online browsers, mobile devices, and even web servers is one of the language's objectives. It goes without saying that we will utilize Dart as we will be utilizing the Flutter framework. There are numerous parallels between Dart and other programming languages including Java, C, Swift, and Kotlin. Since Java is the most familiar language we know, we found that Dart is very comparable to it. This makes learning a new language easier and more straightforward.

4.1.4 Website Side for admin:

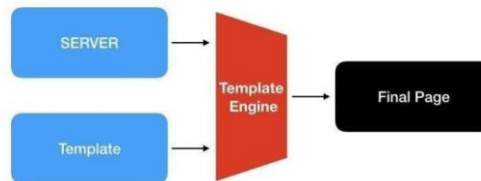
1. **Design:** We developed a dedicated website interface specifically for the administrator role. This platform allows the admin to perform full system management functions, including creating manager accounts and sending login credentials via email, monitoring driver activity, managing trip records, approving or rejecting driver registrations, and controlling access permissions. The website was designed with clarity and efficiency in mind to ensure ease of use and full administrative oversight.
2. **Programming Languages and Frameworks:** The website was built using Flutter, a UI toolkit developed by Google, in combination with the Dart programming language. Dart was chosen for its compatibility with Flutter and its similarity to familiar languages such as Java, which made it easier for the team to learn and adopt. The web application supports modern browsers and is fully responsive, allowing the admin to manage the system effectively across various devices. The backend logic and database connectivity are handled using Node.js and MongoDB, enabling real-time operations, secure role-based access control, and fast data processing.

4.1.5 Server Side:

1. **Frameworks:** On the server side of the TaxiGo system, we used Node.js, a fast, scalable, and cross-platform runtime environment designed for building high-performance server-side and networking applications. Node.js operates on a single-threaded, event-driven, non-blocking I/O architecture, powered by the V8 JavaScript engine, making it ideal for real-time applications such as ride tracking, live chat, and notifications



To build our API, we used the Express.js framework, which simplifies the routing process and allows for efficient server-side rendering and handling of HTTP requests. Express.js supports modular routing and middleware integration, which helped us structure the application logic cleanly and efficiently.



For real-time communication, especially the in-app chat feature between drivers, and managers, we integrated Firebase Realtime Database. This cloud-hosted NoSQL database allows seamless synchronization of JSON-based data across connected clients, enabling real-time updates and message delivery across platforms. Also, track the driver's trip path in real time.

2. **Programming Languages:** The backend of our system is primarily written in JavaScript, leveraging its compatibility with Node.js and various supporting libraries. JavaScript's flexibility allows us to build both frontend and backend components effectively. Its asynchronous capabilities and support for real-time features make it an ideal choice for our needs, especially in functionalities like live ride tracking, status notifications, and user management.

4.1.6 IDEs and Code Editors

To develop the TaxiGo application, we relied primarily on Visual Studio Code (VS Code), a free, cross-platform code editor developed by Microsoft. VS Code provided us with powerful features such as syntax highlighting, integrated terminal, Git support, and extensions for working with Dart, JavaScript, Node.js, and MongoDB. These tools significantly enhanced our development workflow across both mobile and web platforms.

For branding, we used Logo Maker to design a professional logo that represents the identity of the TaxiGo platform.

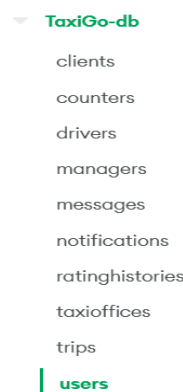
- **Version Control:** Given that we worked as a team, version control was essential to collaborate efficiently. We used GitHub, this allowed us to manage code versions, track issues, and merge features smoothly. We also used Google Drive occasionally to share documentation and media assets.
- **API Endpoints Testing:** Since our application depends heavily on RESTful APIs, we needed to ensure that each endpoint was working correctly before integrating it with the client side. For this purpose, we used Postman, a robust tool for testing HTTP requests such as GET, POST, PUT, PATCH, and DELETE. Postman helped us debug responses, monitor performance, and ensure reliability across all API functionalities, including login, ride creation, notifications, and real-time messaging.

4.1.7 Database Structure

In the majority of applications, the database is crucial, particularly for reading, editing, and removing data. It also gives the application purpose and allows for data backup. Since our application depends directly on the database, which shows data, we chose a NoSQL database since there were few relationships between various collections and it was simple to utilize the API.

4.1.8 Database- NoSQL:

Because it makes it simple for developers to create apps using documents, we decided to construct our schema without a relational database. Unlike XML, MongoDB stores documents in JSON format, which is based on JavaScript. As previously said, Node.js will be used to build the application's backend, and MongoDB is a great tool for Node.js application prototyping. Your data is kept as JSON objects in a MongoDB database. Unlike SQL databases, which involve merging various tables to create a document, most of the time, all of the data related to the object may be found in a single document. A NoSQL database application called MongoDB can handle documents that resemble JSON and have an optional schema.



4.1.9 Features of The Application

4.1.10 Implementation

In this section, we'll dive into the details of each part of the system. All the tools, methods and libraries used are discussed in detail.

4.1.11 Mobile Application

1. **Welcome Screens:** The initial experience of the TaxiGo app starts with a splash screen featuring a yellow background and a simple loading animation, providing a smooth entry into the app. This is followed by three onboarding screens designed to introduce users to the app's core features. The first screen displays the message "Anywhere You Are," emphasizing location flexibility, while the second shows "At Any time," highlighting 24/7 availability. The final screen presents "Book Your Car" with a "Get Started" button that directs users to the main application, where they can begin using the service

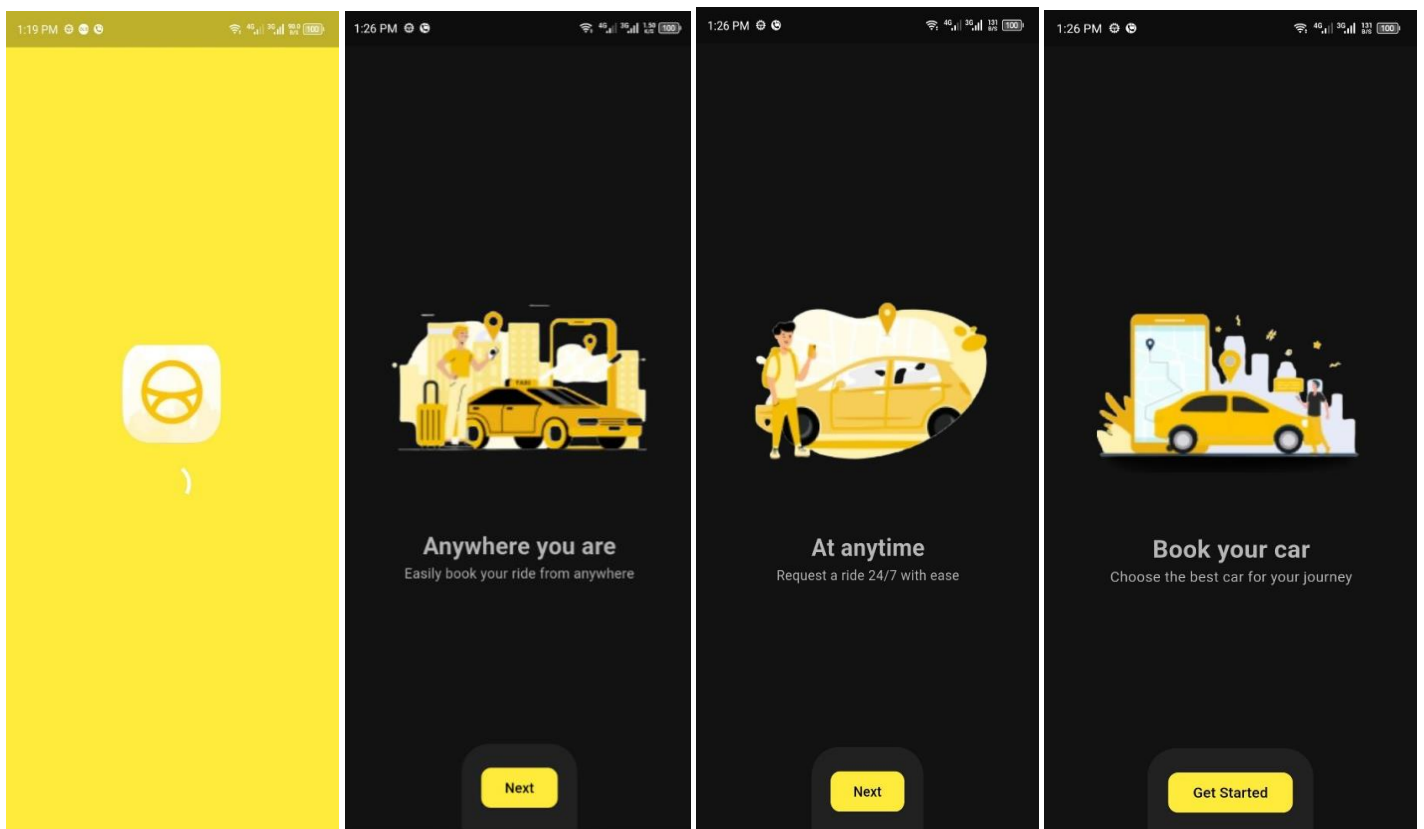


Figure 1 Welcome

2. **Home Page – Visitor Interface:** The home page of TaxiGo serves as a landing page for visitors. At the top, the AppBar includes a button to toggle the side dashboard menu. There is also a single button that dynamically navigates to either the Sign Up or Log In page, depending on the user’s choice. Additional buttons allow users to switch between Dark Mode and Light Mode as well as change the app language between Arabic and English.
3. **Side Menu – Dashboard Overview:** When the side dashboard is opened, it

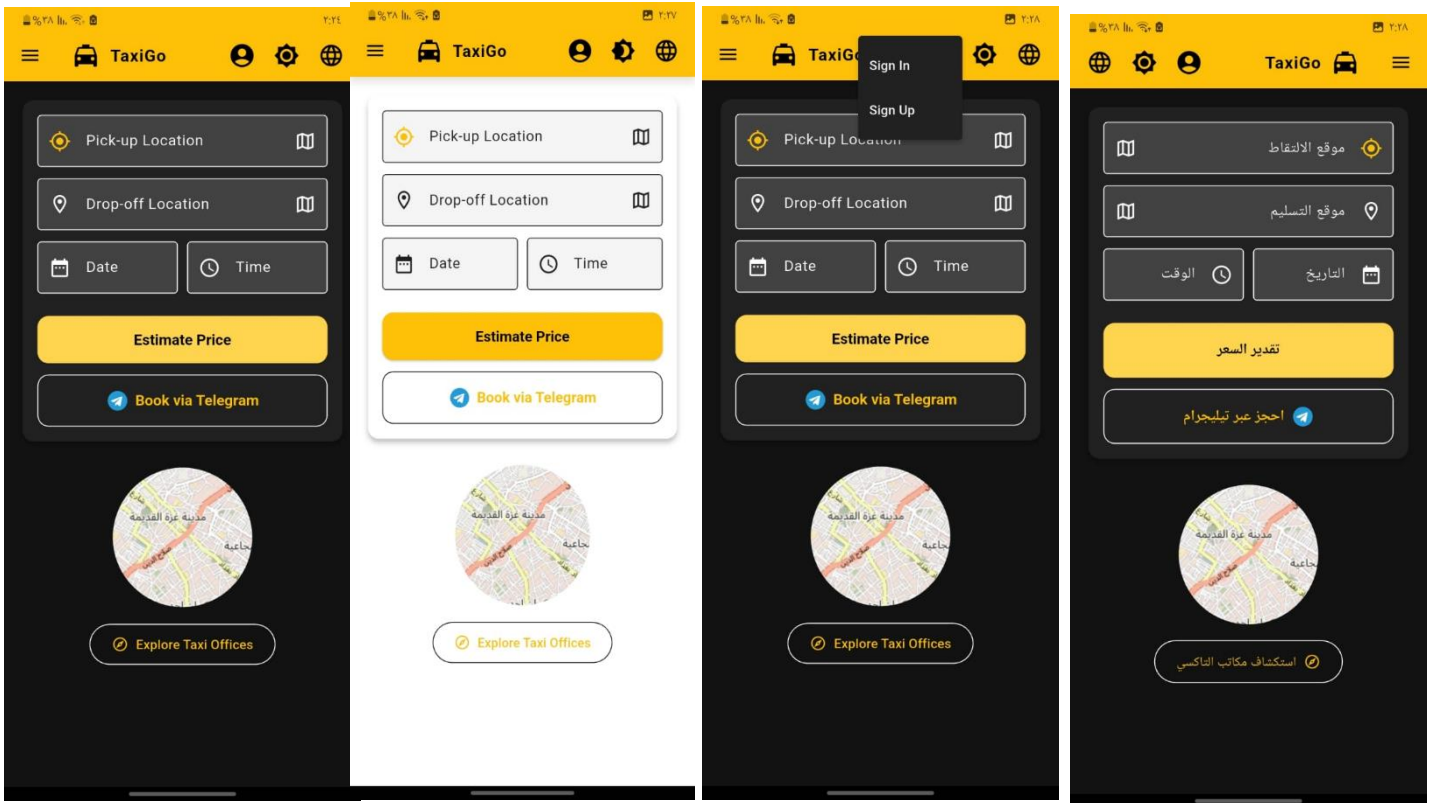
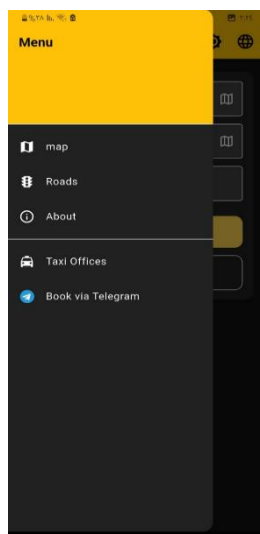
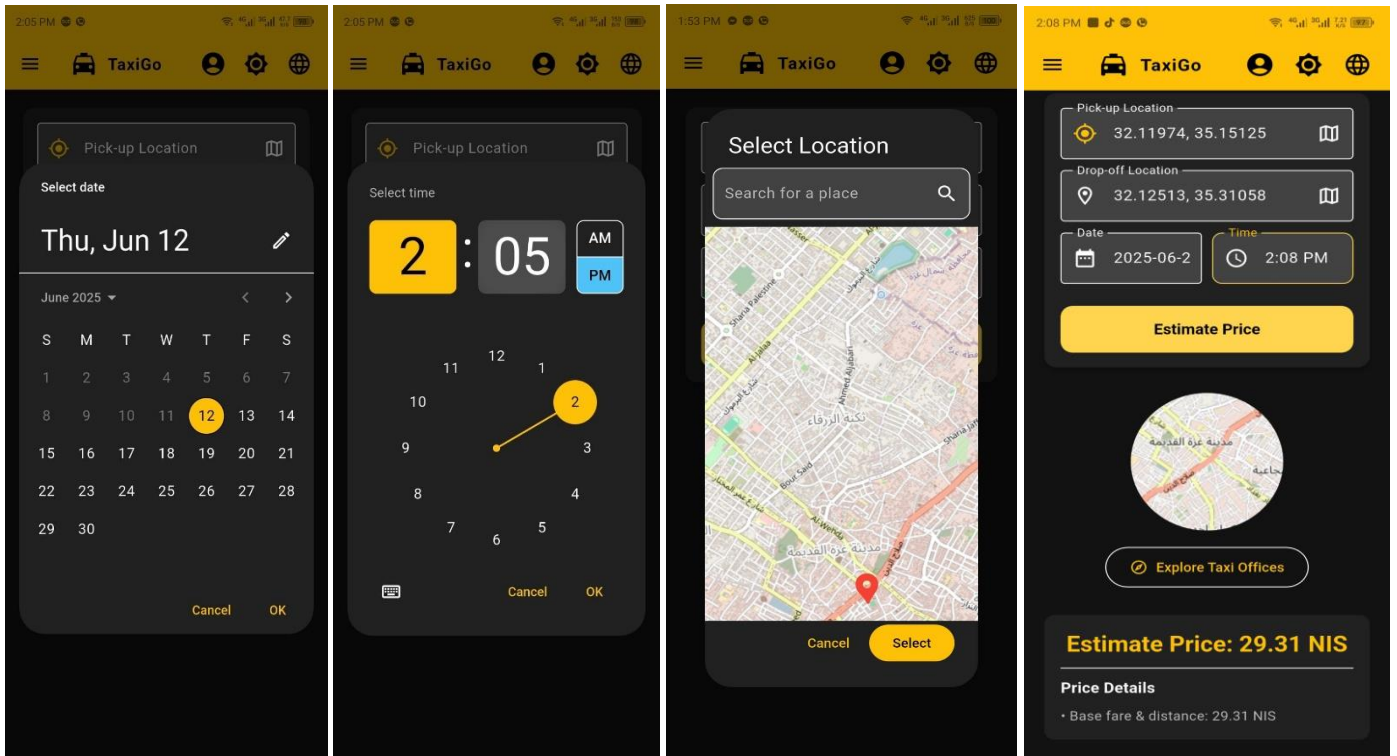


Figure2 Home Page

4. displays four main navigation options that enhance the user’s interaction with the app as a visitor.



5. **Trip Price Estimation Feature:** In the main body of the home page, visitors can estimate the price of a ride. They do this by selecting the start and end locations using an interactive map, and then entering the desired date and time of the ride. Upon clicking the "Estimate Price" button, the estimated cost of the ride is calculated and displayed on the screen.



Trip Price Estimation3 Figure

6. **View Taxi Offices on the Map:** At the bottom of the home page, there is a button that opens a map showing the locations of all taxi offices. Each office is clickable, allowing the visitor to view details about the office and directly call them using a dedicated call button.

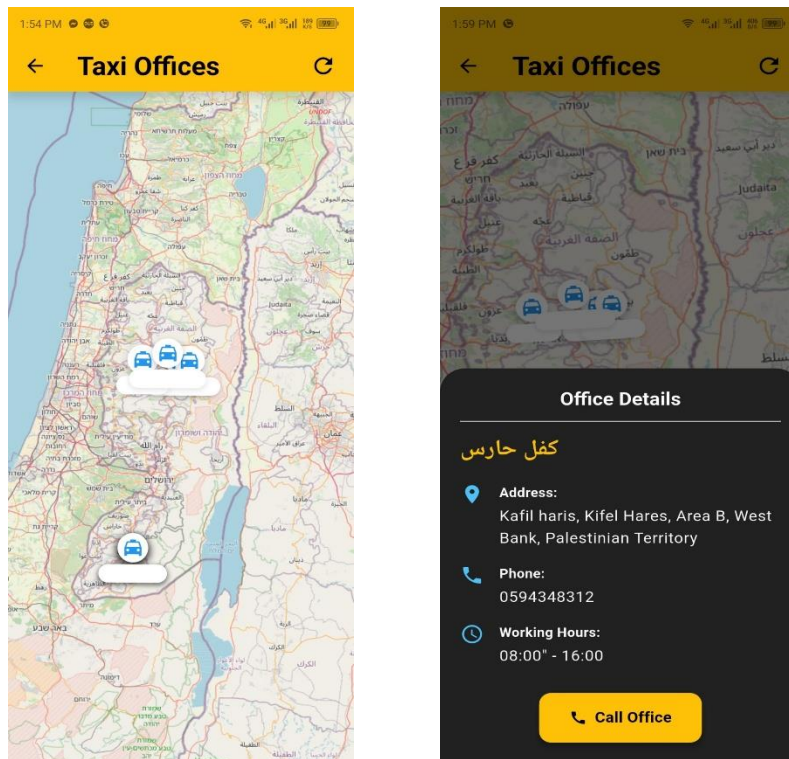


Figure 4 View Taxi Offices

Taxi Offices Page: This final button navigates to the same Taxi Offices map feature found on the home page, allowing users another way to view, locate, and contact registered taxi offices directly through the application.

7. **Map Page:** The first option is "Map", which opens a detailed map using an external API called Open Source Routing Machine (OSRM). It displays the best route between the user's current location and any destination they search for. The map also includes a button that shows real-time road status updates (open, closed, traffic, checkpoints, etc.) based on live information received through a dedicated Telegram channel.

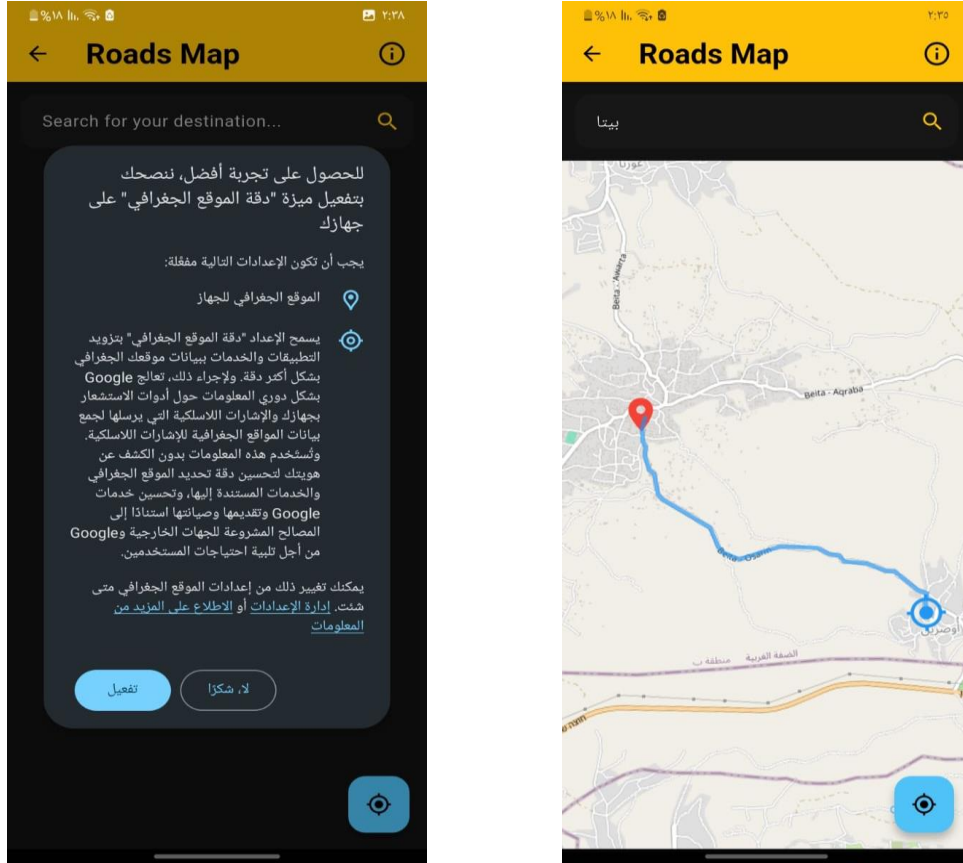


Figure5 Map Page

8. Roads Page: The "Roads" page displays real-time road conditions across major cities in Palestine. These updates are sourced from a Telegram bot/channel and processed using Gemini AI, which intelligently analyzes and filters messages to present accurate and relevant traffic information. This helps visitors stay informed and plan safer travel routes.

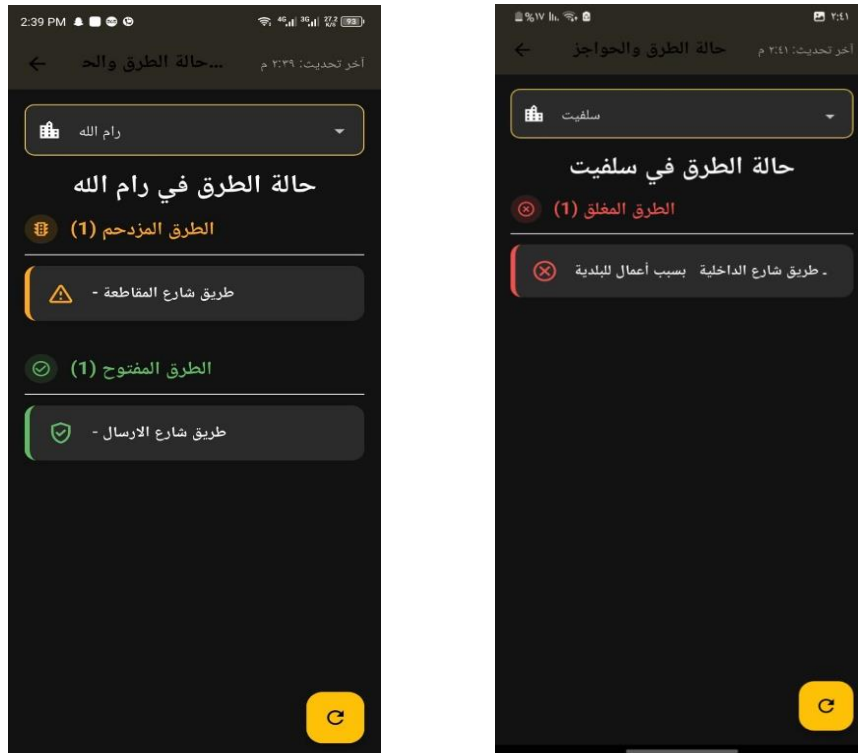


Figure 6 Roads Page

8. About Page: The "About" section provides a brief overview of the TaxiGo project, including its purpose, goals, and the development team behind it. It serves to introduce users to the idea and vision of the platform.

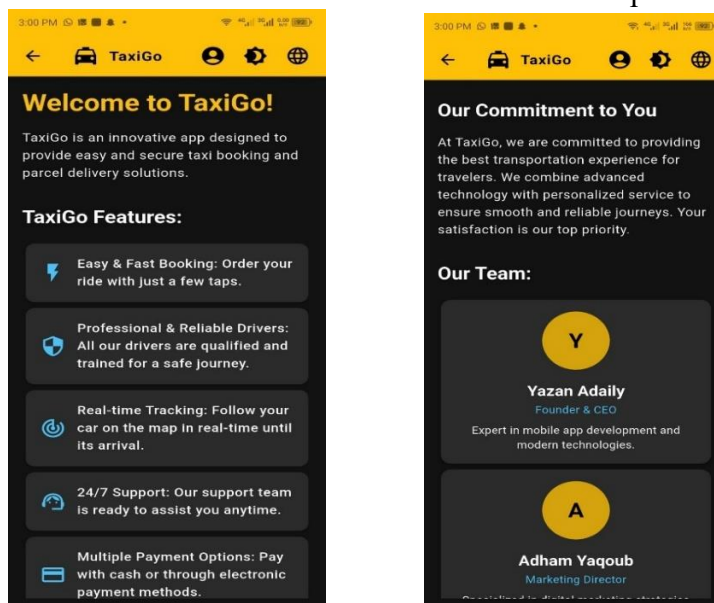


Figure 7 About Page

9. **User Registration Page:** The User Sign Up page in the TaxiGo app allows new clients to register for an account by filling out a simple form. The form collects essential information such as the user's full name, phone number for communication, email address, password (entered twice for confirmation), and gender. To proceed, users must agree to the terms by checking a box, which enables the "Sign Up" button.

Before submission, the system validates the password strength to ensure account security. The password must be at least 8 characters long, contain both uppercase and lowercase letters, and include at least one special character or number. If the password does not meet these criteria, the form displays an error message prompting the user to choose a stronger password.

Upon submitting the form, the system sends an email verification link to the user's email address, and the account remains inactive until the email is confirmed. At the bottom of the page, two helpful links are provided: one that asks "Are you a driver?", which redirects drivers to a separate registration page with different requirements, and another that says "Already have an account?", which navigates users to the Sign In page in case they already have a registered account. This setup ensures a smooth, secure, and role-specific onboarding experience tailored to both regular users and drivers.

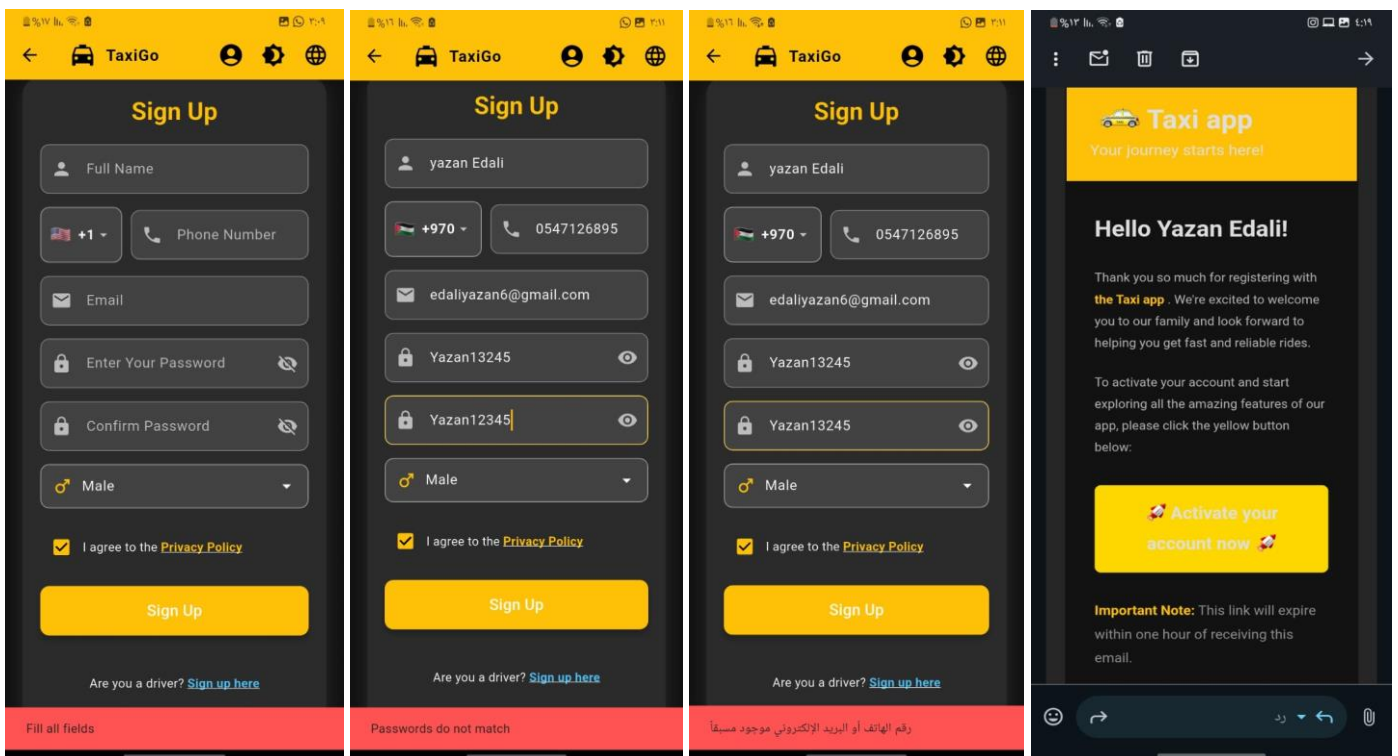


Figure 8 User Registration Page

10. **driver Registration Page:** The Driver Sign Up page in the TaxiGo app is designed specifically for taxi drivers who wish to join the platform. In addition to the standard user registration fields—such as full name, phone number, email, password (with confirmation), and gender—the driver must also provide professional and vehicle-related details. These include the driver's license number, the license expiration date, the car's plate number, vehicle type, vehicle color, and most importantly, the Taxi Office ID. This Taxi Office ID is a unique identifier assigned to each taxi company and is managed by the system administrator (admin), ensuring that each driver is correctly linked to a valid and registered office. Just like regular users, drivers must agree to the terms and verify their email to activate their account. This specialized sign-up process helps maintain professional standards and ensures that all drivers on the platform are properly registered and connected to legitimate taxi offices.

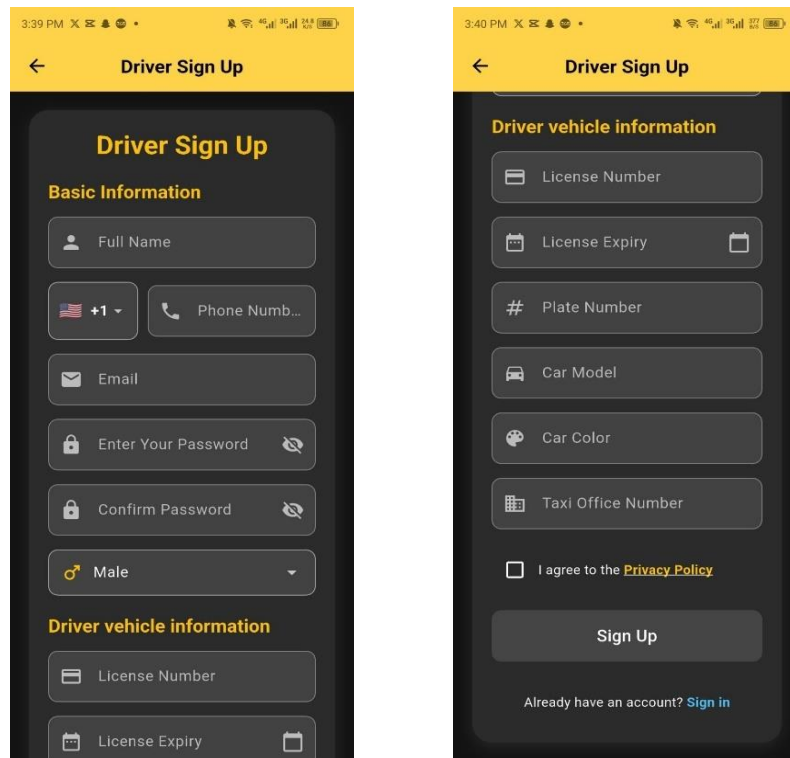


Figure 9 Driver Registration Page

11. **Sign in:** The Sign In page in the TaxiGo app allows existing users—whether clients, drivers, managers, or admins—to log into their accounts using their registered email and password. Once the login credentials are submitted and verified, the system performs an additional security check to ensure that the user’s session is valid and active. In the database, there is a specific field named is Logged, which plays a key role in this process. After a successful login, this field is automatically updated to true, indicating that the user is currently signed in. Conversely, when the user logs out, the system sets is Logged back to false. This mechanism helps prevent multiple or unauthorized sessions under the same account and enhances the overall security and control of user activity within the application. It ensures that users can only access protected features when they are properly authenticated and actively logged in.

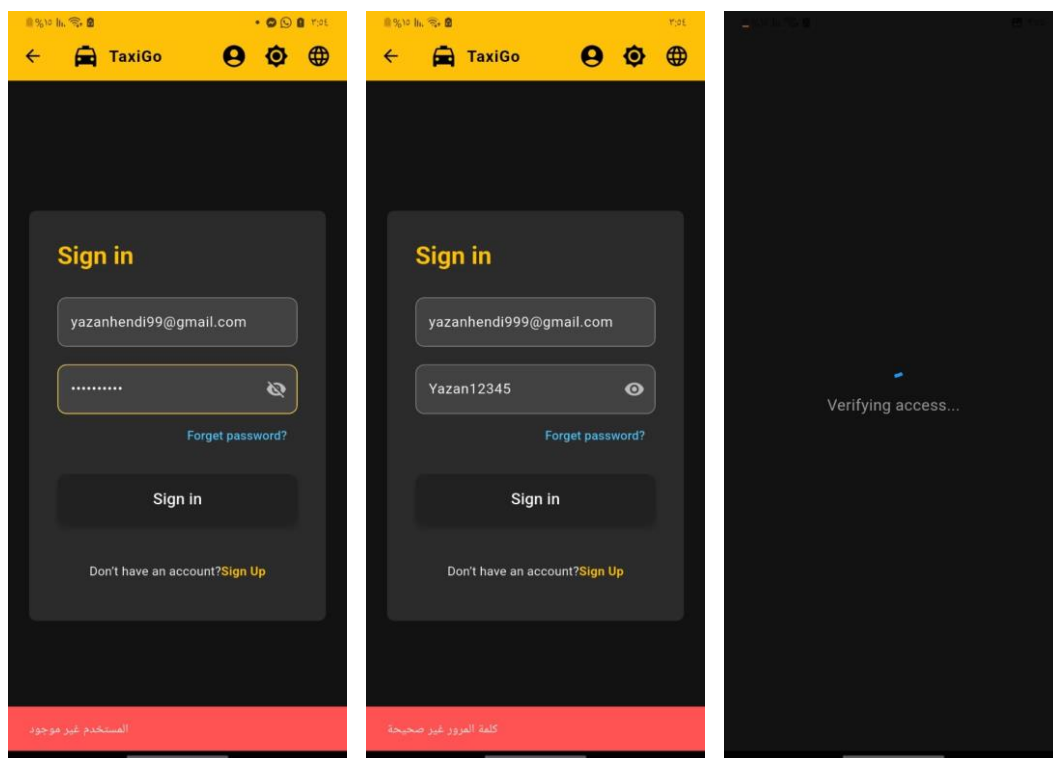


Figure 10 Sign in

12. **Forget Password:** The "Forgot Password" feature in the TaxiGo app provides users with a secure and easy way to reset their password if they forget it. This option, located on the login page, redirects users to a dedicated password recovery page, where they are asked to enter the email address associated with their account. Once the email is submitted, the system sends a password reset link to that email address, allowing the user to create a new password. This process ensures that only the rightful account holder can regain access to the account, maintaining security and user convenience. If we try to access the same link a second time and reset our password, it will not be accepted because the token becomes invalid.

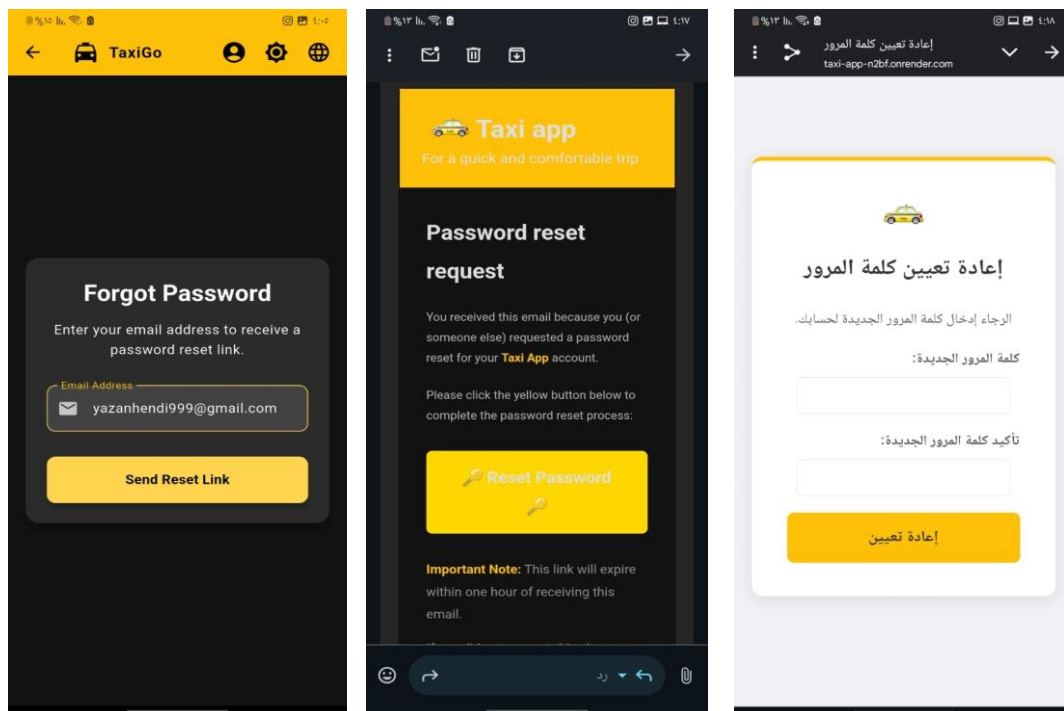
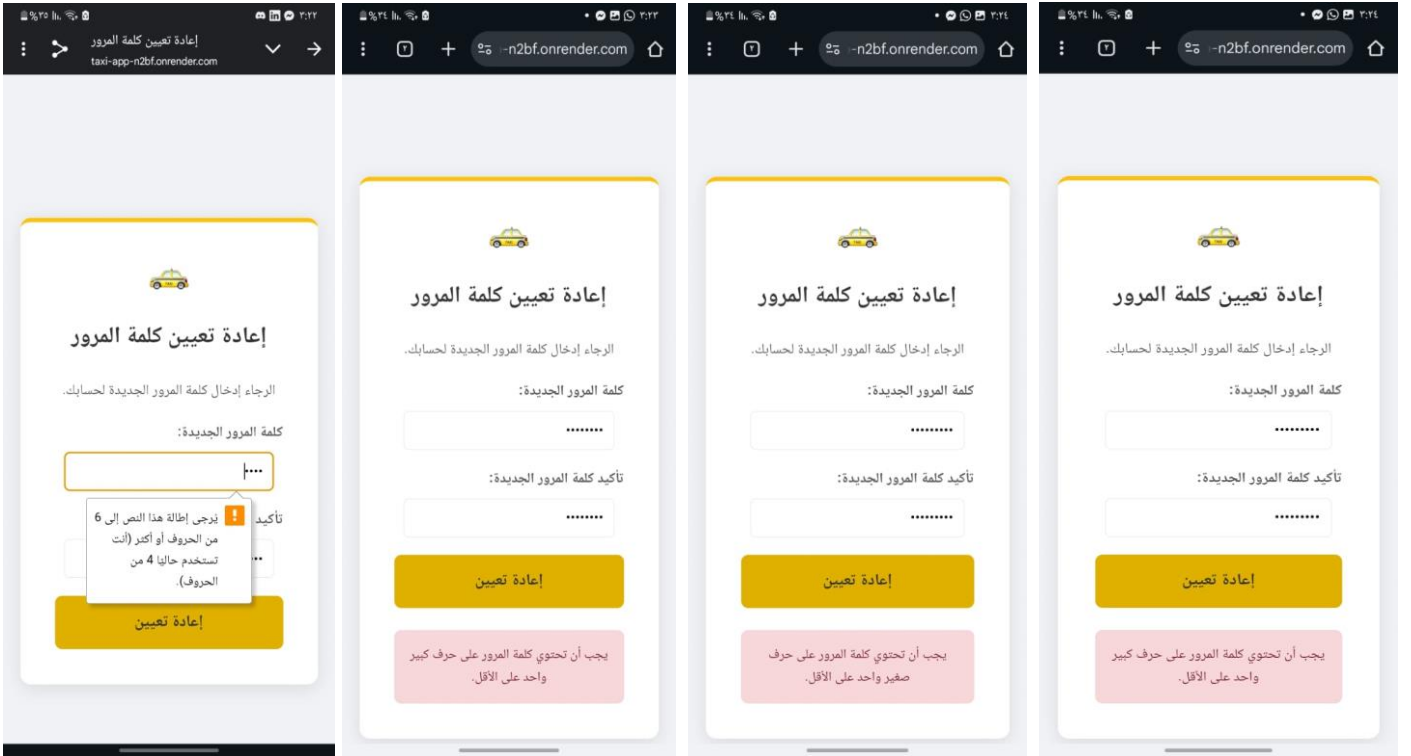
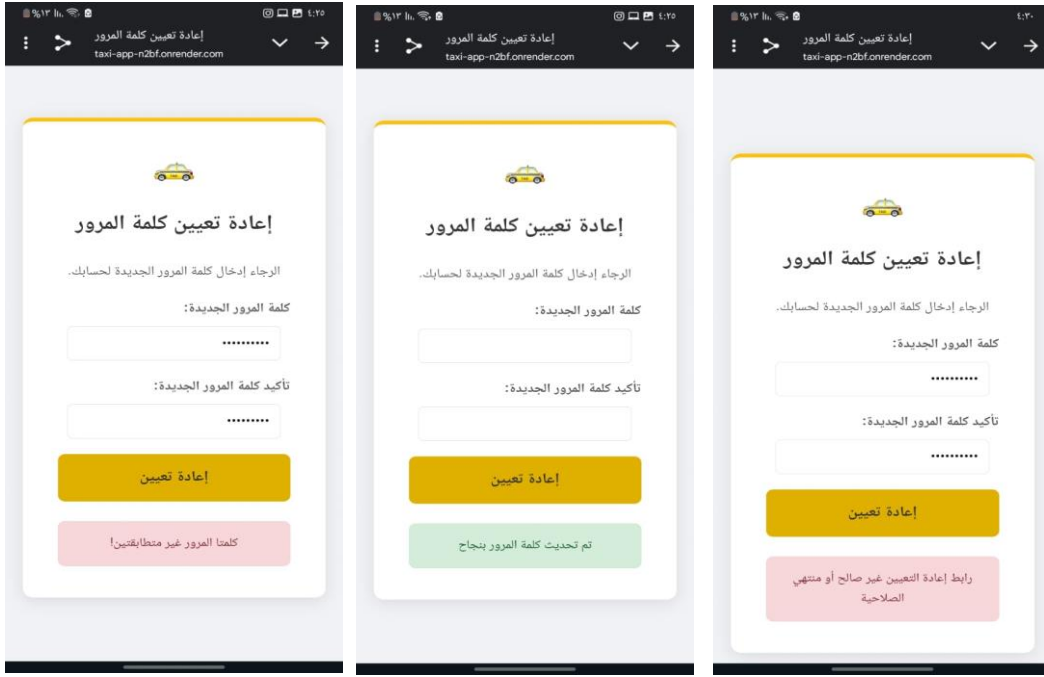


Figure 11 Forget Password



1. **User:** After successfully signing in as a regular user, the application navigates to the User Home Page, which serves as the central hub for booking a ride and accessing key features.

At the top, the AppBar displays the title "User Dashboard", a notifications icon, and a menu button that toggles the navigation drawer (dashboard). This drawer contains essential navigation options such as:

- **Home:** redirects to the main booking screen.
- **Trip History:** allows users to view their past rides.
- **Driver List:** shows all available and approved drivers in the system.
- **Settings:** lets users update their personal preferences.
- **Support:** provides access to help and support services.

User Home Page Functionality: The User Home Page allows users to easily request a new ride. When the page is loaded, the app automatically fetches and stores the user's current location into a text field. The user can then set the destination location by either searching for a place on the map or tapping directly on it. Once both the start and end points are selected, the system calculates and displays the estimated distance and ride cost.

The user is also required to choose a payment method, with three options available:

- **Cash** – pay directly to the driver.
- **Wallet** – a virtual balance stored in the database. The system checks if the user's wallet contains enough funds; if so, the booking proceeds.
- **Card** – although available as a choice, this method is currently non-functional and used for future integration.

After confirming all ride details, the user can press the "Request New Ride" button to submit the booking. This triggers the ride request process and prepares the system to notify nearby drivers.

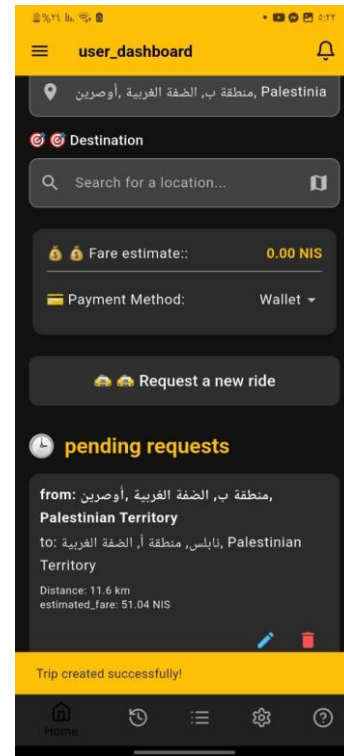
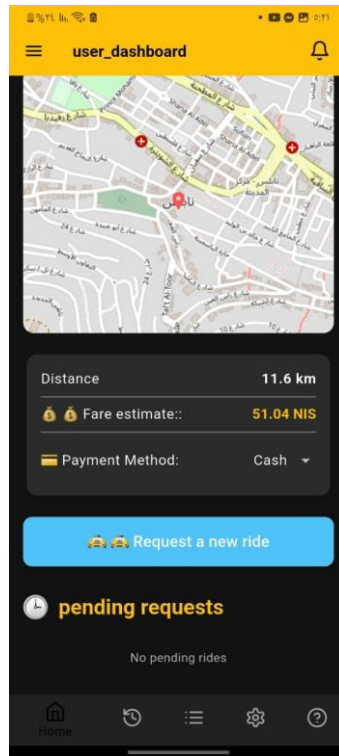
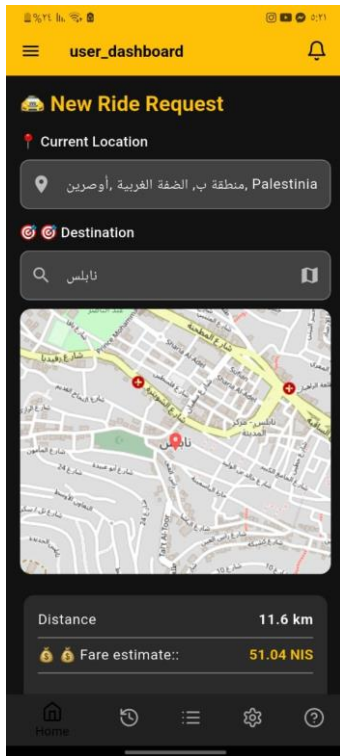
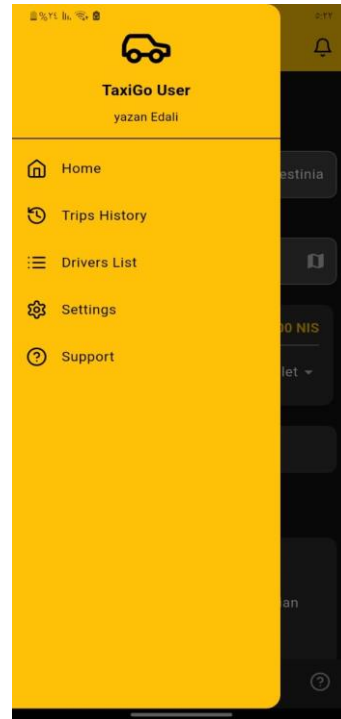
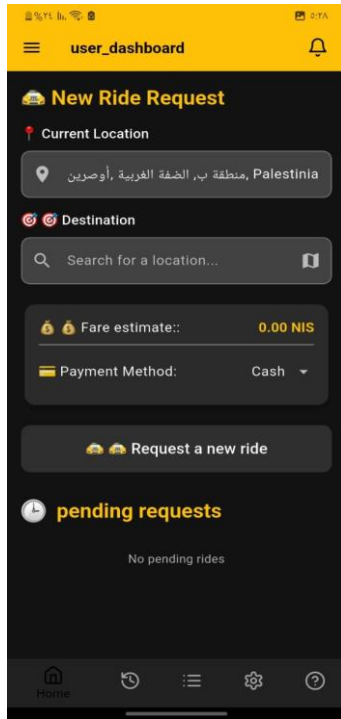


Figure 12 User

Scheduled Trips Page: In addition to immediate ride booking, the app provides a **Scheduled Trips** page where users can plan rides in advance. The user selects a **specific date and time** for the ride along with the usual pickup and destination locations. When the scheduled time arrives, the system **automatically activates the ride request**, making it visible to available drivers. This feature improves flexibility and ensures timely transportation for planned events.

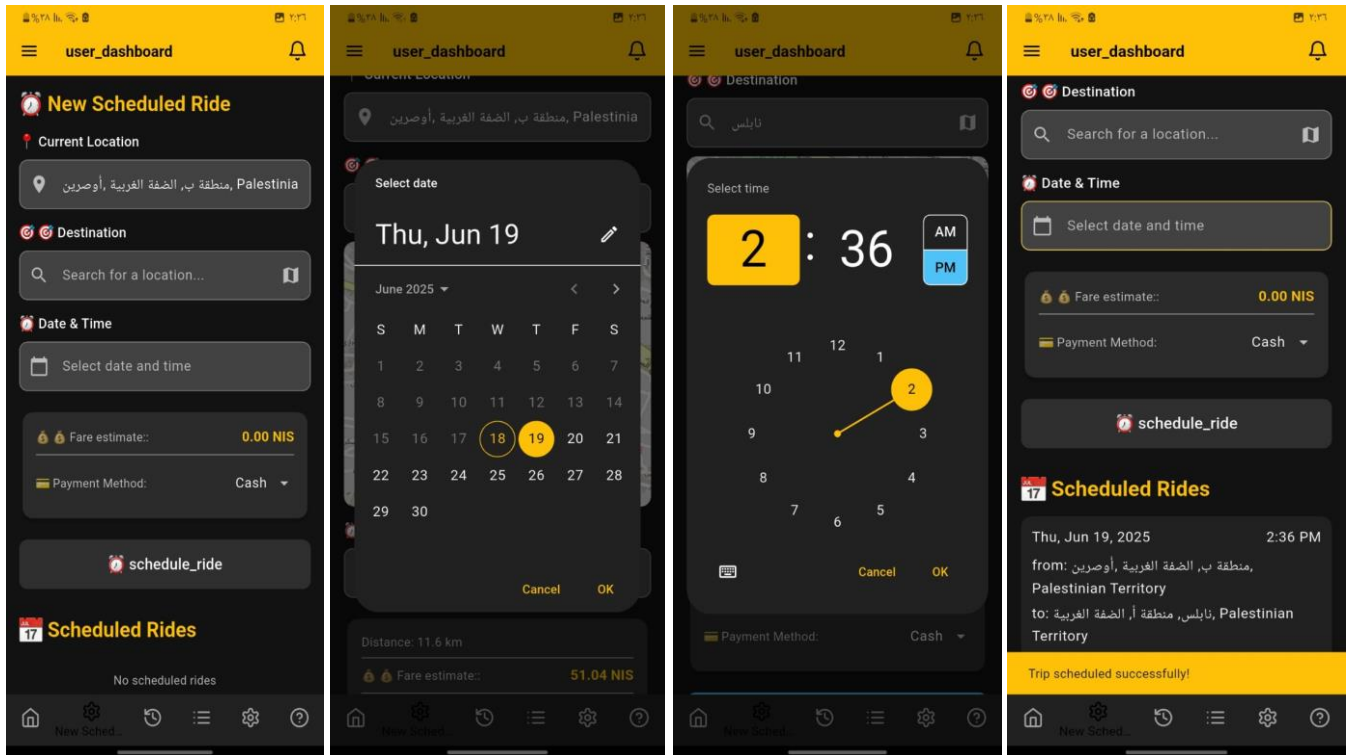
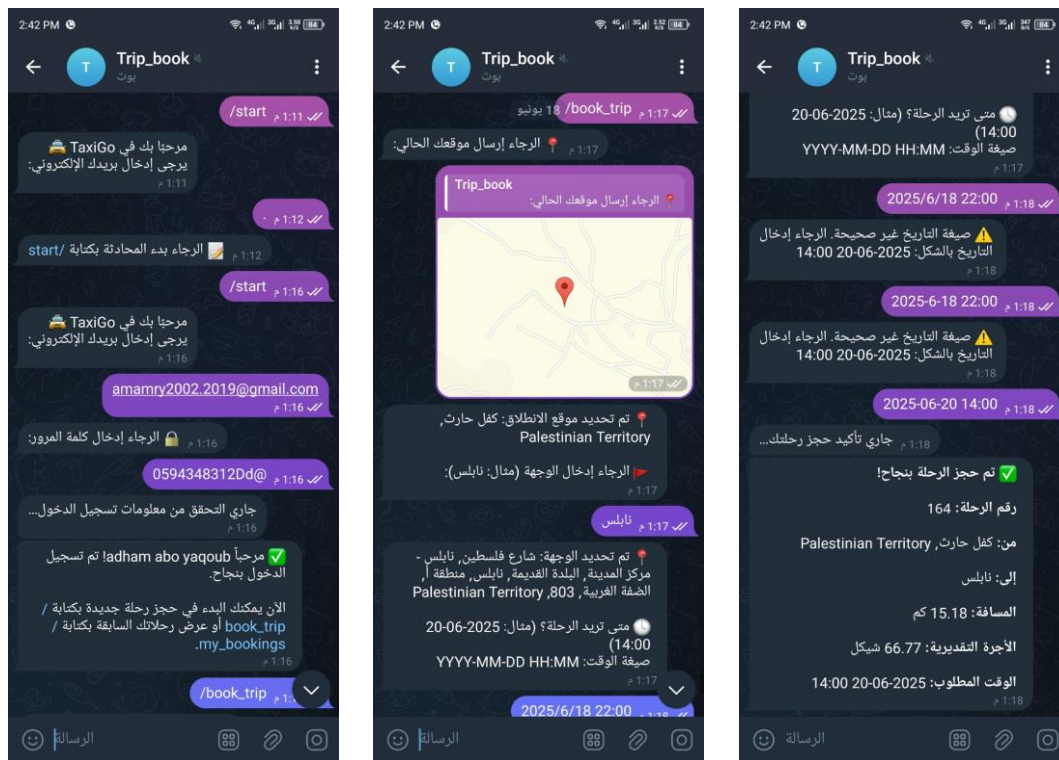


Figure13 Scheduled Trips Page

Scheduled Trip Booking via Telegram Bot: In addition to booking scheduled trips through the mobile app, TaxiGo offers a convenient Telegram bot that allows users to schedule rides directly from the messaging platform. After a one-time login via a secure authentication link, the bot stores the user's credentials, eliminating the need to log in again for future bookings.

Through a simple conversational interface, the user is prompted to provide essential trip details such as the pickup location, destination, and desired date and time. Once confirmed, the scheduled trip is recorded in the system and will be automatically activated at the specified time, just like in-app bookings.

This feature enhances accessibility and user convenience, particularly for users who prefer lightweight, fast interactions without navigating through the full app interface. It also showcases TaxiGo commitment to multi-channel flexibility and seamless user experience.



مرحباً adham abo yaqoub! حسابك مرتبط بالفعل.
الرجاء إدخال أمر لحجز رحلة جديدة /book_trip/
أو /my_bookings/ لعرض رحلاتك.
12:10

Figure 14 Scheduled Trip Booking via Telegram Bot

2. **Trip History Page:** The Trip History page provides users with a list of all their previously requested rides, including trips that are:
 - **Accepted** – The driver has accepted the ride but hasn't started it yet.
 - **In Progress** – The ride is currently ongoing.
 - **Completed** – The ride has finished successfully.

Each trip entry displays summary information such as the trip status, date, and basic route details. When a user clicks on a specific trip, a popup window appears showing more detailed information about that particular ride. This includes the pickup and drop-off locations, trip duration, driver name, payment method, and fare.

This feature allows users to easily track their ride history and review the details of any past trip.

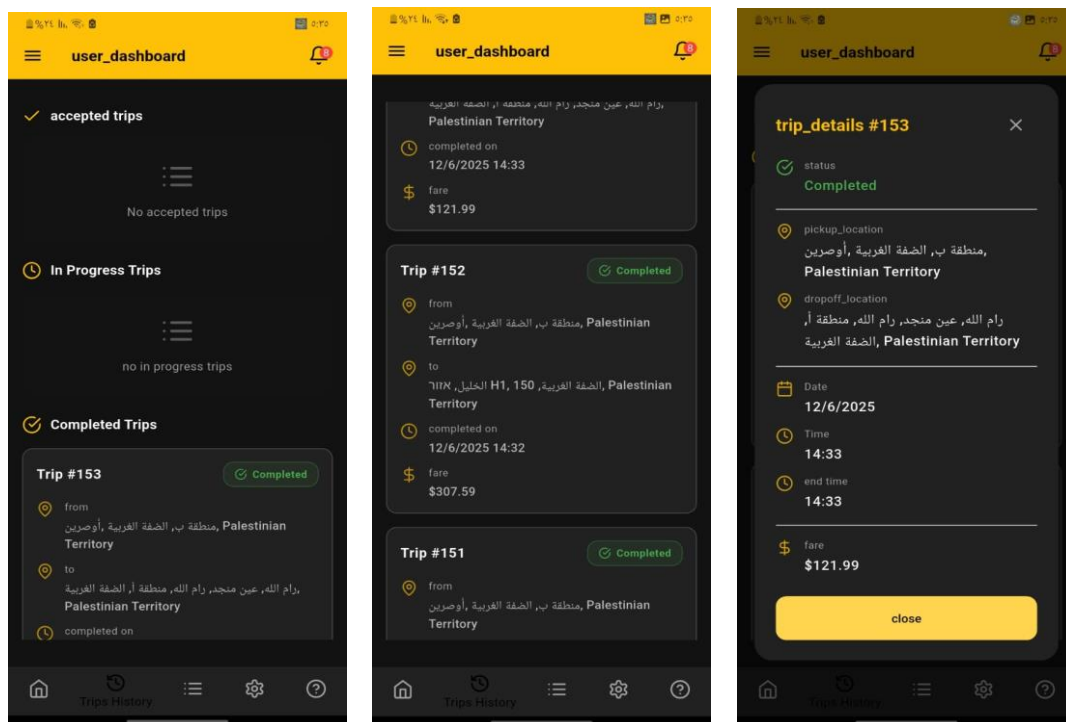


Figure 15 Trip History Page

3. **Driver List Page:** The Drivers page displays a list of all drivers who are currently marked as "Available" in the system. This availability status is managed and controlled either by the admin or the manager responsible for that driver.

Each driver entry may include basic details such as their name, car type, license plate number, and taxi office. Users can view this information to become familiar with the active drivers in the system.

This page helps users explore trusted, approved drivers who are ready to accept trips.

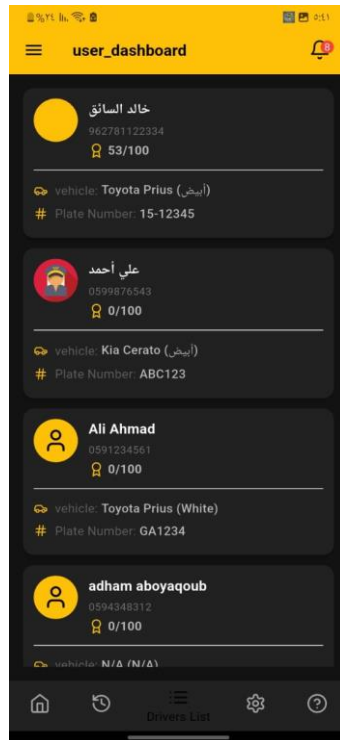
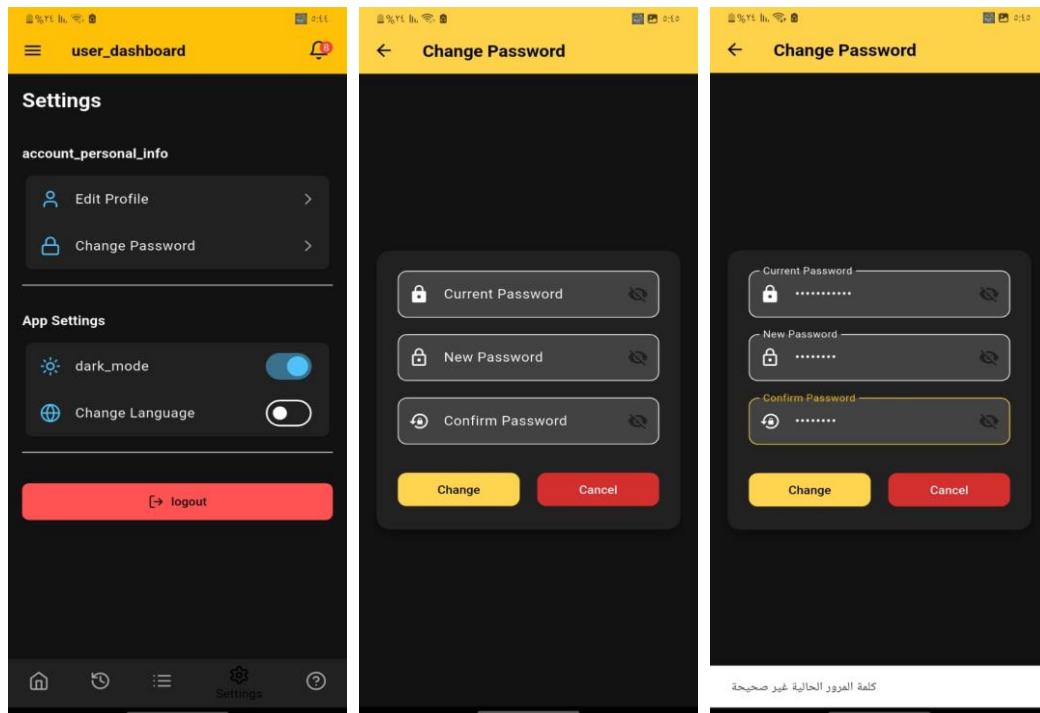


Figure 16 Driver List Page

4. **Settings Page:** The Settings page allows users to manage their personal preferences and account information. It includes four main functionalities:
- **Change Theme & Language:** Users can switch between dark/light themes and toggle the app language between Arabic and English. (Already explained earlier.)
 - **Change Password:** This option opens a form where the user must enter their current password and then input and confirm a new password. This ensures security during password updates.
 - **Edit Profile:** Users can update their profile picture, name, phone number, and email. This helps keep their account details up to date and personalized.



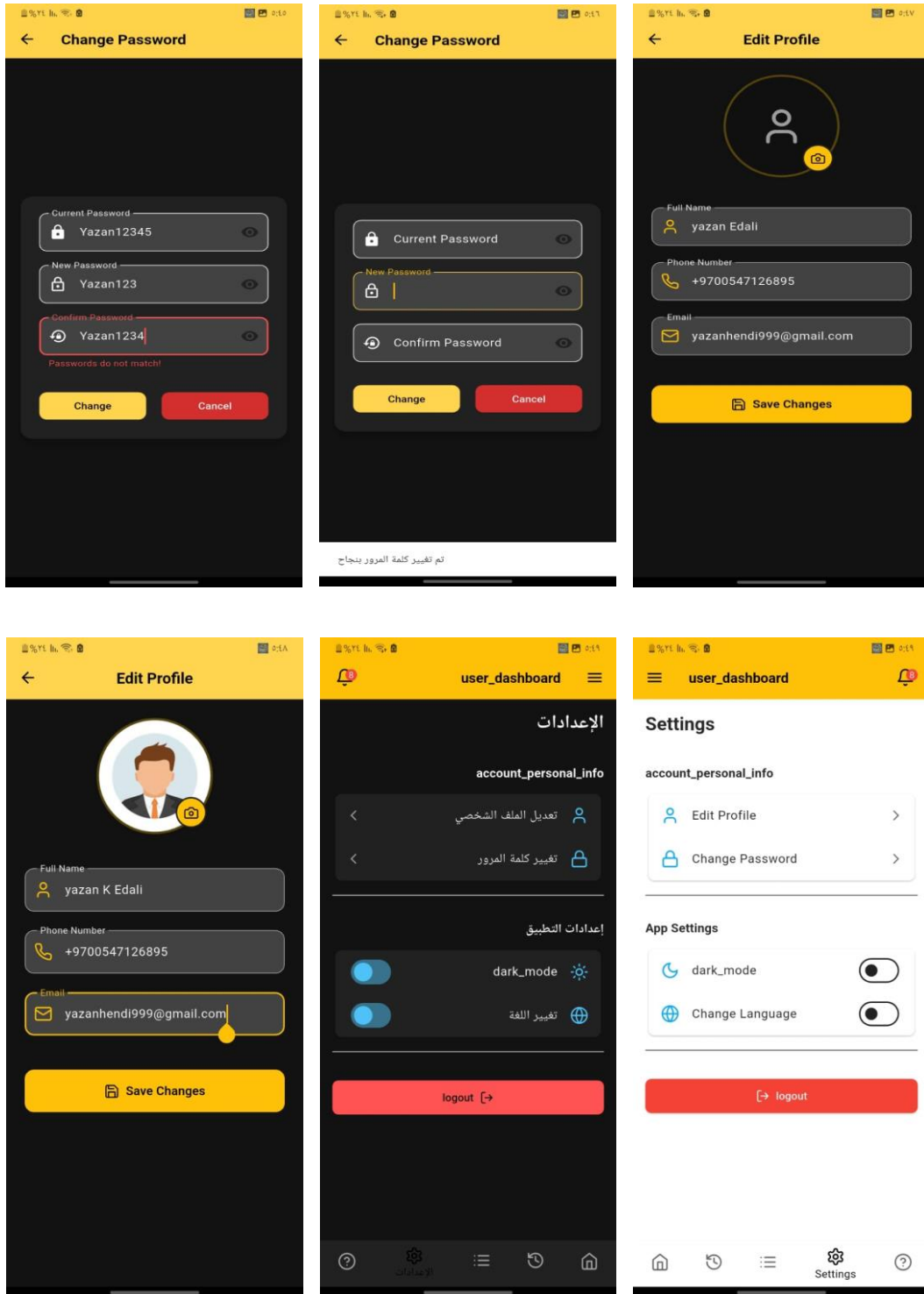


Figure 17 Settings Page

5. **Support Page:** The Support page provides users with quick access to emergency help and general assistance. It includes the following features:

✓ **Emergency Call Options**

Users can quickly request help in urgent situations by choosing to call:

- Police
- Ambulance
- Fire Department

These buttons initiate direct calls to the respective emergency services.

✓ **Support Contacts**

The page also displays special contact numbers for TaxiGo support, allowing users to reach out for help regarding the app or their trips.

✓ **FAQ Section**

At the bottom, there is a Frequently Asked Questions (FAQ) section where users can view common questions and their answers, helping them solve issues without needing direct assistance.

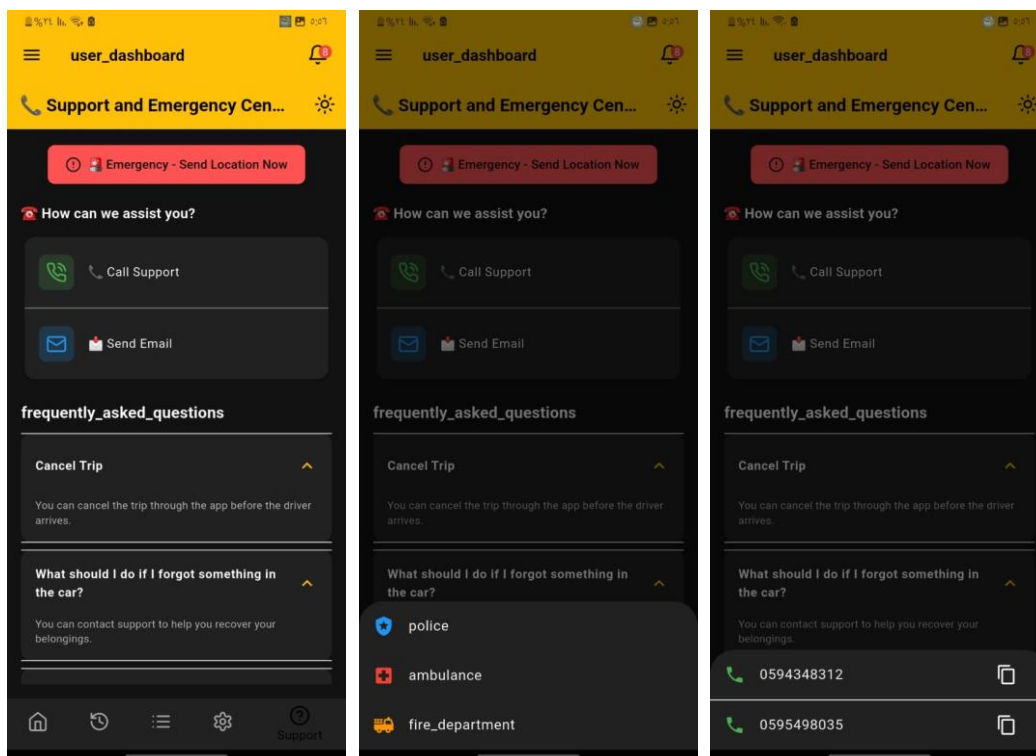


Figure 18 Support Page

6. **Notifications:** The Notifications system delivers real-time updates to the user regarding the status of their trip. These push notifications appear directly in the app and alert the user about important events, including:

- Trip Accepted: The user receives a notification when a driver accepts their trip request.
- Trip Rejected: If a driver declines the request, the user is instantly informed.
- Trip Started: A notification is sent as soon as the trip begins.
- Trip Completed: The user gets notified once the trip ends.

In addition to displaying real-time alerts, each notification is marked as "unread" when received. Once the user clicks on a notification to view its details, the status of the notification automatically changes to "read", helping users manage and keep track of new vs. old updates efficiently. This enhances the user experience by maintaining a clear and organized notification system.

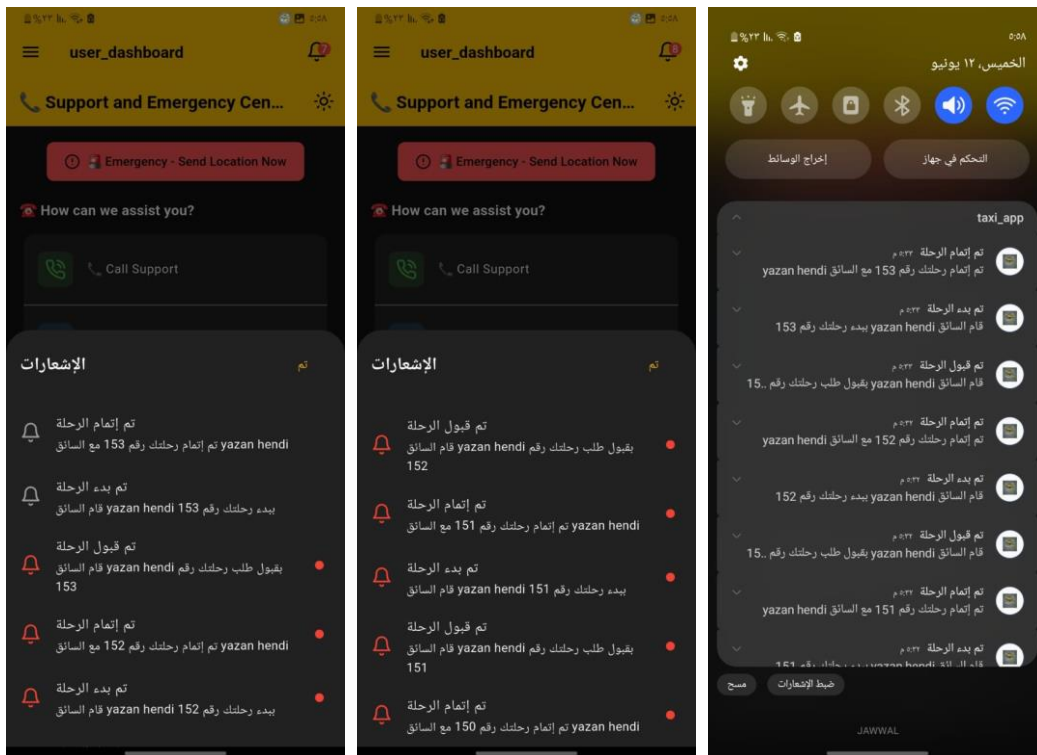


Figure 19 Notifications

1. **Driver Page:** Upon logging in, the driver is taken to the Home Page, which displays their name and basic personal information. One of the most important components on this page is the driver's rating. Unlike traditional systems where users provide feedback manually, this platform automatically calculates the driver's rating using system logic. The rating is based on how the driver interacts with trips. For instance, if a driver accepts a trip but delays starting it, their rating will decrease. On the other hand, starting the trip on time will positively impact the rating. The faster and more punctual the driver is, the better their score becomes. These ratings are computed using fair and logical mathematical formulas designed to reflect real performance.

Additionally, the Home Page displays the recent trips section, which shows the last three trips the driver has completed. This allows the driver to easily review their latest ride activity and stay updated on their performance and job history.

At the top of the driver interface, the App Bar includes a settings icon, a notifications icon, and a menu button. Tapping the menu button opens the Dashboard Sidebar, which contains links to several important pages:

- Home
- Trip Requests
- My Trips
- Earnings
- Settings
- Chat

Each of these pages provides essential tools and data for drivers to manage their rides, track their income, and communicate with support or customers.

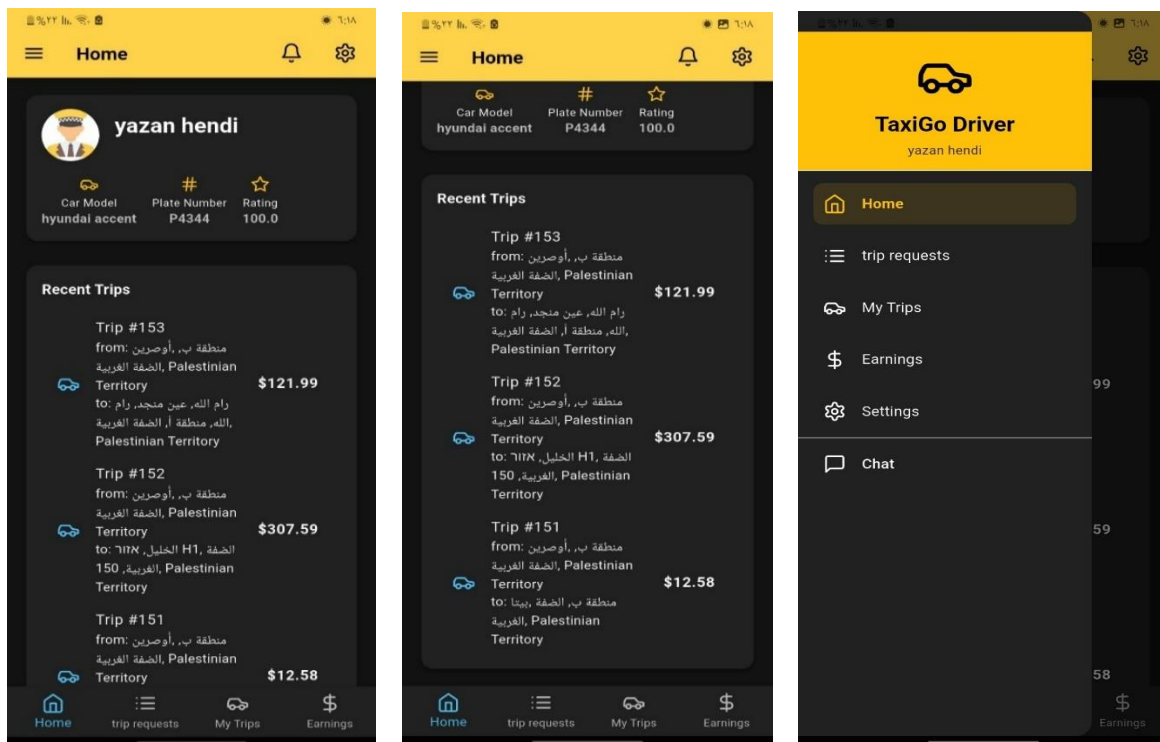


Figure 20 Driver Page

2. **Trip Requests page:** The Trip Requests page in the driver interface is divided into two main sections: Pending Trips and Accepted Trips.

In the Pending Trips section, the system displays ride requests that are geographically near the driver. The search starts by looking for trips within a 5-kilometer radius of the driver's current location. If no trips are found, the system automatically expands the search radius to 15 kilometers, and if necessary, continues increasing the search range until a trip is found. The driver is always informed about the current search distance, with a message indicating how far the nearest trip is from their location.

Each pending trip includes essential details such as pickup and drop-off points, the estimated distance, and the expected fare. The driver has three options:

- Accept the trip, which moves it to the "Accepted Trips" section.
- Reject the trip, which removes it from their view.
- Ignore the trip, in which case it remains visible until it's accepted by another driver or expires.

This smart filtering and distance-based trip matching ensure drivers are only shown relevant and nearby ride requests, helping them respond quickly and efficiently.

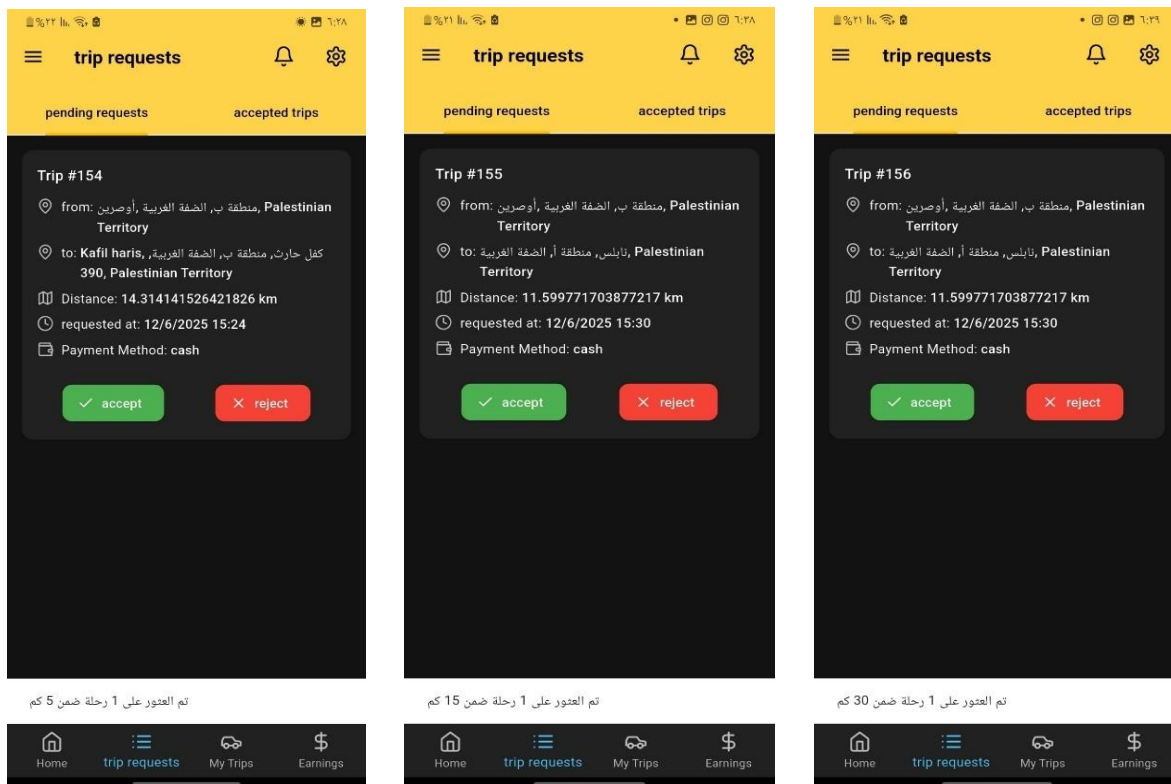


Figure21 Trip Requests page

The Accepted Trips section displays all the trips that the driver has accepted. Once a trip is accepted, the server calculates the estimated time it will take the driver to reach the passenger's pickup location based on the current distance between them. This time is critical — if the driver fails to arrive within the specified window, the trip automatically reverts back to the Pending state, and the driver's rating is reduced accordingly.

Within this section, the driver is given the option to start the trip. When the "Start Trip" button is clicked, a map opens up with accurate routing from the pickup location to the drop-off point. The map includes real-time tracking, which is a crucial feature: it allows the driver to follow the route live and helps the system continuously monitor the trip's progress. This ensures a smooth and transparent experience for both the driver and the passenger.

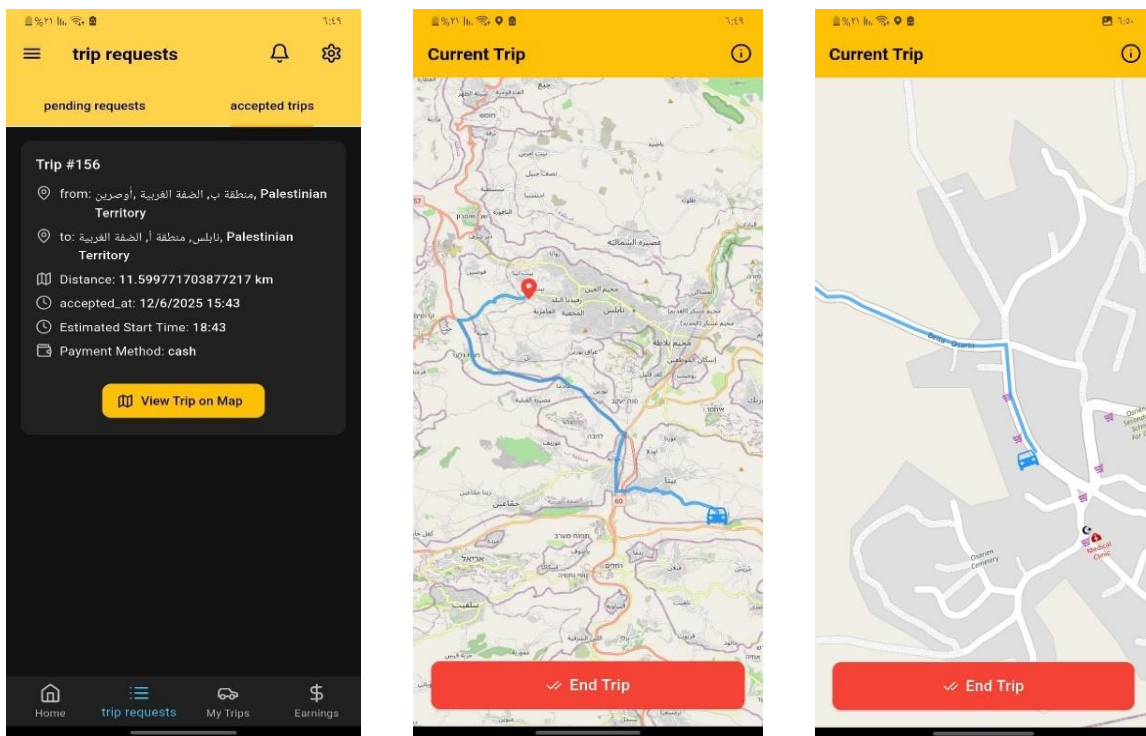


Figure 22 Accepted Trips

3. **My Trips:** This page is very similar to what the driver sees. It shows a list of all trips related to the user, including those that are ongoing or completed.

History Trip: This page provides detailed information about past trips. Each trip can have a status like Accepted, In Progress, or Complete. When the user clicks on any specific trip, a popup appears showing the full details of that trip, giving the user more insight into the journey, including times, locations, and status updates.

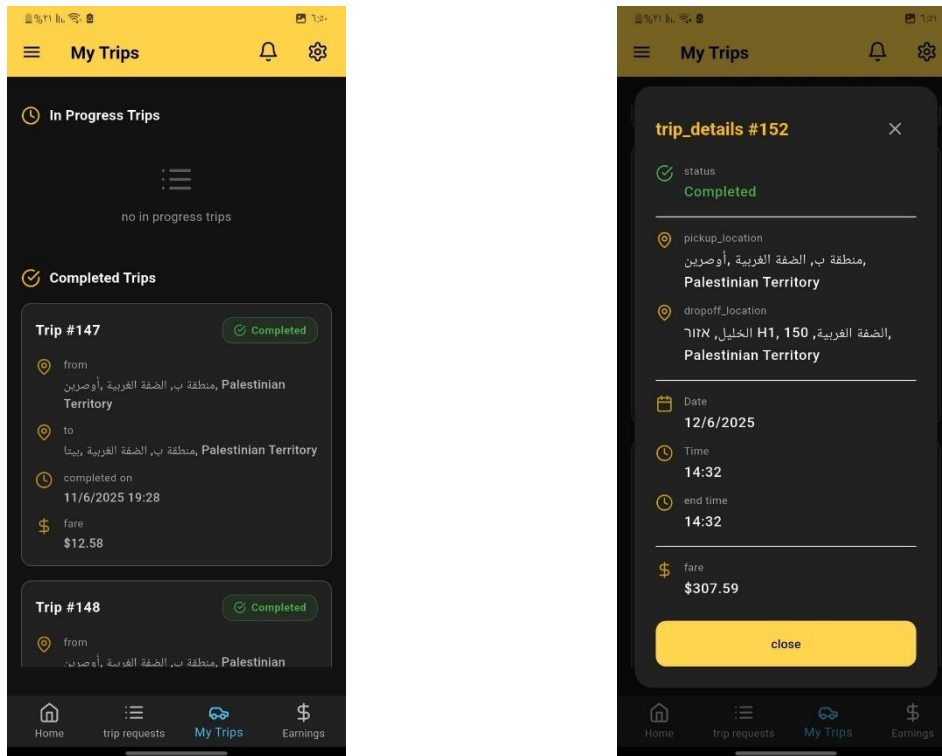


Figure 23 History Trip

4. **Earnings Page:** This page provides the driver with a simple and clear summary of their income. It displays each completed trip along with the amount earned from that trip. This helps the driver easily track their daily, weekly, or monthly earnings in an organized way.

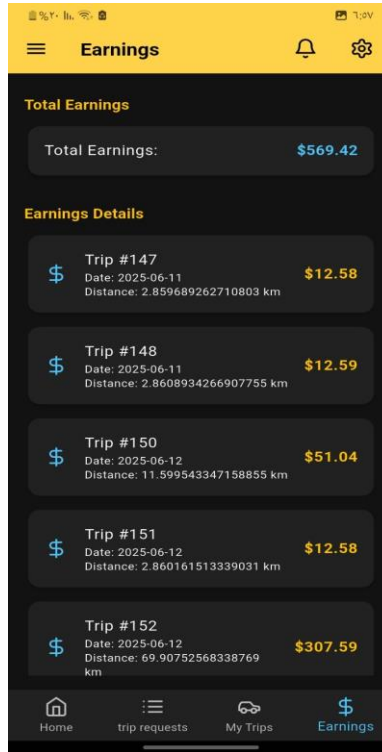


Figure 24 Earnings Page

5. **Settings Page (Driver Panel):** The settings page allows the driver to manage and personalize their account through four main actions:

- **Change Theme and Language:** The driver can switch between different visual themes (e.g., dark/light mode) and select the app language based on preference. This was explained earlier.
- **Change Password:** When choosing to change the password, the driver is prompted to enter their current password followed by the new password, which must be confirmed by entering it twice to avoid mistakes.
- **Edit Profile:** The driver can update their personal information, including changing their profile picture, full name, phone number, and email address.
- **Logout:** The driver can securely log out from their account at any time using the logout option at the bottom of the settings page.

This page helps ensure both personalization and security for the driver's account.

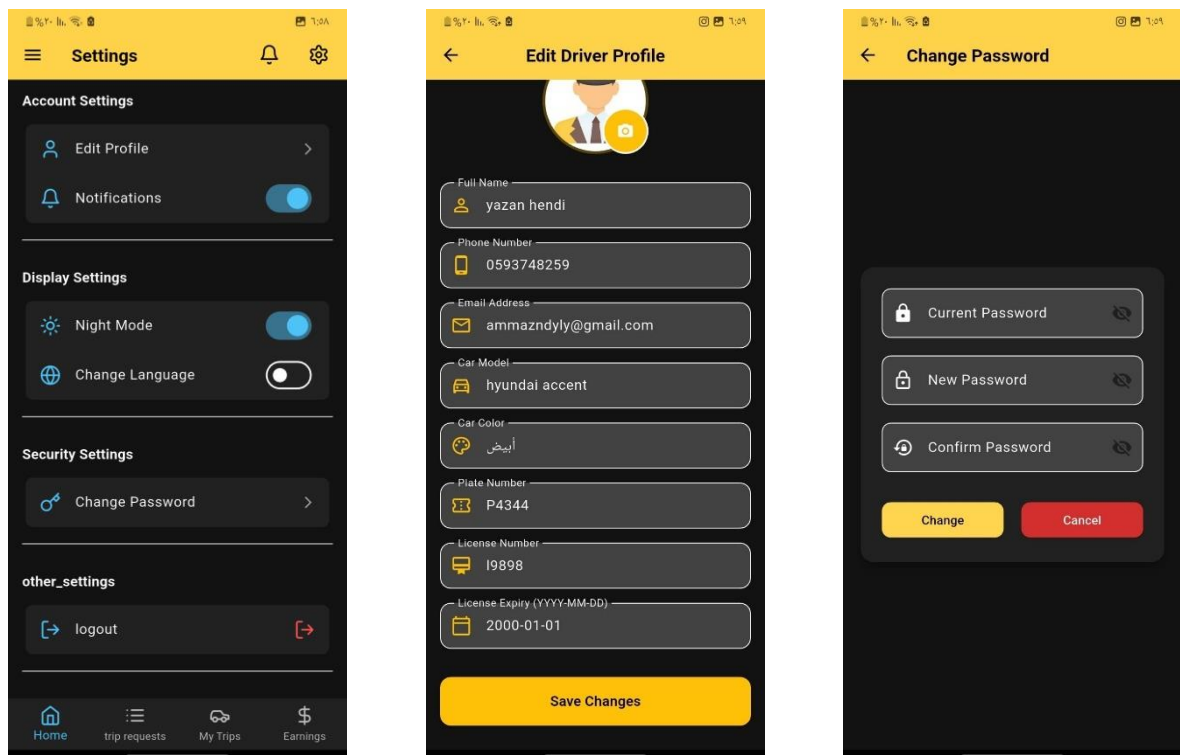


Figure 25 Settings Page

Chat Page (Driver Panel): The chat page enables real-time communication between the driver and their assigned manager. This feature is built using socket technology, which ensures instant message delivery without the need for refreshing or reloading the page. Through this live chat system, drivers can quickly receive guidance, report issues, or coordinate directly with their manager in a seamless and efficient way. Additionally, the driver receives real-time notifications for any incoming messages, ensuring they stay informed and responsive at all times. This functionality enhances support, communication, and overall efficiency within the system.



Figure 26 Chat Page

1. **Manager Panel – Home Page Overview:** At the top of the manager panel, the App Bar contains a notifications icon and a button to open the side dashboard menu. The dashboard menu includes the following sections: Home, Manage Drivers, Settings, and Chat.

The Home Page is designed to provide the manager with a quick overview of daily operations. It displays real-time updates, including:

- The total number of drivers under the manager’s supervision.
- The number of trips completed today.
- The total earnings for the current day.

Additionally, a section titled “Active Drivers Now” shows a live list of drivers who are currently marked as Available, allowing the manager to monitor their status and engagement in real-time. This overview helps the manager stay informed and manage resources efficiently throughout the day.

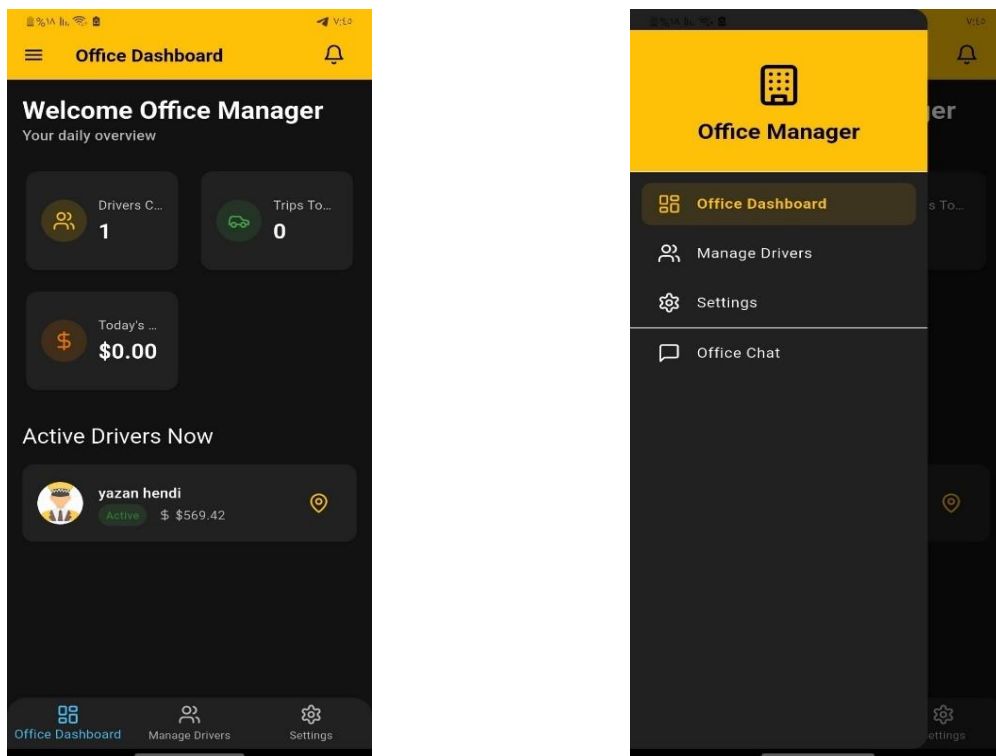


Figure 27 Manager Panel – Home Page Overview

2. **Driver List Page – Manager Panel:** This page displays a comprehensive list of all drivers under the supervision of the current manager. Each driver card includes essential details such as the driver’s name, current availability status (Available/Unavailable), and rating.

The manager can perform several key actions on this page:

Search Functionality: A search bar allows the manager to quickly find a specific driver by name or phone number.

Filtering Options: The manager can filter the list of drivers based on their availability status:

- All Drivers
- Available Only
- Unavailable Only

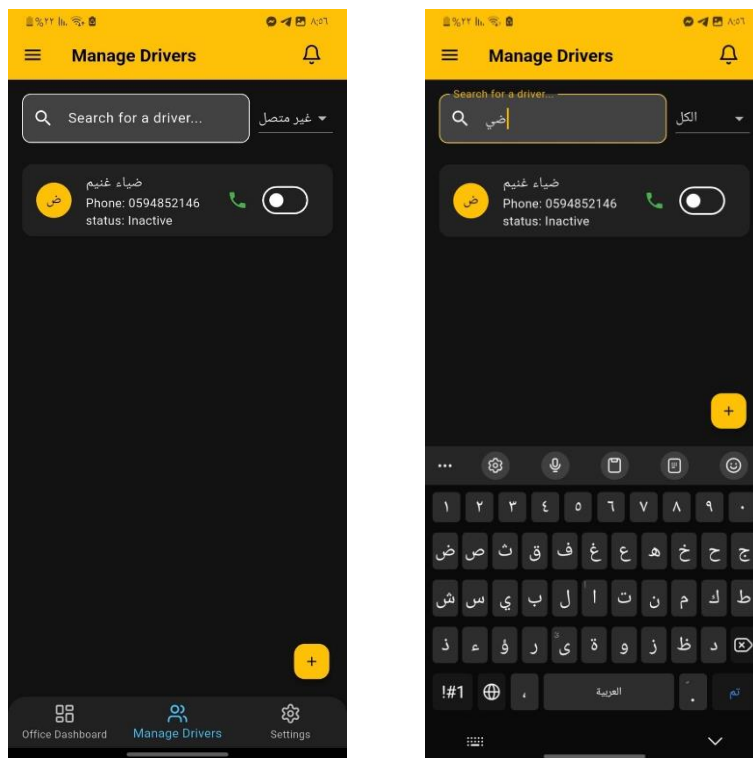
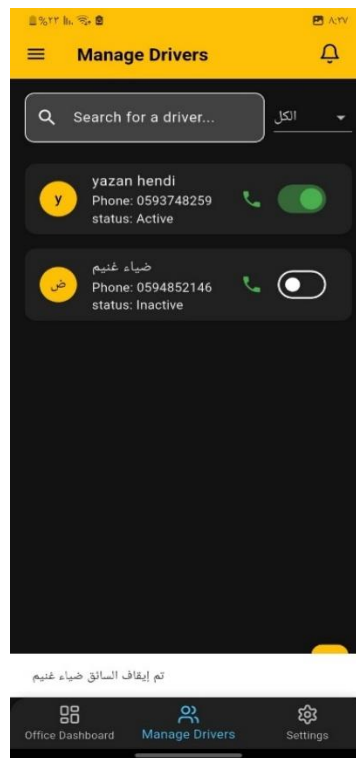
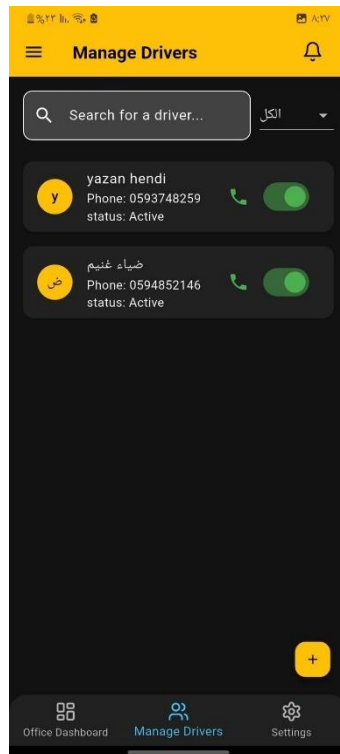


Figure 28 Driver List Page – Manager Panel

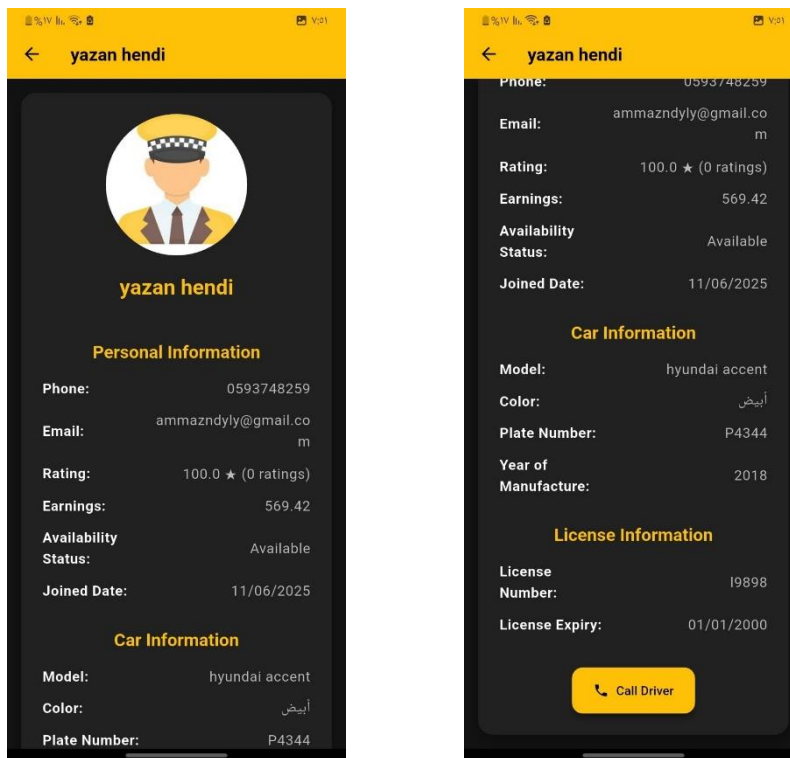
Toggle Driver Availability: The manager has the ability to activate or deactivate a driver's availability status (e.g., set a driver to "Available" or "Unavailable") directly from the list.



Direct Communication: A call button is provided next to each driver, enabling the manager to call any driver directly from within the app.



View Driver Details: By tapping on a driver's card, the manager can access a detailed profile view. This includes information such as full name, phone number, email, number of completed trips, total earnings, and current status.



View Driver Details29 Figure

Add New Driver: There is a button that allows the manager to add a new driver to the system. Tapping this opens a form to enter the new driver's details such as name, phone number, email, vehicle info, and upload documents or images if needed.

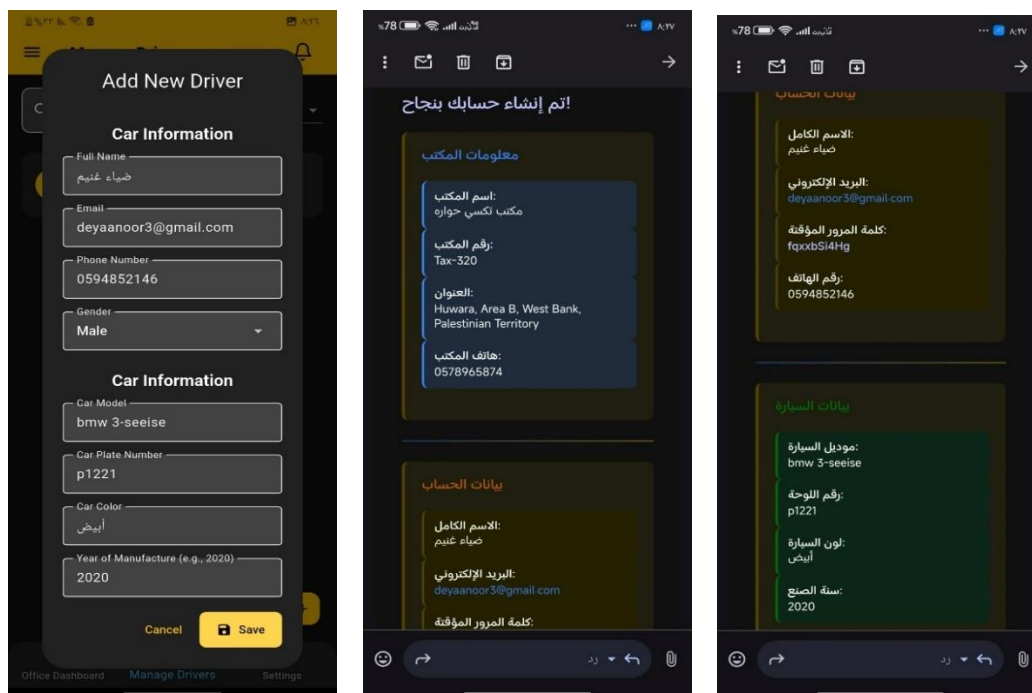


Figure30 Add new Driver

These features give the manager full control and visibility over their team of drivers, streamlining communication, oversight, and operational efficiency.

Chatting: The manager chat page enables real-time communication between the manager and their assigned drivers using WebSocket technology, ensuring instant message delivery. The interface displays a list of drivers on the side, allowing the manager to select and chat with any driver easily. Managers receive live push notifications for new messages, even when navigating other pages, and can view full chat history for reference. This feature supports fast, efficient coordination and improves operational management.

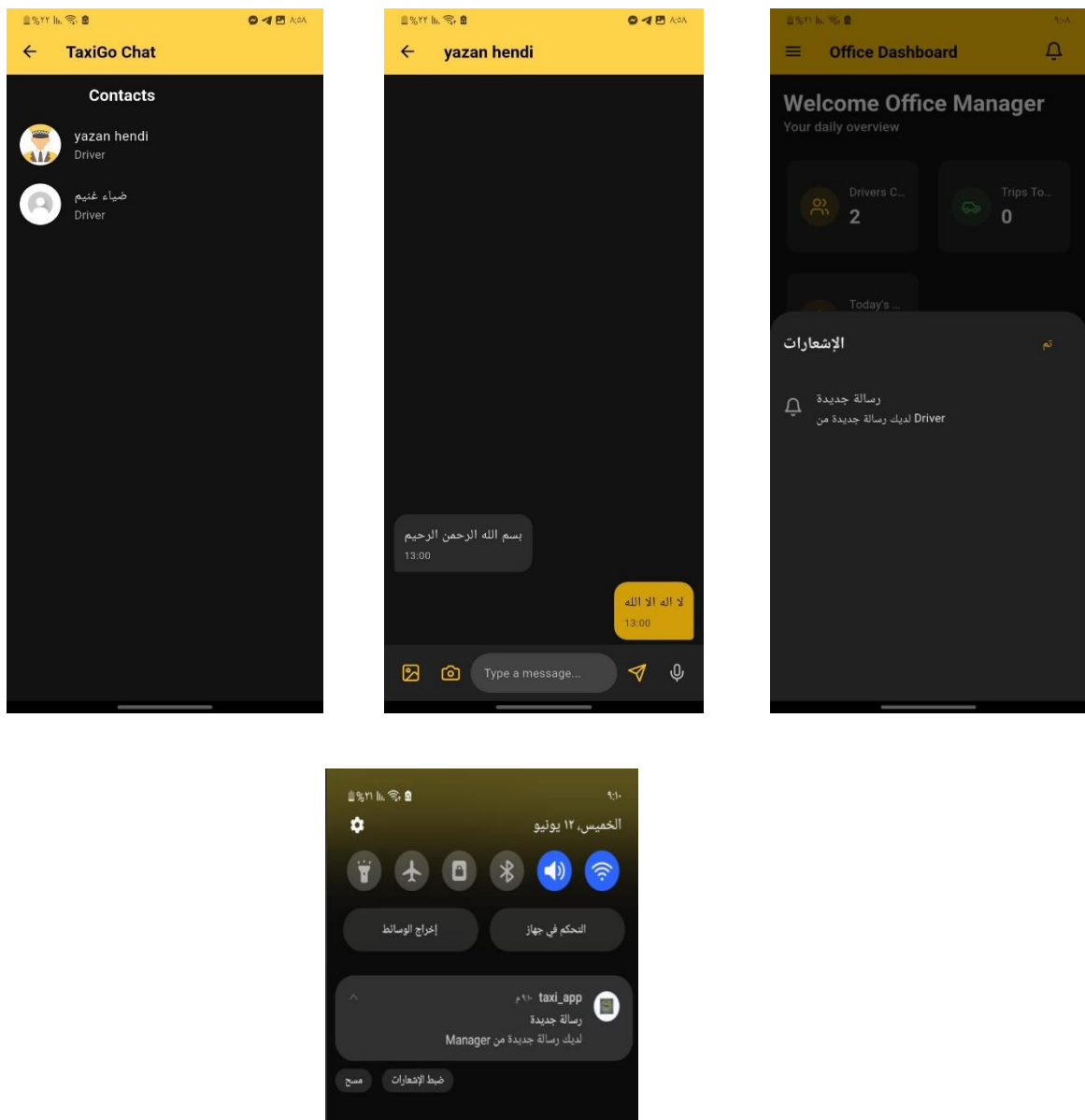


Figure 31 Chatting

3. **Settings Page – Manager Panel:** The Settings page in the Manager Panel provides several important customization and account management options:
 - **Theme Mode (Night Mode):** The manager can switch between light and dark mode to enhance visual comfort based on preference or lighting conditions.
 - **Language Selection:** Allows the manager to change the app language to their preferred one for better usability.
 - **Change Password:** Opens a secure form where the manager must enter their current password and confirm the new password twice before applying the change.
 - **Logout:** Provides a button to securely log out from the application and return to the login screen.

These settings ensure the manager has control over their user experience and account security.

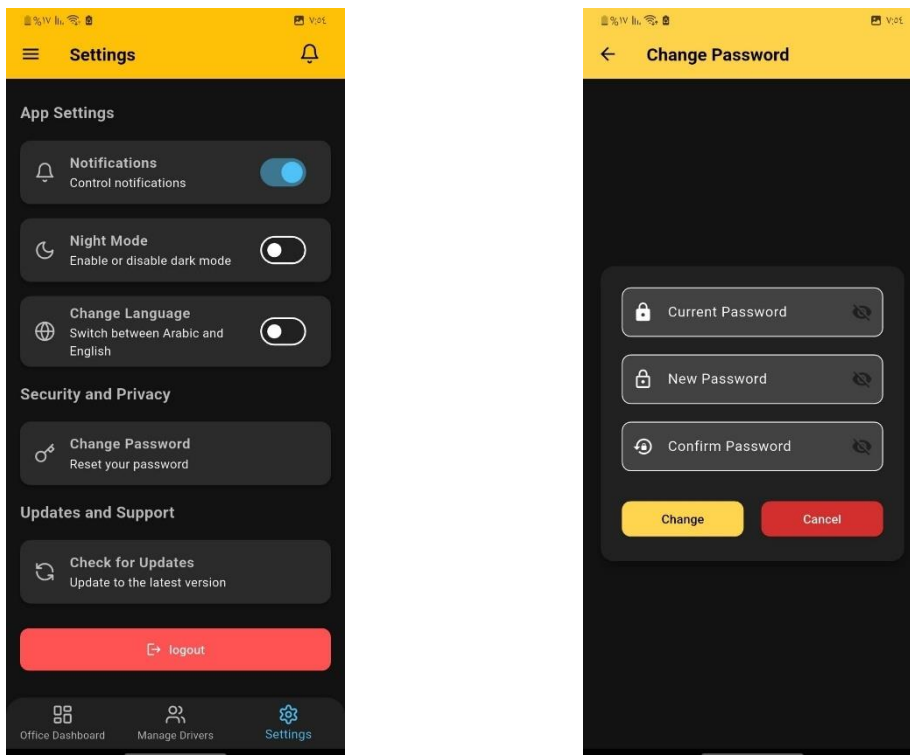


Figure 32 Settings Page – Manager Panel

1. **Admin Page:** When the admin logs into the system, they are directed to the Home Page, where the app bar includes a setting icon and a menu button that opens the Dashboard. The dashboard includes navigation options such as: Home, Drivers, Users, Trip Management, Payment Management, Settings, and Taxi Offices.

The Home Page provides daily statistics, including the number of completed trips, the number of available drivers, newly registered users, and total daily revenue. Additionally, there is a weekly report visualized through a bar chart, showing the number of trips completed each day of the week.

At the bottom of the page, a pie chart displays the percentage of active versus inactive drivers. A driver is considered active when they engage with the system by actions such as accepting trip requests. These visual tools allow the admin to monitor platform performance and driver engagement at a glance.

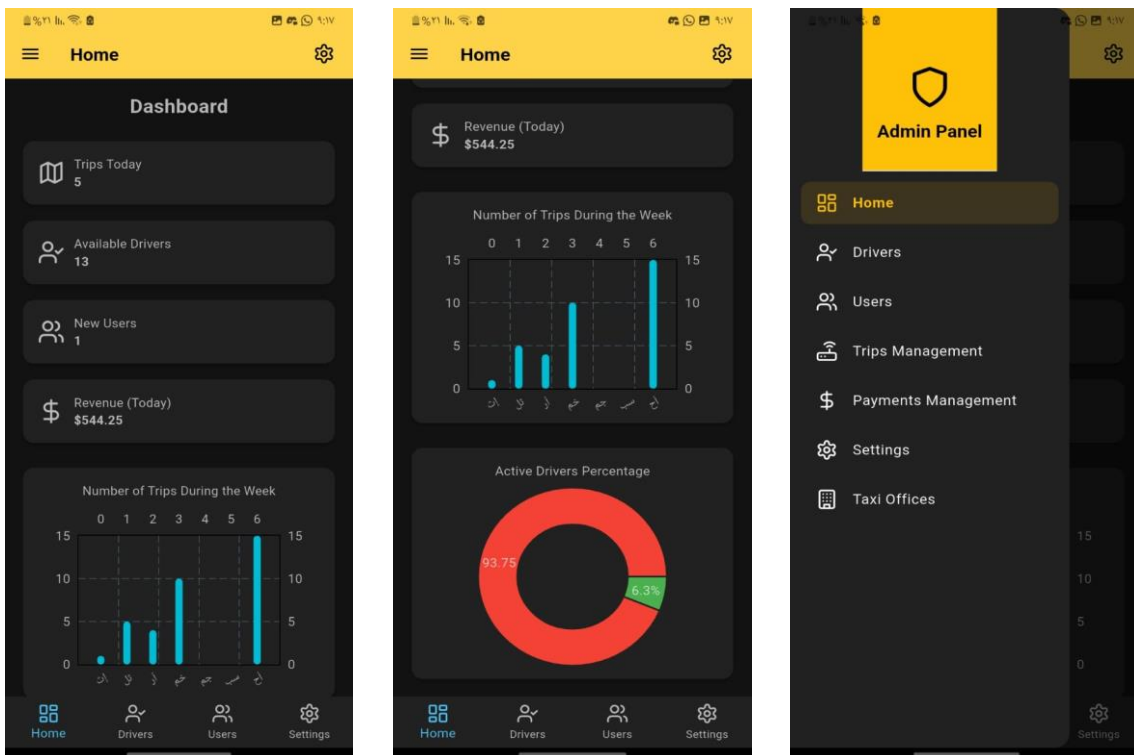


Figure 33 Admin Page

2. **Drivers Page:** The Drivers Page displays a list of all registered drivers in the system. Each driver is shown with basic information, and the admin can tap on any driver to view their full profile and details. From this page, the admin can manage the driver's availability status — enabling or disabling it as needed. Additionally, the admin has the ability to directly contact the driver through the app, making communication fast and efficient. This page gives the admin full control over monitoring and managing driver activity on the platform.

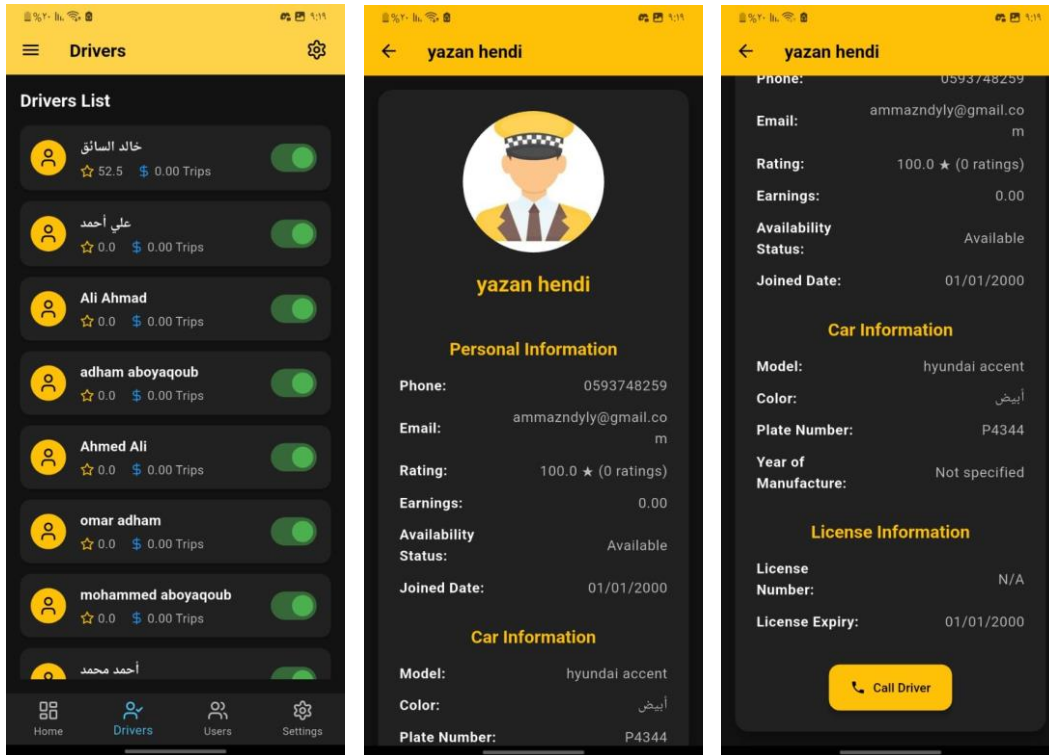


Figure 34 Drivers Page in Admin

3. **Users Page:** The Users Page displays a list of all registered users on the platform. Each user is shown with basic information, and the admin can tap on any user to view detailed profile information. Just like with drivers, the admin can directly contact the user through the app. This page allows the admin to monitor user activity, manage their status if needed, and ensure smooth communication with platform users.

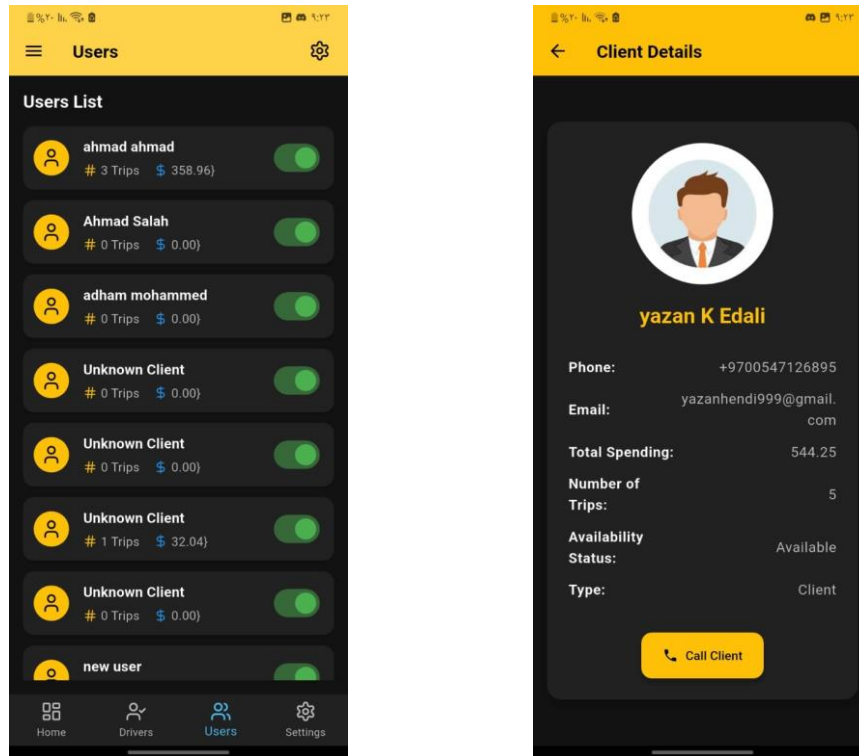


Figure 35 Users Page in Admin

4. **Trip Management:** The Trip Management page provides the admin with a comprehensive overview of all trips across the platform, regardless of their current status — whether Pending, Accepted, In Progress, or Completed. Each trip is listed with key summary details. The admin can click on any trip to open a popup window displaying full trip information, including the rider's and driver's details, pickup and drop-off locations, trip status, distance, and fare. This page helps the admin efficiently monitor, review, and manage trip activities across the system.

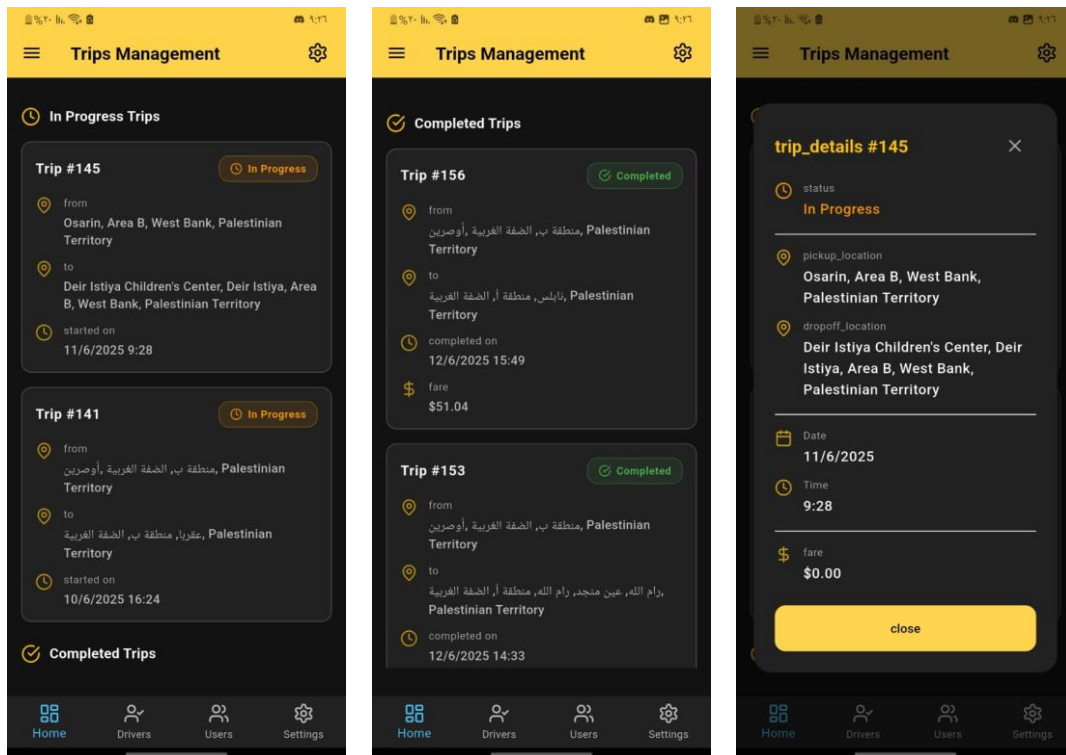


Figure 36 Trip Management

5. **Settings page:** The Settings page for the Admin includes several customization and account management options. The admin can switch between light and dark themes, change the application language, and update their account password by providing the current password and entering a new one. Additionally, the page offers a logout option to securely sign out of the system. These features ensure a personalized, flexible, and secure experience for the admin while using the application.

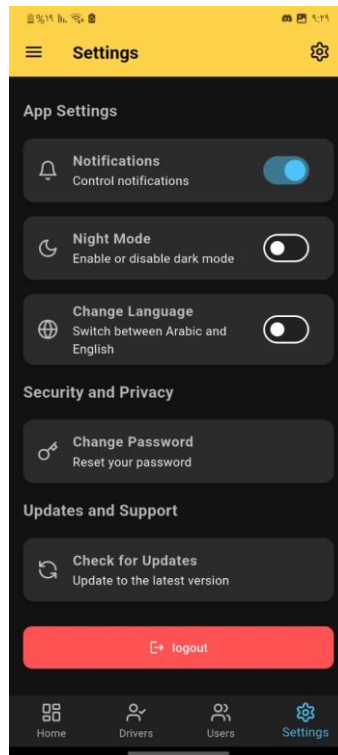


Figure 37 Settings page

6. **Taxi Offices page:** The Taxi Offices page allows the admin to view a list of all existing taxi offices within the system. Each office entry can be clicked to display detailed information about that office. Additionally, the admin has the ability to add a new taxi office. When adding a new office, the system requires the admin to select the office's location using an interactive map, where the location can be searched and marked precisely. Once a new taxi office is created, an automatic email is sent to the assigned manager containing their login credentials. This manager becomes responsible for managing the newly added taxi office.

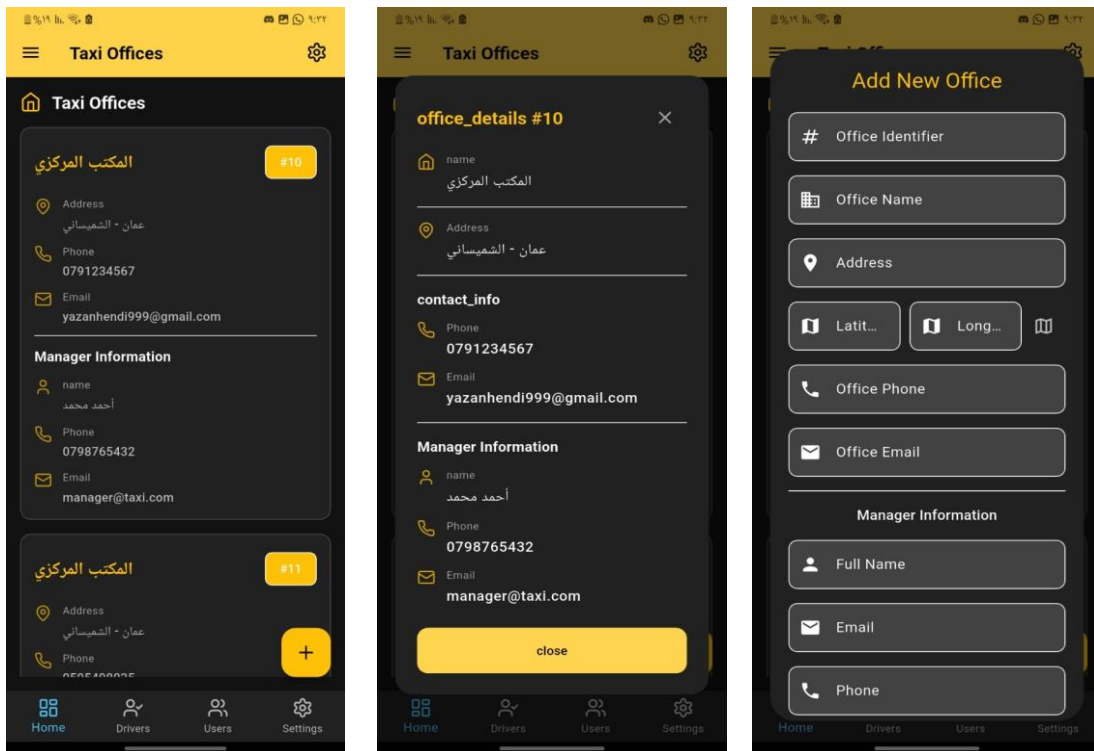
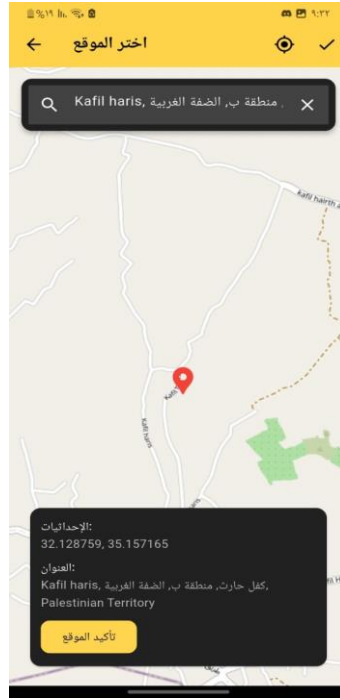
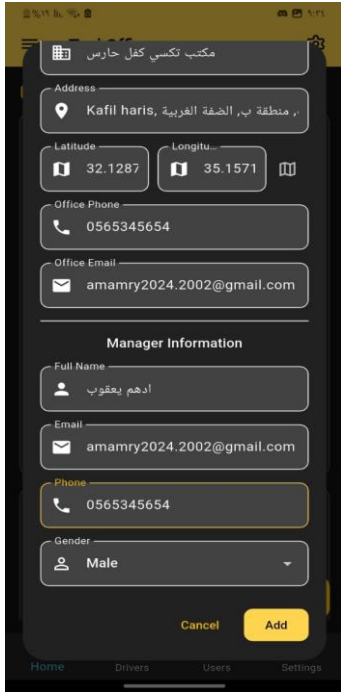


Figure 38 Taxi Offices page



This is the message I received by email:



4.1.12 Website:

Home Page:

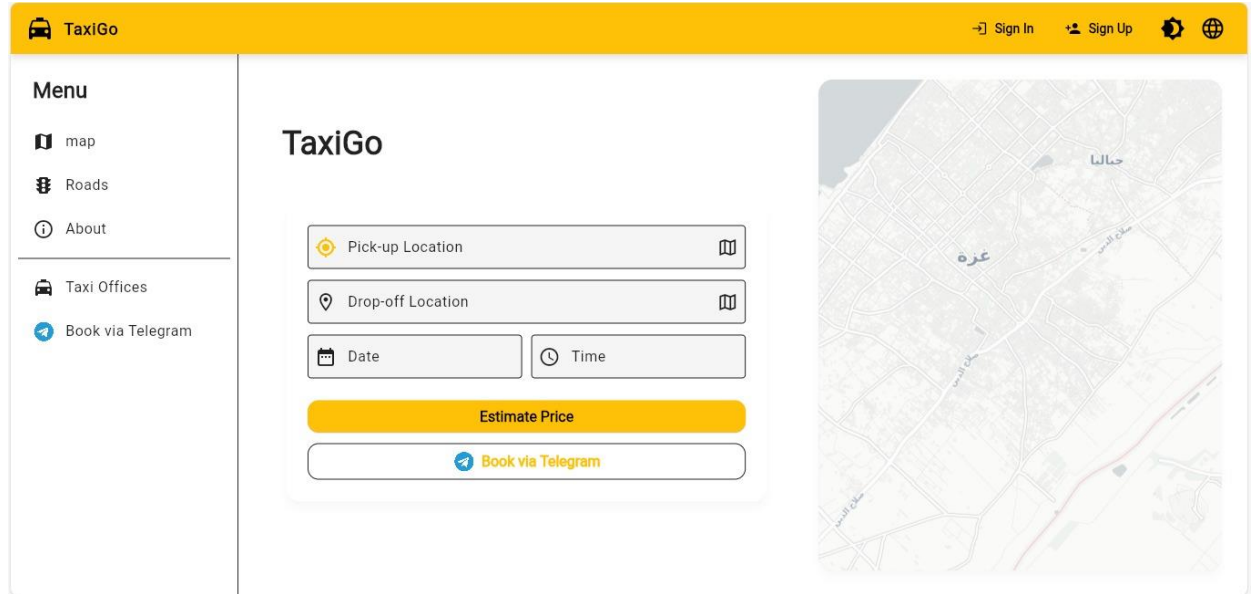
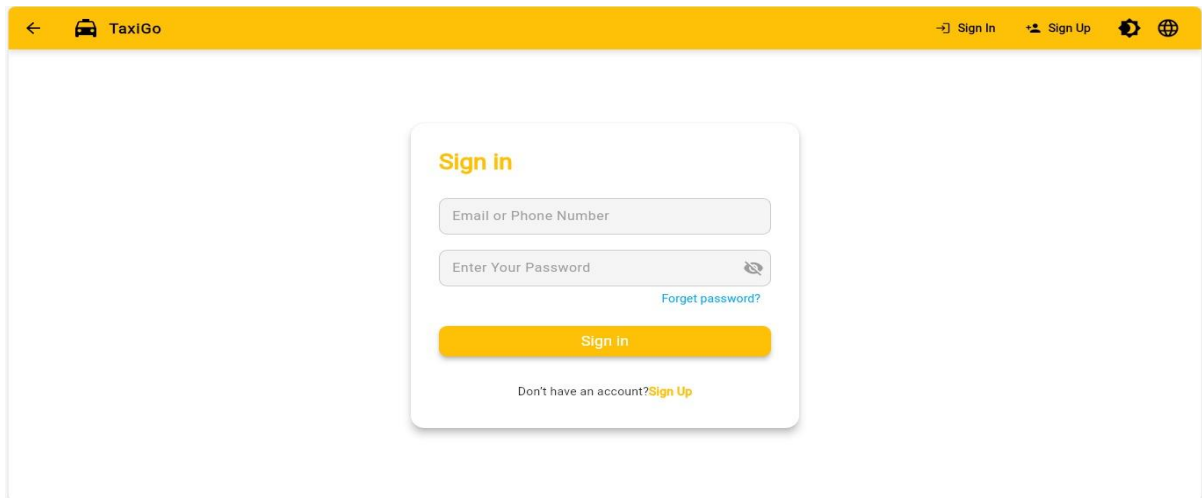


Figure 39 Home Page in Web

Sign in Page:



Forgot Password:

← TaxiGo Sign In Sign Up

Forgot Password

Enter your email address to receive a password reset link.

Email Address

Send Reset Link

Sign-up of User:

← TaxiGo Sign In Sign Up

Sign Up

I agree to the Terms & Conditions

Male

Confirm Password

Enter Your Password

Email

+1 Phone Number

Full Name

← TaxiGo Sign In Sign Up

In Sign Up must contain at least one spatial character and at least 8 character to accept the password:

A screenshot of the Taxigo mobile app sign-up screen. The form fields are: Name (ahmad ahmad), Phone (+970 0598457824), Email (yazanedali@gmail.com), Password (123), Confirm Password (123), Gender (Male), and a checked checkbox for 'I agree to the Privacy Policy'. A yellow 'Sign Up' button is at the bottom. A red error banner at the bottom reads: 'Password must be at least 8 characters long.'

A screenshot of the Taxigo mobile app sign-up screen. The form fields are: Name (ahmad ahmad), Phone (+970 0598457824), Email (yazanedali@gmail.com), Password (123456789), Confirm Password (123456789), Gender (Male), and a checked checkbox for 'I agree to the Privacy Policy'. A yellow 'Sign Up' button is at the bottom. A red error banner at the bottom reads: 'Password must contain at least one uppercase letter.'

A screenshot of the Taxigo mobile app sign-up screen. The form fields are: Name (ahmad ahmad), Phone (+970 0598457824), Email (yazanedali@gmail.com), Password (YAzanEdali123), Confirm Password (YAzanEdali123), Gender (Male), and a checked checkbox for 'I agree to the Privacy Policy'. A yellow 'Sign Up' button is at the bottom. A red error banner at the bottom reads: 'Password must contain at least one special character (e.g. !@#%\$).'

← TaxiGo Sign In Sign Up

ahmad ahmad

+970 0598457824

yazanedali@gmail.com

YAzanEdali

YAzanEdali

Male

I agree to the [Privacy Policy](#)

Sign Up

Password must contain at least one digit.

← TaxiGo Sign In Sign Up

ahmad ahmad

+970 0598457824

yazanedali@gmail.com

123456789Y

123456789Y

Male

I agree to the [Privacy Policy](#)

Sign Up

Password must contain at least one lowercase letter.

Sign-Up for Driver:

← Driver Sign Up

Driver Sign Up

Basic Information

Full Name

+1 Phone Number

Email

Enter Your Password

Confirm Password

Male

Driver vehicle information

License Number

License Expiry

Plate Number

Car Model

Car Color

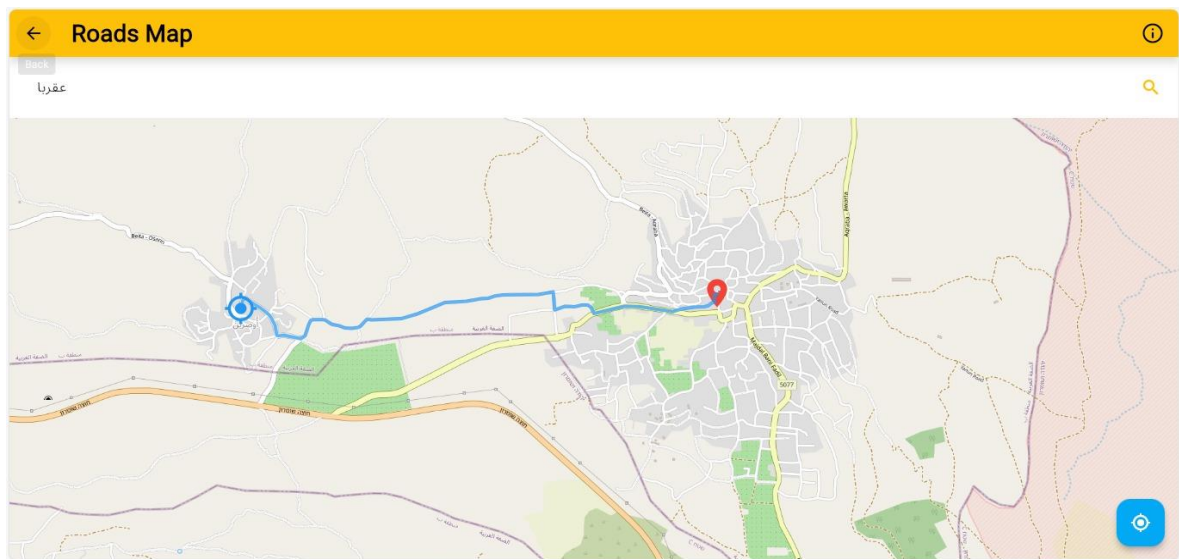
Taxi Office Number

I agree to the [Privacy Policy](#)

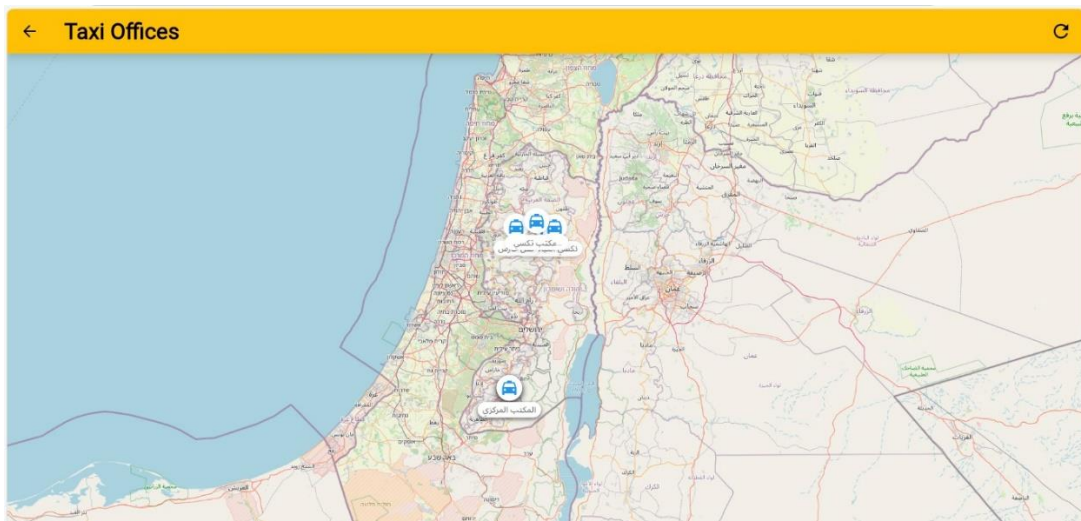
Sign Up

Already have an account? [Sign in](#)

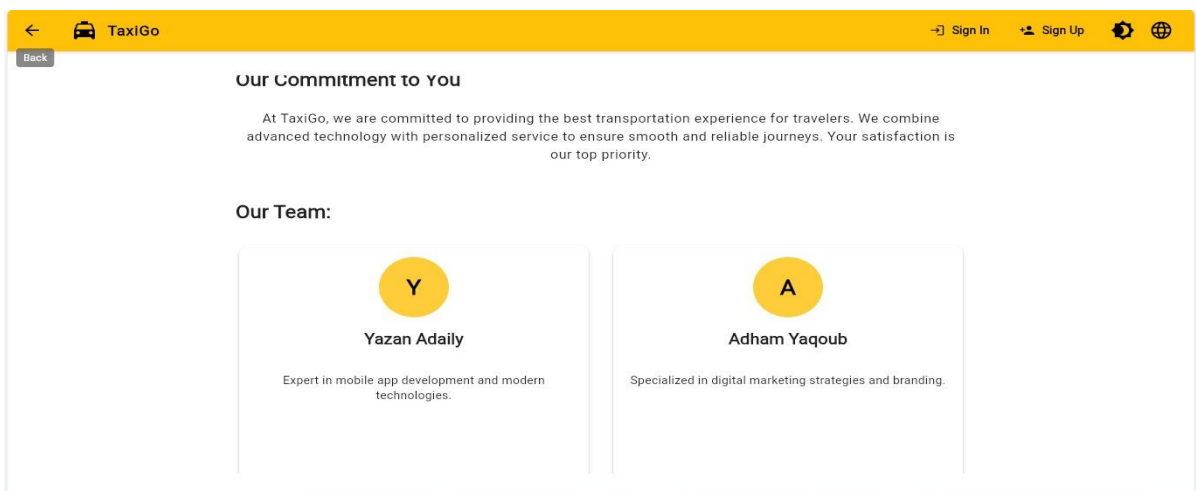
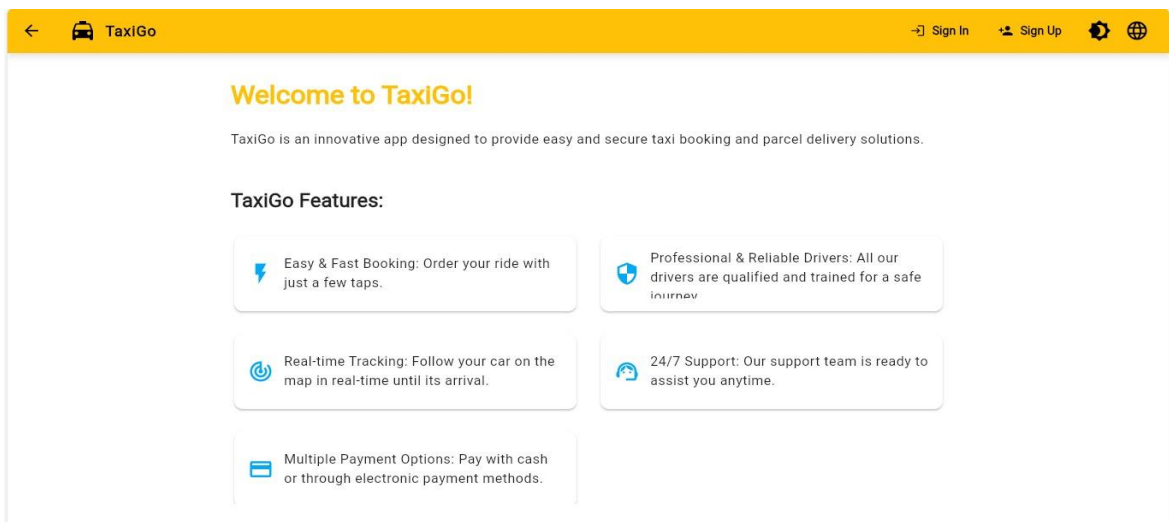
Map:



Taxi Office:



About Page:



Road Page:

← حالة الطرق والحوادث آخر تحديث: 10:10 م

حالة الطرق في رام الله

الطرق المزدحم (1)

⚠ طريق شارع المقاطعة -

الطرق المفتوح (1)

✅ طريق شارع الارسال -

🔄

Home Page of User:

user_dashboard

TaxiGo User
adham abo yaqoub

- Home
- New Scheduled Ride
- Trips History
- Drivers List
- Settings
- Support

New Ride Request

📍 Current Location
Osarin, Area B, West Bank, Palestinian Territory

🎯 Destination
Search for a location...

💰 Fare estimate:: **0.00 NIS**

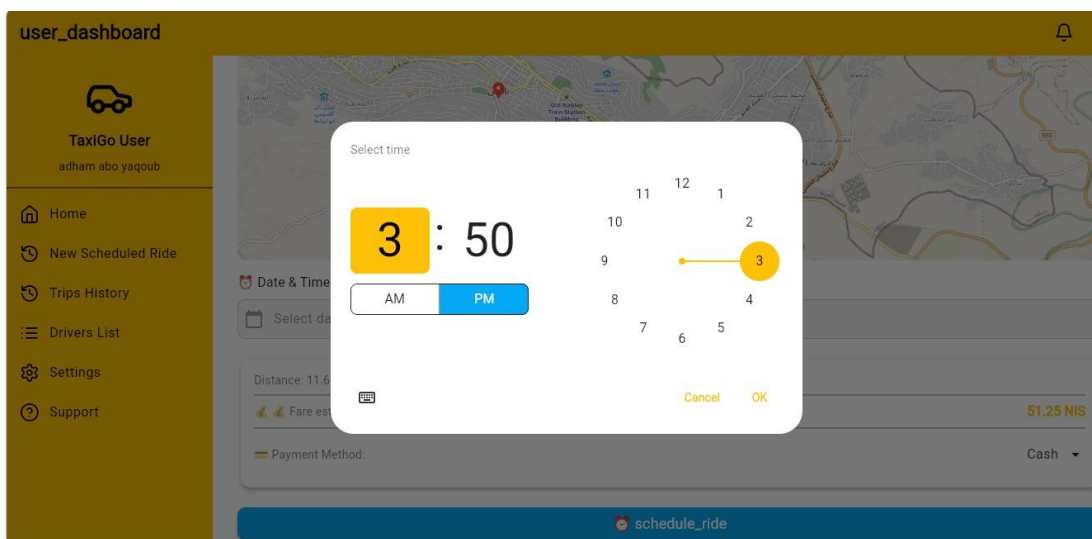
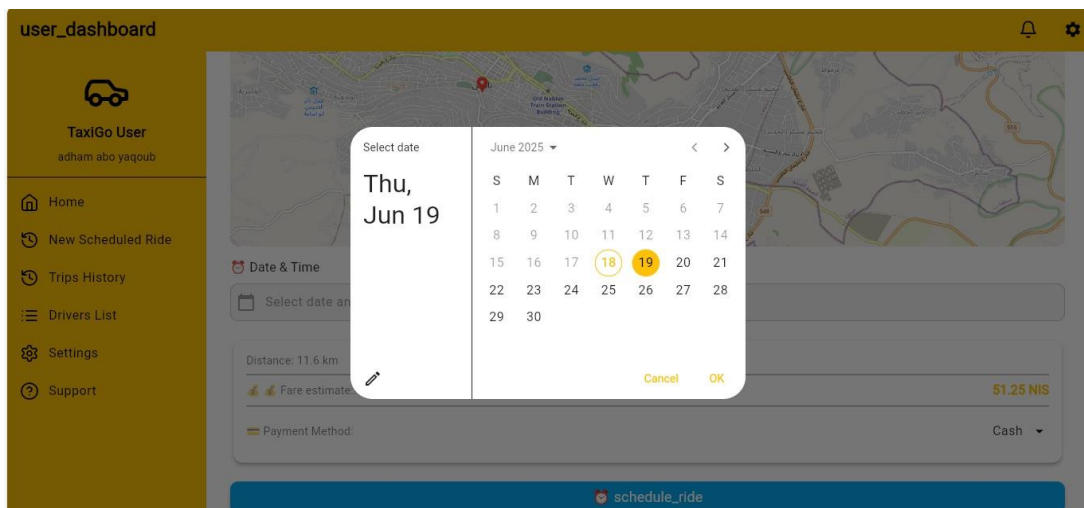
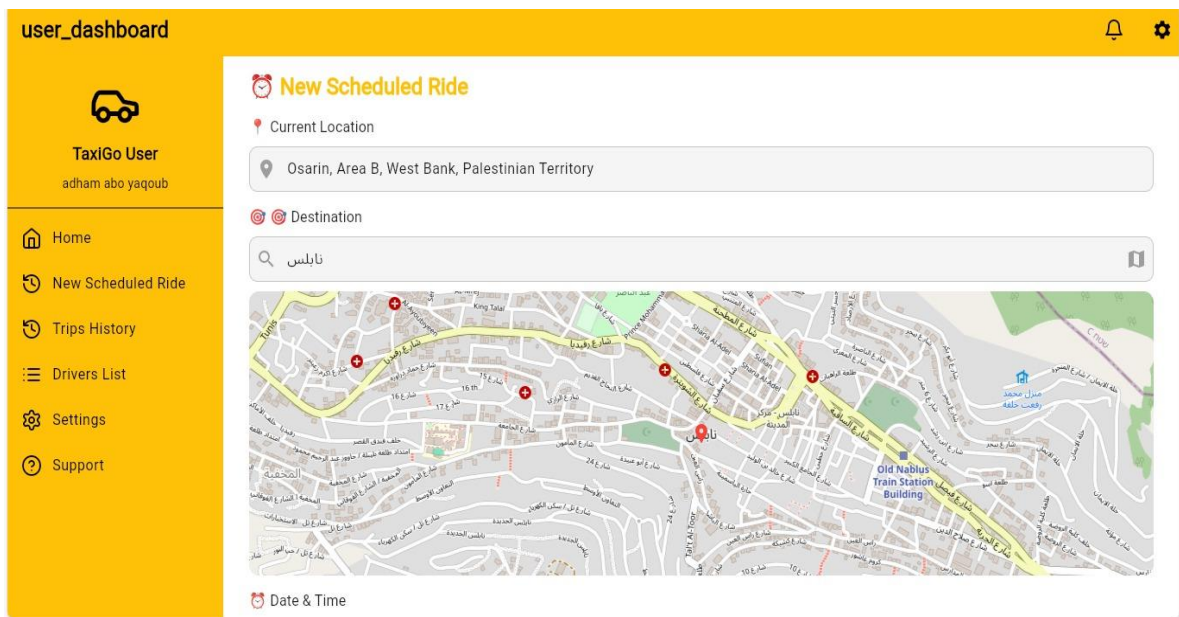
💳 Payment Method: Cash

🚖 Request a new ride

pending requests

from: كفل حارت, Palestinian Territory
to: نابلس

Schedules Ride:



Notification in web:

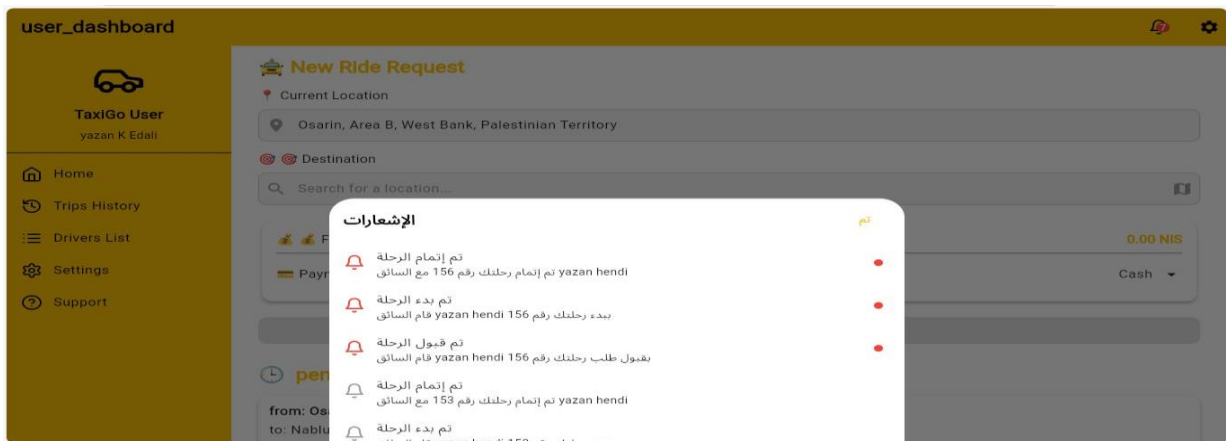


Figure 40 Notification in web

Trip History Table in Web:

user_dashboard

TaxiGo User
yazan K Edali

My Trips

Trip ID	Route	Date	fare	status
#156	أوصرين ↓ نابلس	12/6/2025 15:49	\$51.04	Completed
#153	أوصرين ↓ رام الله	12/6/2025 14:33	\$121.99	Completed
#152	أوصرين ↓ الخليل	12/6/2025 14:32	\$307.59	Completed
#151	أوصرين ↓ بيتا	12/6/2025 14:32	\$12.58	Completed

Pop-Up to show details:

user_dashboard

TaxiGo User
yazan K Edali

trip_details #156

- status: Completed
- pickup_location: منطقة ب. الضفة الغربية، أوصرين, Palestinian Territory
- dropoff_location: منطقة أ. الضفة الغربية، نابلس, Palestinian Territory
- Date: 12/6/2025
- Time: 15:49
- end time: 15:50

Trip ID	Route	Date	fare	status
#151	أوصرين ↓ بيتا	12/6/2025 14:32	\$12.58	Completed

Driver List:

The screenshot shows the 'user_dashboard' interface with a yellow header and a sidebar on the left. The sidebar contains a 'TaxiGo User' profile for 'yazan K Edali' and navigation links for Home, Trips History, Drivers List, Settings, and Support. The main content area displays a grid of six driver profiles, each with a profile picture, name, phone number, rating, and vehicle details.

Driver Name	Phone Number	Rating	Vehicle	Plate Number
خالد السائق	962781122334	53/100	Toyota Prius (أبيض)	15-12345
علي أحمد	0599876543	0/100	Kia Cerato (أبيض)	ABC123
Ali Ahmad	0591234561	0/100	Toyota Prius (White)	GA1234
adham aboyaqoub	0594348312	0/100	N/A (N/A)	N/A
Ahmed Ali	025852014	0/100	Toyota Camry (Red)	12345
omar adham	594348313	0/100	N/A (N/A)	N/A

Settings:

The screenshot shows the 'Settings' page in the 'user_dashboard'. The sidebar is identical to the previous page. The main content area is titled 'Settings' and is divided into two sections: 'account_personal_info' and 'App Settings'. The 'account_personal_info' section includes 'Edit Profile' and 'Change Password' options. The 'App Settings' section includes 'dark_mode' and 'Change Language' options, each with a toggle switch. A red 'logout' button is located at the bottom of the settings area.

Section	Setting	Value/Action
account_personal_info	Edit Profile	>
	Change Password	>
App Settings	dark_mode	Off
	Change Language	Off

← Change Password

Back

Current Password
.....

New Password
123456
Password must be at least 8 characters long.

Confirm Password
123456

Change Cancel

← Change Password

Back

Current Password
.....

New Password
123456789
Password must contain at least one uppercase letter.

Confirm Password
123456789

Change Cancel

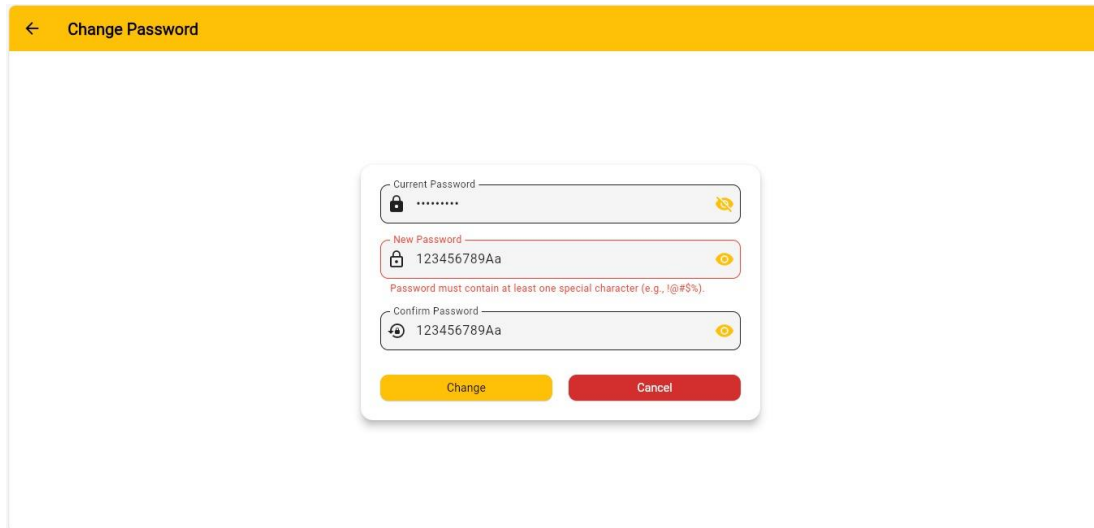
← Change Password

Current Password
.....

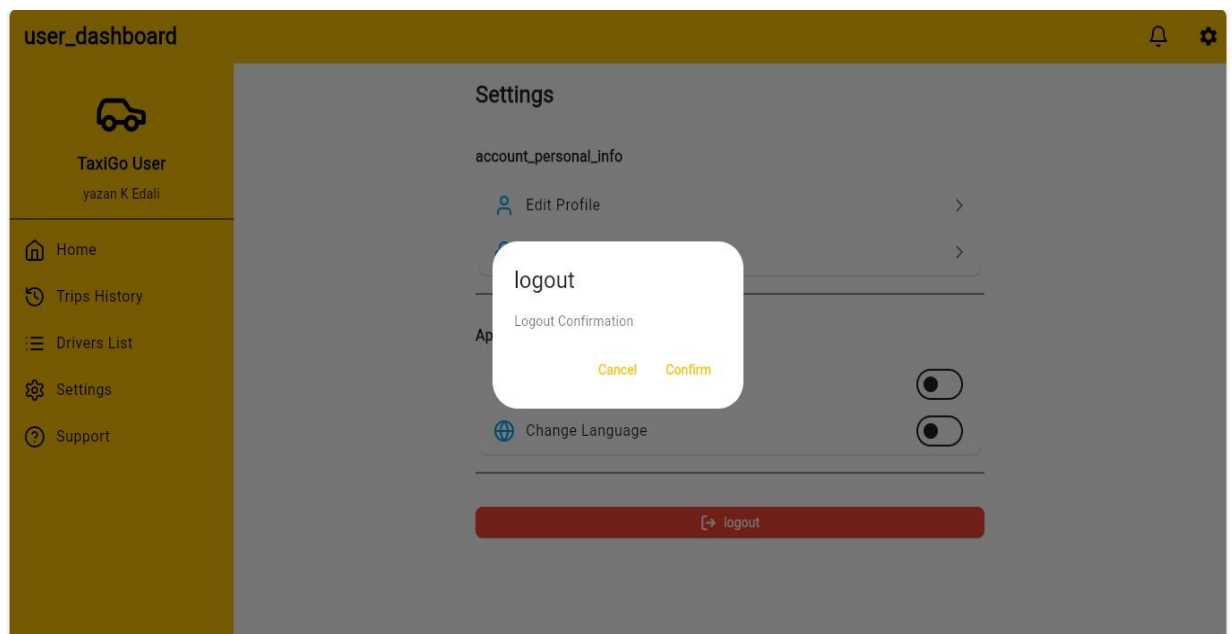
New Password
123456789A
Password must contain at least one lowercase letter.

Confirm Password
123456789A

Change Cancel




Logout:



Home Page of Driver:

Home


TaxiGo Driver
yazan hendi

Home


trip requests

My Trips

Earnings

Settings

Chat

 **yazan hendi**


Car Model: hyundai accent | Plate Number: P4344 | Rating: 100.0

Recent Trips

Trip #156	from: منطقة ب, الضفة الغربية, أوسرين, فلسطين to: نابلس, منطقة أ, الضفة الغربية, فلسطين Territory	\$51.04
Trip #153	from: منطقة ب, الضفة الغربية, أوسرين, فلسطين to: رام الله, عين منجد, رام الله, منطقة أ, الضفة الغربية, فلسطين Territory	\$121.99
Trip #152	from: منطقة ب, الضفة الغربية, أوسرين, فلسطين to: H1, 150, الضفة الغربية, H1, الخليل, أوسر, فلسطين Territory	\$307.59

Trip Requests:

trip requests


TaxiGo Driver
yazan hendi

Home

trip requests

My Trips

Earnings

Settings

Chat

pending requests

accepted trips

Trip #157

from: Osarin, Area B, West Bank, Palestinian Territory

to: Nablus, Area A, West Bank, Palestinian Territory

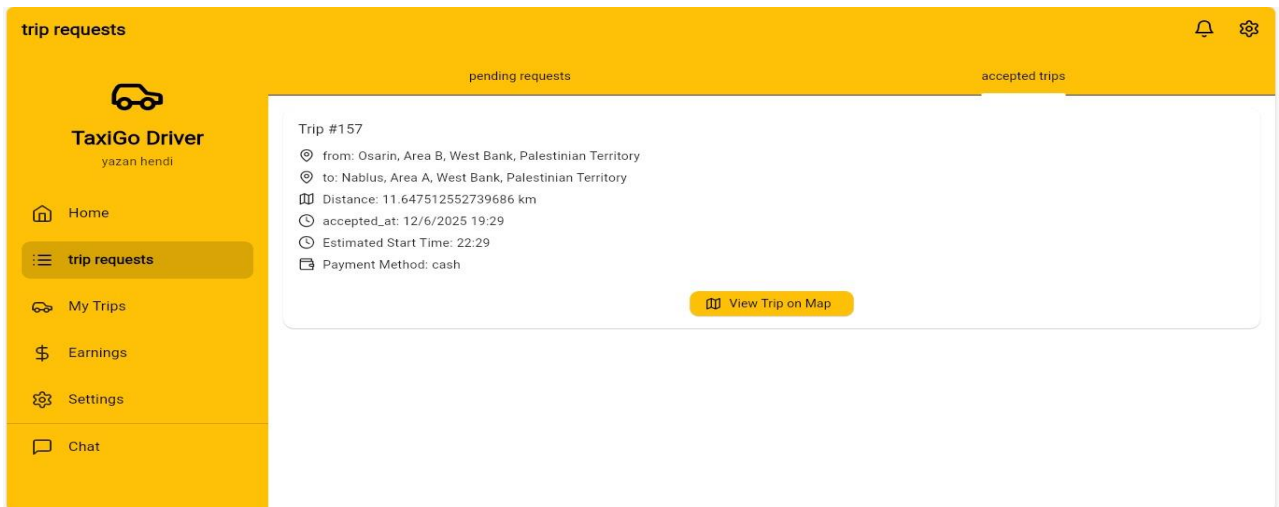
Distance: 11.647512552739686 km

requested at: 12/6/2025 19:17

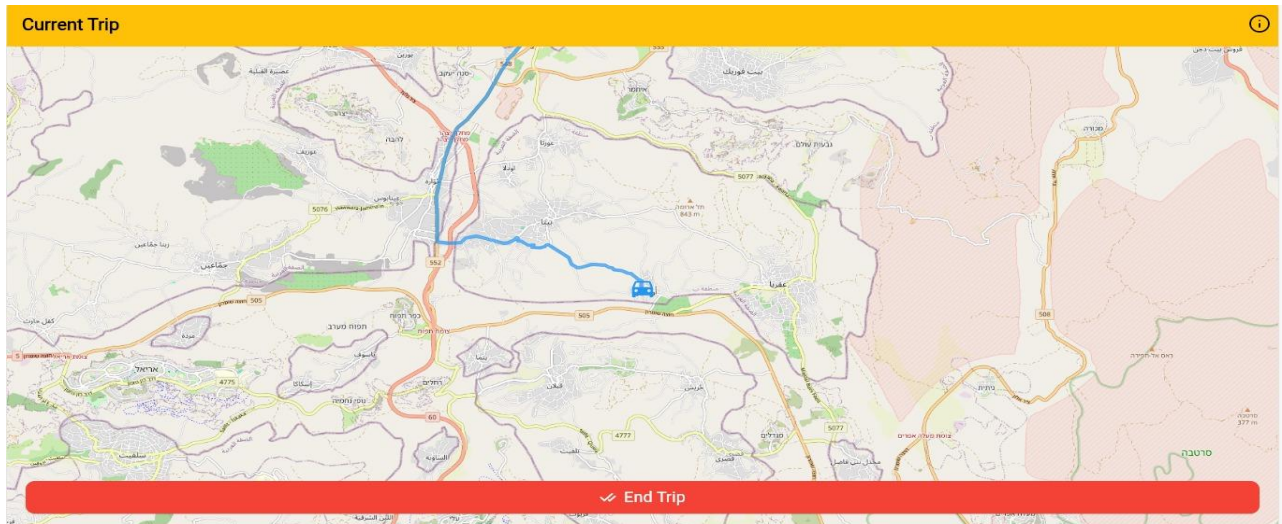
Payment Method: cash

تم العثور على 1 رحلة ضمن 5 كم

Accepted Trip:



Show real time tracking in Map:



Trip in Web for Driver:

The screenshot shows the 'My Trips' interface for a driver. The left sidebar contains navigation options: Home, trip requests, My Trips (selected), Earnings, Settings, and Chat. The main content area displays a table of completed trips.

Trip ID	Route	Date	fare	status
#147	أوصرين ↓ بيتا	11/6/2025 19:28	\$12.58	Completed
#148	أوصرين ↓ بيتا	11/6/2025 19:49	\$12.59	Completed
#150	أوصرين ↓ نابلس	12/6/2025 14:31	\$51.04	Completed
#151	أوصرين ↓ بيتا	12/6/2025 14:32	\$12.58	Completed

Figure 41 Trip in Web for Driver

Earnings Page:

The screenshot shows the 'Earnings' interface for a driver. The left sidebar contains navigation options: Home, trip requests, My Trips, Earnings (selected), Settings, and Chat. The main content area displays the total earnings and a list of earnings details.

Total Earnings

Total Earnings: **\$569.42**

Earnings Details

Trip #147 Date: 2025-06-11 Distance: 2.859689262710803 km	\$12.58
Trip #148 Date: 2025-06-11 Distance: 2.8608934266907755 km	\$12.59
Trip #150 Date: 2025-06-12 Distance: 11.599543347158855 km	\$51.04
Trip #151 Date: 2025-06-12 Distance: 2.860161513339031 km	\$12.58

Setting Page:

Settings

TaxiGo Driver
yazan hendi

Home

trip requests

My Trips

Earnings

Settings

Chat

Driver Settings

Account Settings

Edit Profile

Notifications

Display Settings

Night Mode

Change Language

Security Settings

Change Password

Chat with Manager Page:

ادهم يعقوب

Contacts

ادهم يعقوب
Manager

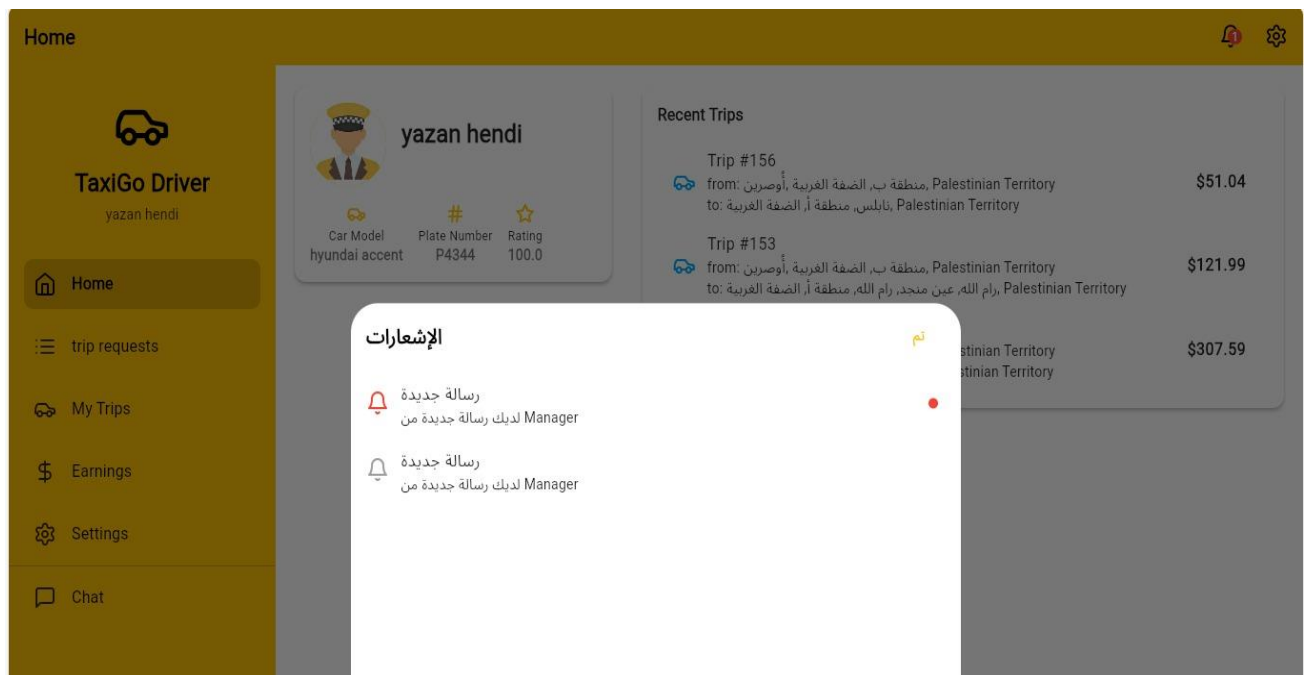
لا اله الا الله
13:00

مرحبا
21:10

بسم الله الرحمن الرحيم
13:00

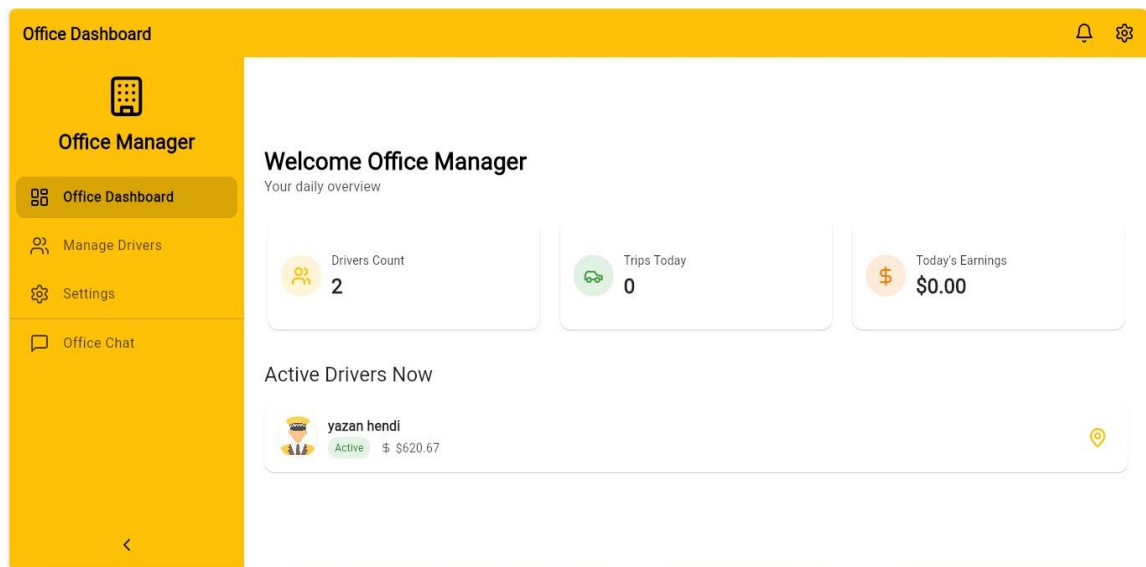
Type a message...

Notification of message for Driver:



Manager Page:

Office Manager:



Manager Driver:

name	Phone	status	details	call	status_change
yazan hendi	0593748259	Active			<input checked="" type="checkbox"/>
ضياء غنيم	0594852146	Inactive			<input type="checkbox"/>

Details of driver:

← yazan hendi

Personal Information

Phone: 0593748259
Email: ammazdyb@gmail.com
Rating: 100 ⭐ (0 ratings)
Earnings: 620.67
Availability Status: Available
Joined Date: 11/06/2025

Car Information

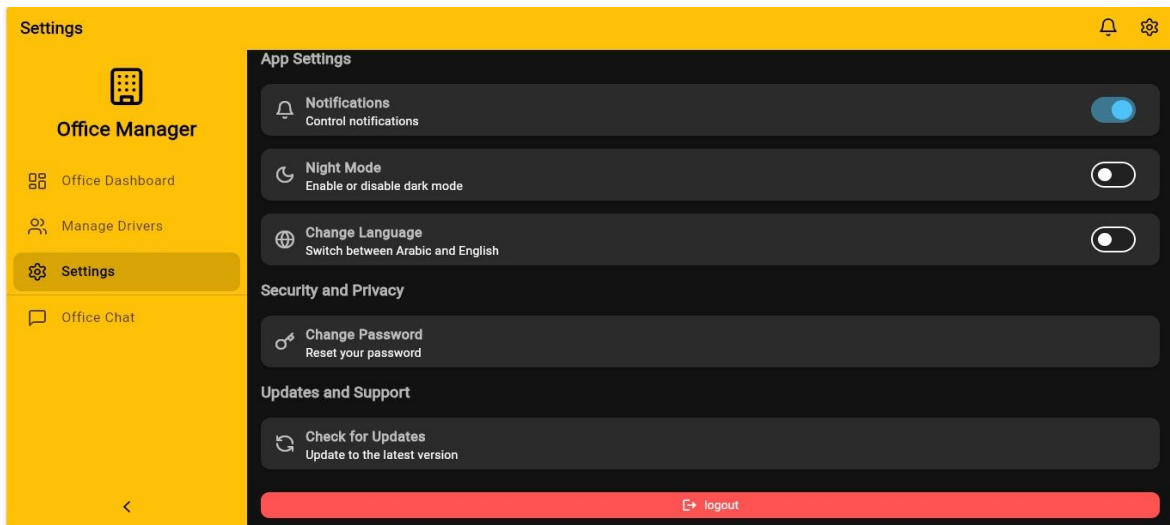
Model: Hyundai accent
Color: ابيض
Plate Number: PA344
Year of Manufacture: 2018

License Information

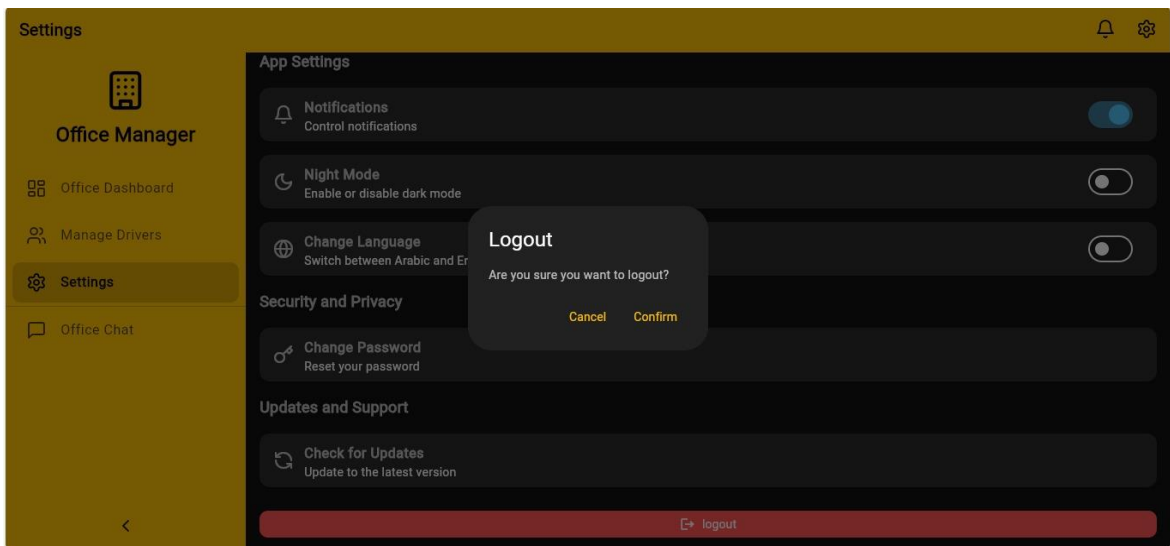
License Number: 15896
License Expiry: 01/01/2030

[Call Driver](#)

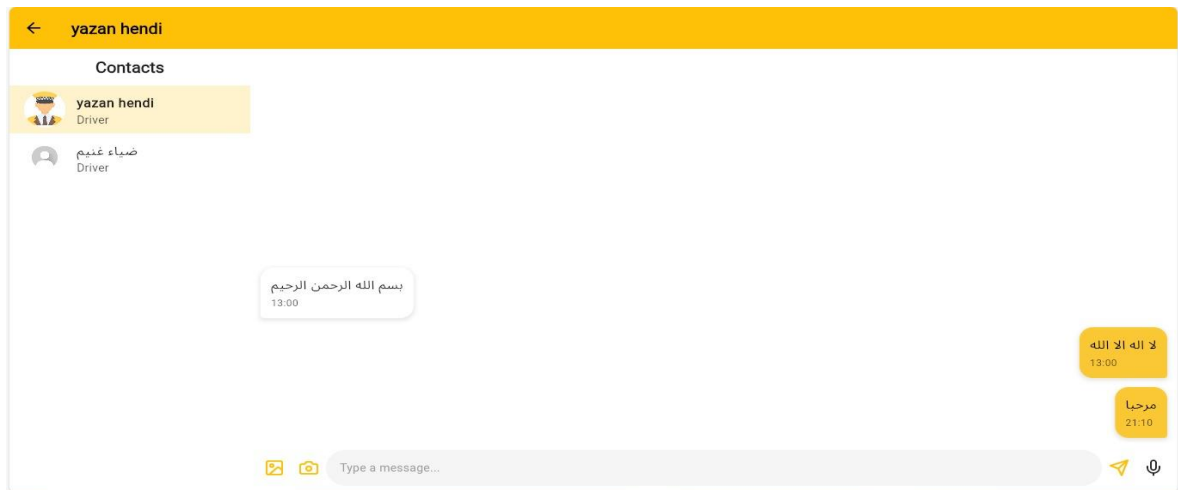
Settings:



Logout:



Chatting for Driver:



Admin Page:

Home Page:

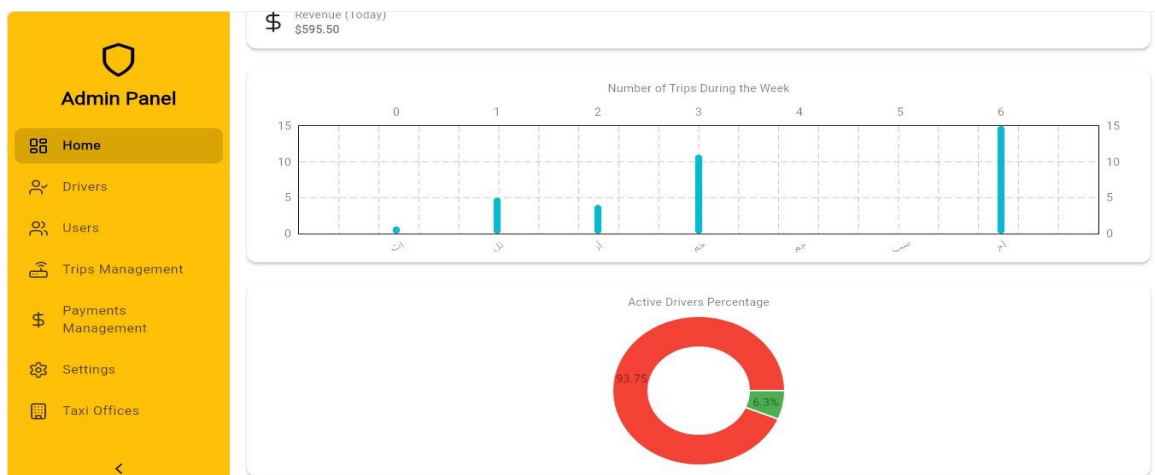
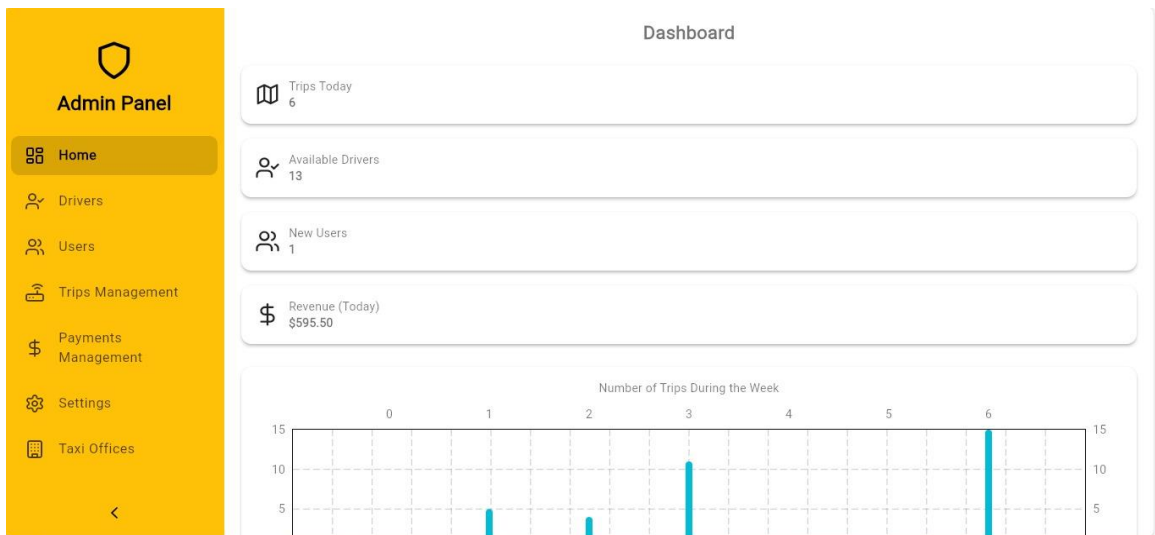










Figure 42 Admin Page in Web

Driver List:

Admin Panel		Drivers List
Home	Drivers	<div>  خالد السائق ☆ 52.5 \$ 0.00 Trips </div>
Users	Drivers	<div>  علي أحمد ☆ 0.0 \$ 0.00 Trips </div>
Trips Management	Drivers	<div>  Ali Ahmad ☆ 0.0 \$ 0.00 Trips </div>
Payments Management	Drivers	<div>  adham aboyaqoub ☆ 0.0 \$ 0.00 Trips </div>
Settings	Drivers	<div>  Ahmed Ali ☆ 0.0 \$ 0.00 Trips </div>
Taxi Offices	Drivers	<div>  omar adham ☆ 0.0 \$ 0.00 Trips </div>
	Drivers	<div>  mohammed aboyaqoub ☆ 0.0 \$ 0.00 Trips </div>

Driver details:

←
خالد السائق



خالد السائق

Personal Information

Phone:	952781122334
Email:	ahmed.driver@gmail.com
Rating:	32.5 (0 ratings)
Earnings:	0.00
Availability Status:	Available
Joined Date:	01/01/2000

Car Information








Model:	Toyota Prius
Color:	أبيض
Plate Number:	15-12243
Year of Manufacture:	Not specified

License Information

License Number:	N/A
License Expiry:	01/01/2000

[Car driver](#)

User List:

Admin Panel		Users List
Home	Users	<div>  ahmad ahmad # 3 Trips \$ 358.96 </div>
Drivers	Users	<div>  Ahmad Salah # 0 Trips \$ 0.00 </div>
Users	Users	<div>  adham mohammed # 0 Trips \$ 0.00 </div>
Trips Management	Users	<div>  Unknown Client # 0 Trips \$ 0.00 </div>
Payments Management	Users	<div>  Unknown Client # 0 Trips \$ 0.00 </div>
Settings	Users	<div>  Unknown Client # 1 Trips \$ 32.04 </div>
Taxi Offices	Users	<div>  Unknown Client # 0 Trips \$ 0.00 </div>

User details:

Client Details

new user

Phone: +9700525750448
 Email: edaliyazan6@gmail.com
 Total Spending: 718.07
 Number of Trips: 10
 Availability Status: Available
 Type: Client

Call Client

Trip Management:

Admin Panel

- Home
- Drivers
- Users
- Trips Management**
- Payments Management
- Settings
- Taxi Offices

All Trips

Trip ID	Route	Date	fare	status
#145	Osarin ↓ Deir Istiya Children's Center	11/6/2025 9:28	\$0.00	In Progress
#141	أوسرين ↓ عقرية	10/6/2025 16:24	\$32.45	In Progress
#157	Osarin ↓ Nablius	12/6/2025 19:29	\$51.25	Completed
#156	أوسرين ↓ نابلس	12/6/2025 15:49	\$51.04	Completed
#153	أوسرين	12/6/2025	\$121.99	Completed

Figure 43 Trip Management in Web

Admin Panel

- Home
- Drivers
- Users
- Trips Management**
- Payments Management
- Settings
- Taxi Offices

All Trips

trip_details #141

status: In Progress

pickup_location: منطقة ب. الضفة الغربية، أوسرين, Palestinian Territory

dropoff_location: عقرية، منطقة ب. الضفة الغربية, Palestinian Territory

Date: 10/6/2025

Time: 16:24

fare: \$32.45

close

Trip ID	Route	Date	fare	status
#152	أوسرين ↓ الجيل	12/6/2025 14:32	\$307.59	Completed

Payments:

The screenshot displays the 'Payments Management' section of an admin panel. On the left is a yellow sidebar with navigation options: Home, Drivers, Users, Trips Management, Payments Management (highlighted), Settings, and Taxi Offices. The main content area is titled 'Payments Management' and contains a 'Transaction History' table. The table has columns for Trip ID, User, Driver, Amount, Payment Method, Date, and status. The status column shows 'Completed' for all entries, with a toggle icon to the right of each status label.

Trip ID	User	Driver	Amount	Payment Method	Date	status
#157	yazan K Edali	yazan hendi	\$51.25	cash	12/6/2025 19:30	Completed
#156	yazan K Edali	yazan hendi	\$51.04	cash	12/6/2025 15:50	Completed
#153	yazan K Edali	yazan hendi	\$121.99	cash	12/6/2025 14:33	Completed
#152	yazan K Edali	yazan hendi	\$307.59	cash	12/6/2025 14:32	Completed
#151	yazan K Edali	yazan hendi	\$12.58	cash	12/6/2025 14:32	Completed
#150	yazan K Edali	yazan hendi	\$51.04	cash	12/6/2025 14:31	Completed
#148	new user	yazan hendi	\$12.59	cash	11/6/2025 19:49	Completed

Figure 44 Payments Table in Web

Its details:

This screenshot shows the same 'Payments Management' interface as above, but with a modal window open for 'Transaction Details #157'. The modal is white with a close button (X) in the top right corner. It lists the following details: User (yazan K Edali), Driver (yazan hendi), Payment Method (cash), Amount (\$51.25), and transaction_date_time (12/6/2025 19:30). The background table is dimmed.

Trip ID	User	Driver	Amount	Payment Method	Date	status
#157	yazan K Edali	yazan hendi	\$51.25	cash	12/6/2025 19:30	Completed
#156	yazan K Edali	yazan hendi	\$51.04	cash	12/6/2025 15:50	Completed
#153	yazan K Edali	yazan hendi	\$121.99	cash	12/6/2025 14:33	Completed
#152	yazan K Edali	yazan hendi	\$307.59	cash	12/6/2025 14:32	Completed
#151	yazan K Edali	yazan hendi	\$12.58	cash	12/6/2025 14:32	Completed
#150	yazan K Edali	yazan hendi	\$51.04	cash	12/6/2025 14:31	Completed
#148	new user	yazan hendi	\$12.59	cash	11/6/2025 19:49	Completed

Settings:

Admin Panel

- Home
- Drivers
- Users
- Trips Management
- Payments Management
- Settings**
- Taxi Offices

App Settings

- Notifications**
Control notifications
- Night Mode**
Enable or disable dark mode
- Change Language**
Switch between Arabic and English

Security and Privacy

- Change Password**
Reset your password

Updates and Support

- Check for Updates**
Update to the latest version

[logout](#)

Taxi Offices:

Admin Panel

- Home
- Drivers
- Users
- Trips Management
- Payments Management
- Settings
- Taxi Offices**

Taxi Offices

id	name	Address	Phone
#10	المكتب المركزي	عمان - الشميساني	0791234567
#11	المكتب المركزي	عمان - الشميساني	0595498035
#13	المكتب المركزي	عمان - الشميساني	0599498035
#14	المكتب المركزي	عمان - الشميساني	0599498034
#15	المكتب المركزي	عمان - الشميساني	0593498034
#16	المكتب المركزي	عمان - الشميساني	0593498034
#17	المكتب المركزي	عمان - الشميساني	0593498034
#18	المكتب المركزي	Osareen	0593498034

[+](#)

Figure 45 Taxi Offices in Web

Admin Panel

- Home
- Drivers
- Users
- Trips Management
- Payments Management
- Settings
- Taxi Offices**

Taxi Offices

office_details #10

name
المكتب المركزي

Address
عمان - الشميساني

contact_info

Phone
0791234567

Email
yazanhendi999@gmail.com

Manager Information

name
أحمد محمد

id	name	Address	Phone
#10	المكتب المركزي	عمان - الشميساني	0791234567
#11	المكتب المركزي	عمان - الشميساني	0595498035
#13	المكتب المركزي	عمان - الشميساني	0599498035
#14	المكتب المركزي	عمان - الشميساني	0599498034
#15	المكتب المركزي	عمان - الشميساني	0593498034
#16	المكتب المركزي	عمان - الشميساني	0593498034
#17	المكتب المركزي	عمان - الشميساني	0593498034
#18	المكتب المركزي	Osareen	0593498034

[+](#)

Add new Office:

Add New Office

Office Identifier

Office Name

Address

Latitude Longitude

Office Phone

Office Email

Manager Information

Full Name

Email

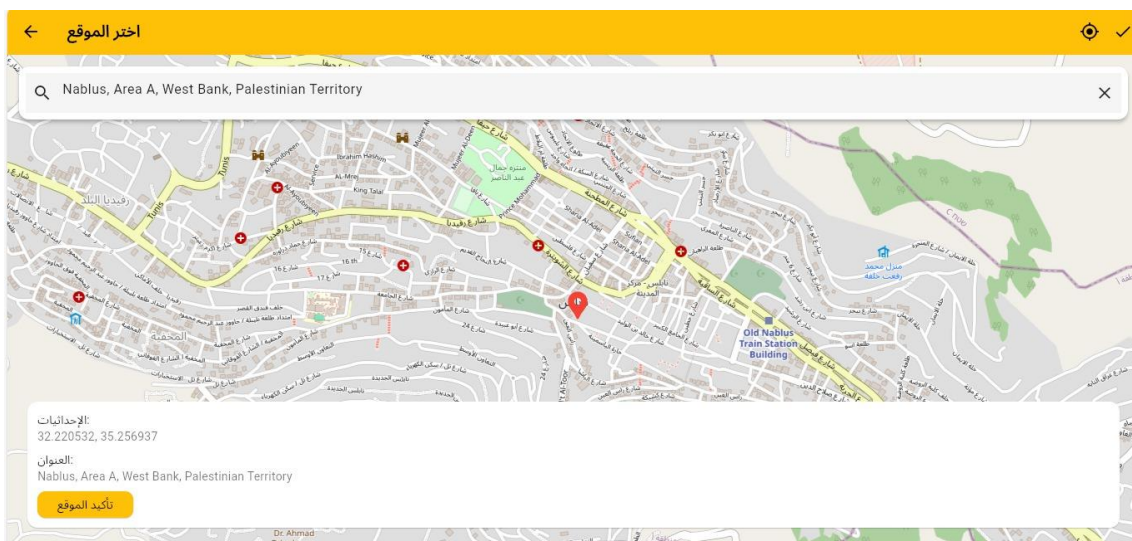
Phone

Gender: Male

Cancel Add

ID	Name	Phone
#10	المركزي	0791234567
#11	المركزي	059498035
#13	المركزي	059498035
#14	المركزي	059498034
#15	المركزي	0593498034
#16	المركزي	0593498034
#17	المركزي	0593498034
#18	المركزي	0593498034
#19	المركزي	0593498034
#20	المركزي	0123456987
#23	المركزي	0594348312
#24	المركزي	0578965874

To choose the site:



4.1.13 Database and API Server

In order to create a high-performance backend server and implement all of the RESTful APIs, we utilized the Node.js framework in our application. Numerous built-in Node.js functions and libraries helped us implement and build the necessary functionality and methodology.

5 Chapter 5

5.1 Result and Analysis

In developing the TaxiGo application, the implementation resulted in a complete, cross-platform transportation management system that effectively supports real-time taxi bookings. Two databases were used: **Firebase** for handling real-time communication features such as chat and ride status updates, and **MongoDB** as the primary data storage system for managing users, drivers, trips, and roles.

The use of **Flutter** allowed us to build both mobile and web interfaces from a single codebase, ensuring consistent UI and faster development. **Node.js** served as the back-end technology, efficiently managing requests and handling interactions with both MongoDB and Firebase.

For hosting, **Render** was used to deploy the server online, making it constantly accessible. Media management was handled using **Cloudinary**, where images (such as driver documents and car photos) were stored externally and their URLs saved in the database for scalability and improved load time.

The result is a **stable, scalable, and user-friendly system** that provides:

- Real-time ride tracking and notifications.
- Role-based access and dashboards (user, driver, manager, admin).
- Admin capabilities to manage all trips, users, and offices.
- A clean interface to manage settings, view statistics, and interact with maps.

This system not only succeeded in its functionality goals but also demonstrated solid integration of multiple complex systems such as geolocation, notification services, and role management.

6 Chapter 6

6.1 Discussion

Building TaxiGo presented numerous **technical and logistical challenges** but also served as a major learning opportunity.

6.1.1 Learning Outcomes

We were introduced to several new technologies:

- **Dart & Flutter** for UI and front-end development across platforms.
- **Node.js** for scalable backend APIs.
- **MongoDB** for NoSQL data storage, and **Firebase** for real-time functionality.
- Cloud platforms like **Render** and **Cloudinary**, used for backend deployment and media hosting respectively.

The experience strengthened our skills in **software architecture, RESTful API design, and real-time communication.**

6.1.2 Key Challenges

- **New technologies:** Dart, Flutter, and Firebase were completely new to us, which extended the learning curve.
- **Dual database management:** Maintaining data consistency between MongoDB and Firebase required careful design and synchronization.
- **Performance issues with Render:** Occasionally we faced slow response times or server downtimes, especially during deployment.
- **Cloudinary integration:** Secure media upload and linking presented initial difficulties until proper configuration was established.
- **Hardware limitations:** Running multiple emulators, development tools, and services caused heavy system load on our laptops, affecting productivity.
- **Mapping services:** We initially planned to use **Google Maps**, but due to cost limitations, we switched to **OpenRouteService** and **OpenStreetMap**, implementing live tracking and route drawing manually, which required significant effort and testing.

Despite these challenges, our team successfully developed a functioning, feature-rich system that aligns with our original vision and offers strong potential for real-world use in Palestine's transportation sector.

7 Chapter 7

7.1 Conclusion and Recommendation

7.1.1 Summary

We have successfully developed a cross-platform application for TaxiGo, a modern ride-hailing service that provides users with a seamless, real-time experience for booking and managing taxi rides. The application allows passengers to book rides easily, track their drivers live, and manage their trip history, while providing drivers with tools to manage availability, trips, and route navigation. The admin dashboard allows administrators to monitor system usage, manage users and drivers, and view statistics through a clean, intuitive interface. With a focus on usability, real-time features, and map integration, TaxiGo effectively connects passengers with drivers, improving urban mobility and service management.

7.1.2 Things We Learned

Throughout the TaxiGo project, we gained valuable technical and practical skills, including:

- Mobile app development using the Flutter framework and Dart language.
- Building REST APIs with Node.js and Express.js using JavaScript.
- Testing and validating endpoints with Postman.
- Integrating real-time chat functionality using Firebase.
- Deploying the back-end with Render, handling server-side operations and database connections.
- Uploading and managing media using Cloudinary, and storing their URLs in MongoDB.
- Working with two different databases (Firebase for real-time features and MongoDB for core storage).
- Using Git and GitHub effectively to manage and collaborate on code.
- Integrating OpenStreetMap manually after challenges with Google Maps pricing, including adding custom routing and markers.
- Debugging and optimizing performance under limited internet connections and hardware constraints.

7.1.3 Recommendations

Mobile app development has become an essential part of software engineering and should be given more attention in academic curriculums. We recommend offering a dedicated mobile development course that focuses on frameworks like Flutter and Firebase, enabling students to build real-world applications with modern tools and deploy them confidently.

7.1.4 Future Work

TaxiGo is just the beginning. We plan to expand and enhance the application in the following ways:

- Enhancing the chat system with support for group conversations, voice messages, image sharing, and real-time notifications.
- Integrating advanced map features with automatic geolocation detection, turn-by-turn navigation, and route optimization using live traffic data.
- Introducing dynamic pricing models, promo codes, and ride sharing.
- Creating user profile systems with ratings, feedback, and ride preferences.

8 Chapter 8

8.1 References

- 8.1.1 [1] GitHub, “Source Code Management and Version Control,” GitHub, 2024. [Online]. Available: <https://github.com>
- 8.1.2 [2] Flutter Dev Team, “Flutter Documentation,” Flutter, 2024. [Online]. Available: <https://docs.flutter.dev>
- 8.1.3 [3] Uber Technologies Inc., “How the Uber App Works,” Uber, 2023. [Online]. Available: <https://www.uber.com>
- 8.1.4 [4] OpenStreetMap, “Free and Open Map Data,” OpenStreetMap, 2024. [Online]. Available: <https://www.openstreetmap.org>
- 8.1.5 [5] OpenRouteService, “Routing and Geocoding APIs,” OpenRouteService, 2024. [Online]. Available: <https://openrouteservice.org>
- 8.1.6 [6] Google Maps Platform, “Maps, Routes & Places APIs,” Google Developers, 2024. [Online]. Available: <https://developers.google.com/maps>
- 8.1.7 [7] Cloudinary, “Media Asset Management and Image Hosting,” Cloudinary, 2024. [Online]. Available: <https://cloudinary.com>
- 8.1.8 [8] Render, “Cloud Hosting and Deployment Platform,” Render, 2024. [Online]. Available: <https://render.com>
- 8.1.9 [9] Firebase, “Realtime Database and Notifications,” Firebase, 2024. [Online]. Available: <https://firebase.google.com>
- 8.1.10 [10] Node.js, “Node.js Runtime and NPM Libraries,” Node.js, 2024. [Online]. Available: <https://nodejs.org>
- 8.1.11 [11] Socket.IO, “Real-time Communication Library for Node.js,” Socket.IO, 2024. [Online]. Available: <https://socket.io>