REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

DESIGNING A CARPOOLING APP TO REDUCE TRAFFIC CONGESTION IN URBAN AREA

Ahmed SAFA

Master's Thesis

Supervisor

Assoc. Prof. Dr. Sefer KURNAZ

İstanbul, 2024

DESIGNING A CARPOOLING APP TO REDUCE TRAFFIC CONGESTION IN URBAN AREA

Ahmed SAFA

Information Technologies and Security

Master's Thesis

ALTINBAŞ UNIVERSITY

2024

The thesis titled DESIGNING A CARPOOLING APP TO REDUCE TRAFFIC CONGESTION IN URBAN AREA prepared by AHMED SAFA and submitted on 10/07/2024 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

Assoc. Prof. Dr. Sefer KURNAZ

Supervisor

Thesis Defense Committee Members:

Assoc. Prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbaş University	
Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Department of Computer Engineering, Altınbaş University	
Asst. Prof. Dr. Serdar KARGIN	Department of Biomedical Engineering, Istanbul Arel University	

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Ahmed SAFA

Signature

DEDICATION

To My dearest parents in the world who worked for my success and who never stopped encouraging and supporting me. May God ensure that this work bears fruit; Thank you for the noble values, education, and ongoing support. I wish you a long life full of happiness.

To my sisters and brothers who have always been there for me, examples of perseverance, courage and generosity, through their continuous encouragement.

To All friends and all those whom I love and are dear to me and whom I have neglected to mention, while wishing them success in everything they undertake.

My thanks go to my advisor, “Assoc. Prof. Dr. Sefer KURNAZ”, for agreeing to direct this work, it is an honor for me. For his multiple advice and efforts to ensure quality work, his support, his skills, and his foresight were invaluable to me.

ABSTRACT

DESIGNING A CARPOOLING APP TO REDUCE TRAFFIC CONGESTION IN URBAN AREA

SAFA, Ahmed

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Assoc. Prof. Dr. Sefer KURNAZ

Date: 07 / 2024

Pages: 51

Urban areas across the globe are negatively impacted by traffic congestion, and Istanbul is no exception. The city's traffic situation is especially daunting due to its large population of 16 million individuals who reside and work there [1]. To address this issue, carpooling has been identified as an effective method for decreasing traffic congestion and promoting sustainable transportation. As a result, we created a carpooling application tailored for Istanbul with the goal of mitigating traffic congestion. The app permits drivers and passengers to generate personal profiles, seek out carpooling companions, and communicate with one another to coordinate rides. In addition, the app offers services like booking rides, processing payments, and assessing and evaluating experiences to guarantee safety and liability. Overall, to ensure the success of the project, the app will be designed to address the specific needs of Istanbul's urban population and to provide a viable solution to the city's traffic congestion problem using the database schema and API architecture using the PHP programming language and MySQL database management system. The app will be also scalable and can be adapted to other urban areas facing similar traffic problems and compatible with multiple platforms.

Keywords: Carpooling, Traffic Congestion, PHP, MYSQL, Urban Area.

ÖZET

ŞEHİR İÇİ TRAFİK SIKIŞIKLIĞINI AZALTMAK İÇİN BİR ARABA PAYLAŞIM UYGULAMASI TASARIMI

SAFA, Ahmed

Yüksek Lisans, Bilişim Teknolojileri, Altınbaş Üniversitesi,

Danışman: Doç. Dr. Sefer KURNAZ

Tarih: 07 / 2024

Sayfa: 51

Dünya genelindeki şehirler trafik sıkışıklığından olumsuz etkilenmektedir ve İstanbul da bir istisna değildir. Şehirde yaşayan ve çalışan 16 milyon bireyin büyük nüfusu nedeniyle, İstanbul'un trafik durumu özellikle zorludur [1]. Bu sorunu çözmek için araba paylaşımı, trafik sıkışıklığını azaltmak ve sürdürülebilir ulaşımı teşvik etmek için etkili bir yöntem olarak belirlenmiştir. Sonuç olarak, İstanbul'a özgü trafik sıkışıklığını hafifletme amacıyla bir araba paylaşım uygulaması oluşturduk. Uygulama, sürücülerin ve yolcuların kişisel profiller oluşturmalarına, araba paylaşımı yapacakları kişileri bulmalarına ve sürüşleri koordine etmek için iletişim kurmalarına olanak tanır. Ayrıca, uygulama, yolculuk rezervasyonu, ödeme işlemleri ve deneyimleri değerlendirme ve derecelendirme gibi hizmetler sunarak güvenlik ve sorumluluğu garanti altına alır. Genel olarak, projenin başarısını sağlamak için uygulama, İstanbul'un şehir nüfusunun özel ihtiyaçlarını ele alacak şekilde tasarlanacak ve şehrin trafik sıkışıklığı sorununa uygulanabilir bir çözüm sunmak için PHP programlama dili ve MySQL veritabanı yönetim sistemi kullanılarak veritabanı şeması ve API mimarisi ile oluşturulacaktır. Uygulama ayrıca ölçeklenebilir olacak ve benzer trafik sorunları yaşayan diğer kentsel alanlara uyarlanabilecek ve birden fazla platformla uyumlu olacaktır.

Anahtar Kelimeler: Araba Paylaşımı, Trafik Sıkışıklığı, PHP, MYSQL, Kentsel Alan.

TABLE OF CONTENTS

Pages

ABSTRACT ..... **vi**

ÖZET ..... **vii**

LIST OF FIGURES ..... **x**

ABBREVIATIONS ..... **xii**

1. INTRODUCTION **1**

1.1 PROBLEM STATEMENT 2

1.2 AIM OF STUDY 3

1.3 THESIS LAYOUT 3

2. GENERAL INFORMATION **5**

2.1 CARPOOLING 5

2.1.1 Static Carpooling 6

2.1.2 Dynamic Carpooling 7

2.2 LITERATURE REVIEW 8

2.2.1 Carpooling App for a Green Social Solution to Traffic and Congestions (GA) 8

2.2.2 A Trust-Based Cooperative Social System 9

2.3 CONTRIBUTION 10

3. METHODOLOGY **14**

3.1 INTRODUCTION 14

3.2 TECHNOLOGIES 14

3.3 ANALYSIS AND DESIGN 19

3.3.1 Analysis and Specification of Functional Requirements 19

3.3.2 Use Case Diagram 20

3.3.3 Class Diagram 21

3.3.4 Sequence Diagram 22

3.4 PROJECT MANAGEMENT 23

3.4.1 Gantt Chart Overview 24
3.4.2 GitHub for Version Control 25
4. IMPLEMENTATION 28
4.1 INTRODUCTION 28
4.2 PRESENTATION OF HOMEPAGE 28
4.3 PRESENTATION OF THE APP 32
5. CONCLUSION 41
REFERENCES 42

LIST OF FIGURES

Pages

Figure 2.1: A Carpooling System Where Three Requests are Served by a Single Driver from Their Sources (S) to Their Destinations (D) (Silwal et al., 2019). 6

Figure 3. 1: Architecture MVC 19

Figure 3.2: Use Case Diagram. 21

Figure 3.3: Class Diagram. 22

Figure 3.4: Sequence Diagram. 23

Figure 3.5: Project Management. 24

Figure 3.6: Gantt Chart. 25

Figure 4.1: Homapage. 28

Figure 4.2: About US. 29

Figure 4.3: Welcome to TURKAR. 29

Figure 4.4: Improve İstanbul's Commute. 30

Figure 4.5: Services. 30

Figure 4.6: Reviews. 31

Figure 4.7: Team. 31

Figure 4.8: Pricing. 32

Figure 4.9: Frequently Asked Questions. 32

Figure 4.10: Authentication page. 33

Figure 4.11: Registration Page. 33

Figure 4.12: Homepage. 34

Figure 4.13: Notifications. 34

Figure 4.14: Languages. 35

Figure 4.15: User. 35

Figure 4.16: Profile. 36
Figure 4.17: Trips 36
Figure 4.18: New Trips……37
Figure 4.19: Show Trip……37
Figure 4.20: Edit Trip……38
Figure 4.21: Locations……..…38
Figure 4.22: Vehicles. 39
Figure 4.23: Show Vehicles. 39
Figure 4.24: Reviews. 40

ABBREVIATIONS

AJAX	:	Asynchronous JavaScript and XML
API	:	Application Programming Interface
CO2	:	Carbon dioxide
CSS	:	Cascading Style Sheets
GA	:	Genetic Algorithm
GPS	:	Orthogonal Frequency Division Multiplexing
HTML	:	HyperText Markup Language
JS	:	JavaScript
JSON	:	JavaScript Object Notation
MVC	:	Model – View – Controller
MYSQL	:	Structured Query Language
PHP	:	Hypertext Preprocessor
UML	:	Unified Modeling Language

1. INTRODUCTION

Man has had a strong desire to travel since the widening of distances and the introduction of modes of transportation, whether public or private. The demand for movement has implications, as seen by air pollution, climate change, and other factors.

Recent years have seen a dramatic surge in trip demands. This expansion necessitates the development and ongoing improvement of modes of transportation. Despite its ongoing improvement, public transportation remains less competitive in terms of user demand. Private automobiles continue to be the most popular means of transportation due to their availability and efficiency. However, increased human mobility and the widespread use of private vehicles have had a negative impact on the environment and generated concerns about quality of life.

Indeed, excessive usage of private cars contributes to pollution, traffic congestion, and stress. The transition to new means of transportation is necessary, which led to the development of the concept of carpooling, which involves linking non-professional drivers with carpooling offers with possible passengers who want to travel together.

Initially, carpooling networks served persons whose starting and finishing locations of travel were reasonably close together. Such static methods have proven successful for some types of ride-sharing solutions, but they do not operate well when static information is not accessible prior to making a decision.

Furthermore, the information technology revolution is transforming transportation, as it does in so many other parts of daily life. These technology advancements have necessitated the development of a new sort of carpooling known as dynamic carpooling, which must be capable of responding to requests in milliseconds and providing more comprehensive automatic matching than a simple radial search around trip origins and destinations.

Urban centers worldwide, including Istanbul, face a significant problem of traffic congestion. Starting in the 1950s, Istanbul became the primary destination for a significant migration of people from rural to urban areas. Since then, the city has undergone rapid industrialization and unplanned urbanization, resulting in the emergence of numerous critical problems such as increased density, congestion, traffic jams, pollution, and a shortage of

urban space [2]. This not only results in delays and annoyance for commuters but also contributes to amplified levels of pollution, fuel consumption, and accidents. Carpooling has the potential to alleviate traffic congestion by encouraging ride-sharing and decreasing the number of vehicles on the roads. The purpose of this project is to create a carpooling application employing the PHP (MYSQL) programming language, with the aim of tackling the issue of traffic congestion in Istanbul.

This project consists of four steps: planning, designing, development, deployment, and maintenance.

a. Planning consists of the following steps: defining the purpose and goals of the carpooling web application and determining the target audience's needs. Gather application requirements, such as features, functionality, and user interface design.

b. Designing entails creating wireframes and user interface mockups. Create database schema and application architecture. Determine the application's technology stack and platform.

c. The development phase consists of creating the application's front-end and back-end functionality. Put the database schema and data models into action. Conduct testing and quality assurance to ensure that the application is working as it should.

d. Deployment entails the following steps: Deploy the application to a production environment, Configure the application to be scalable, secure, and fast. Final testing and quality assurance should be performed to ensure that the application is ready for release.

e. Maintenance entails monitoring the application for problems and errors. Continuously update and improve the application's functionality and user interface. Maintain the application's security and stability by performing regular backups and maintenance tasks.

1.1 PROBLEM STATEMENT

Urban areas globally face considerable issues as traffic congestion worsens, resulting to environmental pollution, increased fuel use, and longer commuting times. Istanbul, with its large population and growing urbanization, exhibits these issues, resulting in significant societal and environmental consequences. Despite different initiatives to reduce traffic congestion, there is still a gap in implementing efficient, sustainable transportation systems

that can adapt to the unique requirements and habits of urban populations. This thesis seeks to close this gap by creating a carpooling application adapted to Istanbul's specific urban milieu. The proposed approach aims to use technology to facilitate ride-sharing, reduce the number of vehicles on the road, and create a more environmentally friendly urban transportation system. The effectiveness of such technology interventions in considerably reducing traffic congestion, however, necessitates further examination to determine their potential implications and scalability within Istanbul's urban infrastructure.

1.2 AIM OF STUDY

The purpose of this research is to design, develop, and deploy a user-centric carpooling online application to alleviate traffic congestion in Istanbul through effective ride-sharing. This project aims to better understand and meet the transportation needs of Istanbul citizens by offering a scalable, secure, and user-friendly platform that supports sustainable urban mobility. This study aims to provide a practical solution to traffic congestion and its associated environmental impacts by taking a comprehensive approach that includes planning, design, development, deployment, and continuous maintenance. Furthermore, this study seeks to assess the application's usability, efficacy, and influence in promoting carpooling as a viable alternative to individual car usage in Istanbul's urban context.

1.3 THESIS LAYOUT

Following this chapter, the thesis is structured as follows:

In Chapter Two, we present general information about carpooling, including an examination of static and dynamic carpooling ideas. A detailed literature analysis is undertaken to place our research in the context of existing solutions, identifying the gaps that our initiative seeks to remedy.

In Chapter Three, we describe our technique, including the technology tools and frameworks used. We dig into the analysis and design processes, explaining how to create functional requirements, use case diagrams, class diagrams, and sequence diagrams. Project management solutions, such as using Gantt chart and GitHub for version control, are also discussed.

In Chapter Four, we go over the creation of the carpooling web application, from the homepage design to the complete app interface and functionality. This chapter focuses on the practical implementation of the design and development methodologies discussed in earlier chapters.

In Chapter Five, we wrap up the thesis by reviewing the research findings, discussing the carpooling application's impact on traffic congestion in Istanbul, and suggesting future research directions. The references and Appendix A are supplied at the conclusion to provide comprehensive assistance and additional resources for the study.

2. GENERAL INFORMATION

2.1 CARPOOLING

Carpooling consists of the joint use of a vehicle by a non-professional driver and one (or more) passenger(s) with the aim of making all or part of a common journey. This form of transport has contributed to improving the environmental, social and economic image of the car by reducing CO2 emissions, car dependency, transport costs incurred in the context of individual mobility, etc.

Carpooling also allows us to restore a certain level of communication that has disappeared in public transport by strengthening social bonds by grouping people together in the same car.

Furuhata et al., (2013) [3] define carpooling as a "mode of transportation in which individual travelers share a vehicle for a trip and share travel-related costs such as gasoline, tolls, and parking fees with others who have a similar route and schedule."

This definition of carpooling includes two forms of carpooling as Furuhata et al. (2013) mention it:

a) Disorganized carpooling where participants organize themselves directly among themselves in order to carpool without the help of a third party. This form of carpooling has a long history and usually involves family members, neighbors, friends, work colleagues, or even strangers.

b) Organized carpooling is operated by agencies which offer matching opportunities between participants who a priori have no common history and who wish to establish a carpooling group. This type of carpooling allows a large-scale move given that the potential participants do not only belong to the direct environment of the participants. An important part of this type of carpooling is the presence of arrangements prior to the carpooling activity which are made by the agency in order to match carpooling requests and offers.

(Figure 1.1) shows an example of a carpooling system where three requests can be served by a single driver. This consequently reduces travel time and thus alleviates traffic jams (Silwal et al., 2019) [4].

Figure 2.1: A Carpooling System Where Three Requests are Served by a Single Driver from Their Sources (S) to Their Destinations (D) (Silwal et al., 2019).

There are several variations of the carpooling problem, most of which are developing efficient mobility systems that allow travelers to share their rides and thus improve their travel experience. Carpooling trip planning can be classified into “static” carpooling and “dynamic” carpooling.

In static ridesharing, the supply and demand of drivers and passengers, respectively, is known in advance (i.e., origins, destinations, departure and arrival times of travelers are data in advance) and can thus be used to plan their shared journeys. These pre-established services are mainly used to plan daily trips (home-to-work) as well as shared long-distance trips (e.g. intercity trips). However, long-distance commutes generally have more flexible schedules than commutes. This uses the second variant of carpooling, namely dynamic carpooling which makes it possible to match peers (offers, requests) in real time or almost.

2.1.1 Static Carpooling

Carpooling is often static in the sense that it needs substantial planning and is difficult to change once arranged. Carpooling does not lend itself well to spontaneous travel. Work hours, errands, and unforeseen events necessitate transportation systems that allow for ad hoc travel. [5]

Static carpooling is ensured by a non-automated system accessible via the web, integrating the functionalities for managing carpooling offers and requests. Most existing systems fall into this category. The use of such systems requires prior registration in order to have access to the various functionalities offered through these supports. Unfortunately, in the general case, these functionalities boil down to a simple submission or consultation of a carpooling offer or request depending on whether the Internet user is a driver or passenger. Indeed, it is assumed that all offers and demands of drivers and passengers are known in advance before executing a matching process.

2.1.2 Dynamic Carpooling

Dynamic carpooling is characterized by great flexibility of use and less interdependence than static carpooling. This involves being able to find a shared route within a short period of time (less than half an hour for example) for a given route and depending on the position of the vehicles of potential drivers. Access to the service and its mode of operation must be easy and flexible.

Ride-sharing is reckoned to be an automated system where a service connects drivers and passengers based on similar itineraries and timings for carpooling in a personal vehicle. These systems are inherently dynamic, and flexible to adapting to the ever-changing needs and schedules of its users (Prieto et al., 2017) [6]. The major complexity of these systems lies in the need to align individuals' spatial and temporal restrictions. Both drivers and passengers must set their requirements — specific pick-up times, and start and end points — before a ride can be arranged and carried out. Typically, drivers adhere to predetermined paths and schedules. Consequently, ride-sharing services must be versatile in order to accommodate occasional deviations from these paths to collect and drop off passengers, provided these diversions remain within the drivers' acceptable range of additional travel (Cici et al., 2015) [7].

A typical scenario of a dynamic carpooling service can be described as follows:

a. The driver informs the service via an application on his Smartphone that he is offering a trip as a driver; he indicates his time and point of departure, his destination as well as the number of passengers he can accommodate in his vehicle.

b. A few minutes before departure, the passenger who requests to take a carpooling route contacts the service via the same application to indicate their request.

c. The service then looks for a driver present on the requested route and who respects the time constraints. Once found, the service connects driver and passenger. All the driver has to do is finalize the route by determining the meeting point and using the satellite geopositioning (GPS) of their phone.

Despite the advancements witnessed in technology and the efforts invested therein, a bunch of systems that provide real-time carpooling services are still in their embryonic phases, primarily due to security concerns and gaps in automation. Moreover, while platforms that facilitate dynamic carpooling provide the benefit of accessing a real-time list of available rides, they often disregard the optimization of ride assignments. Unlike static systems, these dynamic platforms do not encapsulate search algorithms that could assign vehicles and queries based upon optimization of specific factors such as cost, trip duration, vehicle capacity, and the coordination of transfers.

2.2 LITERATURE REVIEW

Traffic congestion is a significant issue in urban areas, leading to increased travel time and carbon emissions. Carpooling is a potential solution to reduce traffic congestion and promote sustainable transportation. The review draws upon two studies in the field, Boukhater et al. (2013) and Caballero-Gil el.al (2017), to provide insights into carpooling app design, user behavior, and benefits.

2.2.1 Carpooling App for a Green Social Solution to Traffic and Congestions (GA)

In [8] the authors propose a (genetic algorithm) GA with a customized fitness function that searches for the solution with minimal travel distance, efficient ride matching, timely arrival, and maximum fairness while taking into account the riding preferences of the carpoolers. Moreover, addressing the problem of traffic congestions, parking problems, inordinate fuel consumption, and excessive pollution. The carpooling problem has been divided into two: a Daily Carpooling Problem (DCPP) or a Long-term Carpooling Problem (LTCPP).

The solution space is represented as a set of potential carpool configurations, each evaluated based on a composite fitness function that includes a number of key costs:

Distance Cost (D): This cost determines the additional distance a driver must travel to enable carpooling, with the goal of minimizing unnecessary detours.

Time Cost (T): It calculates the variation from preferred arrival times for all participants, stressing punctuality and schedule adherence.

Fairness Cost (F): This statistic provides an equitable division of driving responsibilities among participants, which promotes fairness.

Preferences Cost (P): Reflects individual preferences such as non-smoking, maximum passenger capacity, and so on, with the goal of aligning carpool arrangements with user demands.

Their proposal models introduces three sections: 1) server using SQLITE 2) client that is the user's portal to the system and Scheduler to ensure that all participants secure a ride. After different cases results of a survey they did, the preferences were given different priorities, which specify their impact on the solution quality. Smoking has the highest priority, followed by car capacity and age group. When preferences were taken into consideration, only 11.77% of the users had their smoking preferences unmet, versus 21.01% for the first case.

Moreover, 16.8% were unsatisfied with the age group, versus 26.05% for the first case. The car capacity numbers are close: 11.43% for the first case versus 18.75% for the second case.

The genetic algorithm iteratively refines carpool arrangements via selection, crossover, and mutation processes, driven by the fitness function, which averages the aforementioned costs. The goal is to evolve to an optimal set of carpooling arrangements that meet all participants' limits and preferences while reducing overall trip distance and ensuring fairness.

2.2.2 A Trust-Based Cooperative Social System

Caballero-Gil el.al (2017) [9] presents a trust-based cooperative social system applied to a carpooling platform for smartphones. The aim of the system is to increase the efficiency of carpooling services and to reduce the negative impact of individual car usage on the environment.

The proposed system uses trust as the foundation of its social interactions, enabling users to establish relationships based on mutual trust and cooperation. The system incorporates a range of features designed to foster trust, including user reputation and feedback mechanisms, social network analysis, and game theory techniques.

The carpooling platform provides users with a range of functionalities, including ride requests and offers, route optimization, and fare calculation. The system incorporates a range of algorithms designed to ensure the fairness of transactions and the safety of users. They have proposed an android application under the name of "Carpoolap" in which each user can see the routes he/she proposes as a driver, and whether potential passengers exist for those routes. The server was developed using JavaScript technologies by frameworks like ‘node.js’ and ‘express.js’. As a database for all the centralized data on this server, a No SQL database, such as MongoDB. The app proposed a trust algorithm.

The article presents the results of a series of simulations conducted to evaluate the effectiveness of the proposed system. The simulations demonstrate that the trust-based cooperative social system can significantly increase the efficiency of carpooling services and reduce the negative impact of individual car usage on the environment.

Overall, the article presents a novel approach to carpooling services that integrates trust and cooperation as core principles, offering a promising solution to the environmental and social challenges posed by individual car usage.

2.3 CONTRIBUTION

Designing a carpooling app is an effective way to reduce traffic congestion in urban areas such as Istanbul. Carpooling, also known as ride-sharing, allows individuals who are traveling to the same destination to share a car, thus reducing the number of cars on the road. This can help to reduce traffic congestion, air pollution, and travel time, while also promoting social interaction and cost-sharing among users.

The use of programming language PHP (MYSQL) in designing the carpooling app can be a good choice as it is a widely-used language for web development and has many features that can be helpful for designing a carpooling app. The MYSQL database can also be used to store user data and other relevant information needed for the app.

We can develop an example that estimates the reduction in emissions achieved by sharing a trip among numerous users. We'll take four users (A, B, C, and D) who intend to undertake the identical trip separately and compare them to the situation in which they share the trip using your app.

Assumptions:

a. Each user drives a similar car with an average CO2 emission rate of 120 g/km (a common value for a petrol car).

b. The distance of the trip for each user if traveled separately is 50 km.

c. When sharing a trip, we assume only one car is used, and the trip length remains the same.

Calculation of Pollution Generated by Individual Trips:

Individual CO2 Emissions: For each user driving separately, the CO2 emissions can be calculated as follows:

$$Emissions_{individual} = Distance \times Emission\ Rate \quad (2.1)$$

$$Emissions_{individual} = 50Km \times 120\ g/km \quad (2.2)$$

Total Emissions for 4 Separate Trips:

$$Emissions_{total,separate} = 4 \times Emissions_{individual} \quad (2.3)$$

Calculation of Pollution Generated by Shared Trip:

Shared Trip CO2 Emissions: Since all four users share a single car, the emissions are only for the one trip:

$$Emissions_{shared} = Distance \times Emission\ Rate \quad (2.4)$$

$$Emissions_{shared} = 50Km \times 120\ g/km \quad (2.5)$$

Results Calculation:

Let's calculate these emissions to quantify the pollution reduction.

a) Individual CO2 Emissions: Each user, when driving separately for a 50 km trip, would generate:

$$Emissions_{individual} = 50\text{km} \times 120\text{g/km} = 6000\text{g (or 6 kg) of CO2} \quad (2.6)$$

b) Total Emissions for 4 Separate Trips:

$$Emissions_{total,separate} = 4 \times 6000\text{g} = 24000\text{g (or 24 kg) of CO2} \quad (2.7)$$

c) Emissions for Shared Trip:

$$Emissions_{shared} = 6000\text{g (or 6 kg) of CO2} \quad (2.8)$$

Pollution Reduction:

By comparing the total emissions from individual trips with the emissions from a shared trip, the reduction in pollution can be calculated as:

$$Reduction = Emissions_{total,separate} - Emissions_{shared} \quad (2.9)$$

$$Reduction = 24000\text{g} - 6000\text{g} = 18000\text{g (or 18 kg) of CO2} \quad (2.10)$$

The example clearly demonstrates that by allowing four users to carpool for a single trip, our software can dramatically cut CO2 emissions. Instead of generating 24 kg of CO2 by driving alone, the users produce only 6 kg of CO2 by sharing a ride. This leads in a significant CO2

savings of 18 kilograms for a single shared journey. When multiplied across numerous trips and users, the potential impact on air pollution reduction is significant.

This tangible example highlights the environmental benefits of carpooling and the importance of our app in supporting sustainable travel choices. It makes a persuasive case for the use of carpooling to reduce air pollution in metropolitan areas.

3. METHODOLOGY

3.1 INTRODUCTION

In this chapter, we describe the application work and implementation details of our work. We will first describe the technologies and implementation language. Subsequently, we present the database used to validate and train our system. We then detail the management of the project by including the Gantt chart and the management of GitHub.

3.2 TECHNOLOGIES

a) HTML [10] is a language used for structuring web pages, and its more stringent variant, XHTML, is also a markup language for the web. Previously, HTML was employed for both organizing the page structure and managing its visual style. Nowadays, these tasks are clearly separated, with X/HTML strictly used for defining the page's structure, including elements like titles, subtitles, paragraphs, images, input forms, and hyperlinks. It often serves as the primary or sole framework recognized and utilized by software that accesses web pages, such as search engines or text-based browsers. Pages whose HTML code does submit to any changes through JavaScript or PHP – either before or after being displayed in the browser – are referred to as “static” pages.

b) CSS [11] is known to be a style sheet language used to enhance the presentation of web pages. Through Cascading Style Sheets (CSS), developers can alter the appearance of HTML elements, including their color, size, and font, as well as their positioning, width, height, and stacking order on the page — essentially, all aspects that define the layout of an HTML document. This capability allows the same X/HTML document to appear distinctly without altering its underlying structure, by simply changing the CSS rules applied to it. Such separation between structure and presentation not only

facilitates the construction of web pages but also their ongoing maintenance and development.

c)  JavaScript [12] client-side programming language JavaScript is a language that is read and executed by your browser – the client – (therefore directly on your computer) and which allows you to create reactions in response to events on the page or to user actions. Previously, JavaScript code was often integrated into HTML code, but again, today we recommend the separation of the two languages for reasons of organization of the code, but also of accessibility of the site. JavaScript itself is capable of writing HTML or CSS code, and therefore of completely modifying the basis of a Web page.

d)  PHP [13] is considered a server-side programming language that operates on the server hosting the X/HTML page. It processes requests before they are sent to the browser that requested them. To illustrate, when a user fills out a form and clicks on a button, the server receives this data, and PHP processes it to generate the X/HTML for the new web page, which is then sent back to the browser. Therefore, the X/HTML content of a page can be completely generated by the means of PHP, or PHP can be utilized to insert specific snippets of code within a page to achieve HTML, CSS, or JavaScript code.

e)  MySQL [14] is a database management system frequently used in conjunction with the PHP programming language. It utilizes Structured Query Language

(SQL) to perform operations such as querying, populating, or updating databases. Within the context of a website, PHP typically manages the connection to the database and sends the SQL commands required for database operations. MySQL then processes these commands and sends back the resulting data or information, which is further treated by the PHP code. It is noteworthy that there other remarkable database systems which encompass Oracle and Microsoft SQL Server.

f)  Unified Modeling Language (UML) [15] is a graphical modeling language that uses pictograms to offer a standardized way to visualize the design of a system. This language is widely used in the fields of software development and object-oriented design, providing a clear and organized method to diagram the structure and behaviors of systems.

j)  Visual Studio Code [16] is a versatile and efficient source code editor that operates on desktop environments and supports Windows, macOS, and Linux. Basically, it integrates support for languages like JavaScript, TypeScript, and Node.js, and boasts an extensive array of extensions that accommodate additional programming languages (such as C++, C#, Java, Python, PHP, Go) and various runtimes (including .NET and Unity).

h)  Git [17]: Git Bash is a Microsoft Windows application designed to offer an emulation layer for Bash, which stands for Bourne Again Shell. A shell is considered to be a terminal application that facilitates interaction with an operating system. Git Bash encompasses a package that installs Bash, several standard Bash utilities, and Git on a Windows system, enhancing the user's ability to manage version control and run Bash commands within a Windows operating system.

i)  GitHub [18] is a software development and service company headquartered in the United States. GitHub notably develops the GitHub platform, the Atom text editor and the Electron framework.

j)  PhpMyAdmin [19] It is a management web application for MySQL database management systems made in PHP. This practical interface makes it possible to execute, very easily and without much knowledge of databases, queries such as data table creations, insertions, updates, deletions and modifications of database structure, as well as the granting and revocation of rights and import/export. This system allows you to conveniently save a database as a .SQL file and transfer your data to it, even without knowing SQL. PhpMyAdmin comes with MAMP and XAMPP and WAMP.

k)  Bootstrap [20] is a comprehensive toolkit used for developing the design aspects of websites and web applications, including graphics, animations, and page interactions. This toolkit encompasses a collection of HTML and CSS-based design templates for elements like forms, buttons, and navigation tools, along with optional JavaScript extensions. Bootstrap is renowned for its popularity and is one of the most actively developed projects on the GitHub platform.

l)  jQuery [21] is a versatile, open-source JavaScript library that is free to use and functions across multiple platforms. It is compatible with all major web browsers, including Safari, Chrome, and Firefox to name but a few. Created in 2006, jQuery simplifies the process of scripting. It is the most recognized and extensively utilized JavaScript

framework. This library enables manipulation of HTML, CSS, JavaScript, and AJAX codes and primarily operates on the client side.

m)  WampServer (Windows Apache MySQL Php)[**22]:** is a Windows web development platform for dynamic web applications, It enables you to build up web servers on your local devices in order to test websites in a full development environment using Apache2 server, PHP scripting language and MySQL database. It also has PhpMyAdmin to manage your databases more easily.

n) AJAX (asynchronous JavaScript and XML) [23]: is not a technology and even less a programming language. It is in fact a web programming concept that is based on several technologies such as XML, JSON and JavaScript. The objective of AJAX is to ensure that a web page communicates with the web server without causing the page to reload. This is why AJAX's preferred language is JavaScript. AJAX allows you to create more interactive and responsive pages by revitalizing and refreshing certain parts of these pages without having to completely reload the whole thing for each new user request. It is a set of new technologies intended to promote the transfer of data between the server and the web browser.

o) MVC (Model – View – Controller) [24]: A design pattern that offers a general solution to the problem of structuring an application. MVC defines rules that determine in which layer of the architecture, and in which (object-oriented) class of this layer, a specific functionality must be integrated. An application that complies with these rules is easier to understand, manage and modify. These rules are the result of a process of experimentation and development of best practices which resulted in a standard architecture.

Here is a simple illustration to help you quickly understand the logic of this architecture:

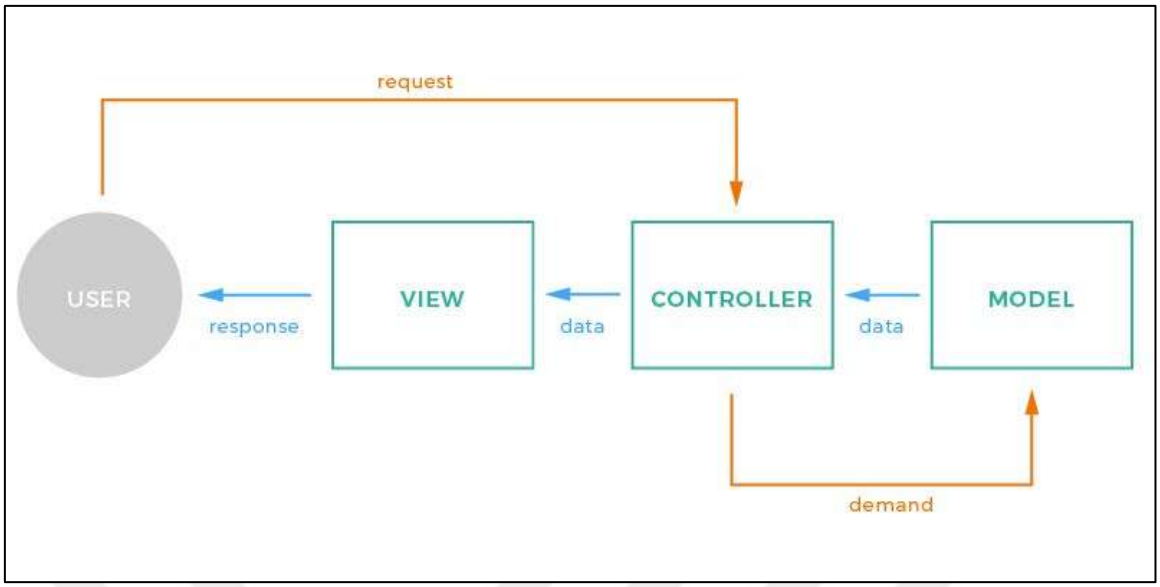

Figure 3. 1: Architecture MVC

Each user action goes through the controller which sends requests to manipulate an object to the model. The model makes changes to the object and returns it to the controller which in turn passes it to the view. So, these three poles have different responsibilities:

a) The model contains the data and its logic.

b) The View contains the graphical presentation to be returned to the user.

c) The controller processes user actions (via requests), asks the model to make changes, then passes the data to the view.

The controller therefore has a central place in this architecture. It is the bridge between user interactions and data processing. Everything goes through him in order to be able to control everything.

3.3 ANALYSIS AND DESIGN

3.3.1 Analysis and Specification of Functional Requirements

The specification of functional needs will allow us to have a better approach to users and their functionalities.

Actors:

In this phase we will identify the actors interacting with the system and who are as follows:

Admin:

a) Login/Logout.
b) Add, Edit, Delete, Disable Vehicles.
c) View Reports.
d) Add, Edit, Delete users.
e) Validation of users.
f) Add, Edit, Delete simple Locations.

User:

a) Register/Login/Logout: Create/ Update an account or log in.
b) View trips.
c) Add, Edit, Delete, Requests for trips.
d) Add, Edit, Delete, Review for trips.
e) Add, Edit, Delete vehicles.

3.3.2 Use Case Diagram

In UML, use case diagrams are deployed to model the behavior of a system and are instrumental in capturing system requirements. These diagrams provide a high-level overview of the system's functions and scope. Additionally, use case diagrams delineate the interactions between the system and its various actors, helping to clarify how users and other systems are affected by each other. [25]

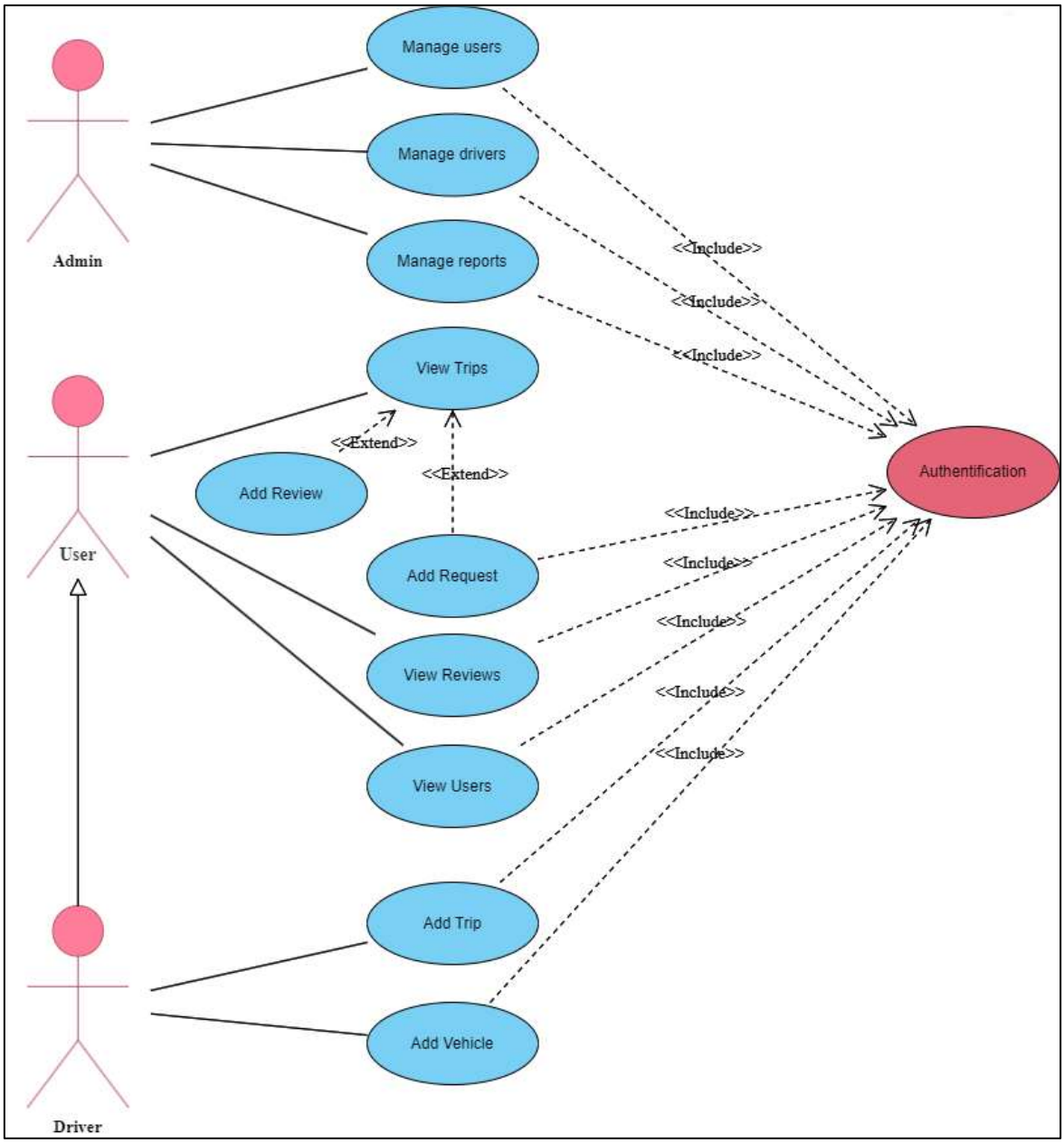

Figure 3.2: Use Case Diagram.

3.3.3 Class Diagram

In software design and modeling, a class diagram serves as a fundamental tool to describe classes and their relationships. This type of diagram enables high-level modeling of software architecture, allowing designers to visualize the structure without having recourse to the source code. Each class represented in a class diagram directly corresponds to a class in the source code, facilitating a clear grasp of the software's design and organization.

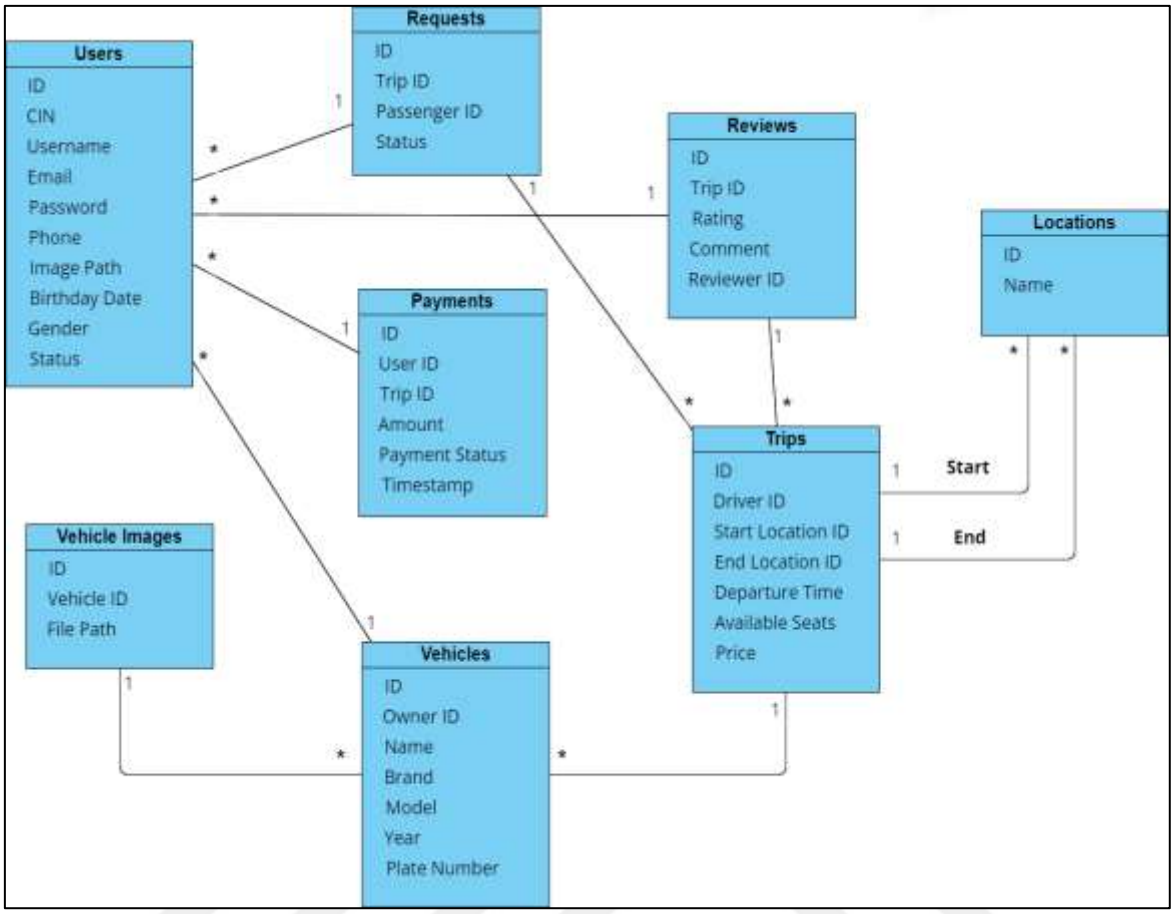

Figure 3.3: Class Diagram.

3.3.4 Sequence Diagram

UML sequence diagrams are reckoned to be a type of interaction diagram that illustrates how operations are carried out through detailed interactions between objects within a collaboration context. These diagrams are focused on the timing of interactions, using the vertical axis to represent time. They visually demonstrate the sequence of interactions by showing the order in which messages are sent and received, effectively mapping out the flow of communication over time.

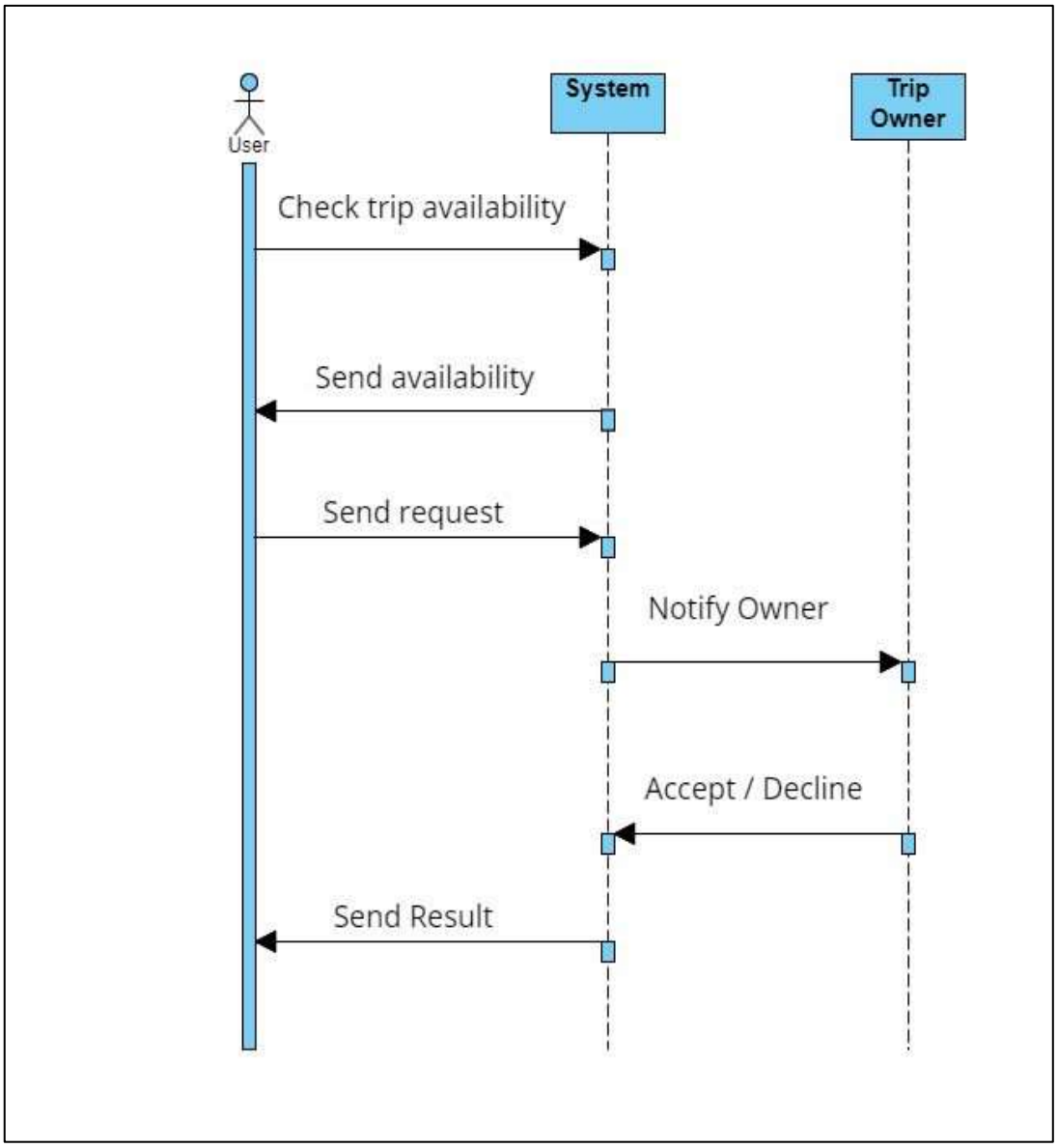

Figure 3.4: Sequence Diagram.

3.4 PROJECT MANAGEMENT

To manage our project, we used the GitHub task management platform in the web applications to divide tasks and know the progress of the project and moreover to accelerate development.

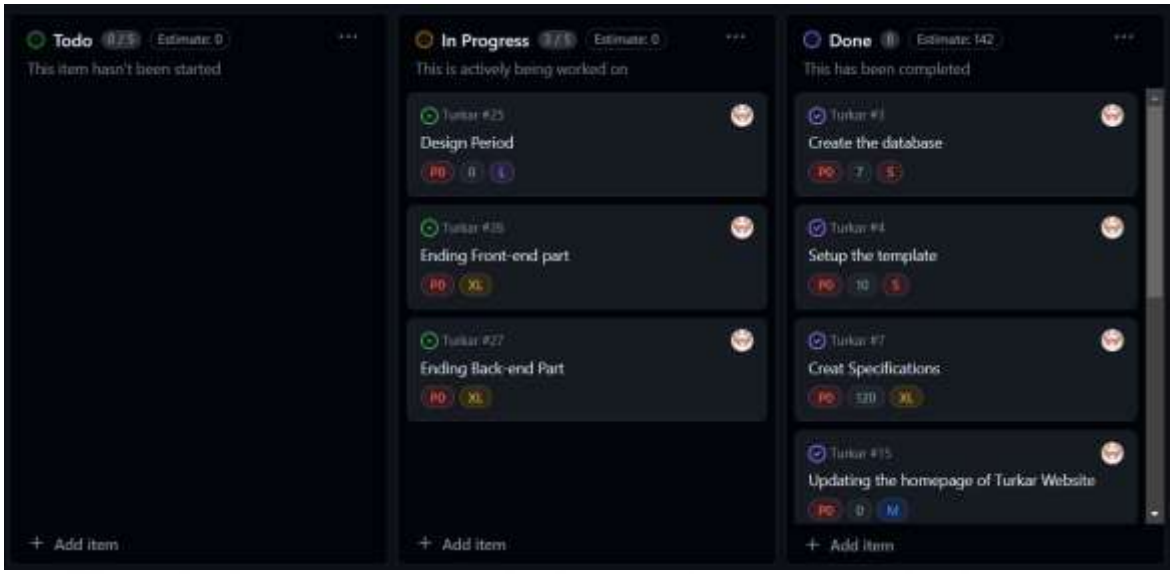

Figure 3.5: Project Management.

3.4.1 Gantt Chart Overview

The Gantt chart is a fundamental visual tool in project management, designed to display the progress of various activities or tasks within a given project. It features a diagram where tasks are listed in the left column, whereas the header row indicates the units of time appropriate for the project, such as days, weeks, or months. Each task is epitomized by a horizontal bar; the placement and length of the bar show the task's start date, duration, and completion date. This setup allows for immediate visualization and tracking of the project's timeline and progress. [26]

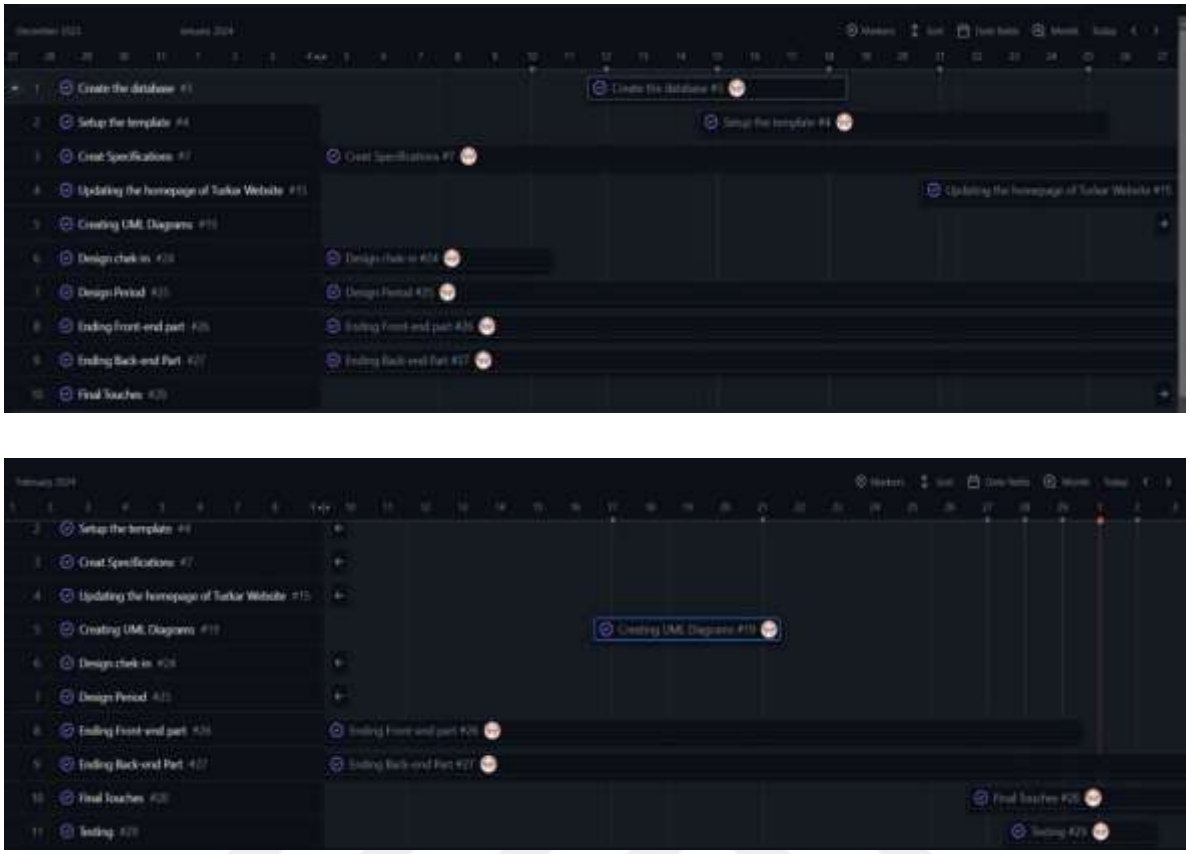

Figure 3.6: Gantt Chart.

3.4.2 GitHub for Version Control

Effective version control is critical during the development of our carpooling platform to maintain code integrity, cooperation, and traceability. To accomplish this, we use GitHub, a popular platform for hosting Git repositories and managing collaborative software development workflows.

Benefits of GitHub Integration:

GitHub provides various benefits for version control in our project:

a. Centralized Repository: GitHub serves as a centralized site for storing our project's source code, making it available to all team members from anywhere with an internet connection.

b. Version History: GitHub's version control system allows us to keep track of any codebase changes, including who made them and when. This allows us to follow the project's evolution over time and revert to prior versions as needed.

c. Collaborative Development: GitHub allows numerous developers to work on the same codebase at the same time, facilitating team cooperation. Branches and pull requests enable developers to propose changes, review each other's code, and seamlessly merge contributions.

d. Issue Tracking: With GitHub's integrated issue tracking system, we can easily log and prioritize tasks, issues, and feature requests. This improves communication and guarantees that no concerns are missed throughout development.

Workflow Overview:

Our GitHub workflow is organized to manage code changes and streamline collaboration:

a. We use a branching technique in which each feature, bug repair, or enhancement is developed in a different branch. This method enables us to isolate and test modifications individually before incorporating them into the main codebase.

b. Pull Requests: When a developer finishes working on a branch, they create a pull request (PR) on GitHub to submit the modifications for review. PRs serve as a tool for peer review, allowing team members to provide feedback, debate potential improvements, and assure code quality prior to merging.

c. Continuous Integration (CI): We include CI solutions like GitHub Actions into our workflow to automate testing and code validation processes. Each pull request triggers an automated CI pipeline that runs tests, lints code, and performs other checks to maintain code quality and avoid regressions.

d. Code Reviews: As part of our workflow, team members thoroughly examine each other's code modifications. GitHub's built-in review tools enable reviewers to offer comments, suggest improvements, and approve changes prior to merging.

Conclusion:

By using GitHub for version control, our project gains access to a rich and collaborative development environment. GitHub's features enable our team to work effectively, maintain code quality, and ensure the success of our carpooling platform's development path.

4. IMPLEMENTATION

4.1 INTRODUCTION

After presenting the project and the tools used then comes the study and testing and validation phase of the project in which we will present the platform created and its basic functionalities through screenshots.

4.2 PRESENTATION OF HOMEPAGE

Homepage:

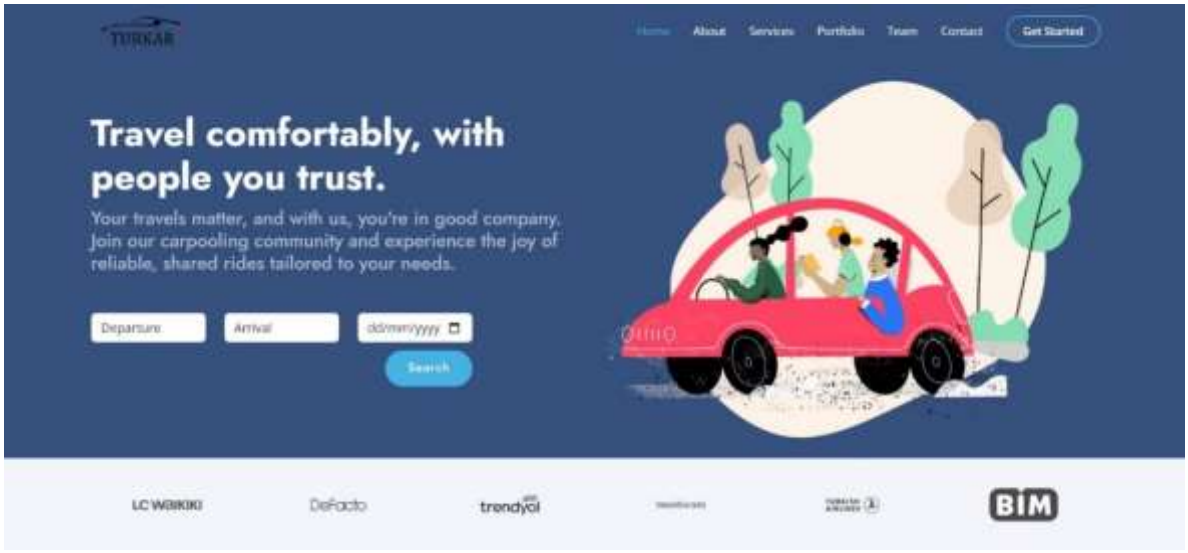

Figure 4.1: Homapage.

This home page contains a header with a search form for customer offers, popular trips, recent offers, featured regions.

About Us:

Figure 4.2: About US.

Welcome to TURKAR:

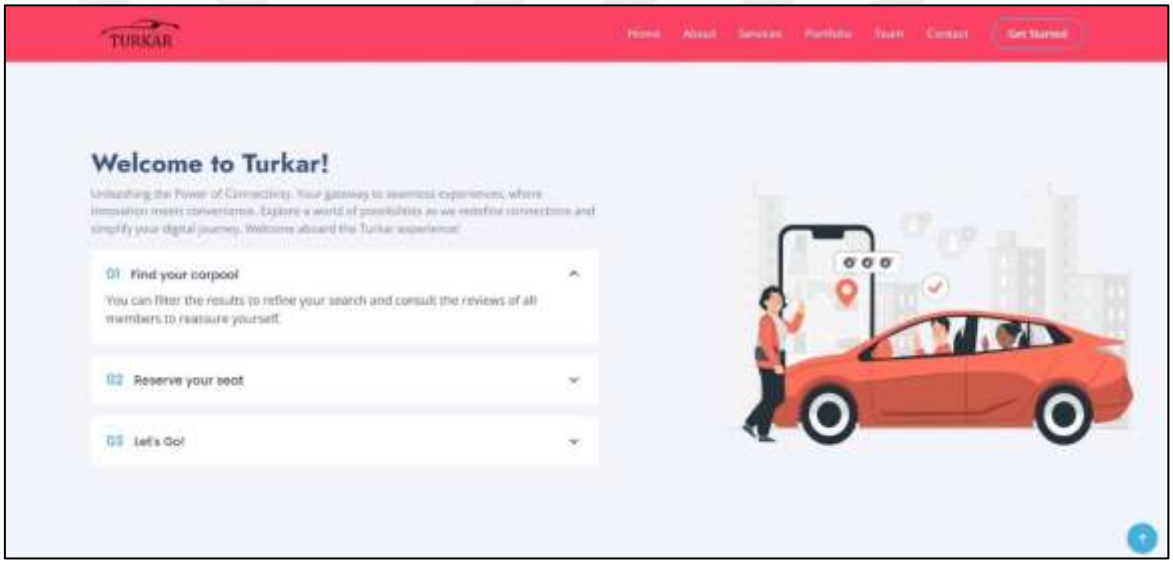

Figure 4.3: Welcome to TURKAR.

Improve İstanbul's commute:

Figure 4.4: Improve İstanbul's Commute.

Services:

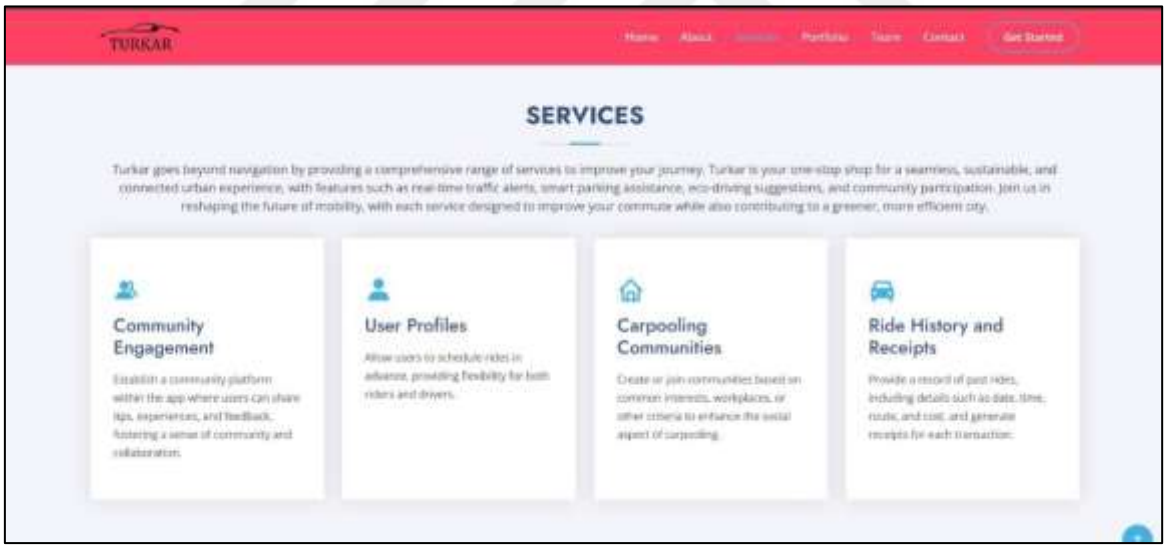

Figure 4.5: Services.

Reviews:

Figure 4.6: Reviews.

Team:

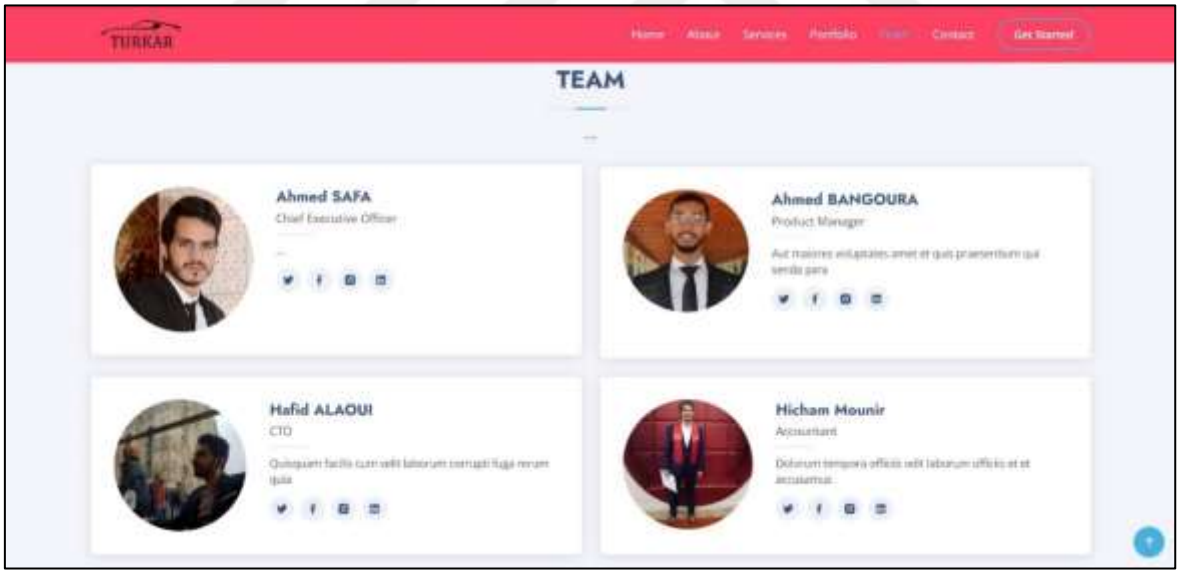

Figure 4.7: Team.

Pricing:

Figure 4.8: Pricing.

Frequently Asked Questions:

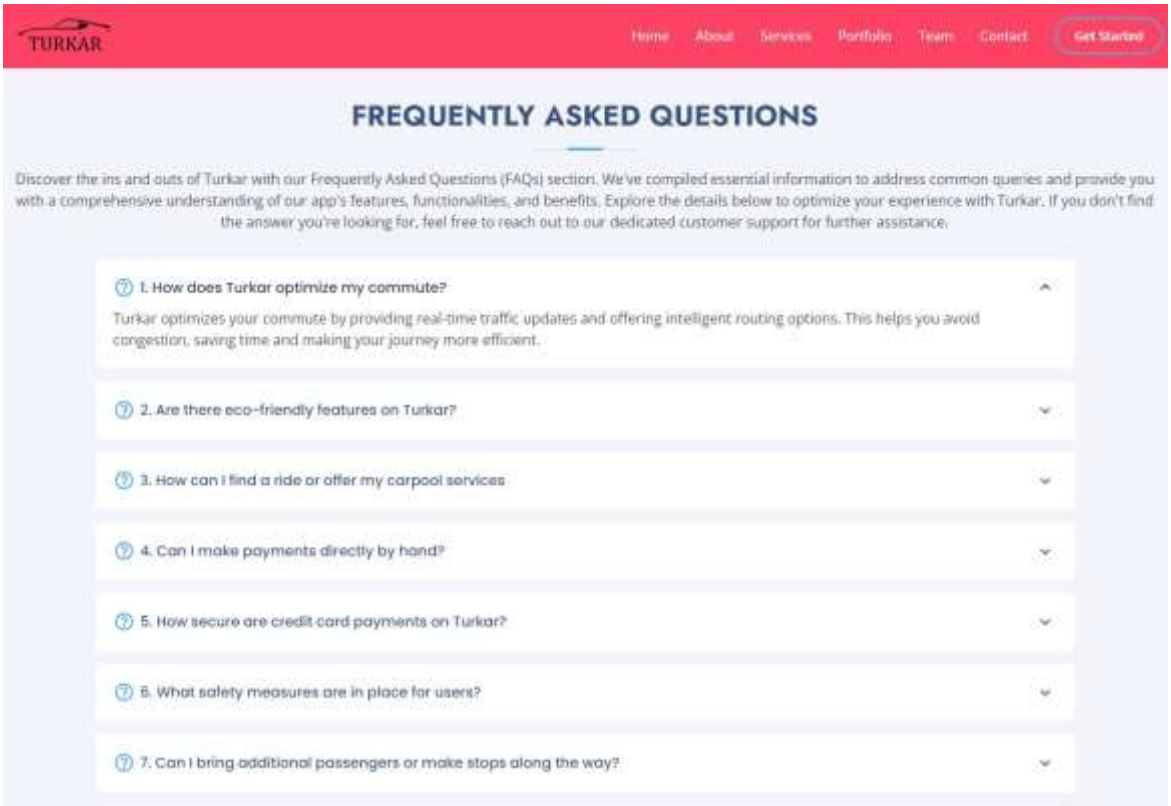

Figure 4.9: Frequently Asked Questions.

4.3 PRESENTATION OF THE APP

Authentication page:

The authentication interface which allows you to connect by entering your email and password. Or you can choose to connect using your Facebook or Google Account.

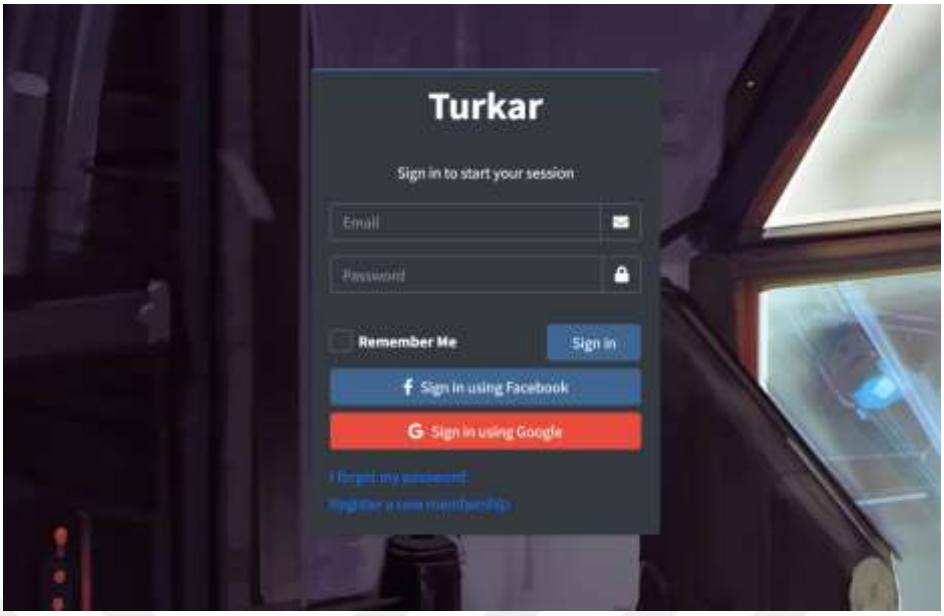

Figure 4.10: Authentication page.

Registration Page:

This interface allows you to register new membership.

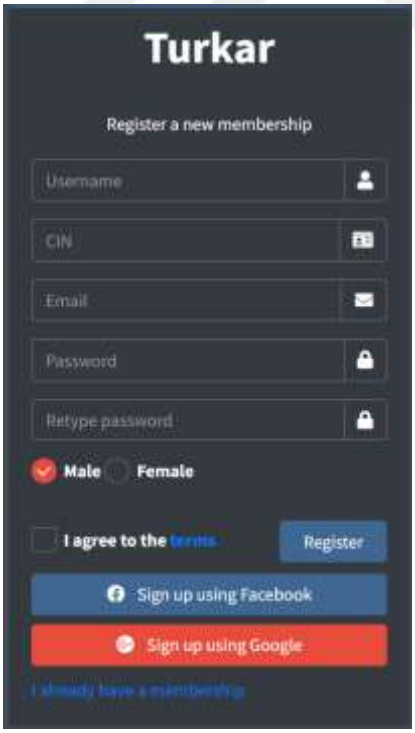

Figure 4.11: Registration Page.

Homepage:

this is the homepage, first page that is displayed after authentication, you will find in this page a statistic of the number of vehicles plus, trips plus the number of users in our platform. you can also see a list of the latest trips created which are not yet started with 2 buttons one to create a new trip and the other to see all the trips. and last you can see a list of famous trips organizers.

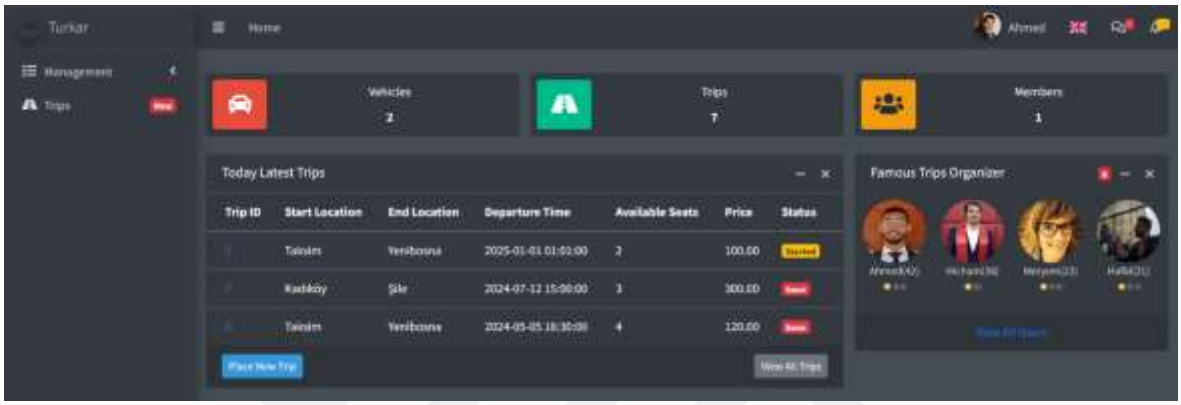

Figure 4.12: Homepage.

You can also see that there is a menu that takes you to different management pages, as well as a button to access the list of routes, where you can apply different actions to these routes.

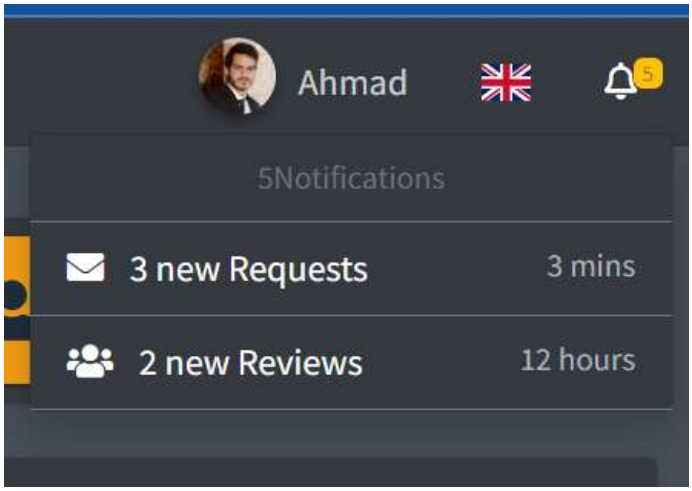

Figure 4.13: Notifications.

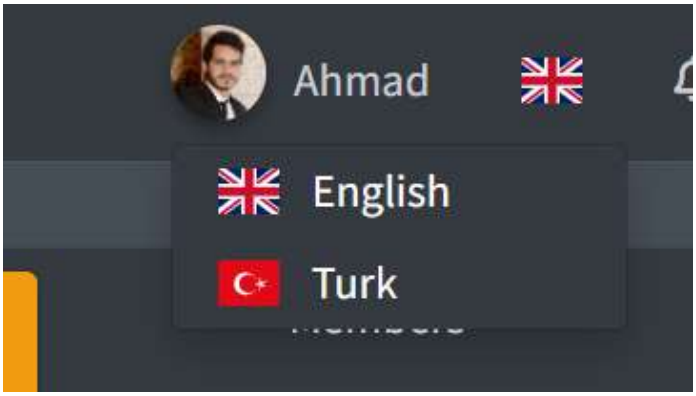

Figure 4.14: Languages.

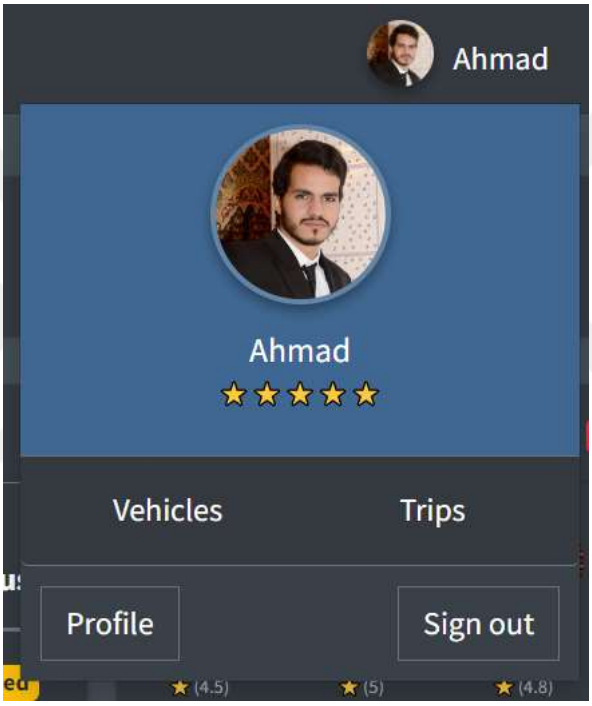

Figure 4.15: User.

You can also see at the top of the platform a navigation bar that has too many options like a button that can display a small notifications window plus a button to select a language also the user button that displays a small window that allows you to navigate to your vehicles or trips also your profile and finally you can find the logout button.

Profile:

The Profile page lets you view and update all your information.

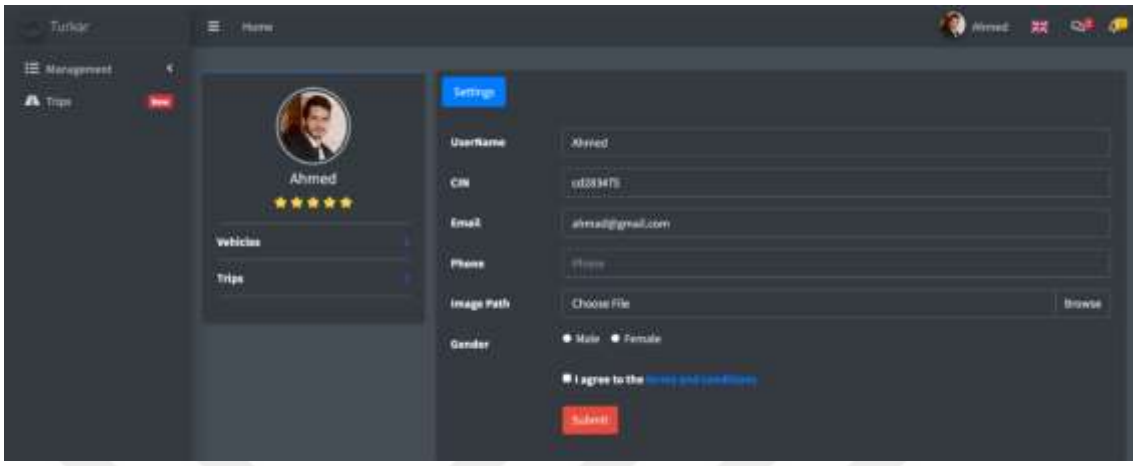

Figure 4.16: Profile.

Trips:

When you click on the Trip button in the menu, a list of trips is displayed, with a text field for searching, or by clicking on the header of each column, you can sort the column, you can find also buttons in each row to request to join a trip or add a payment, or to view trip details, and a button to add a review of the trip. You can also find another options like download the list as PDF,CSV,Excel,….

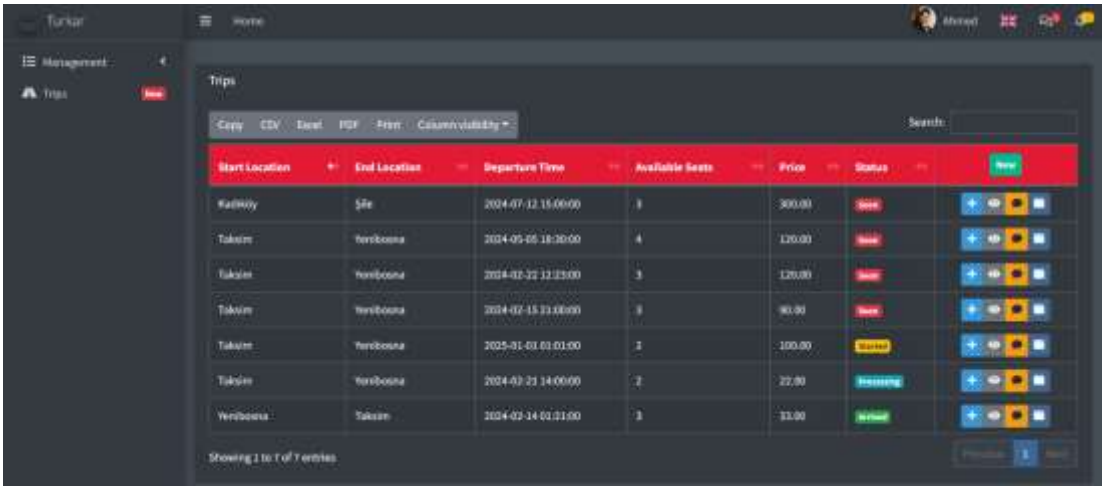

Figure 4.17: Trips

when you click on the new button in the trip list, you'll be redirected to this form, where you can add a new trip.

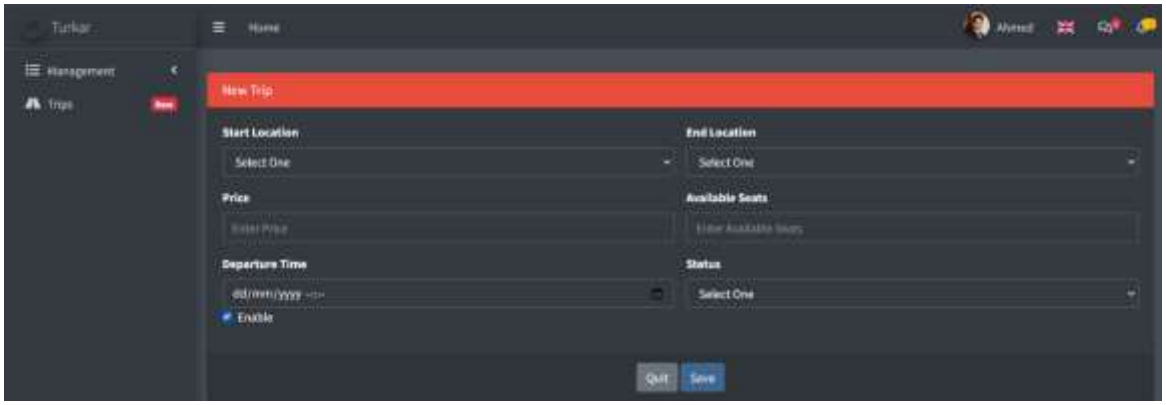

Figure 4.18: New Trips.

Clicking on the "eye" button in the trip list will take you to a page where you can see the details of this trip, its owner and the list of passengers who will be sharing the trip with you.

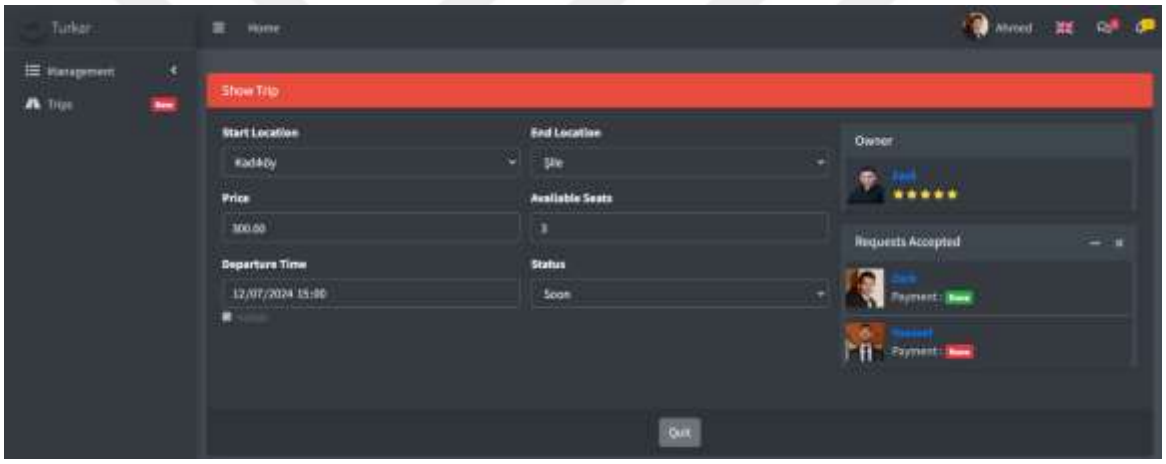

Figure 4.19: Show Trip.

Clicking on edit trip you can see the requests of the people who will join the trip and after that you will decide whether to accept them or not.

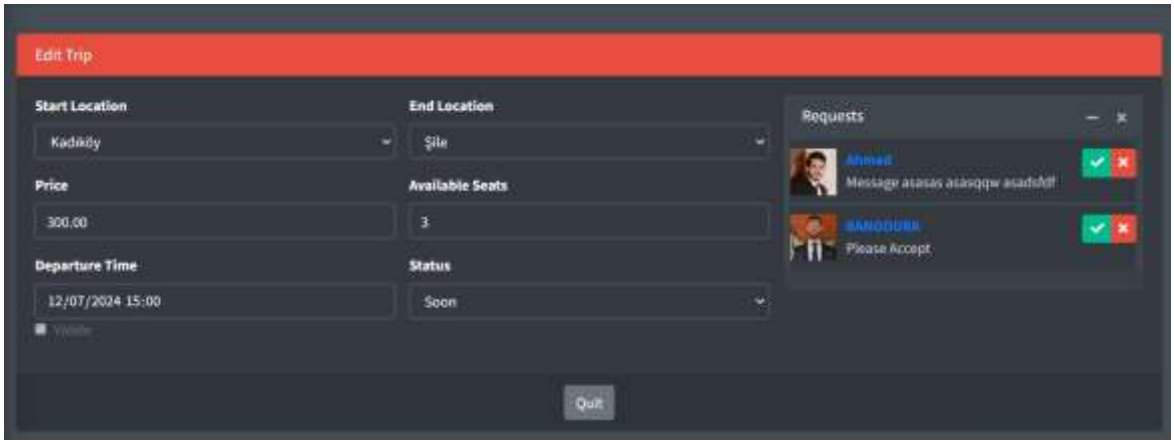

Figure 4.20: Edit Trip.

Locations:

When you click on the Locations button in the menu, a list of Locations is displayed, with a text field for searching, or by clicking on the header of each column, you can sort the column, you can find also buttons in each row to manage those locations. You can also find another options like download the list as PDF,CSV,Excel,….

Figure 4.21: Locations.

Vehicles:

By clicking on the Vehicles button in the menu, a list of Vehicles is displayed, with a text field for searching, or by clicking on the header of each column, you can sort the column, you can find also buttons in each row to manage those Vehicles. You can also find another options like download the list as PDF,CSV,Excel,….

Figure 4.22: Vehicles.

Clicking on the "eye" button in the Vehicles list will take you to a page where you can see the details of this vehicle.

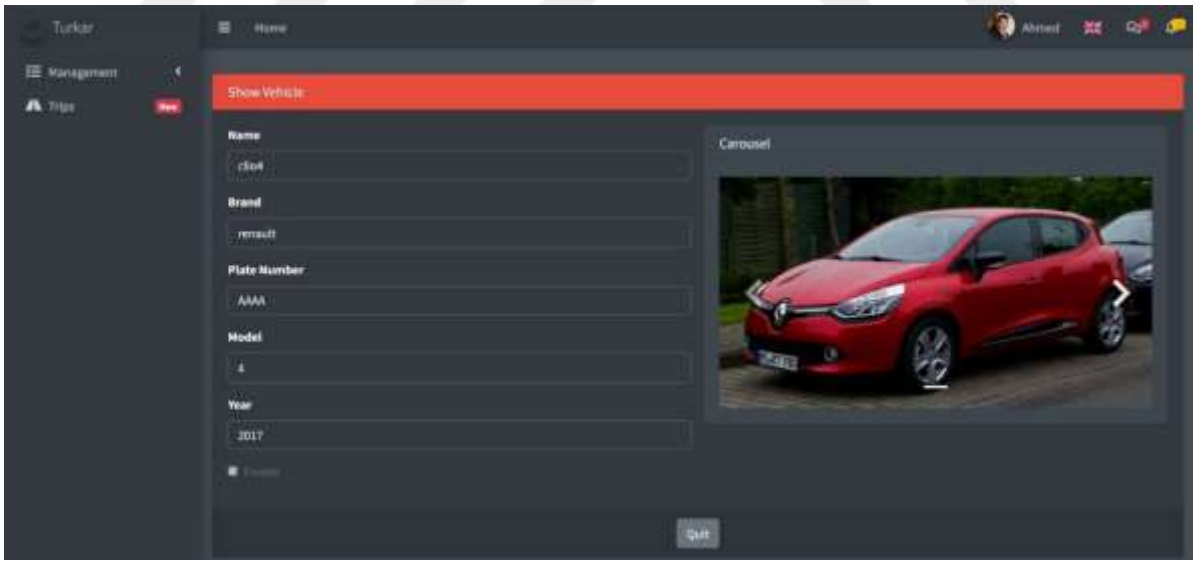

Figure 4.23: Show Vehicles.

Reviews:

Clicking on a user's name takes you to that user's profile. You can then see the user statistics, rating, personal information (address, phone number and national ID Carde) and a list of user reviews so you can have an idea about that user whether he is a good or bad fit for your trip.

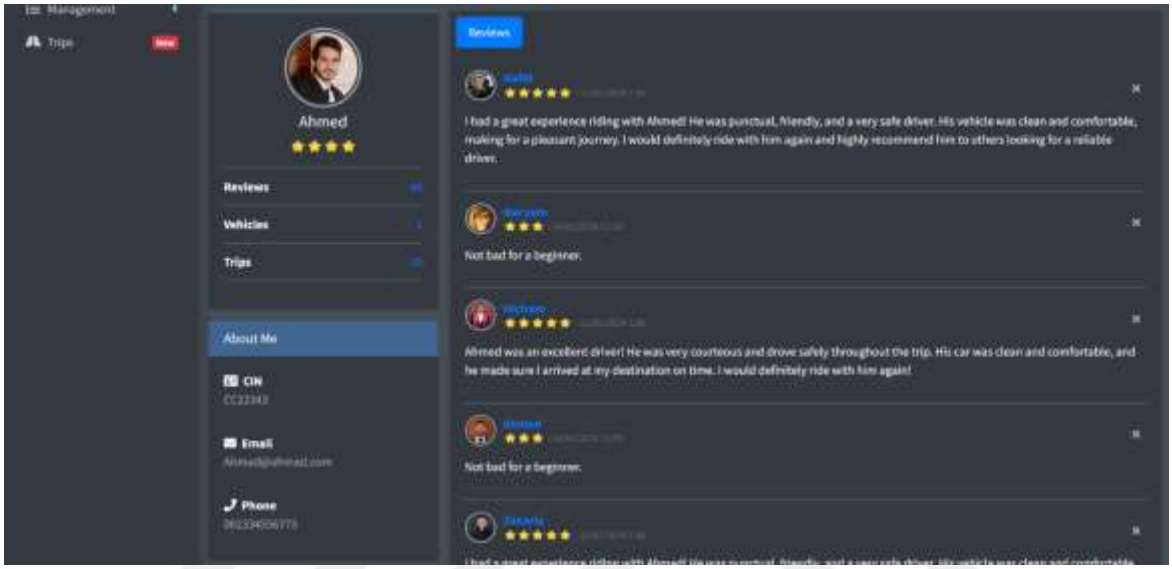

Figure 4.24: Reviews.

In this chapter, we have documented everything about our project. And finally, we gave an overview of the interfaces created in the TURKAR platform.

5. CONCLUSION

The creation of this web platform allowed us to live an experience full of ambitions, complex and interesting. I am happy to have had this subject which I find very interesting in terms of production. It allows me to try several technologies at the same time and deepen my knowledge of certain more complex technologies and even train myself to broaden my experience in technical terms and even work tools (software).

This project also calls on my creativity which appears over time as well as the development of a high sense of reflection on how to better detect user needs. This allows us to realize the level of difficulty of a project between what we want to achieve and the constraints we must respect and adapt in any case. This project will bring us a lot in terms of technical knowledge.

The project requires creating a carpooling platform and collecting user information to create more reliable, smoother, easier, and more comfortable trips. In this project, I had to achieve three objectives: study new technologies, create a platform with these technologies and secure this platform.

Today, traffic congestion has become a major problem in many urban areas all over the globe, which leads to economic, social, and political impacts. Carpooling aims to lower the cost of travel for commuters, which not only saves them money but also reduces the use of the most important non-renewable resource we have, as well as providing a transportation alternative that can potentially lessen traffic congestion [26].

This carpooling application is made by PHP (MYSQL) as a programming language, and it is fast, scalable, expandable, and highly available. It also safeguards the privacy and security of the users' data. The application is also easily maintainable, since it can be improved in a variety of ways.

Overall, the development of carpooling software intends to provide a practical solution to the problem of traffic congestion in Istanbul. The software is expected to inspire more people to share rides, reduce the number of cars on the road, and contribute to a more sustainable urban transportation system in the long run.

REFERENCES

[1] https://worldpopulationreview.com/world-cities/istanbul-population, published in 2023.

[2] "The solutions to traffic congestion in Istanbul," by Emrah Akyuz, published in the Journal of Academic Social Sciences, 3(16), pages 442-449, September 2015.

[3] Ridesharing: “The state-of-the-art and future directions”. Transportation Research Part B: Methodological, 57, 28-46. By Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M. E., Wang, X., & Koenig, S. (2013).

[4] “A survey of taxi ride sharing system architectures”. In 2019 IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 144-149). IEEE. By Silwal, S., Gani, M. O., & Raychoudhury, V. (2019, June).

[5] “Interactive System for Real Time Dynamic Multi-hop Carpooling”. Philip A. Gruebele September 2, 2008.

[6] “Car sharing adoption intention in urban areas: What are the key sociodemographic drivers? Transportation Research Part A: Policy and Practice, 101, 218-227.” Prieto, M., Baltas, G., & Stan, V. (2017).

[7] “Designing an on-line ride-sharing system”. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems (pp. 1-4). Cici, B., Markopoulou, A., & Laoutaris, N. (2015, November).

[8] "An intelligent carpooling app for a green social solution to traffic and parking congestions," by Oussama Dakroub, Carl Michael Boukhater, Fayez Lahoud, Mariette Awad, Hassan Artail, published by an International IEEE Conference on Intelligent Transportation Systems, October 2013.

[9] "Trust-Based Cooperative Social System Applied to a Carpooling Platform for Smartphones," by Cándido Caballero-Gil 1, Pino Caballero-Gil, Jezabel Molina-Gil, Francisco Martín-Fernández and Vincenzo Loia, published by Sensors (Basel, Switzerland), 27 January 2017.

[10] HTML.com. (n.d.). HTML Tutorial. HTML.com. Retrieved March 23, 2023, from https://html.com/

[11] MDN Web Docs. (n.d.). CSS (Cascading Style Sheets). MDN Web Docs. Retrieved March 23, 2023, from https://developer.mozilla.org/en-US/docs/Web/CSS

[12] MDN Web Docs. (n.d.). JavaScript. MDN Web Docs. Retrieved March 23, 2023, from https://developer.mozilla.org/en-US/docs/Web/JavaScript

[13] Hypertext Preprocessor. (n.d.). PHP. Retrieved March 23, 2023, from https://www.php.net/

[14] MySQL: MySQL Community Downloads. (n.d.). MySQL. Retrieved March 23, 2023, from https://dev.mysql.com/downloads/

[15] UML.org. (n.d.). UML Tutorial. UML.org. Retrieved March 23, 2023, from https://www.uml.org/.

[16] Visual Studio Code. (n.d.). Visual Studio Code Documentation. Visual Studio Code. Retrieved March 23, 2023, from https://code.visualstudio.com/docs

[17] Git. (n.d.). Git Documentation. Git. Retrieved March 23, 2023, from https://git-scm.com/doc

[18] GitHub Docs. (n.d.). GitHub Documentation. GitHub Docs. Retrieved March 23, 2023, from https://docs.github.com/en

[19] phpMyAdmin. (n.d.). phpMyAdmin Documentation. phpMyAdmin. Retrieved March 23, 2023, from https://www.phpmyadmin.net/docs/

[20] Bootstrap. (n.d.). Bootstrap Documentation. Bootstrap. Retrieved March 23, 2023, from https://getbootstrap.com/docs/5.1/getting-started/introduction/

[21] jQuery Foundation. (n.d.). jQuery Documentation. jQuery Foundation. Retrieved March 23, 2023, from https://api.jquery.com/

[22] WampServer. (n.d.). WampServer Documentation. WampServer. Retrieved March 23, 2023, from http://www.wampserver.com/en/

[23] AJAX (Asynchronous JavaScript and XML). (n.d.). AJAX Documentation. AJAX. Retrieved March 23, 2023, from https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX

[24] WalkerSpider. (n.d.). Middlewares in Laravel. WalkerSpider. Retrieved March 23, 2023, from https://walkerspider.com/cours/laravel/middlewares/

[25] Visual Paradigm. (n.d.). Model-View-Controller (MVC) Explained. Visual Paradigm. Retrieved March 23, 2023, from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-mvc-model-view-controller/

[26] "Carpooling: A Step to Reduce Traffic Congestion in Sri Lanka" by Amila Sandaruwan, Tharuka Karunarathne, Jayalath Edirisinghe, Vasantha Wickramasinghe, published in the 13th International Conference of Eastern Asia Society for Transportation Studies (EASTS), Vol.12, September 2019.