

An-Najah National University

بسم الله الرحمن الرحيم



Faculty of Engineering and Information Technology  
Computer Engineering Department  
Graduation Project I  
**ProCar**



**Marwan Sobuh**

**SuperVisor : Dr.Sufyan Samara**

**Presented in partial fulfilment of the requirements for the degree of  
Bachelor of Science in Computer Engineering  
March 5, 2024**

## Acknowledgements

First, last and always, thanks and gratitude to God, had it not been for his blessing that accompanied us in every step we took, I would not have completed this project successfully.

My dear parents and my beautiful family, thank you for all the support that always raised our spirits, and for the beautiful words that have always been a wall on which to lean and strengthen me. My great supervisor, Dr. Sufyan Samara, thanks for the advice that was always in the right place, and thanks for the enrichment, comments, and constructive criticism that contributed to increasing the project's features. All thanks to the Computer Engineering Department at An-Najah National University, and to the friends who accompanied us on this journey. Last but not least, thanks to everyone who had a hand in providing any kind of help, it means the world to me.

## Disclaimer

This report was written by Marwan Sobuh at the 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 Author's. 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.

# Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Problem :	9
1.2	Objectives of the work:	9
1.3	Scope of the work:	9
1.4	Significance of the work:	9
1.5	Organization of the report	9
<b>2</b>	<b>Constraints, Standards/ Codes and Earlier course work</b>	<b>10</b>
2.1	Constraints :	10
2.1.1	time limit :	10
2.1.2	limited experience :	10
2.1.3	Lack of members	10
2.2	Standards/Codes:	11
2.2.1	MVC (Model View Controller)	11
2.2.2	Agile Model	12
2.2.3	code	13
2.3	Earlier coursework:	14
<b>3</b>	<b>Literature Review</b>	<b>14</b>
<b>4</b>	<b>Methodology</b>	<b>15</b>
4.1	Mobile application	15
4.1.1	Frameworks	15
4.1.2	Programming Language :	16
4.2	Website for admin and company	16
4.3	Server side	16
4.3.1	Frameworks :	16
4.3.2	Programming Languages :	17
4.4	Database :	18
4.5	IDEs and Code Editors	19
4.6	API testing	19
4.7	Implementation	19
4.7.1	Mobile Application	20
4.7.2	Website for Admin	34
4.7.3	Website for Company	37
4.7.4	Database and API Server	43
<b>5</b>	<b>Results and Analysis</b>	<b>44</b>
5.1	Results :	44
5.2	Analysis	44
<b>6</b>	<b>Discussion</b>	<b>44</b>
<b>7</b>	<b>Conclusion</b>	<b>45</b>



## List of Figures

1	MVC Structure of the system. . . . .	11
2	Agile model . . . . .	13
3	Welcome Screen . . . . .	20
4	Sign In Screen . . . . .	21
5	Sign Up Screen . . . . .	22
6	Home Page . . . . .	23
7	Detail Screen . . . . .	24
8	Search Screen . . . . .	25
9	Personal Page . . . . .	26
10	Edit Information Screen . . . . .	27
11	Company Sign Up . . . . .	28
12	Company Sign in . . . . .	29
13	Company Home Screen . . . . .	30
14	Company Drawer . . . . .	31
15	Add Product Page . . . . .	32
16	Product Detail Page . . . . .	33
17	Web Sign In . . . . .	34
18	Admin Home . . . . .	35
19	Companies Table . . . . .	35
20	Users table . . . . .	36
21	Web Sign In . . . . .	37
22	Company Home Page . . . . .	38
23	Company Chart Page . . . . .	39
24	Web Add Product Page . . . . .	40
25	Company Request Page . . . . .	41
26	Company Shipment Page . . . . .	42

## List of Tables

1	Admin API's . . . . .	43
2	Company API's . . . . .	43
3	User API's . . . . .	44

# Abstract

This project is important because it meets the rising demand for convenient online access to a wide range of car accessories. It provides accessibility, convenience, and competitive options for customers, contributing to economic growth while aligning with environmental sustainability goals.

In my point of view, several important aspects should be covered in the project to ensure its success:

- User Experience (UX) Design\*: Creating an intuitive and user-friendly interface is crucial. Ensure easy navigation, responsive design, and clear product categorization to enhance the overall user experience.
- Product Catalog\*: Comprehensive and well-organized product listings with detailed descriptions, images, and compatibility information are essential to help customers make informed choices.
- Implement robust security measures to protect customer data and financial transactions, instilling trust and confidence among users.
- Search and Filter Functionality: Implement powerful search and filtering tools to help customers quickly find the products they need, enhancing their shopping experience.
- Customer Reviews and Ratings\*: Allow customers to leave reviews and ratings for products, fostering transparency and trust in your offerings.
- Mobile Responsiveness\*: Ensure that the website is fully responsive and functional on mobile devices, as many users shop from smartphones and tablets.



# **1 Introduction**

## **1.1 Problem :**

The problem addressed in this report is the need for a user-friendly and efficient mobile application for finding car cosmetics. The current market for car cosmetics often lacks a personalized and seamless experience for users, and its not common in Palestine. The design problem addressed in this project is to create an app that addresses these issues and provides a more enjoyable and efficient shopping experience.

## **1.2 Objectives of the work:**

The main objective of this work is to design and develop a mobile application that allows users to easily search, filter, and buy car cosmetics. Additionally, I will design a website for admin to control the system, and for companies with the same features as the app, considering that companies usually have employees work on a computers not on phones.

## **1.3 Scope of the work:**

The scope of this work is focused on the design and development of a mobile application. The app includes features such as searching and filtering , requesting and tracking the request , and viewing product details. Additionally, the app also includes web pages for admins to monitor the application. This work does not cover the detailed design and piping of the application.

## **1.4 Significance of the work:**

The significance of this work lies in its potential to improve the car cosmetics shopping experience for users. By providing a personalized and seamless experience for requesting and tracking the products, the app addresses a need in the market for more efficient and user-friendly shopping apps.

## **1.5 Organization of the report**

This report is divided into several chapters, including an abstract, introduction, Constraints, literature review, methodology, results and analysis, Discussion , and conclusion. Each chapter provides a detailed explanation of the design and development of the mobile application.

## **2 Constraints, Standards/ Codes and Earlier course work**

### **2.1 Constraints :**

#### **2.1.1 time limit :**

The main constraint faced during the development of this project was the limited time frame for completion. To overcome this constraint, I prioritized the most important features and focused on their implementation first. Additionally, I utilized agile development methodologies to ensure that the project was completed on time.

#### **2.1.2 limited experience :**

Another constraint faced during the development of this project was the limited experience with the technologies used in the project. To overcome this constraint, I enrolled into a two months training for mobile development using Flutter, I also had the chance to train on the node JS with a Asal ,I also had the help and consultation from my teachers at the University to ensure a successful implementation.

#### **2.1.3 Lack of members**

Unfortunately ,I had a problem with the partnership in this project ,it appeared that he does not have any intentions to work , or to learn , or even to respond , so I had to complete this project alone .

I solved this problem , by prioritizing the most important features and focused on their implementation first. Additionally, I utilized agile development methodologies , worked on the front-end and the back-end synchronously .

## 2.2 Standards/Codes:

### 2.2.1 MVC (Model View Controller)

The Model View Controller pattern was employed in our system. We could split the whole project into three parts so that it would be easier to track the flow of work. These components are as follows:

- The Model: The database model which is a noSql database called MongoDB ,I created several documents for users,products, companies, notifications, ext...
- The View:It represents the GUI that the users(Admin, Companies, Users) will use for monitoring, requesting, tracking and adding products .
- The Controller: It represent the back-end server which built with Node JS for the app and website. The controllers handles the requests , and the Authorization , the controllers also provided with services contains methods to monitor the processes on the database and provide the results for the users, Also it has a socket implementation for real time notifications and data update .

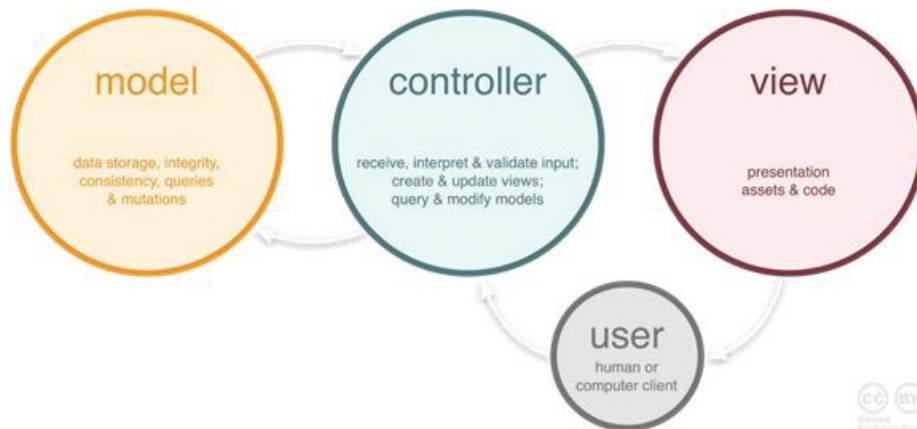


Figure 1: MVC Structure of the system.

### 2.2.2 Agile Model

I used Agile model methodology to model and document the software system. I divided the activities we had to complete into five stages:

- Stage 1: Requirement analysis:  
At this stage, I identified the basic features that we want to provide through the application.
- Stage 2: Planing:  
I have set a time plan for the completion of the checks. Each task has a specific time to be completed, and upon completion it is discussed with the project supervisor.
- Stage 3: Design the requirements:  
I designed the UML diagram to demonstrate the functionality of new features and how they will interact with my system.
- Stage 4: Development:  
for this part, I started writing the code for the front end and back end of each page synchronously.
- Stage 5: Testing:  
I tested every controller using the postman first , ensured that the controller behave as required , then i deploy the connection between the controller and the function in the front , after that ,I edit the User Interface to be more acceptable and user-friendly.

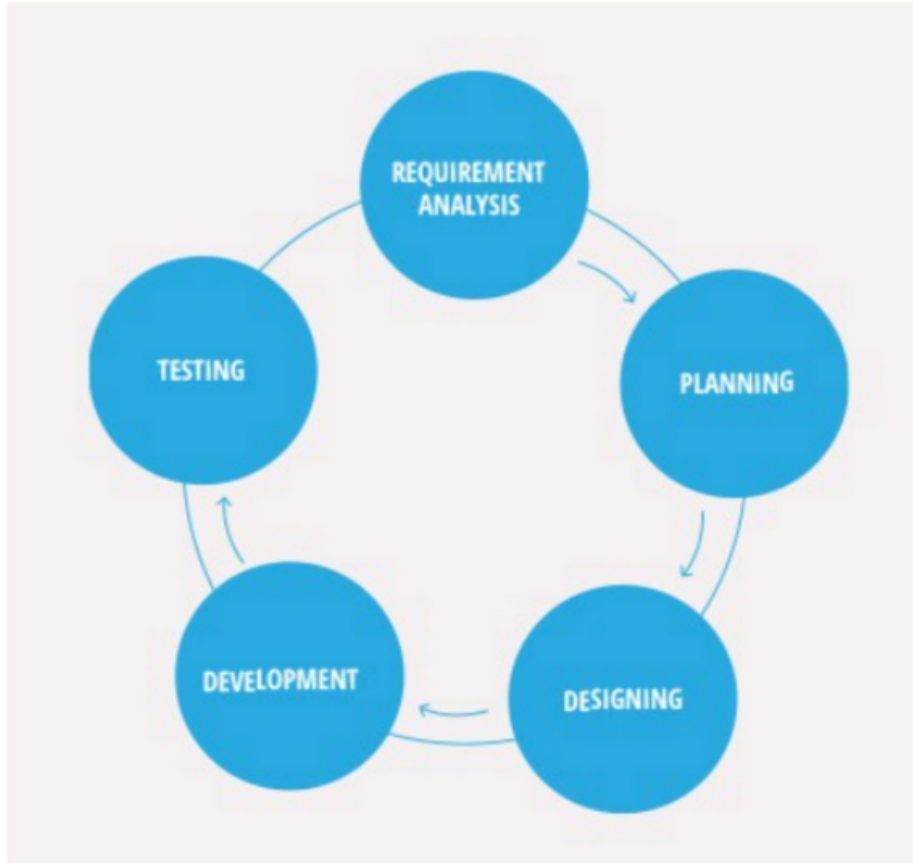


Figure 2: Agile model

### 2.2.3 code

The mobile application was developed in compliance with the Google Material Design Guidelines, which provide a set of guidelines for the design and user experience of mobile applications. Additionally, the application was developed in compliance with the Flutter framework, which is an open-source mobile application development framework created by Google.

The server-side of the application was developed in compliance with the TypeScript language and Node.js applications using TypeScript. Additionally, the server-side of the application uses MongoDB as its database management system, and Passport.js for Authorization.

### 2.3 Earlier coursework:

The development of this project builds upon the knowledge and skills acquired in previous coursework, including OOP, software development, Algorithm, Web Development and database management. My earlier coursework in these areas provided a strong foundation for the successful development and implementation of this project. Additionally, my experience with Agile development methodologies, user interface design, and web development helped to ensure the successful completion of this project.

## 3 Literature Review

Flutter is Google's portable user interface structure for making top notch local interfaces on Android and iOS in specific time. Flutter operates with existing code, is utilized by engineers and associations around the globe, and it is release and not closed source (Arb, G. and Al-Majdi, K., 2020). Flutter is an express stage structure that goes for growing elite mobile applications.

It was launched in 2016 by GOOGLE as an open. Flutter Apps can be run on Android, IOS and Fuschia operating systems resource (Arb, G. and Al-Majdi, K., 2020). Flutter has its very own user interface parts, alongside a motor to deliver them on the iOS and android. Flutter has different advantages among other languages, they can be summarized as: codebase improve for Android and iOS compared to a solitary codebase, more progress with less symbols, experience by changing symbol and reloading as the resulted app runs, resolve crashes and keep troubleshooting from the latest relevant point of interest, benefit from a high arrangement of material structure and iOS season gadgets assembled utilizing Flutter's very own system (Flutter, 2022; Arb Al-Majdi, 2020).

TS.ED (TypeScript Express Decorators) is a Node.js framework built on top of TypeScript and Express. It was first published in 2018 as an open-source project on GitHub. The framework aims to provide a set of decorators to simplify the development of RESTful APIs using TypeScript and Express. It allows developers to write clean and maintainable code by reducing boilerplate and providing a more structured way of defining controllers, routes, and middlewares. TS.ED also provides a set of features such as validation, pagination, and error handling that are commonly used in RESTful APIs. This framework has gained popularity among developers due to its ability to improve code readability and maintainability, making it a popular choice for building robust and scalable web applications.

Authentication and Authorization:

Authentication is the process of verifying the identity of a user or system. Authorization is the process of granting access to resources based on the authenticated identity. In this project, Passport.js was used as the authentication middleware for the backend. Passport.js supports a wide range of authentication strategies, including local authentication, OAuth, and OpenID Connect.

## 4 Methodology

The methodology used to build the mobile application and the backend server involved the use of several tools and technologies.

There are essential parts that cannot be dispensed with in building mobile applications and websites, such as frameworks, programming languages, and tools. All these details will be mentioned and talked about more.

### 4.1 Mobile application

#### 4.1.1 Frameworks

I decided to use Flutter which is open source framework by Google, after researching well and knowing its features, the most important of which is its effective performance. A cross-platform framework allows code to be shared between target platforms. Apart from Flutter, there are no other application frameworks that allow you to share both the user interface code and the user interface itself. Therefore, there is no need to scatter between more than one code and between different format for both of them, it is the same, and this is amazing, useful, shortening a lot of time, suffering and effort, and the result is perfect as expected which reduce the need for previous native programming for these two platforms, and saving a substantial amount of time and effort in cross-platform software development. In addition, it is open source, meaning there are many documents and lessons that can be learned and benefited from, which facilitates and speeds up the learning process, as well as this facilitates the solution of errors that programmers often face.

Flutter is, in a summary, the easiest method to create a faster code developer with high-performing cross-platform mobile app.

Here are some packages I used to develop my application :

- **Responsive** :The design of the application was made responsive by using the ScreenUtil library, which adjusts the layout of the application depending on the screen size of the device.  
Simply , you provide the code with the size of the screen that you use for styling the screens , then the package can use it to edit the sizes in the code it self when the app is running on different screen .
- **State Management** :For state management, the Provider package was used, which allows for easy management of the application's state. Basically , all the data that controls the building of the widgets are provided by

the Provider ,in case any data changed , the provider will automatically rebuilds the widgets according to the new value .

- Localization : The application also includes localization support, which was implemented using the Easy Localization package. Simply this package provides easy way to translate the texts shown in the application . It reads the translation from json files and provide the right phrase for the user according to language he preferred .
- HTTP requests : For handling the http requests, http package was used , simply it's a package provides the coder with the basic functions of the api requests, it's easy to use and to handle the request body and processing the response from the server .

#### **4.1.2 Programming Language :**

Dart is a programming language designed to develop web applications, Android and iOS applications, developed by Google, which targets web and mobile application developers. One of the goals of the language is for it to run on all advanced web browsers, mobile devices, and even web servers. Since we're going to be using flutter framework, it's clear that we're going to be using Dart. Dart has a lot of things in common with other languages like Java, C, Swift, and Kotlin. Since we often deal with Java, which is the most familiar language we have, we saw that Dart is very similar to it and this reduces the difficulty of a new language, and makes it straightforward and easy.

## **4.2 Website for admin and company**

I used the Angular Framework in website , it helps that Angular support designing websites . and its very rich in packages and uses typescript as a programming language.

## **4.3 Server side**

### **4.3.1 Frameworks :**

ts.ed is a Node.js framework built on top of TypeScript that simplifies the development of RESTful APIs. It is built on top of popular Node.js libraries such as Express and Mongoose, and it utilizes the power of TypeScript to provide a strong type system and improved code organization.

One of the main features of ts.ed is its ability to automatically generate a set of CRUD (Create, Read, Update, Delete) endpoints for a given Mongoose model. This allows developers to quickly set up a basic API for their application without having to write repetitive code. Additionally, it provides a set



of decorators and utility functions that make it easy to add custom routes and middleware to the API.

Another key feature of ts.ed is its built-in support for authentication and authorization. It provides a set of decorators and utility functions that make it easy to add authentication and authorization checks to routes and controllers. This makes it easy to ensure that only authorized users can access certain parts of the API.

ts.ed also provides a set of utility functions and decorators for input validation and error handling, which helps to ensure that the API inputs are valid and that errors are handled in a consistent way.

Overall, ts.ed is a powerful framework that can help developers to quickly and easily build robust and scalable RESTful APIs in TypeScript. Its support for Mongoose, TypeScript, and built-in authentication and authorization features make it a great choice for any project that requires a strong and flexible API.

#### **4.3.2 Programming Languages :**

TypeScript is a programming language that is a strict syntactical superset of JavaScript. It adds optional static typing and other features to the language, making it more suitable for large-scale software development.

One of the main benefits of TypeScript is its ability to catch type errors at compile-time, rather than at runtime. This can help to identify and fix bugs before they make it into production, leading to more stable and reliable software. Additionally, TypeScript's static typing can make it easier to understand and navigate large code bases, as it provides more information about the data and functions that are being used.

TypeScript also includes features such as interfaces, classes, and decorators, which can be used to build more modular and reusable code. This can help to improve the maintainability and scalability of a project, as well as making it easier to add new features and functionality.

Overall, TypeScript is a powerful tool for building large-scale and maintainable software. Its static typing and additional features can help to improve the quality and reliability of the code, while its compatibility with JavaScript makes it easy to adopt into existing projects.

#### 4.4 Database :

The server uses MongoDB as the database management system. MongoDB is a widely used, open-source NoSQL document-based database. It is a popular choice for web and mobile applications because it allows for easy scalability and the ability to handle large amounts of unstructured data.

One of the main features of MongoDB is its document-based data model. Instead of using tables and rows like a traditional relational database, MongoDB stores data in semi-structured BSON (binary JSON) documents. This allows for greater flexibility in terms of data structure and the ability to easily store and query complex data.

Another key feature of MongoDB is its horizontal scalability, which is achieved through sharding. Sharding is the process of distributing data across multiple servers, allowing the system to easily handle large amounts of data and handle high levels of read and write traffic. This makes MongoDB well-suited for use in high-traffic web and mobile applications.

MongoDB also offers robust indexing and querying capabilities, allowing developers to easily search, sort and filter data. Additionally, it has built-in support for geospatial indexing and querying, which is useful for applications that deal with location-based data.

In summary, MongoDB is a powerful NoSQL document-based database that offers features such as flexible data modeling, horizontal scalability, robust indexing and querying capabilities and support for geospatial data. Due to its scalability and flexibility, MongoDB is a popular choice for web and mobile applications that need to handle large amounts of unstructured data.

## 4.5 IDEs and Code Editors

To design our app we utilized visual studio code editor , It was developed by Microsoft, it supports many programming languages, also supports cross-platform, and it is a freeware editor. It comes with many of built-in tools, class designer, and comes with very rich and useful extensions.

For testing, Android Studio emulator is used, which is highly efficient compared to other emulators.

## 4.6 API testing

As we all know, we need to test portions of the server on a regular basis to determine if the API is usable and what happens when we submit the necessary data. All of this is done with Postman, which is a development tool for software that allows the user to check the validity of the request he wants to send to the server. The request is sent to a specific web server address And then the response that arrived from the server is shown.

## 4.7 Implementation

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

#### 4.7.1 Mobile Application

##### - User Screens

- Welcome Screen : This message will show to user when he runs the application , he can access as a company or as a user .

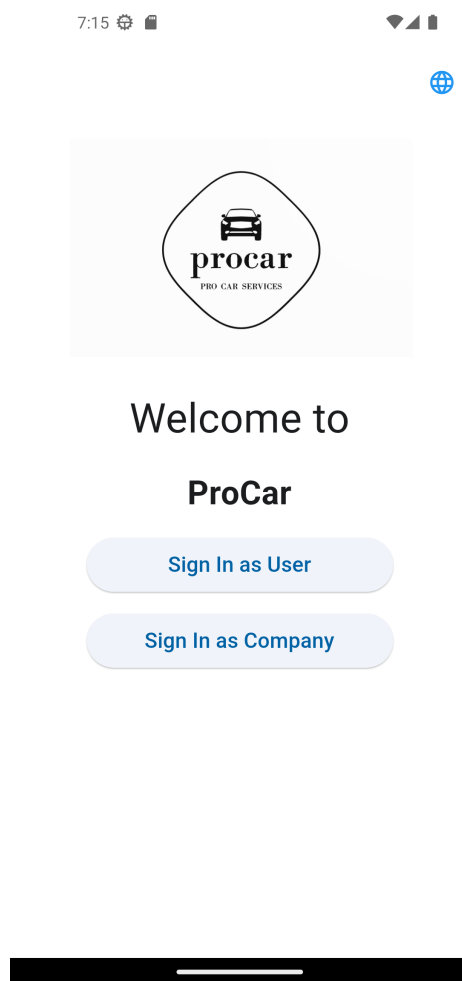


Figure 3: Welcome Screen

- sign in screen : this screen is to sign in for both users of companies , it also has a button to take the user for sign up page :

7:15

procar  
PRO CAR SERVICES

Email

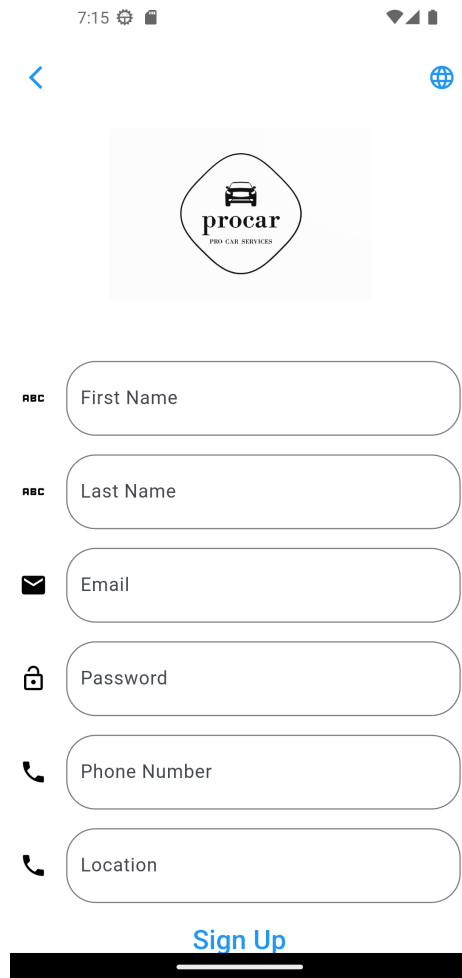
Password

Sign in

Dont have an account ? [Sign Up](#)

Figure 4: Sign In Screen

- Sign Up screens : If the user needs to sign up , he can press the sign up button and be routed to sign up pages according to his type , as user or company ,he should enter a valid email , password and other information , then he will be routed to his home page :



The image shows a mobile application interface for signing up. At the top, there is a status bar with the time 7:15 and various icons. Below the status bar, there is a navigation bar with a blue back arrow on the left and a blue globe icon on the right. The main content area features the procar logo, which consists of a car icon inside a diamond shape, with the text "procar" and "PRO CAR SERVICES" below it. Below the logo, there are six input fields, each with a label and an icon to its left: "First Name" (ABC icon), "Last Name" (ABC icon), "Email" (envelope icon), "Password" (lock icon), "Phone Number" (phone handset icon), and "Location" (phone handset icon). At the bottom, there is a black bar with the text "Sign Up" in blue, which is underlined.

Figure 5: Sign Up Screen

- Home Screen : In the home page, the user have four pages, a list with the newest products ,and a list with the requested products to track the packages, a search screen and a profile screen :

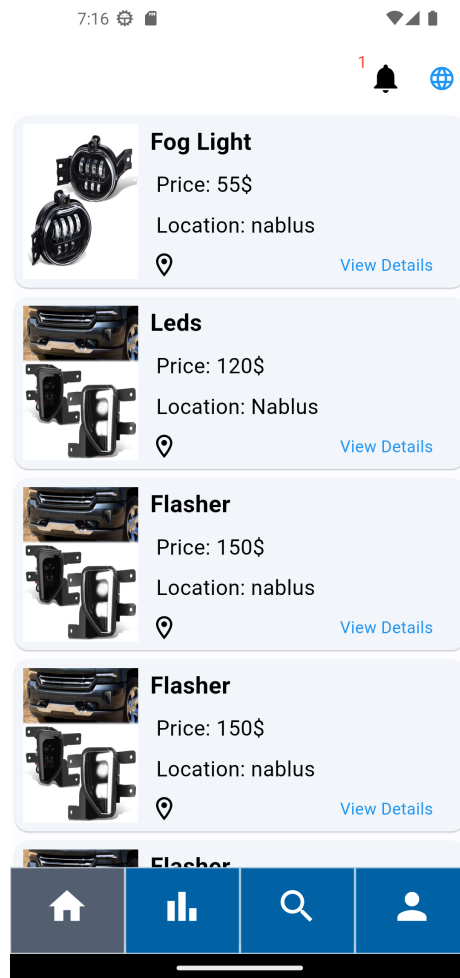


Figure 6: Home Page

- The detail page : where the user can request the products :

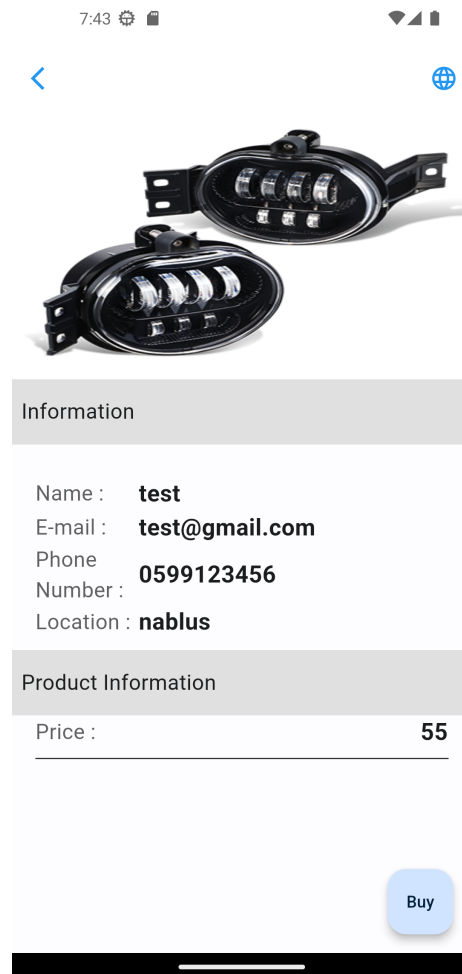


Figure 7: Detail Screen



- Search Screen : in this screen , the user can search in the products by entering their names , this filter service can provide the user an easier searching to get what he desire :

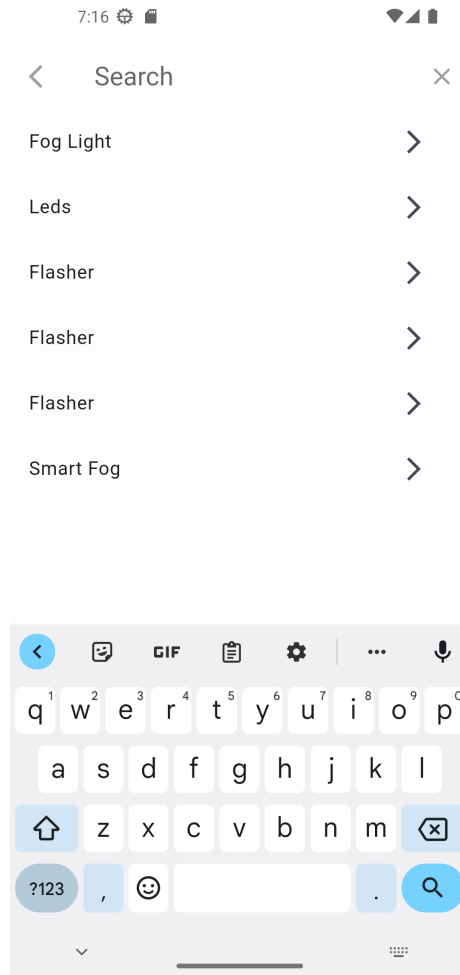


Figure 8: Search Screen

- Personal page : this page contains the features that the user might need , for example , he can change the language , set the dark mode, also there is a few pages for his personal use , such as the products he requested , or the products he bookmarked :

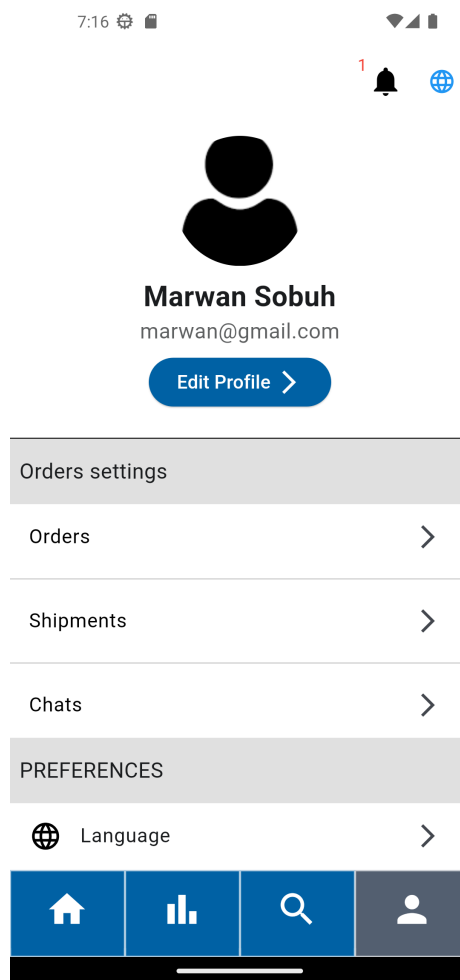




Figure 9: Personal Page


- Edit profile screen : here , the user can edit his information and upload a personal picture :


7:51


< 





[Edit Picture](#)

 first Name

 last Name

 Location

 Email

 Password


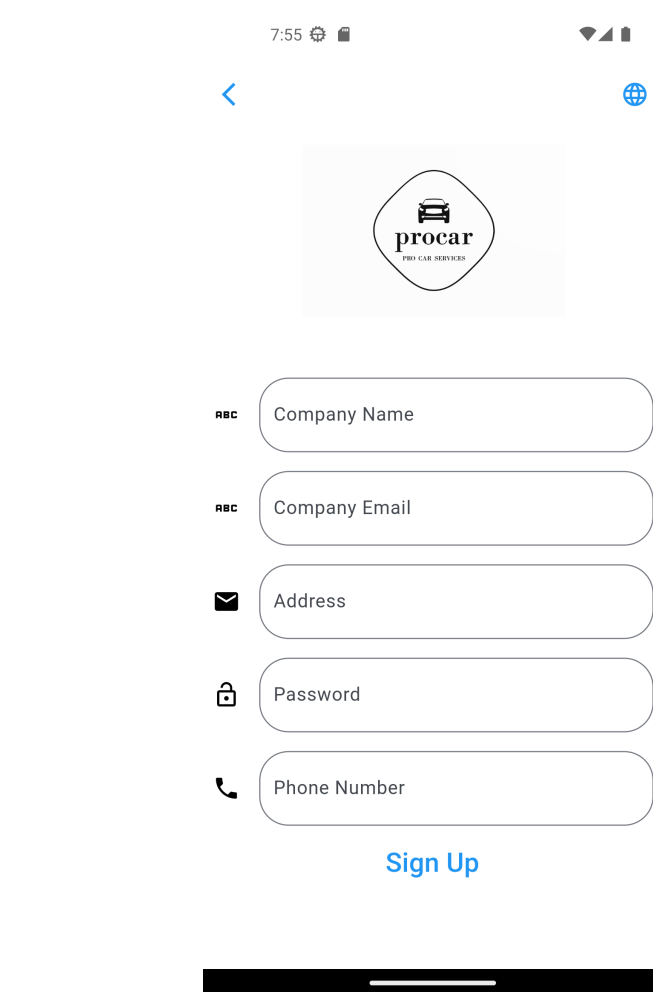
 Phone Number

Figure 10: Edit Information Screen

- Company Screens :

- Sign Up Screen : Of course , A screen to create an account for company , it follows the same steps as entering a unique email :



The image shows a mobile application interface for a company sign-up screen. At the top, there is a status bar with the time 7:55, a gear icon, a battery icon, and a signal strength indicator. Below the status bar, there is a navigation bar with a blue back arrow on the left and a blue globe icon on the right. The main content area features the procar logo, which consists of a car icon inside a diamond shape, with the text "procar" and "PRO CAR SERVICES" below it. Below the logo, there are five input fields, each with a label and an icon to its left: "Company Name" with an "ABC" icon, "Company Email" with an "ABC" icon, "Address" with an envelope icon, "Password" with a lock icon, and "Phone Number" with a phone icon. Below the input fields, there is a blue "Sign Up" button. At the bottom of the screen, there is a black bar with a white horizontal line in the center, indicating the home indicator for a mobile device.

Figure 11: Company Sign Up

- A Login Screen : Similar to the one for the normal user :

7:15 ⚙️ 🔋 🌐

**procar**  
PRO CAR SERVICES

✉ Email

🔒 Password

[Sign in](#)

Dont have an account ? [Sign Up](#)

Figure 12: Company Sign in

- Home Screen : which contains the products that the company has published , Also a floating button to add new products :

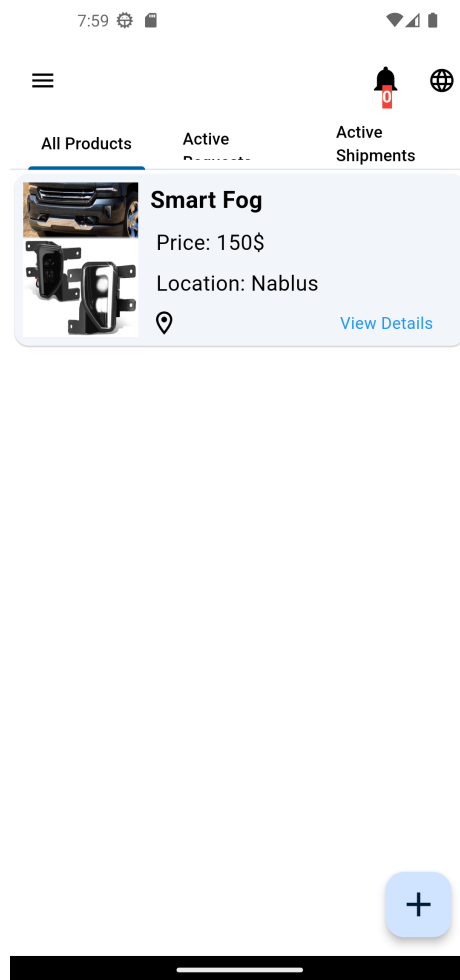


Figure 13: Company Home Screen

- Drawer : A Drawer for fast access to other services :

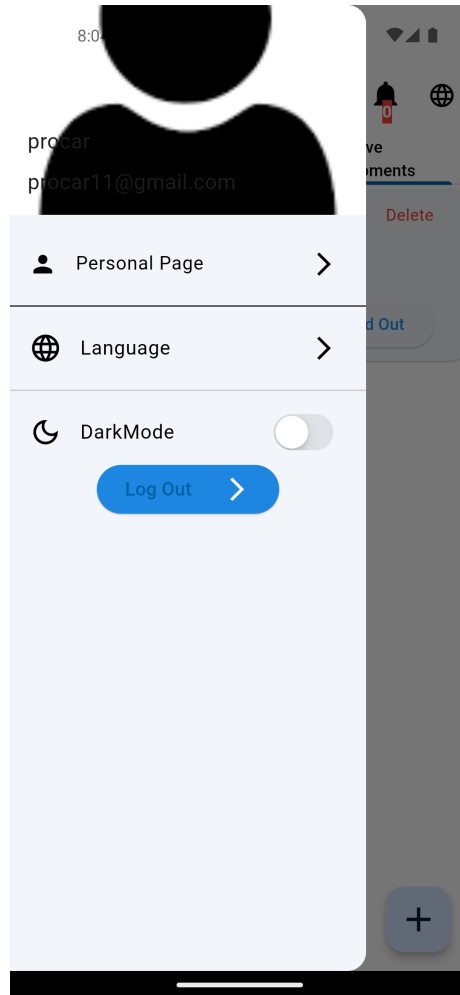


Figure 14: Company Drawer

- Add Product Page : when the user presses on the floating button , a pages show up to fill as a new product details :

7:59

<

*Pick Image*

ABC Product title

# Product Quantity

ABC Product Description

\$ Product Price

Add

Figure 15: Add Product Page



- Detail Page : when the user presses on View Details :

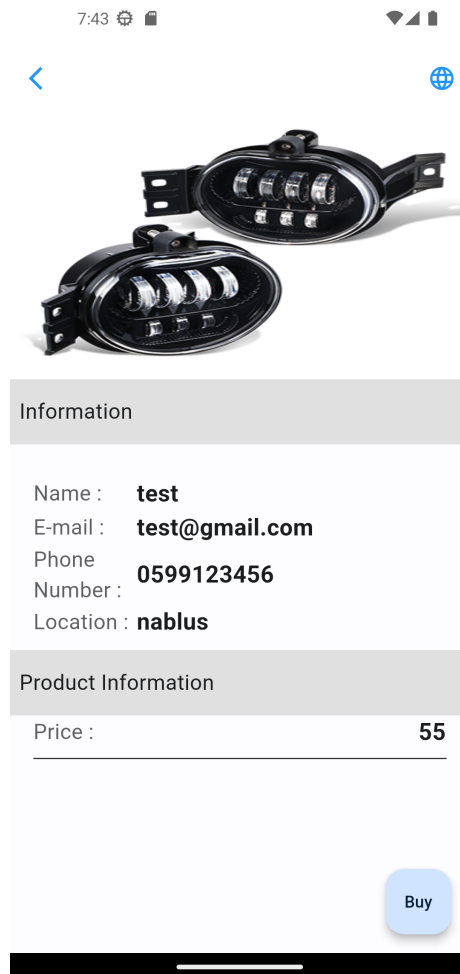


Figure 16: Product Detail Page

#### 4.7.2 Website for Admin

- Sign In Screen : The same as the application , a sign in page for authentication , when the user enters a correct value , a token is sent to him as a respond , the Provider use it later for any requests that needs authorization , which are all the requests other than login and sign up in general :

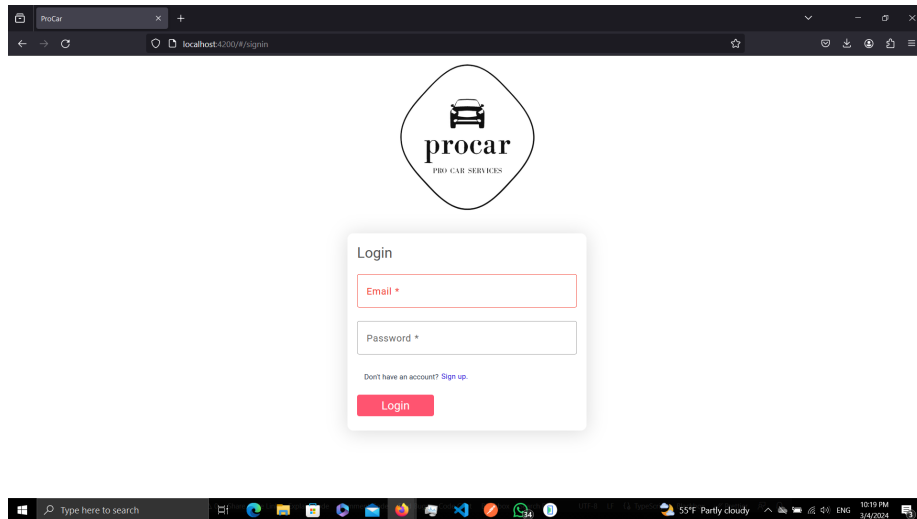
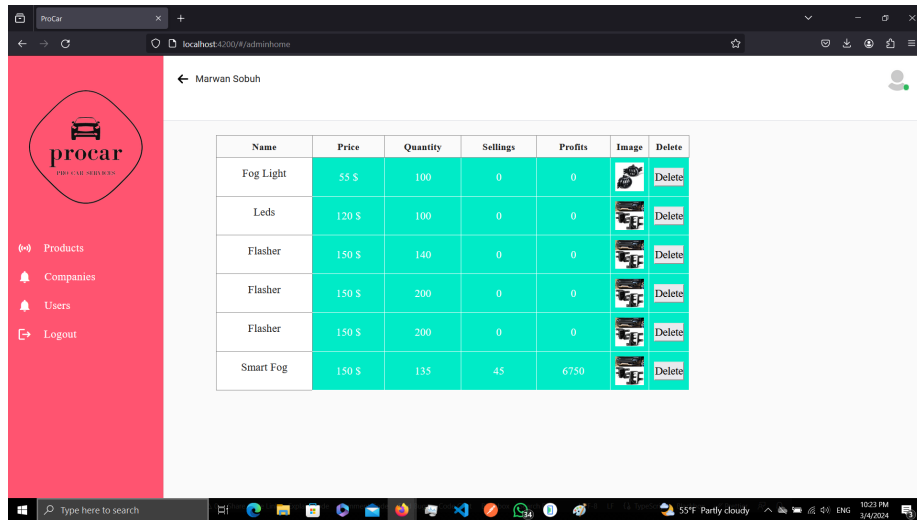


Figure 17: Web Sign In

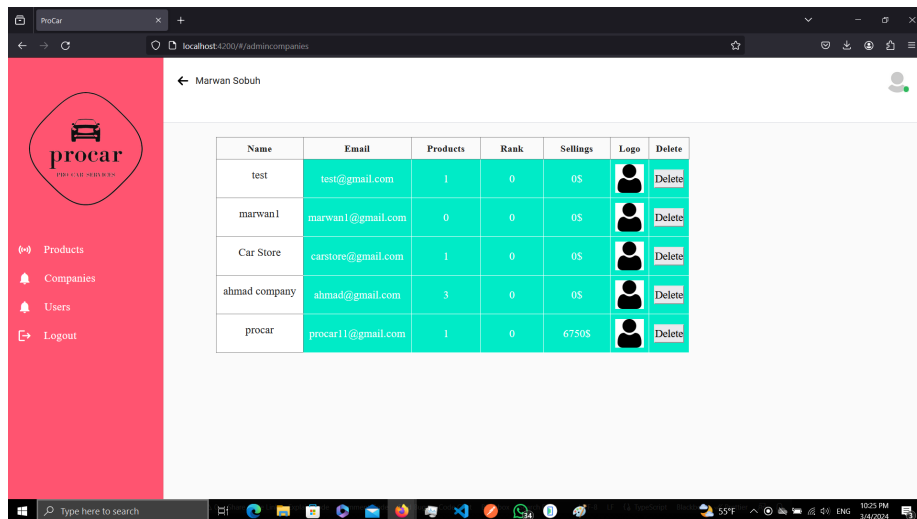
- The Admin has the control over everything in the apps , he can delete users, companies or product :



Name	Price	Quantity	Sellings	Profits	Image	Delete
Fog Light	55 \$	100	0	0		Delete
Leds	120 \$	100	0	0		Delete
Flasher	150 \$	140	0	0		Delete
Flasher	150 \$	200	0	0		Delete
Flasher	150 \$	200	0	0		Delete
Smart Fog	150 \$	135	45	6750		Delete

Figure 18: Admin Home

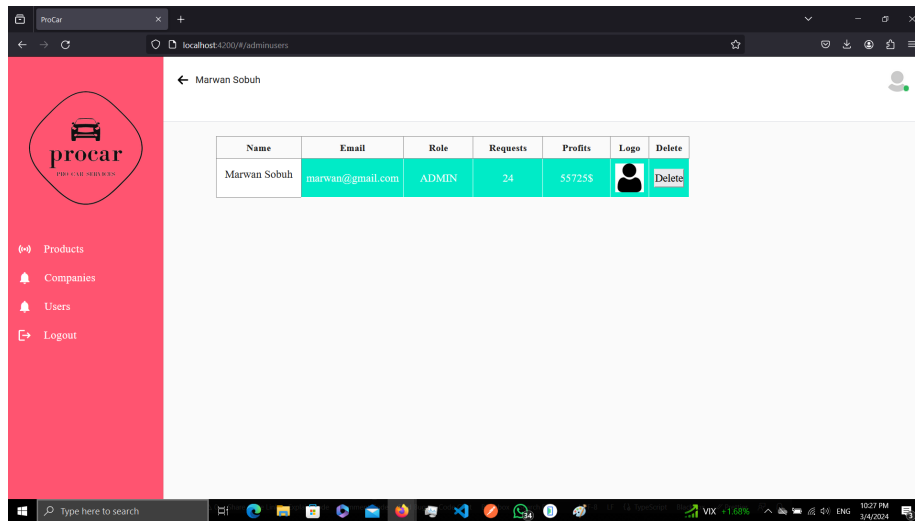
- Companies : In this table , the admin controls the companies , he can have a full view to all companies in the application , and have the ability to delete any of them :



Name	Email	Products	Rank	Sellings	Logo	Delete
test	test@gmail.com	1	0	0\$		Delete
marwan1	marwan1@gmail.com	0	0	0\$		Delete
Car Store	carstore@gmail.com	1	0	0\$		Delete
ahmad company	ahmad@gmail.com	3	0	0\$		Delete
procar	procar11@gmail.com	1	0	6750\$		Delete

Figure 19: Companies Table

- Users : In this table , the admin controls the users , he can have a full view to all users in the application , and have the ability to delete any of them :




Name	Email	Role	Requests	Profits	Logo	Delete
Marwan Sobuh	marwan@gmail.com	ADMIN	24	\$57255		Delete

Figure 20: Users table

### 4.7.3 Website for Company

- Sign In Screen : The same as the application , a sign in page for authentication , when the user enters a correct value , a token is sent to him as a respond , the Provider use it later for any requests that needs authorization , which are all the requests other than login and sign up in general :

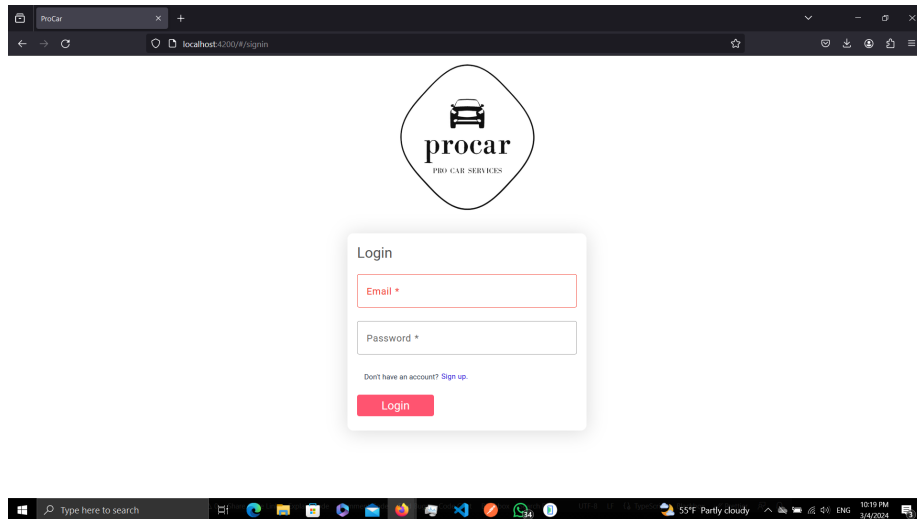


Figure 21: Web Sign In

- Home Screen: in this page, the company can control the Product status and delete it, also when the user press on the product name it routes him to the chart page :

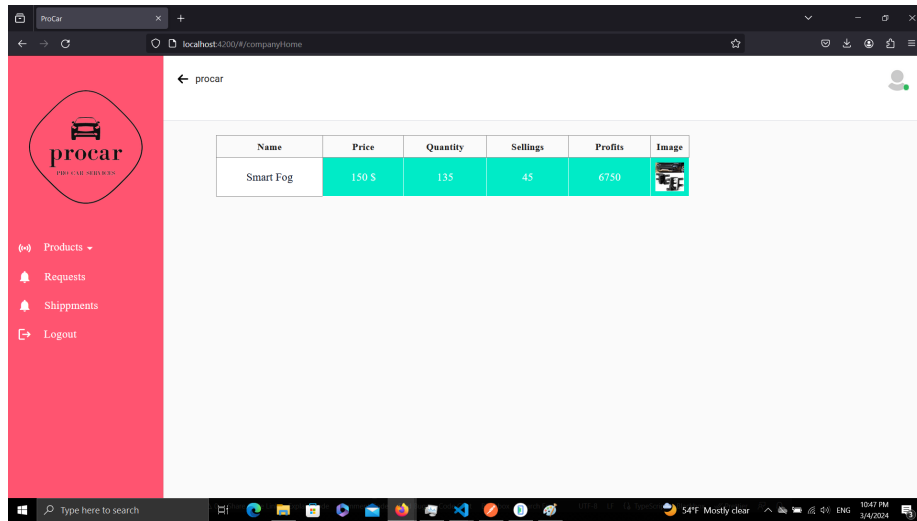


Figure 22: Company Home Page

- Chart Screen: in this page, the company can see an overview for Product sellings with the relation of quantity and the total price for the request :

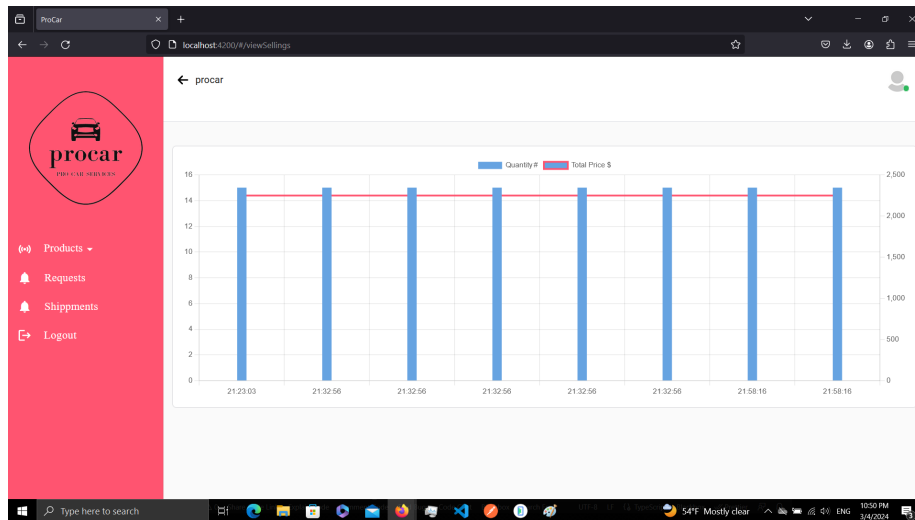


Figure 23: Company Chart Page

- Add Product Screen: in this page, the company can add a new Product same as the mobile app :

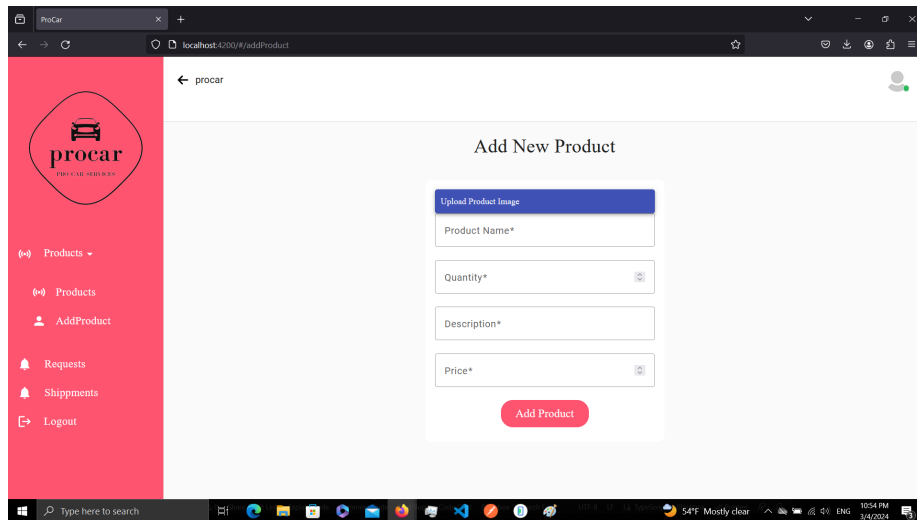


Figure 24: Web Add Product Page



- Requests Screen: in this page, the company can control the request status and accept or delete it, when the company accept it it moves to the shipments table :

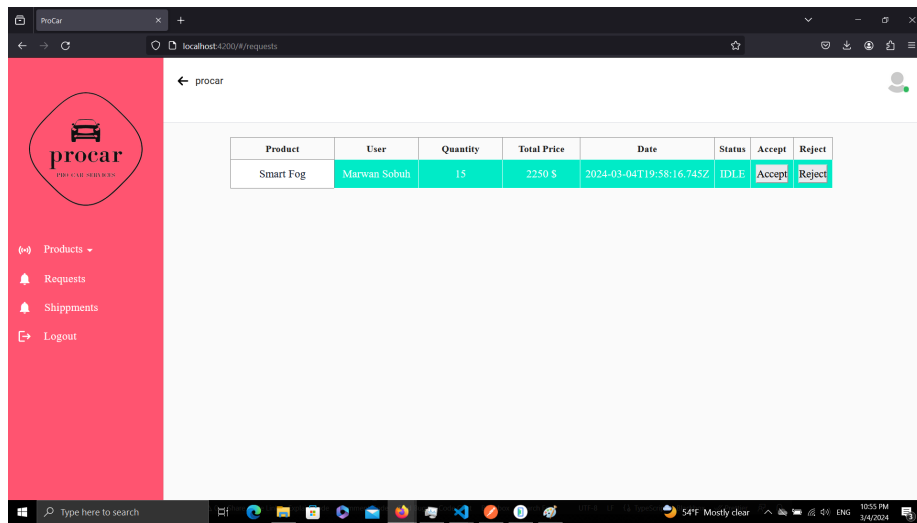


Figure 25: Company Request Page

- Shipment Screen: in this page, the company can control the accepted request status to ship it and sent it using Ship button, also the company cannot delete the request after accepting it :

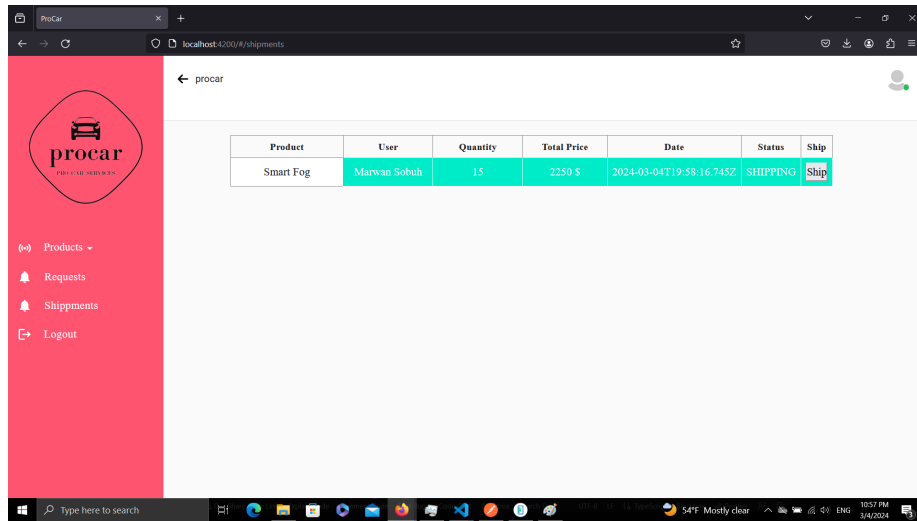


Figure 26: Company Shipment Page

#### 4.7.4 Database and API Server

I used Node JS to help me build a high performance backend server and implement all the RESTful APIs and real time Sockets. Nodejs has many built in functions and libraries that helped me in implementing and developing the needed features and methods.

Here are the list of API's I implemented in the backend server :

- Admin API's :

Path	Type	Function	Need Authorization	Parameter
/getAllProducts	Get	fetch all Products	Yes	-
/getAllUsers	Get	fetch all users	Yes	-
/getAllCompanies	Get	fetch all Companies	Yes	-
/login	Post	Authentication	No	Email and password
/deleteCompany	Delete	delete A company	Yes	Company id
/deleteTProducts	Delete	delete a Products	Yes	Products id
/deleteUser	Delete	delete a user	Yes	User id

Table 1: Admin API's

- Company API's :

Path	Type	Function	Need Authorization	Parameter
/getRelatedProducts	Get	fetch Related Products	Yes	-
/getInfo/:id	Get	fetch company info	Yes	-
/login	Post	Authentication	No	Email and password
/signUp	Post	New Account	No	Company Model
/addNewProducts	Post	publish Products	Yes	Products Model
/updateProducts	Put	update Products	Yes	Products Model
/updateProducts	Put	Update Info	Yes	Company Model
/deleteProducts	Delete	delete a Products	Yes	Products id

Table 2: Company API's

- User API's :

Path	Type	Function	Need Authorization	Parameter
/getAllProducts	Get	fetch Approved Products	Yes	-
/getInfo/:id	Get	fetch User info	Yes	-
/login	Post	Authentication	No	Email and password
/signUp	Post	New Account	No	User Model
/requestProduct	Post	request a Products	Yes	Products id and user id
/updateUser	Put	Update Info	Yes	User Model

Table 3: User API's

## 5 Results and Analysis

### 5.1 Results :

I used many tools and frameworks in the frontend and backend , it helped me to create a user-friendly and efficient system for travelling service , in this project , I learnt how build a user-friendly frontend , an efficient and secure server .

### 5.2 Analysis

## 6 Discussion

In this section, the results of the ProCAR application are discussed and analyzed. The main objectives of the project, which were to create a responsive, user-friendly, and localized web application for requesting and tracking products, have been successfully achieved.

The application's design and user interface have been optimized for ease of use, with the use of the screen util library for responsive design and the Easy Localization package for localization. The use of the Provider package for state management has also contributed to the smooth functioning of the application.

The use of the Node.js runtime environment and the MongoDB database have also been successful in providing a reliable and efficient backend for the application. The use of the ts.ed framework for type-safe and easy development of the server-side logic has also been beneficial.

Overall, the ProCar application has proven to be a functional and efficient solution for requesting and managing products. However, there are still some limitations to the application that can be addressed in future work. For example, the possibility of implementing a real-time chat system for communication between users and companies, and the possibility of providing a mobile appli-

cation for a better user experience.

In conclusion, the ProCar application has successfully achieved its objectives and has proven to be a useful tool for requesting and tracking products. The use of various technologies such as Node.js, MongoDB, and ts.ed have been instrumental in the development of the application. However, there is still room for improvement and further study in the future.

## 7 Conclusion

In conclusion , the following points should be highlighted :

- The problem of providing an efficient and user-friendly platform for requesting and tracking products was successfully addressed through the use of the Flutter framework and the Dart programming language.
- The application's functionality was enhanced through the use of various libraries and packages, such as screenutil for responsiveness, provider for state management, and EasyLocalization for localization.
- The use of MongoDB as the database management system allowed for the efficient storage and retrieval of large amounts of user and products data.
- The application's performance was further improved through the use of TypeScript and the ts.ed framework, which helped in maintaining a consistent and well-structured codebase.

In terms of future work, it is suggested to add more functionalities, like adding payment gateway, or integrating more third party APIs for providing more services for the user. Additionally, further testing and optimization of the recommendation system could be undertaken to enhance its performance.

## 8 References

### References

- [1] Arb, G. I., Al-Majdi, K. (2020). A Freights Status Management System Based on Dart and Flutter Programming Language. Journal of Physics: Conference Series, 1530.
- [2] Flutter. (2023, 01 14). Flutter architectural overview. Retrieved from : <https://docs.flutter.dev/resources/architectural-overview>
- [3] Ts.ed. (2023, 01 13). Ts.ed overview. Retrieved from : <https://tsed.io/getting-started/>