Pick & Share - Car and Ride Sharing System

AL MAMUN
Roll No: 2181150
NU Registration No: 17502005152

BACHELOR OF SCIENCE(Hons.)
IN
COMPUTER SCIENCE AND ENGINEERING

National University, Bangladesh - 2021

Approval

This project report entitled **“PICK & SHARE”** Prepared and submitted by **AL MAMUN**, Roll No: 2181150, Registration: **17502005152** to the Department of Computer Science and Engineering, the National University of Bangladesh in the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering, has been examined and recommended for approval and acceptance as to its style and contents.

Supervisor,

_____

MD. MOBINUL ISLAM
Lecturer
Department of
Computer Science and Engineering
Northern College Bangladesh

DECLARATION

I solemnly declare that the work presented in this project paper has been carried out by us and has been previously submitted to any University/Organization for academic or professional qualification. I hereby ensure that the work that has been presented does not breach existing copyright.

I further undertake to indemnify the university against any loss or damage arising from a breach of the foregoing obligation.

Supervised by

_____
Name
Asst. Professor
Department of Computer Science and Engineering
Northern College Bangladesh, Dhaka, Bangladesh

Submitted by

_____
Al Mamun
Roll Number: 2181150
NU Reg Number: 17502005152

Dedication

THIS PROJECT IS DEDICATED

TO

MY **Parents,**

Especially to my dearest **Mother**

INDEX

Acknowledgments

To complete this work, we received immense support from a multitude of people to whom we owe gratitude.

First, I'm very grateful to the Almighty ALLAH for giving me the strength, without which I couldn't afford to attempt this research work.

Secondly, I thank our parents and support during this study course, who have given me all their time, encouragement, and financial support to pursue this course. Last but not least, sincere appreciation goes to my supervisor MD. Mobinul Islam, Lecturer, Department of Computer Science and Engineering, Northern College Bangladesh for the guidance he has given me through the completion of this project report. We are grateful to him for his cooperation throughout my thesis work.

I would like to thank all the members of the board of examiners for their precious time in understanding my work and for their insightful comments. I would like to thank all of my friends and colleagues/ team members for their cooperation. Last but not least I am grateful to my parents for their continuous support and cooperation.

Abstract

The center of this study was to develop and implement an Online Local Car Share Management System and Centralized Database System that helps capture ride information from Customers and their Booking requests. This was developed to reduce the effect of the Manual System of Registering customers while booking. Therefore the Project offers extensive knowledge regarding Online **Pick & Share** from Bangladesh.

This project is based on the popular framework of PHP (LARAVEL) and uses PHP, MySql, HTML5/CSS3, JavaScript and various tools for development purposes like, Webpack, NPM, SASS, babelJS, Composer, Git, Github, Framework integrated Artisan etc.

Here we propose that **Pick & Share** allows users to request their destination and current place details with a summary of journey details, and the service provider Driver can bid/request (must be verified driver) on the site. Then the users can log in to the portal of the site then the user can show the request details page about the budget and all responses of drivers whose responses with their budget and some of the short details. When the budget matches users and service providers/drivers, the user would make an online-based deal. Through the deal, passengers can journey with a shareable car in the country.

Chapter 1

Introduction

1.0 Introduction

This chapter covers the background of the project, the statement of the problem, the objectives, the scope, and the significance of the project. Our first purpose is to provide the best car rental services to comfortable clients. We offer many types of cars by Driver. Driver will offer the card details and price while responding to requests for any trip while on contract. To provide the most excellent client services and quality Cars.

1.1 Related Research Works

Other car rental companies in this country have some issues. People have limitations to rent them. Big prices or prices are increasing as per traffic jams for wasting time and the main problem is there are no service providers available to share rides for long drives for the district to district contract based service of rental cars. In Bangladesh has lots of card ride sharing which called Rental Car Service all are individual company and there are many various types of price, cars and price are depend on their cars type and quality but when users gone another Rental company then see their are another various types of services and price and customers/users are confusion about this. So, while **Pick & Share** Car Rental is an independent rental car agency based in Bangladesh, it will solve these problems in one system. There will be various types of drivers and cars available and price will show individual also, users can see all drivers responses prices and all of the drivers and public users can also see this. For while the users can choose their comfort car and also price, then compare all of the responses then can make an online based contract.

Apart from our cheap rent a car service, we provide a car lease service for both personal and business travel requirements within Bangladesh.

- According to a 2021 survey, 76.8% of ride sharing users in Bangladesh use the service on an average of two to three times per month..
- The ride sharing industry in Bangladesh is expected to continue to grow in the coming years. This is due to a number of factors, including the increasing availability of smartphones and internet connectivity, the rising disposable income of Bangladeshis, and the growing traffic congestion in major cities.

Current Status of Ride-Sharing Services in Bangladesh

Figure 1 shows the experience of the ride-sharing service of the respondents. Around four-fifths of the respondent cohort had experienced a ride-sharing service in the past while only one-fifth of them never used any of the ride-sharing services in Bangladesh. This finding is very common as people show interest in ride-sharing services based on different reasons and preferences even if they all have the same access to the service. Participants in this service sector may articulate marketing strategies to attract more of these potential consumers and increase the pie for higher revenue. Moreover, based on comfort, flexibility, and convenience, people vary in the selection of transportation modes while they frequently experience rideshare. Figure 2 shows that around 37% of respondents prefer a car only, while 33% of respondents prefer a motorbike only. There are also 27% of respondents who are indifferent between these two modes based on their situation and convenience.

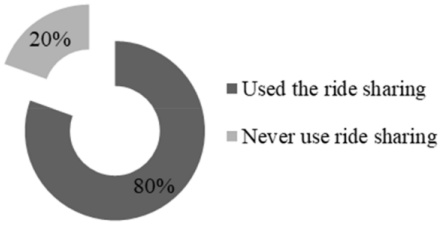


Figure 1. Experience of ride-sharing services

In addition to this, Figure 3 shows the most preferred service providers in the ride-sharing business of Bangladesh. Though this percentage does not classify the different modes of transportation, it reflects a rough estimation of the current market shares of the ride-sharing service providers. Among these few offer both cars and motorbikes. Figure 3 shows that consumers prefer mostly Uber (40%) and Pathao (38 %) among all ride-sharing services though they were free to select more than one service provider. All other ride-sharing services hold an insignificant portion of the market share (around 12%) currently.

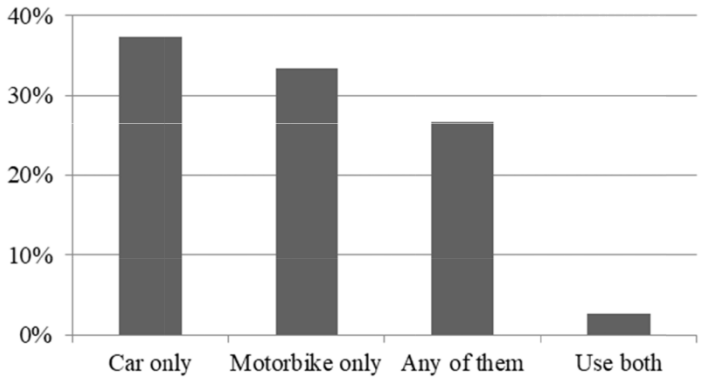


Figure 2. Preferred transportation mode

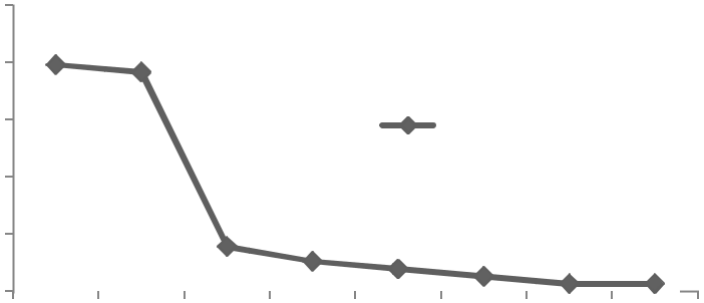
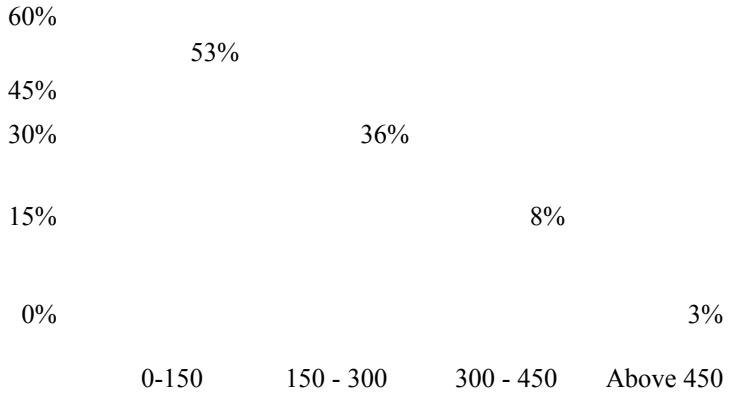
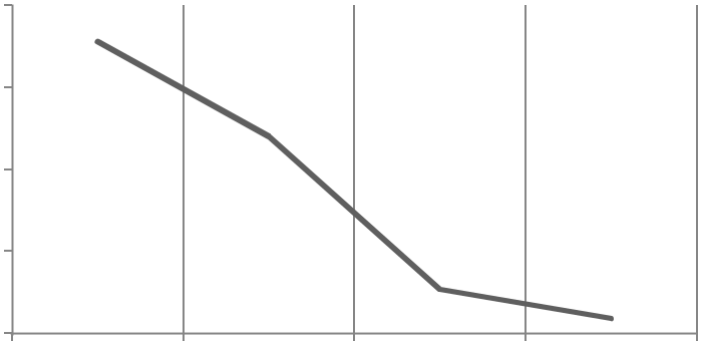


Figure 3. Preferred service provider

However, respondents also reflect the reasons why they do not prefer ride-sharing service. Around 20% of respondents never feel interested in using any of the ride-sharing services, while another certain percentage had experienced the ride-sharing but not interested in selecting this as their primary mode of transportation. Respondents were free to select more than one option, which reflects their situation most. Figure 4 shows that most of the people (26%) avoid ride-sharing service for the concern over security. Though the ride-sharing service operates through GPS tracking and all detailed information is recorded in real time, people often felt insecure for various reasons, especially women in the night time. People also ignore the ride-sharing service equally (21%) when they find a physical meeting point as a challenge and reluctant to provide detailed real-time information. Few others also constrained by technological requirement and find using an app for transportation quite tricky.





Range of Payment (in taka per trip)

Figure 4. Range of average payment per trip

Comparison of Ride-sharing Service with Traditional Modes of Transportation

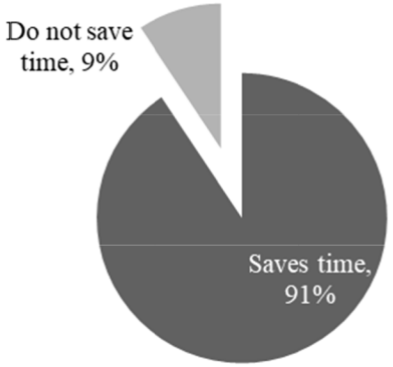


Figure 5. Preference of ride-sharing services on time savings

1.2 Methodology

1. **Problem definition:** The problem that the Pick & Share project is trying to solve is the lack of a centralized system for car sharing in Bangladesh. Currently, there are many different car rental companies, each with their own pricing and services. This can be confusing for users, who may not know which company to choose or what the best deal is. Pick & Share aims to solve this problem by providing a single platform where users can compare prices and services from different car rental companies.
2. **System design:** The Pick & Share system will be designed to be user-friendly and efficient. Users will be able to easily search for cars and book rentals. The system will also provide a variety of features, such as the ability to compare prices, read reviews, and book rentals in advance.

3. **Data collection:** The Pick & Share system will collect data from users, car rental service providers (Drivers), and other sources. This data will be used to improve the system and provide users with the best possible experience.
4. **Data analysis:** The data collected by the Pick & Share system will be analyzed to identify trends and patterns. This information will be used to improve the system and make it more user-friendly.
5. **System implementation:** The Pick & Share system will be implemented using a variety of technologies, including PHP **(Framework Laravel)**, MySQL, HTML5/HTML, CSS3/CSS, and JavaScript, JSON. The system will be hosted on a **cloud server (AWS)** - amazon web service to ensure high availability and scalability. Server used a Nginx **web server**, **Cloudflare** for DNS manager of domain, PHP7.4-fpm socket and all of the modules/libraries which are strongly required.
6. **Testing:** The Pick & Share system will be thoroughly tested to ensure that it is bug-free and meets all of the requirements. The system will be tested by both manual and automated methods.
7. **Deployment:** The Pick & Share system will be deployed to a production environment once it has been thoroughly tested. The system will be monitored on an ongoing basis to ensure that it is performing as expected.

The following are some of the key challenges that the Pick & Share project will face:

- Data collection: The Pick & Share system will need to collect a large amount of data from users, car rental companies, and other sources. This data can be difficult and expensive to collect.
- Data analysis: The Pick & Share system will need to analyze a large amount of data to identify trends and patterns. This can be a complex and time-consuming process.
- System implementation: The Pick & Share system will need to be implemented using a variety of technologies. This can be a challenging task, as the system needs to be reliable and scalable.
- Testing: The Pick & Share system will need to be thoroughly tested to ensure that it is bug-free and meets all of the requirements. This can be a time-consuming and expensive process.
- Deployment: The Pick & Share system will need to be deployed to a production environment once it has been thoroughly tested. This can be a complex and risky process.

1.3 System Development Life Cycle (SDLC)

The System Development Life Cycle involved stages through which the system passed before completion, where waterfall application design was used for the development of the entire System.

The Waterfall Model

The waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance. To follow the waterfall model, one proceeds from one phase to the next in a purely sequential manner.

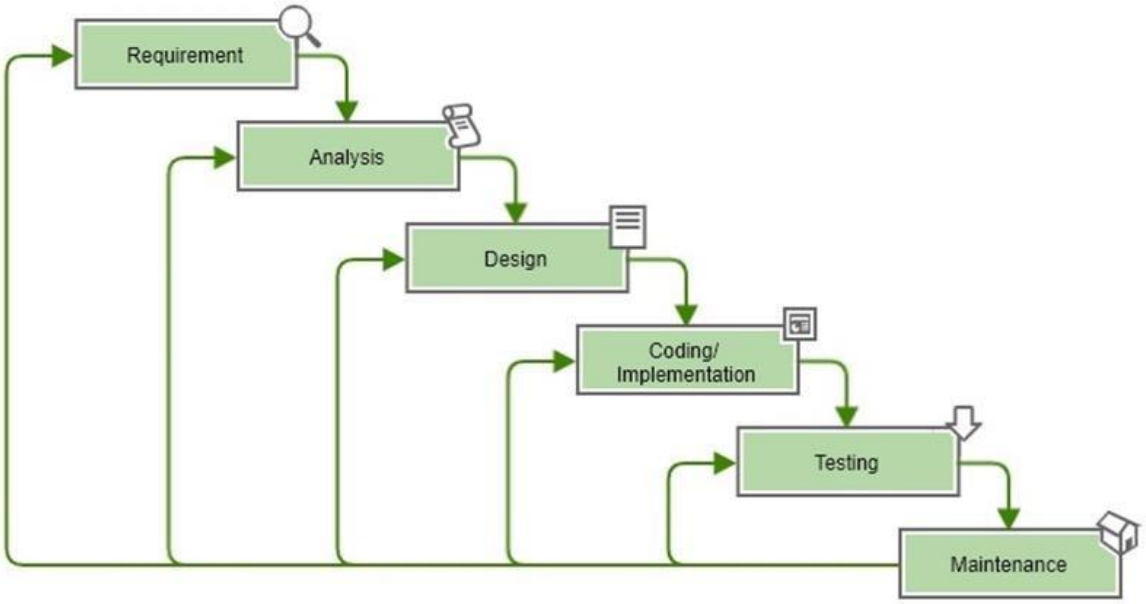


Figure 6: Waterfall model

1.4 Prototyping

This was designed and built to a scaled-down working version of a desired System. A prototype was developed with a **Figma** online tool. This application made the system development life cycle steps easier and more creative by spotting the design of the screen and reporting the system. The initial designed system was found to input and output the required results.

Chapter 2

Requirements

2.0 Introduction

The requirements section typically outlines the functional and non-functional specifications that the system needs to fulfill. These requirements help guide the design and development process to ensure that the final product meets the intended goals and user needs. Here's what a "Requirements" section might include:

1. Functional Requirements:

Detailed descriptions of the functionalities the system should offer. For example:

- User registration and login: Users can create accounts and log in to access the system.
- Booking a ride: Users can request a ride by specifying their destination and pickup location.
- Driver response: Drivers can view ride requests and submit bids with details and prices.
- Price comparison: Users can compare bids from different drivers.
- Online contract: Users and drivers can finalize a contract online when terms are agreed upon.

2. User Requirements:

Detailed descriptions of what various types of users (e.g., passengers, drivers) can do in the system. For example:

- Passengers can request rides, view driver responses, and choose the best offer.
- Drivers can view ride requests, submit bids, and finalize contracts.

3. Non-Functional Requirements:

These specify qualities that the system must possess, often related to performance, security, and usability. Examples include:

- Performance: The system should respond to user actions within 2 seconds.
- Security: User passwords should be encrypted before storage.
- Usability: The user interface should be intuitive and easy to navigate.

4. Technical Requirements:

Specifications related to the technology stack and implementation details. For example:

- The system should be developed using PHP (Laravel) for the backend and HTML5/CSS3 for the front end.
- The system should be compatible with popular web browsers (Chrome, Firefox, Safari).

5. Constraints:

Limitations that need to be considered during development. These could be technical, budgetary, or time-related. For example:

- The project must be completed within six months due to budget constraints.
- The system should support a maximum of 1000 concurrent users.

6. Dependencies:

External elements or systems that the project depends on. For example:
The system will use **Gravatar API** for profile picture management. It’s a third-party profile pictures api service provider.

7. Assumptions:

Factors that are assumed to be true but may have an impact on the project. For example:
Users have access to a reliable internet connection for using the application.

8. Use Cases or User Stories:

- Detailed scenarios that describe how users will interact with the system and accomplish specific tasks.

Sure, here's an improved version of the "Platform Requirement" section with some clarifications and a bit of restructuring for better readability:

2.1 Platform Requirements

Operating System and Server (Windows/Linux)

For both development and production environments, we have opted for a Linux server. After local development, all data will be securely stored and managed using Git. The project's data will reside in a public GitHub repository, as the project follows an open-source approach. To ensure a streamlined process, we employ GitHub Actions (Continuous Integration) to automate the deployment of data to the cloud server. This automation encompasses the seamless migration of data files through the CI tool.

2.2 Software Requirements

The **"Pick & Share"** project requires a specific set of software components to ensure successful development and operation. These components encompass both development and production environments:

Development Environment:

1. Text Editor/IDE: Visual Studio Code is recommended for coding and project management.
2. Version Control: Git is used for managing source code versioning.
3. Server: Local development will involve using a development server, which includes:
   - ➔ Web Server: Nginx
   - ➔ Programming Language: PHP (Version 7.4)
   - ➔ Laravel (Version 8) and its required extensions/plugins
   - ➔ Node.js (Version 18.x), npm, npx for frontend development
   - ➔ Frontend Tools: Webpack, Webpack-CLI, Tailwind CSS, BabelJS, SASS, CSS Loader, PostCSS, lodash, etc.

Production Environment:

1. Server: The production environment aligns with the local development server configuration.
2. Cloud Hosting: The cloud production server will be hosted using a reliable cloud service (e.g., AWS) for scalability and availability.
3. Database: MySQL is chosen as the relational database management system for both development and production.
4. Web Server: Nginx is utilized to serve the application in both environments.
5. Automation: GitHub Actions (Continuous Integration) is used to automate the deployment of code and data to the cloud server.
6. AI Tools: Chat GPT AI and Google Bard assist in bug identification and resolution.

By ensuring that the required software components are correctly installed and integrated, the project can progress smoothly from development to production, ultimately delivering a functional and reliable "Pick & Share" system.

This concise breakdown conveys the key software components and their roles in your project's development and deployment.

Laravel is chosen for several key reasons:

1. Clean Syntax: Its concise syntax makes code more readable.
2. Efficient Development: Tools like Artisan enhance productivity.
3. Modular Structure: MVC pattern enables organized code.
4. Eloquent ORM: Simplifies database interaction with objects.
5. Database Management: Migrations ensure seamless schema updates.
6. Security: Offers built-in protection against vulnerabilities.
7. Community: Active support, abundant tutorials, and packages.
8. RESTful Routing: Simplifies API endpoint creation.
9. Template Engine: Blade streamlines dynamic view creation.
10. Testing: Built-in suite for code quality assurance.
11. Scalability: Adaptability for projects of varying sizes.

Ultimately, the choice of Laravel depends on factors like project requirements, developer familiarity, and the framework's suitability for your application's goals. It's important to evaluate different frameworks based on your project's needs and consider factors such as community support, documentation, and available resources.

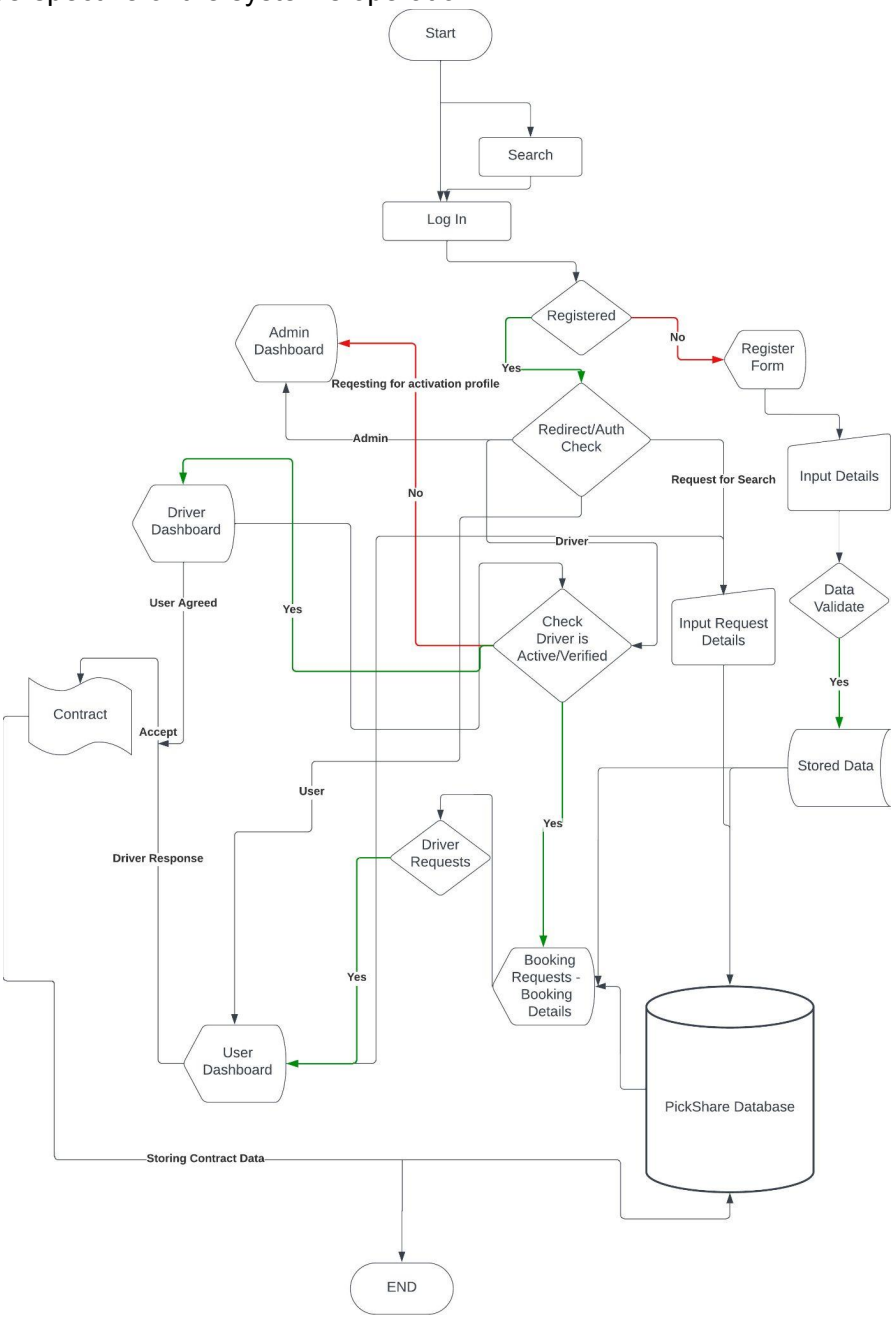
MySQL is chosen for these reasons:

1. Community: Large user base, strong support.
2. Reliability: Stable, proven for production.
3. Performance: Efficient data handling.
4. Ease of Use: Simple setup and management.
5. Scalability: Adaptable for various workloads.
6. Security: Robust authentication, encryption.
7. Compatibility: Works with different languages.
8. ACID Compliance: Ensures data integrity.
9. Cost-Effective: Open-source, economical.
10. Features: Replication, clustering options.

Chapter 3
SOFTWARE DESIGN

3.0 Introduction
Software design is the process of defining the architecture, components, modules, interfaces, and data for a software system. It's a blueprint for creating software that meets specified requirements.

3.1 System Flow Diagram
The System Flow Diagram provides an overview of how data and control flow within the software system. It shows the high-level processes and their interactions, offering a big-picture perspective of the system's operation.



.Fig 7: System Flow Diagram

3.2 ER Diagram

The System Flow Diagram provides an overview of how data and control flow within the software system. It shows the high-level processes and their interactions, offering a big-picture perspective of the system's operation

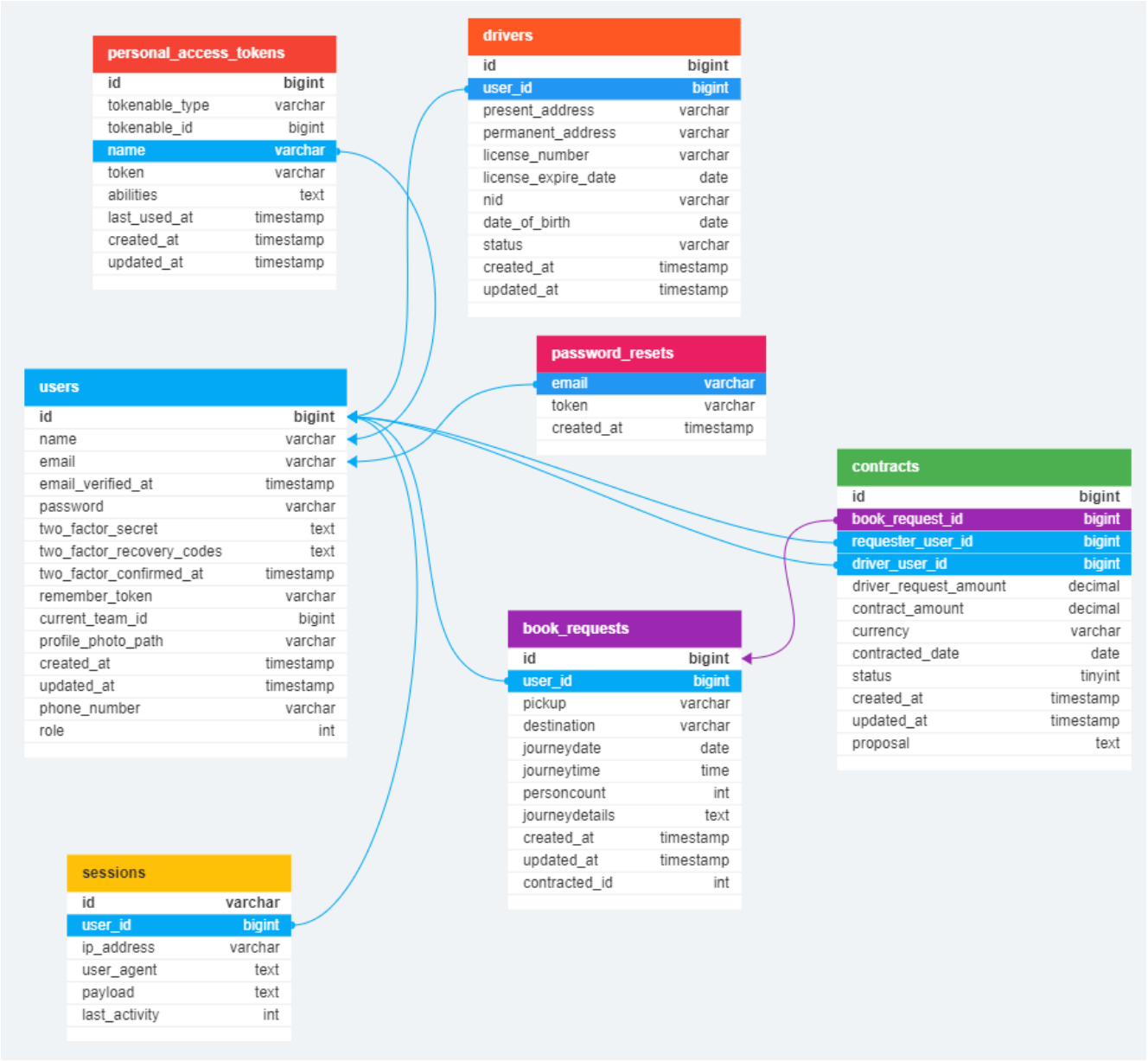


Fig 8: E-R Diagram

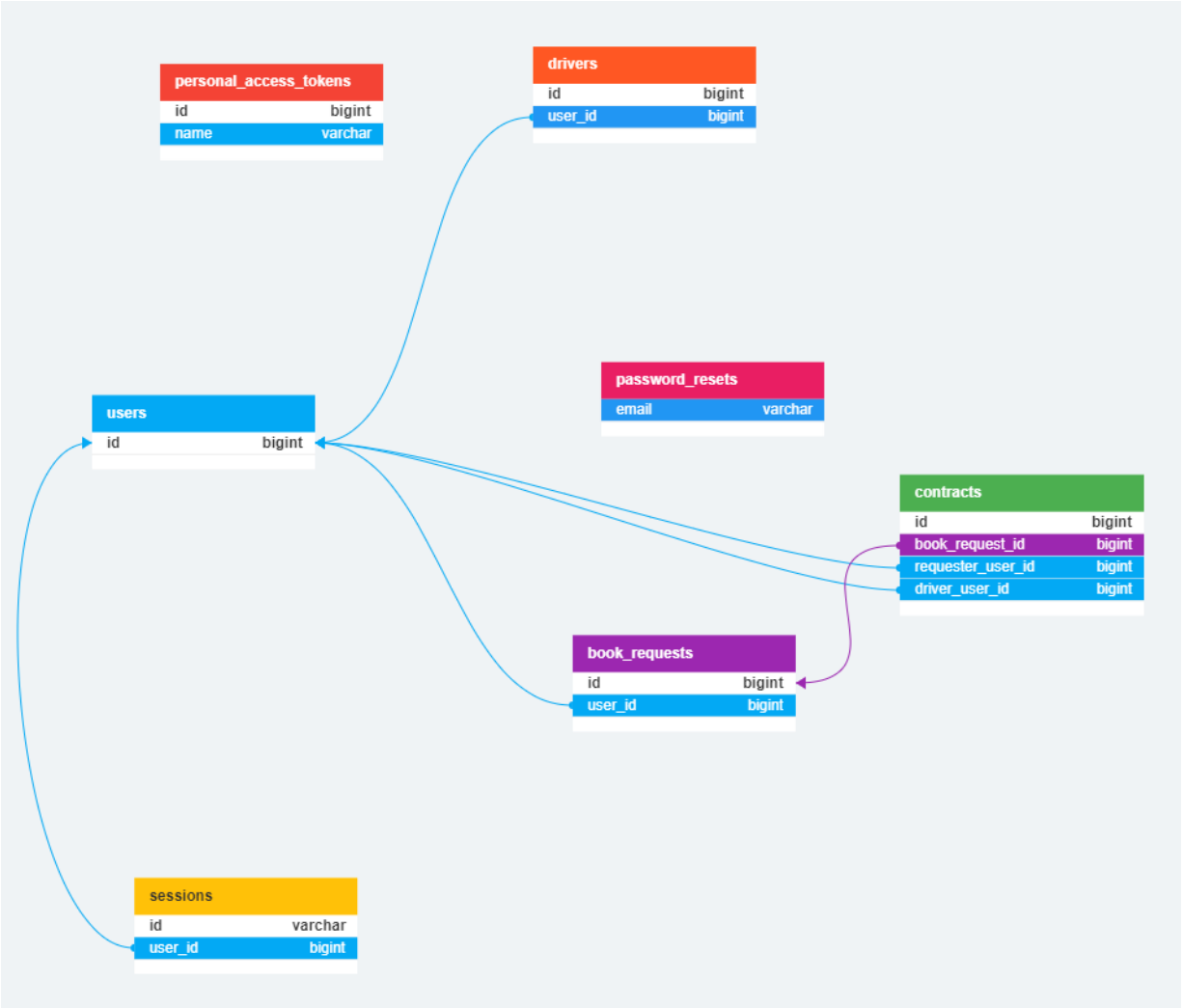


Fig 9: E-R Diagram Only Relation

3.3 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) represents how data moves through the system, from inputs to processes and outputs. It provides insights into how data is processed, transformed, and stored within the system.

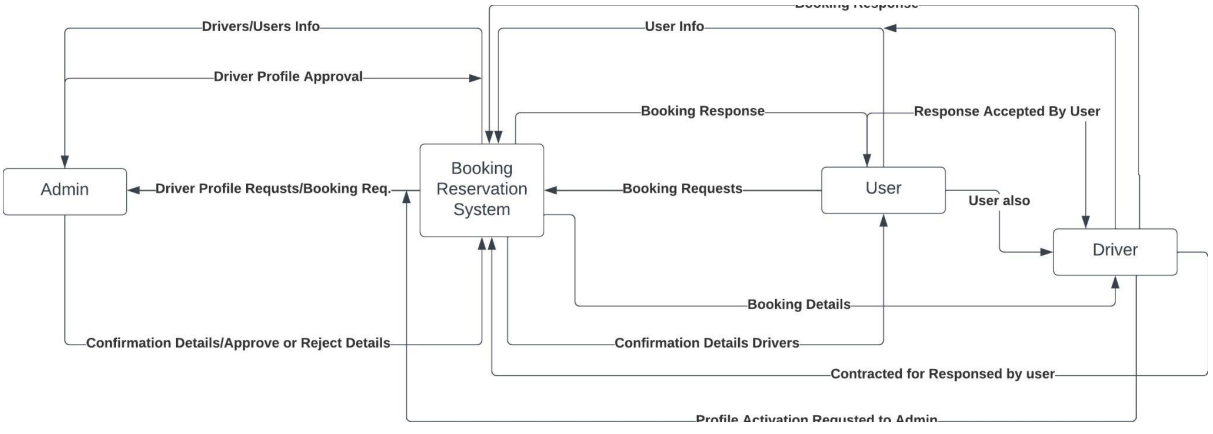


Fig 10: Data Flow Diagram

3.4 Use Case Diagram

The Data Flow Diagram (DFD) represents how data moves through the system, from inputs to processes and outputs. It provides insights into how data is processed, transformed, and stored within the system.

Fig 11: Use Case Diagram

Chapter 4

IMPLEMENTATION

4.0 Introduction

Implementation is the phase in the software development process where the designed system is translated into actual code. It involves writing, testing, and integrating code modules to bring the software to life. This phase marks a crucial step towards turning concepts into functional software applications.

4.1 Folder Architecture

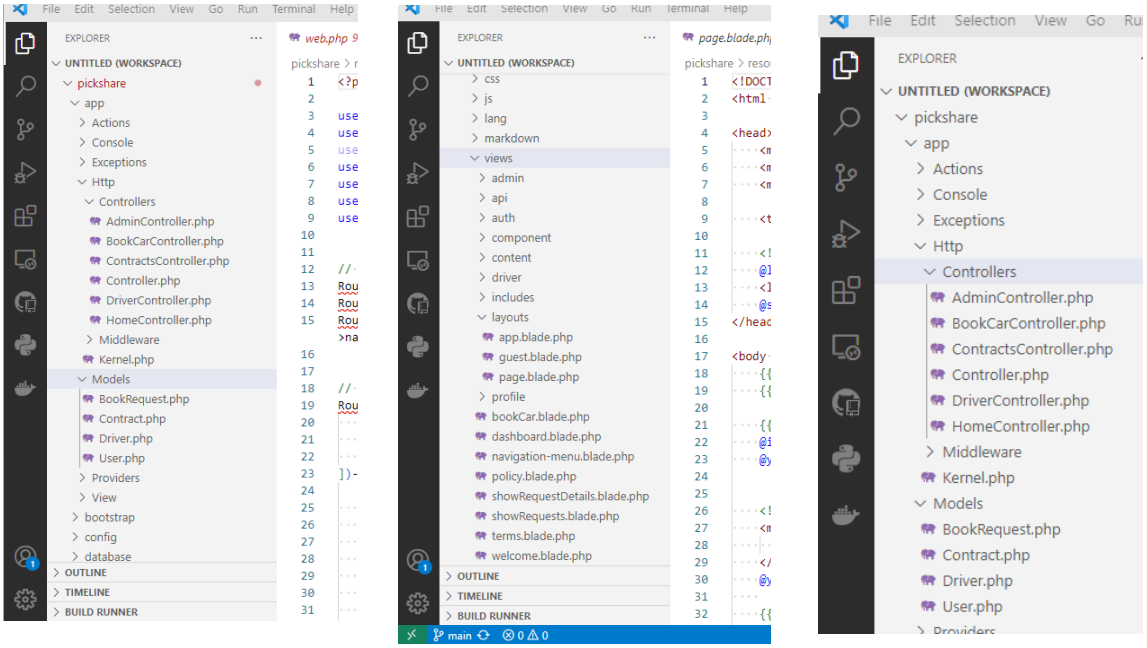


Fig 12: MVC Pattern FOlder Architecture

4.2 Page Layout

page.blade.php

pickshare > resources > views > layouts > page.blade.php > ...

```
<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="csrf-token" content="{{ csrf_token() }}">

    <title>@yield('pgTitle')</title>

    <!-- Styles -->
    @livewireStyles
    <link rel="stylesheet" href="{{ asset('css/app.css') }}">
    @stack('css')
</head>

<body class="min-h-screen bg-gray-100">
    {{-- <x-jet-banner /> --}}
    {{-- @livewire('navigation-menu') --}}

    {{-- Header --}}
    @include('includes.header')
    @yield('pgAfterHeader')


    <!-- Page Content -->
    <main>
        @yield('main')
    </main>
    @yield('pgAfterMain')

    {{-- Footer --}}
```

Fig 13: Main Layout

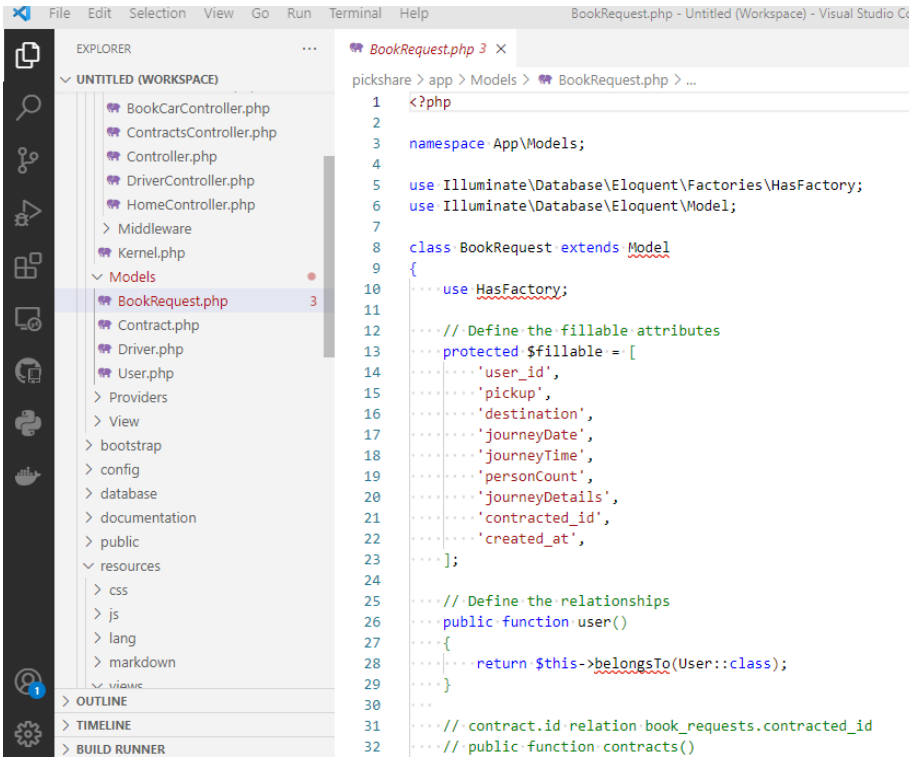


Fig 14: DB Middleman (MODEL)

There are also some of models for CRUD

4.3 Databases Migrations and Manage

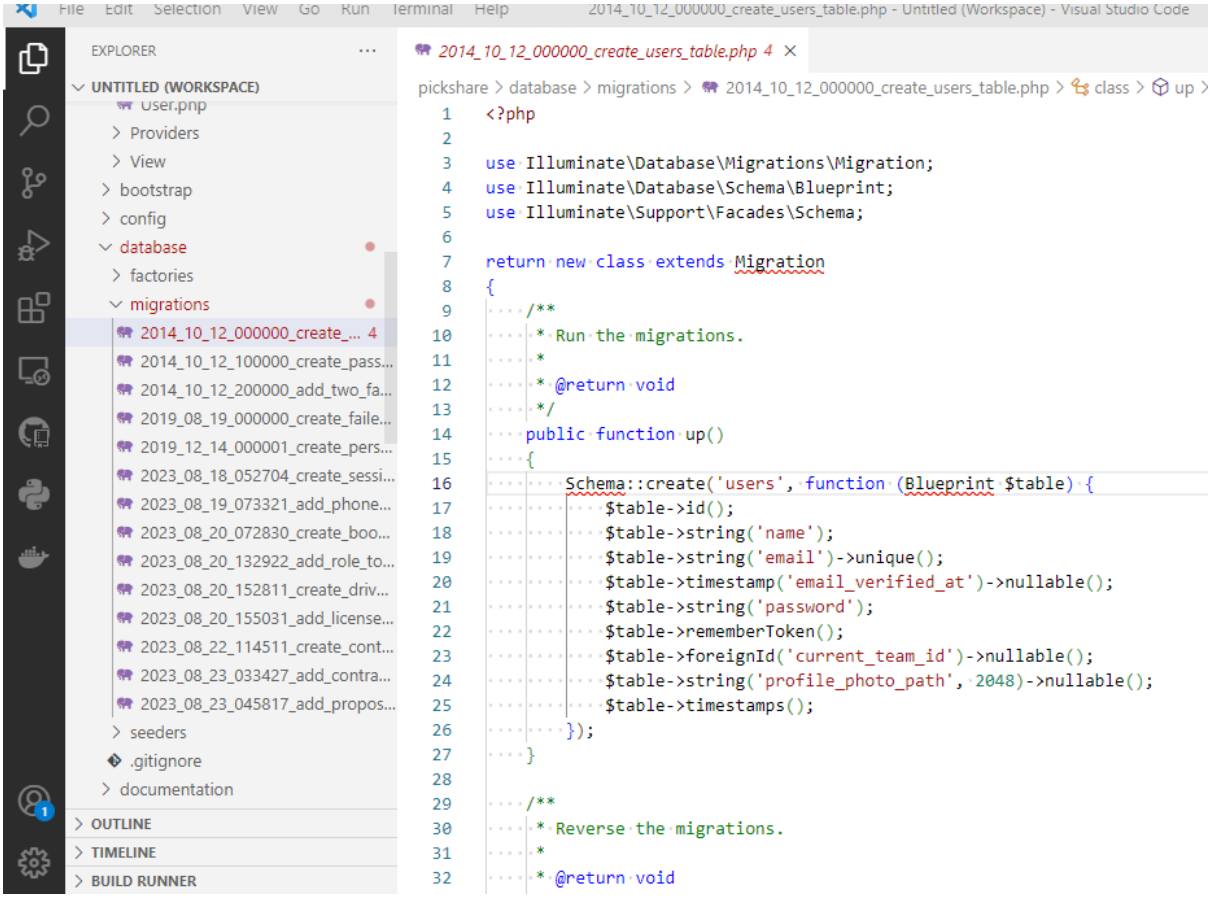


Fig 15: DB Migration FIles (with one sample)

4.4 Deployment Server

```
RAFI@DESKTOP-FAFLUJF MINGW64 ~
$ ssh asifulmamun@pickshare.asifulmamun.info.bd
asifulmamun@pickshare.asifulmamun.info.bd's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1009-aws aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Aug 30 06:15:05 UTC 2023

  System load:  0.04296875         Processes:             141
  Usage of /:   14.9% of 28.90GB   Users logged in:       0
  Memory usage: 41%                IPv4 address for ens5: 172.31.38.25
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

3 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


*** System restart required ***
Last login: Tue Aug 29 17:47:18 2023 from 103.187.66.32
asifulmamun@ip-172-31-38-25:~$ cd /var/www/html/
asifulmamun@ip-172-31-38-25:/var/www/html$ ls
README.md  artisan    composer.json  config    documentation  package-lock.json  phpunit.xml        public     routes      storage              tests   vite.config.js
app        bootstrap  composer.lock  database  node_modules   package.json       postcss.config.js  resources  server.php  tailwind.config.js  vendor  webpack.mix.js
asifulmamun@ip-172-31-38-25:/var/www/html$
```

Fig 16: Production Folder Architecture - AWS (Ubuntu server)

4.5 Authentication

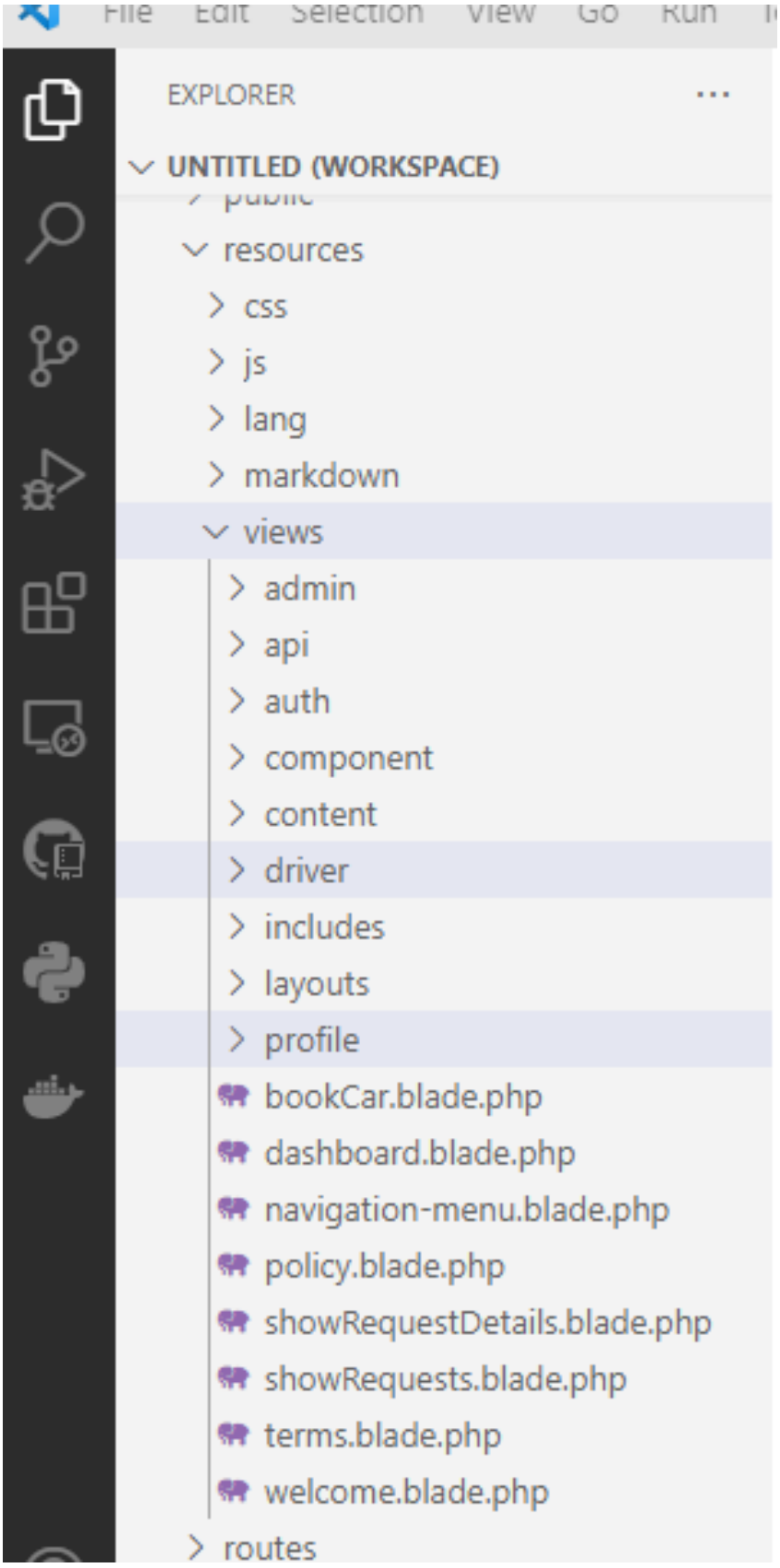


Fig 17: Different VIew for different roles

4.6 Middleware

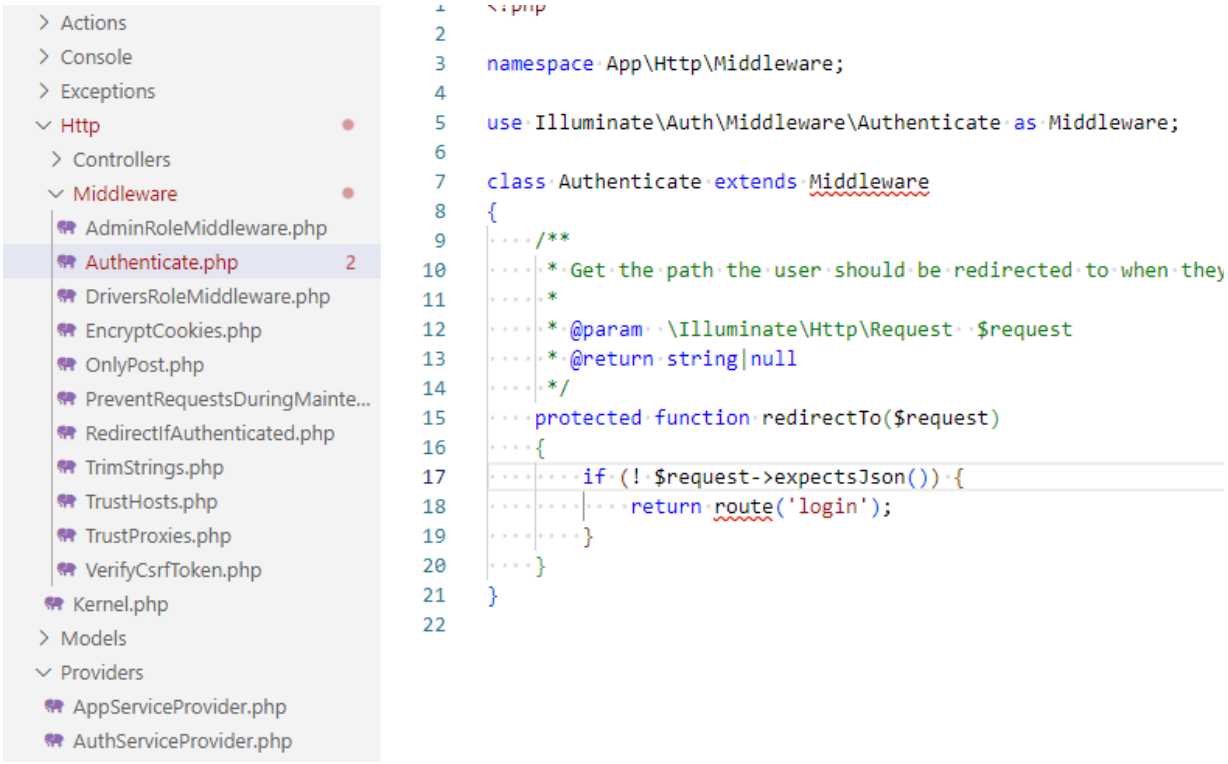


Fig 18: Logged User Middleware

Fig 19: Driver Check Middleware

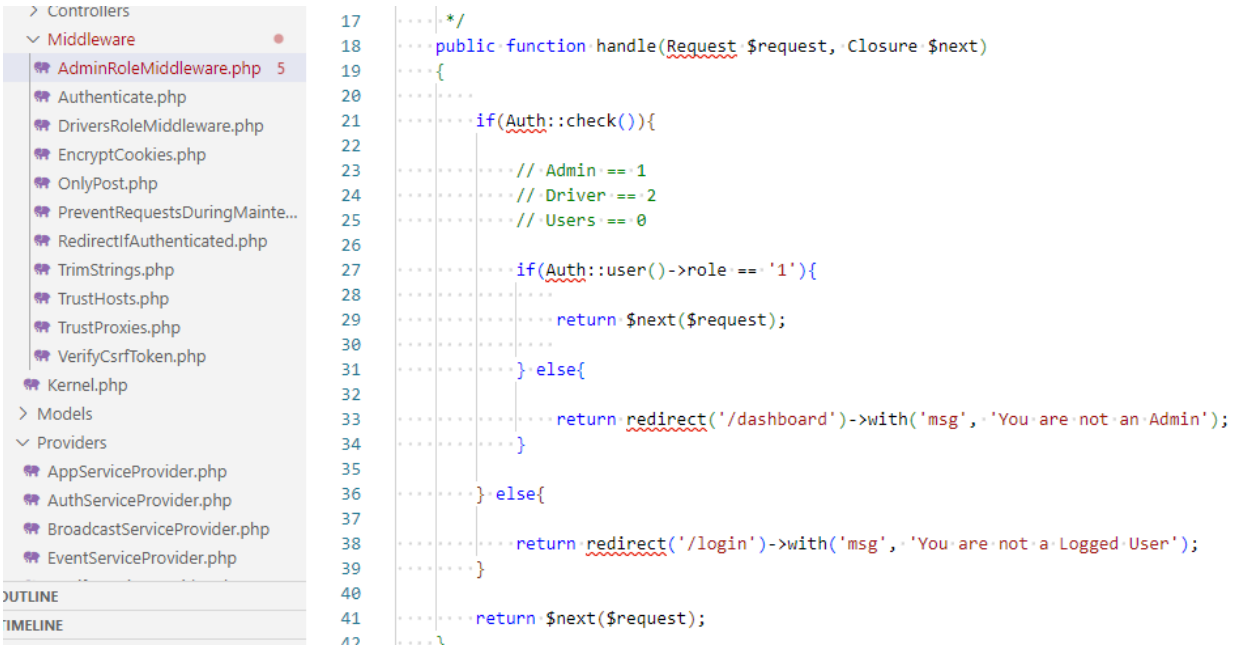


Fig 20: Admin Check Middleware

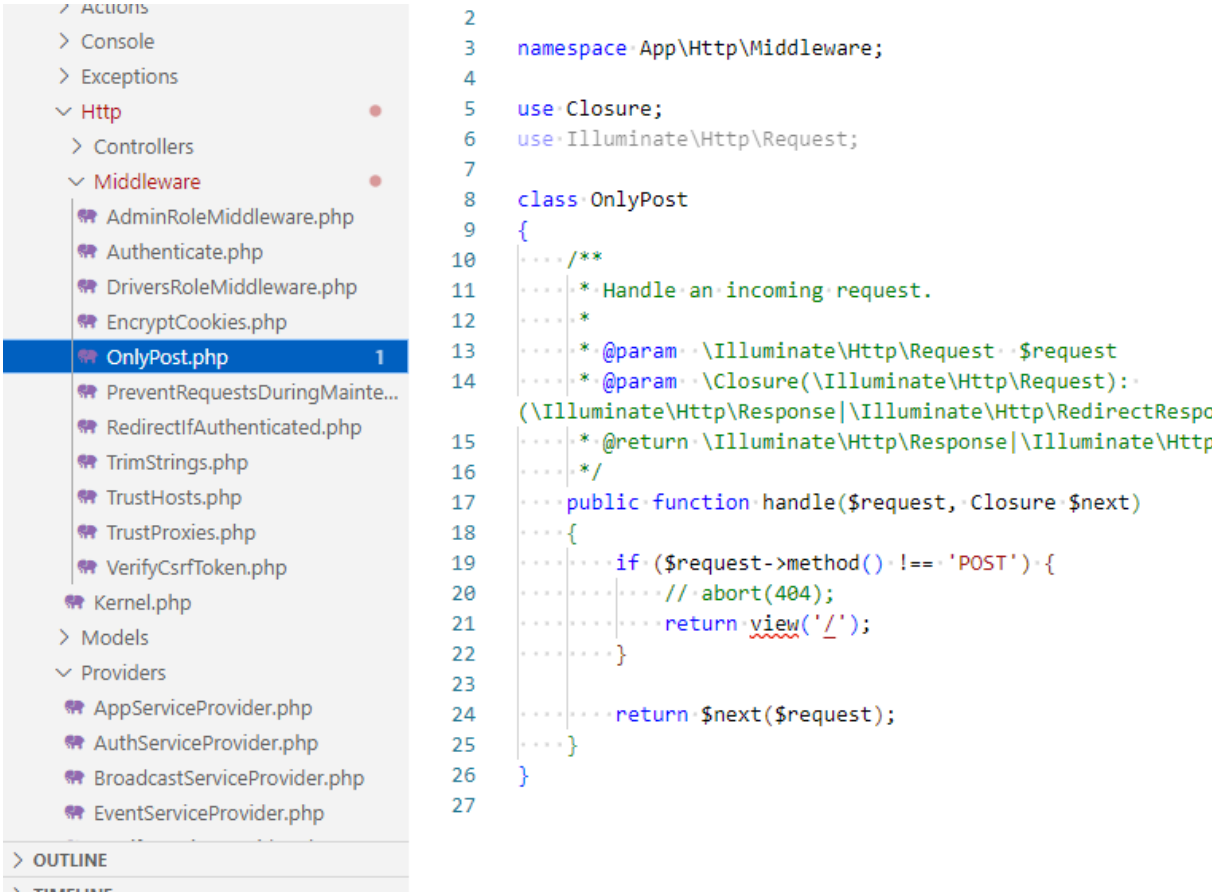


Fig 21: Route (URL) check only post method - Middleware

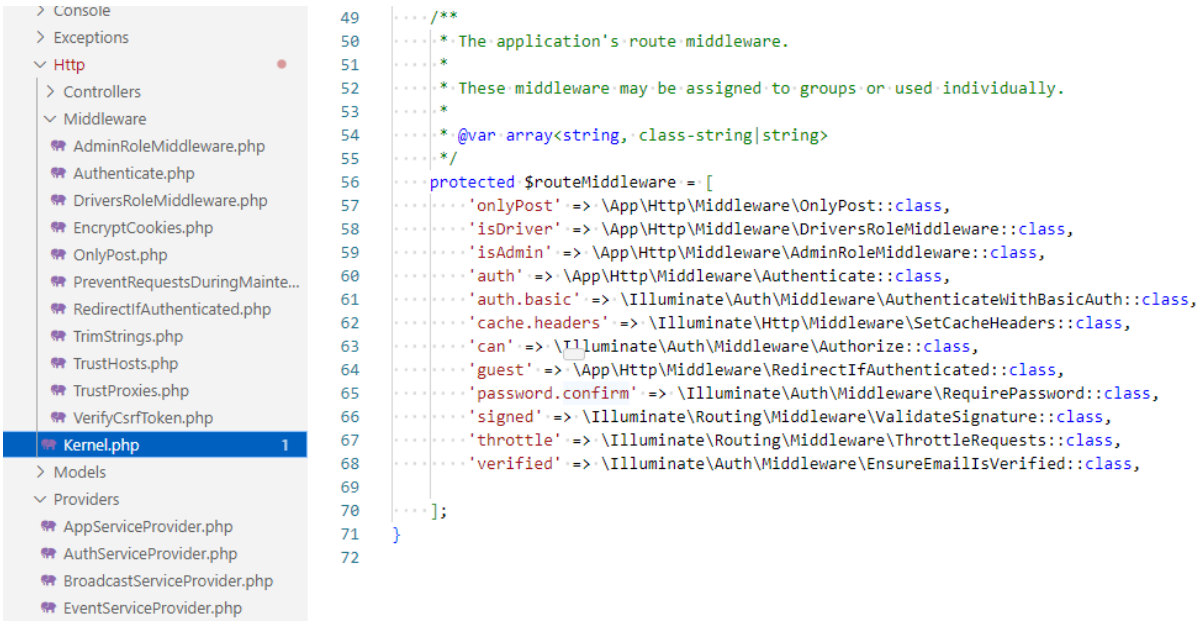


Fig 22: All middleware register in kernel

4.7 Database Tables & Relations

```
*** System restart required ***
Last login: Tue Aug 29 17:47:18 2023 from 103.187.66.32
asifulmamun@ip-172-31-38-25:~$ cd /var/www/html/
asifulmamun@ip-172-31-38-25:/var/www/html$ ls
README.md  artisan    composer.json  config    documentation  package-lock.json  phpunit.xml      public     routes      s
app        bootstrap  composer.lock  database  node_modules   package.json       postcss.config.js  resources  server.php  t
asifulmamun@ip-172-31-38-25:/var/www/html$ mysql -u asifulmamun -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 235
Server version: 8.0.34-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| pickshare          |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use pickshare;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+------------------------+
| Tables_in_pickshare    |
+------------------------+
| book_requests          |
| contracts              |
| drivers                |
| failed_jobs            |
| migrations             |
| password_resets        |
| personal_access_tokens |
| sessions               |
| users                  |
+------------------------+
9 rows in set (0.00 sec)

mysql>
```

Fig 23: MySql Database Tables

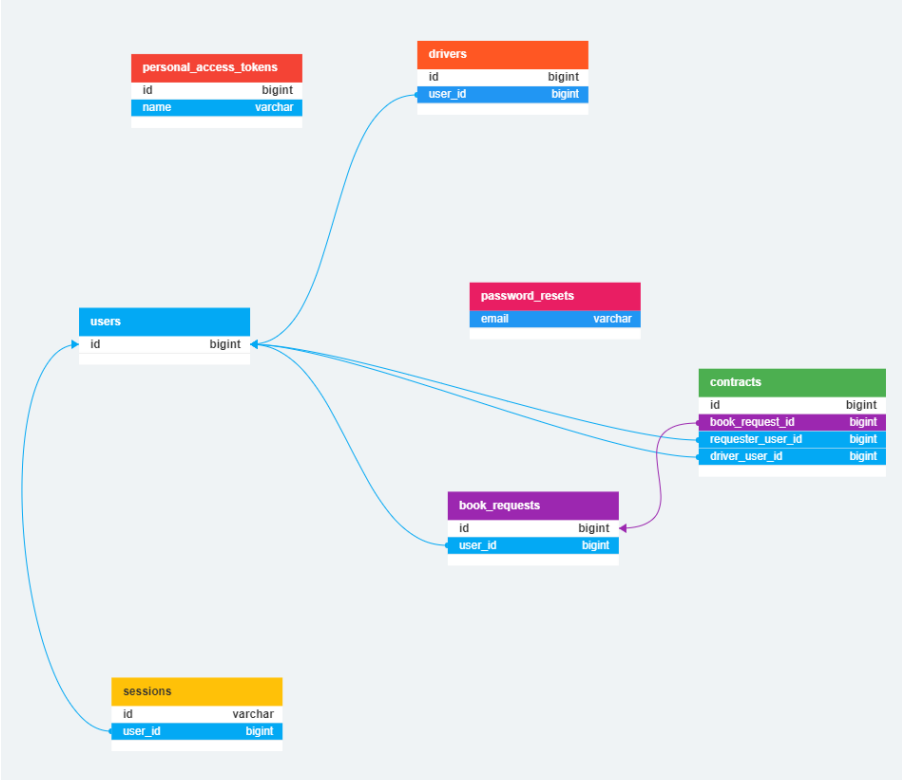


Fig 24: MySql DB relations only via minimal E-R Diagram

4.8 Tools & Automation CI

package.json

pickshare > {} package.json > {} scripts > serve

```
{
    "private": true,
    Debug
    "scripts": {
        "dev": "npm run development",
        "development": "mix",
        "watch": "mix watch",
        "watch-poll": "mix watch --watch-options-poll=1000",
        "hot": "mix watch --hot",
        "prod": "npm run production",
        "production": "mix --production",
        "serve": "clear && php artisan cache:clear && php artisan config:clear && php artisan route:clear && php artisan view:clear && php artisan optimize && php artisan serve",
        "clear": "clear && php artisan cache:clear && php artisan config:clear && php artisan route:clear && php artisan view:clear && php artisan optimize"
    },
    "devDependencies": {
        "@tailwindcss/forms": "^0.5.2",
        "@tailwindcss/typography": "^0.5.0",
        "alpinejs": "^3.0.6",
        "autoprefixer": "^10.4.7",
        "axios": "^0.21",
        "laravel-mix": "^6.0.6",
        "lodash": "^4.17.19",
        "postcss": "^8.4.14",
        "tailwindcss": "^3.1.0"
    }
}
```

Fig 25: Package Used (Developer Dependencies)

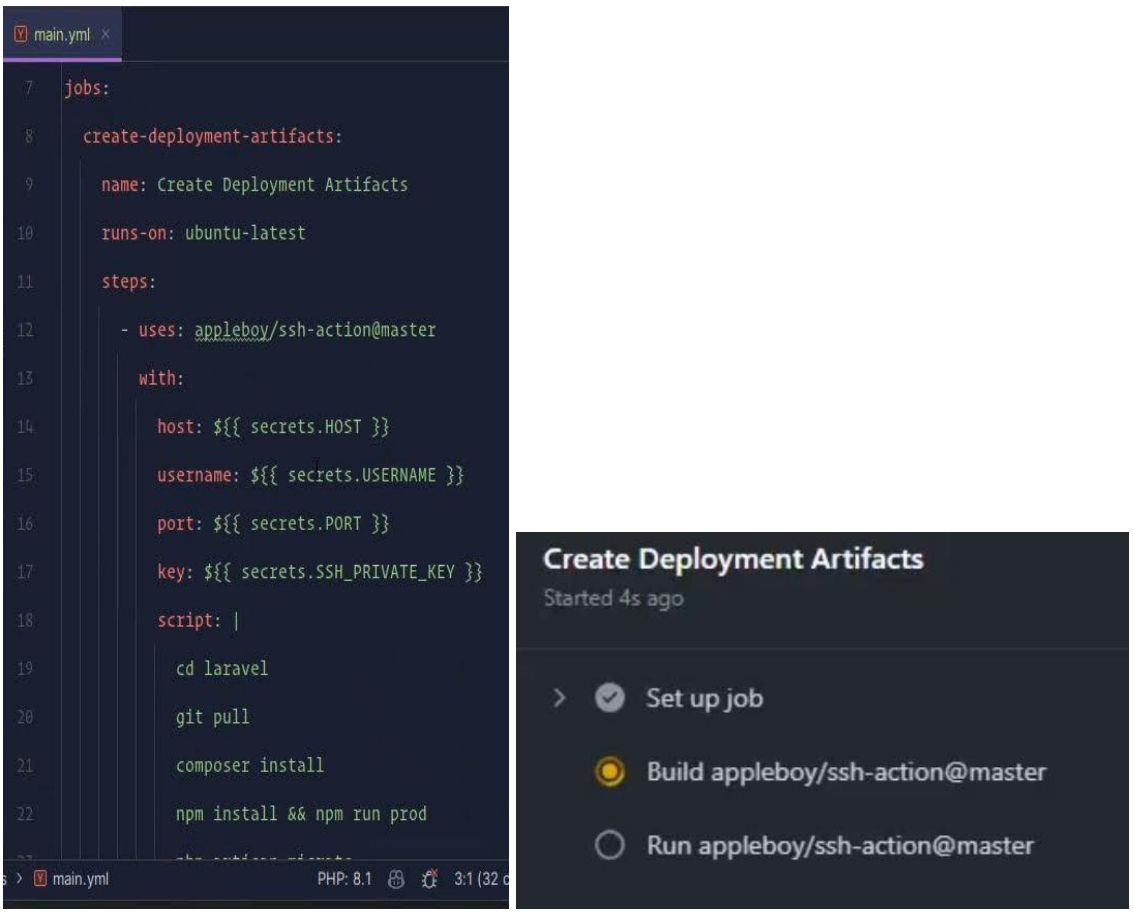


Fig 26: GIthub action for CI Automation

Chapter 5

Results

5.0 Introduction

In this chapter, we unveil the outcomes of our project, showcasing the tangible results achieved through the implementation of our software system. These findings validate the effectiveness of our solution and its alignment with the project's objectives. For a detailed exploration of these results, please visit the following link: Result Page.
Result: http://pickshare.asifulmamun.info.bd

5.1 Home Page

The home page features a header with navigation links, a search section for car rentals, and quick links to routes. Users can input pickup and destination details and click "SEARCH" to find rentals. They can also explore predefined routes and learn about service features. The page is styled using CSS and appears responsive to different screen sizes. Additional details and functionality would likely be implemented using server-side logic and JavaScript.

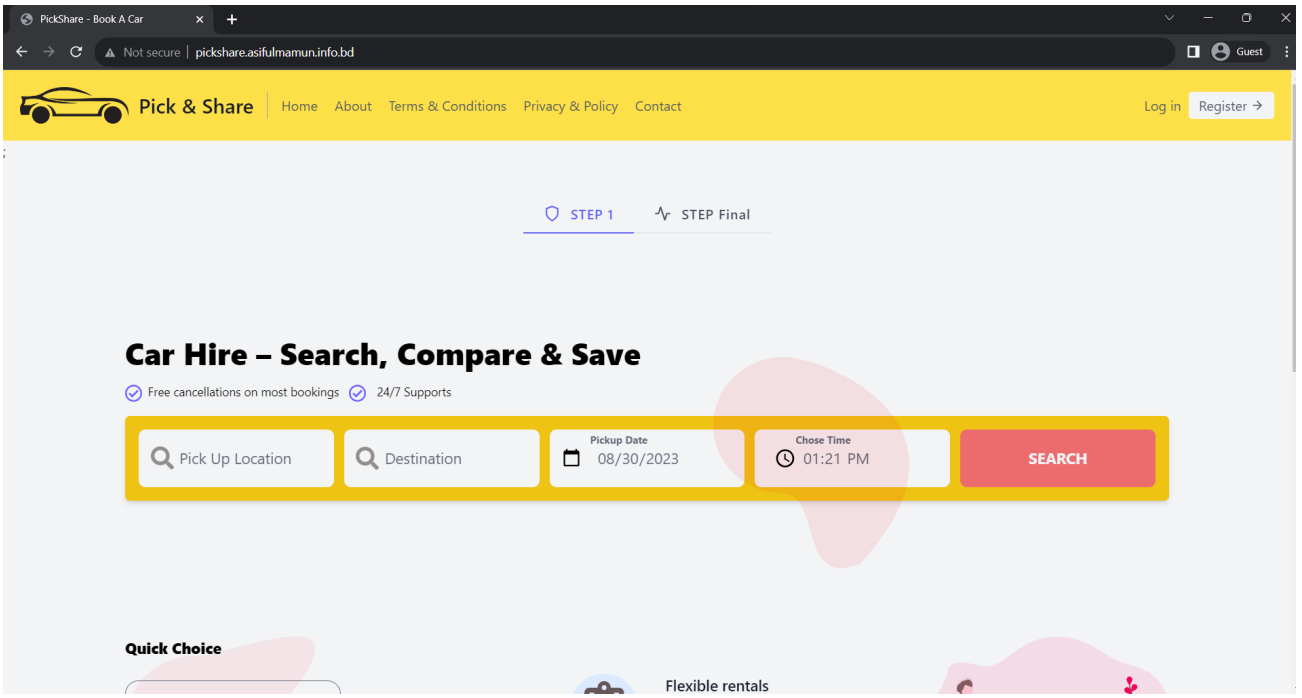


Fig 27: Home Page

5.2 Authentications Pages

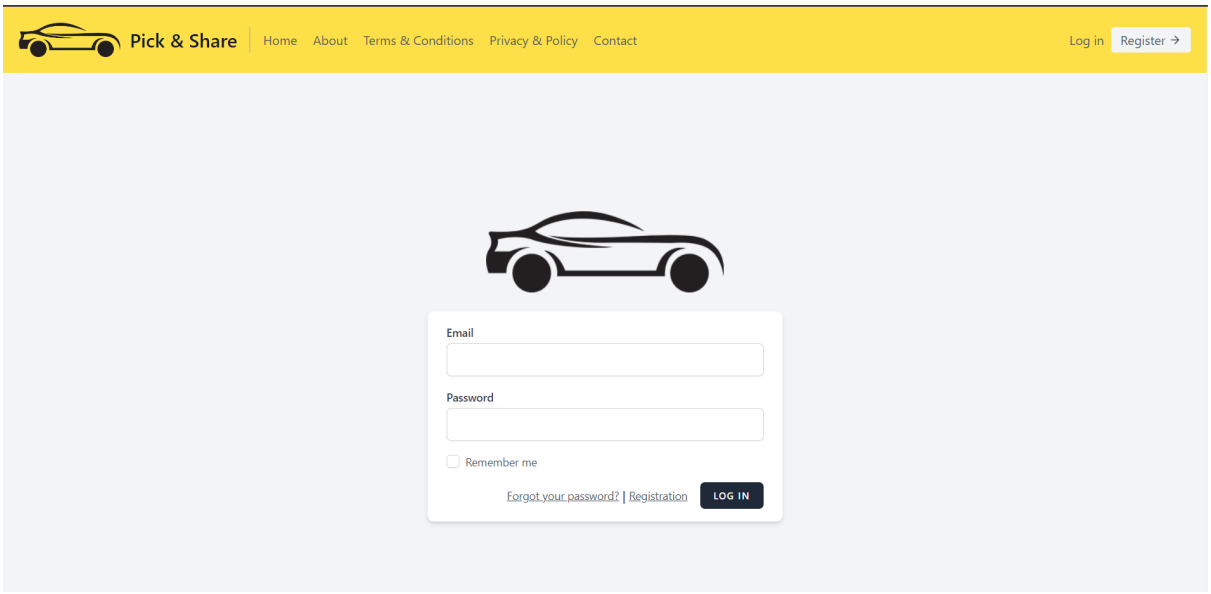


Fig 28: Login Page

5.3 Registration Page

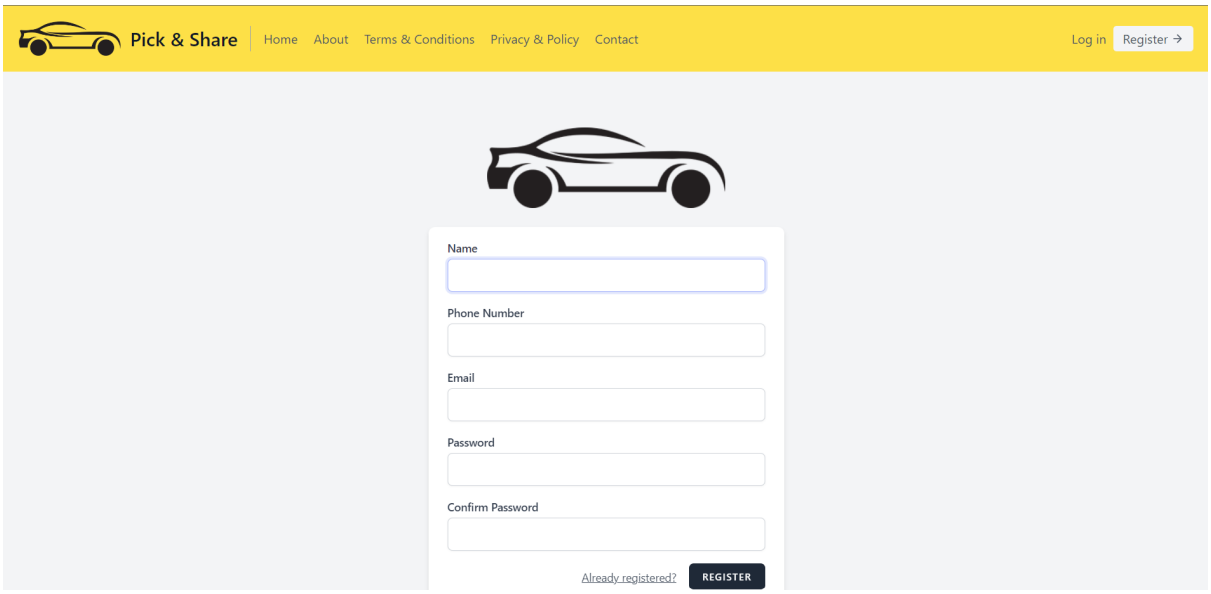


Fig 29: Registration Page

5.4 Dashboards Users and Drivers

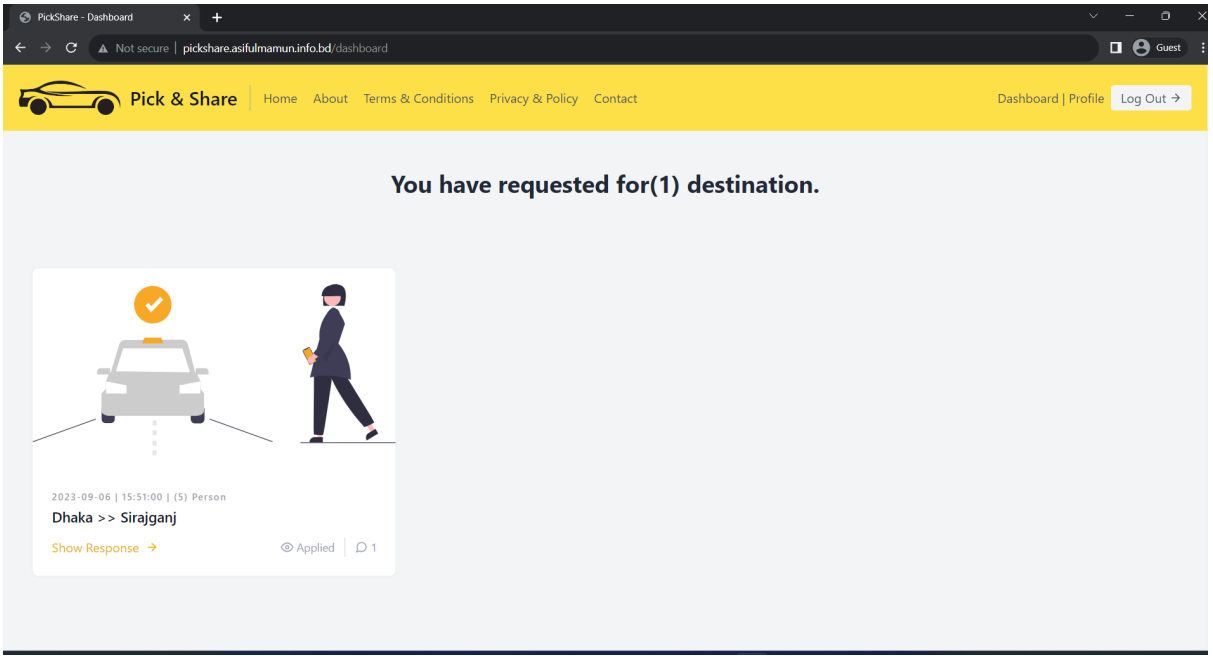


Fig 30: User Dashboard - (also available for driver)

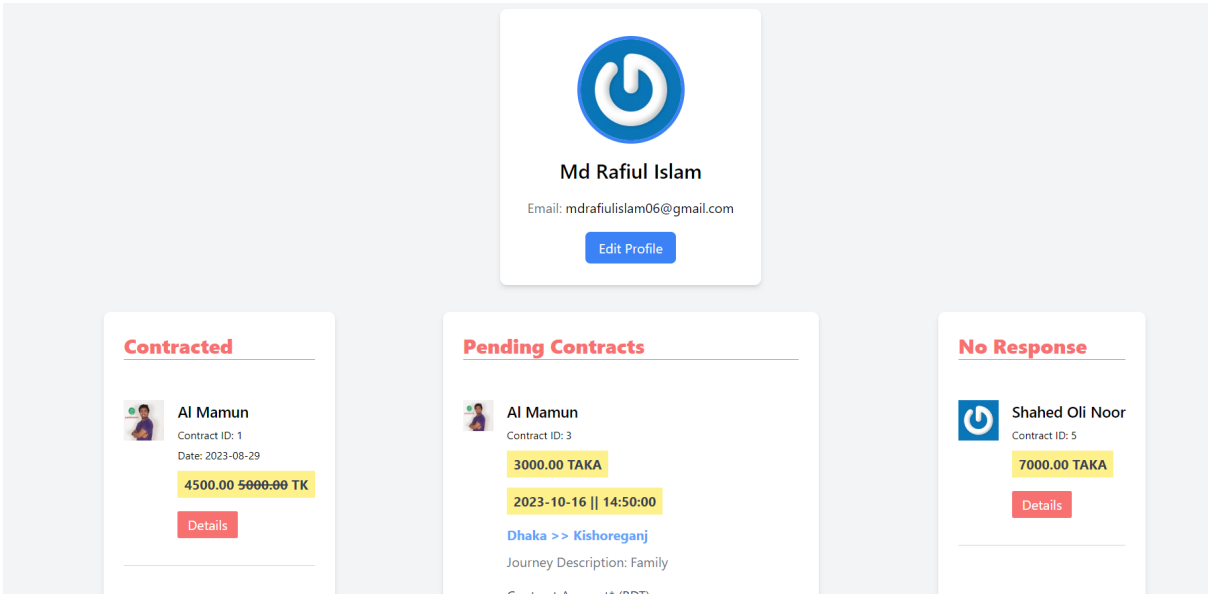


Fig 31: Driver Dashboard

5.5 Admin Dashboard

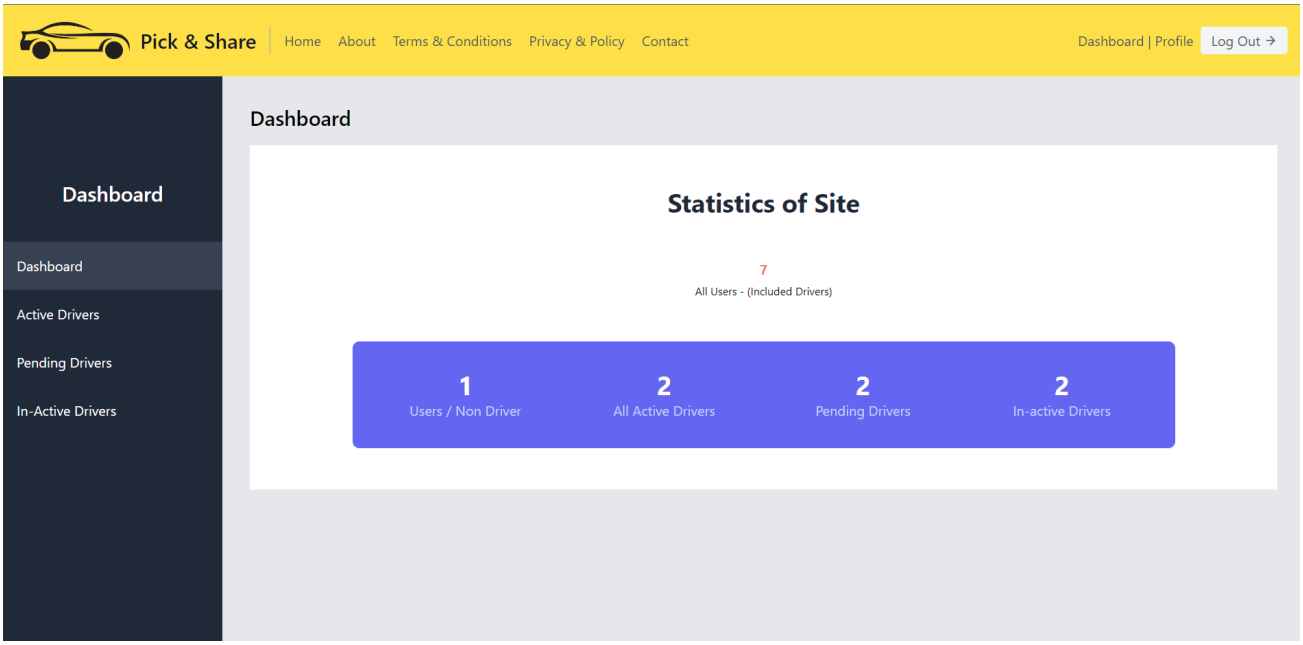


Fig 32: Admin Dashboard

5.6 Admin Operations

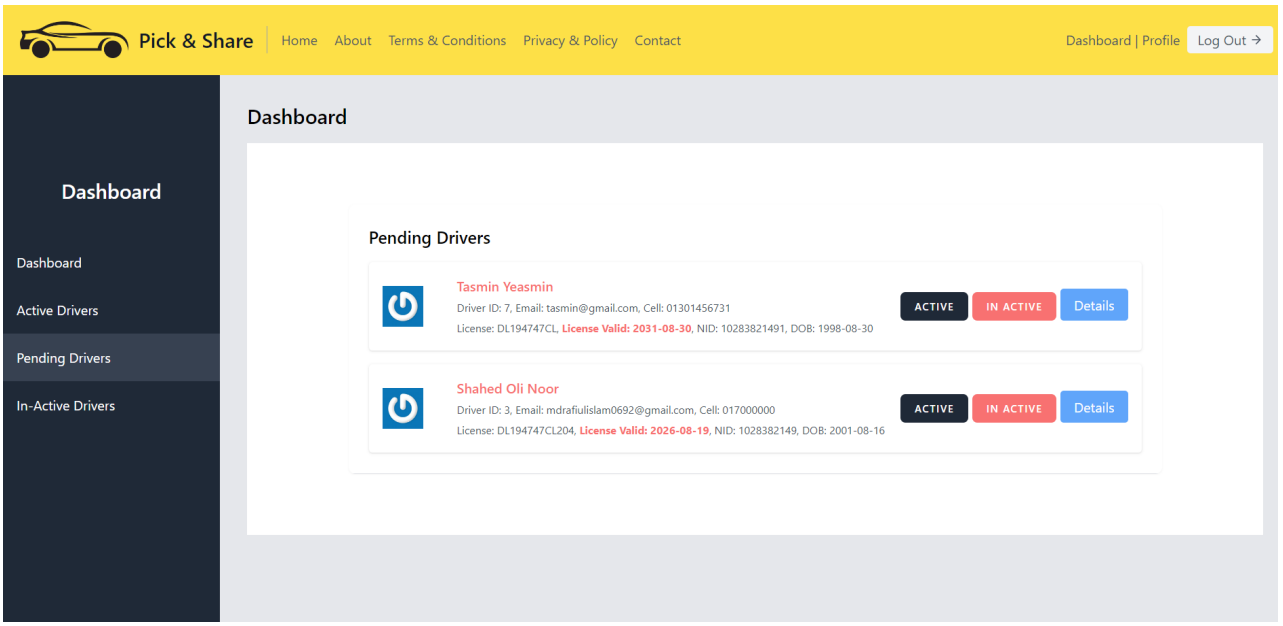


Fig 33: In-Active Drivers List

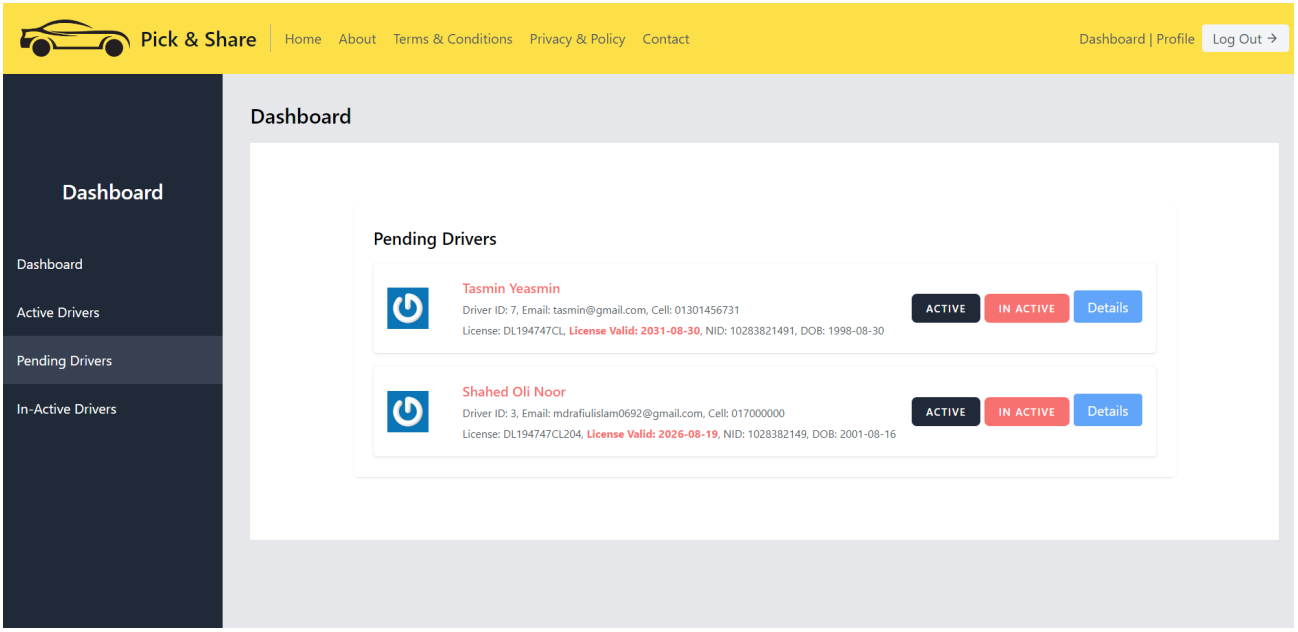


Fig 34: Pending Drivers List

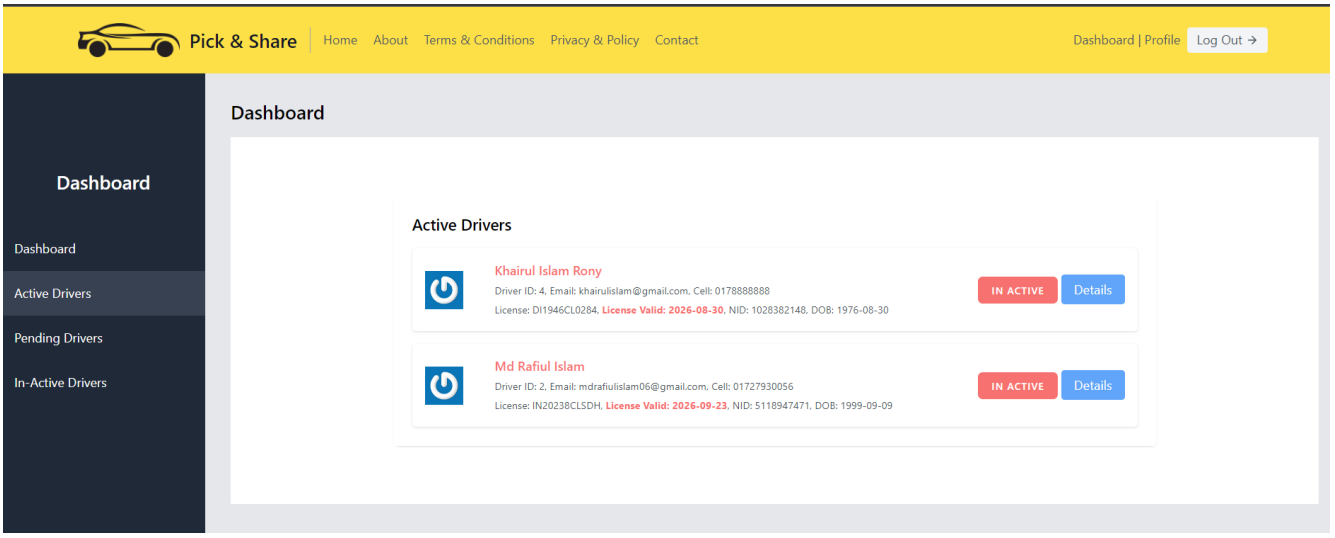


Fig 35: Activated drivers list

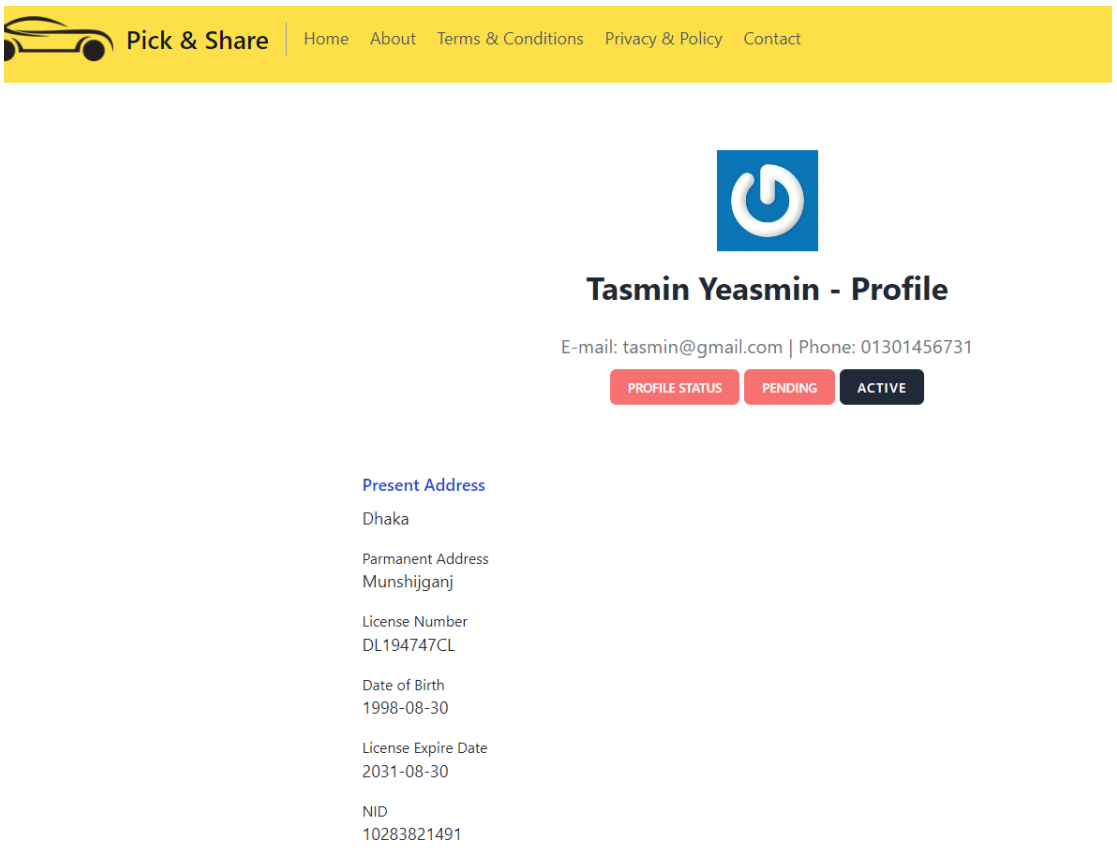


Fig 36: Pending driver profile for activation

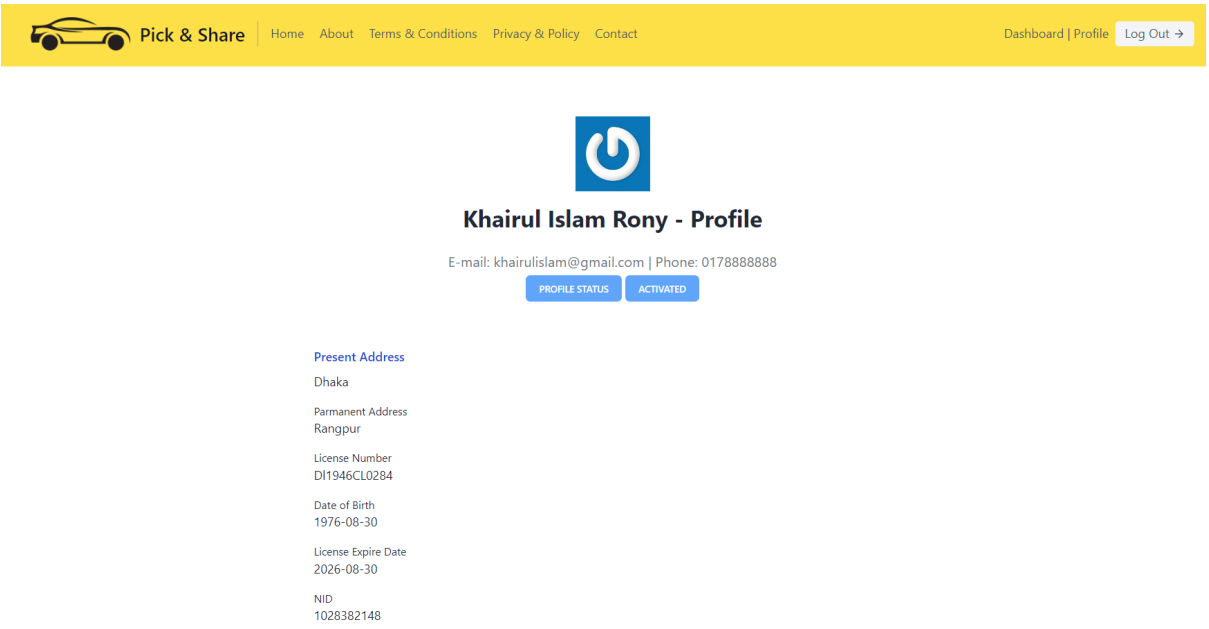


Fig 37: Activated Driver Profile

5.7 Driver Operations

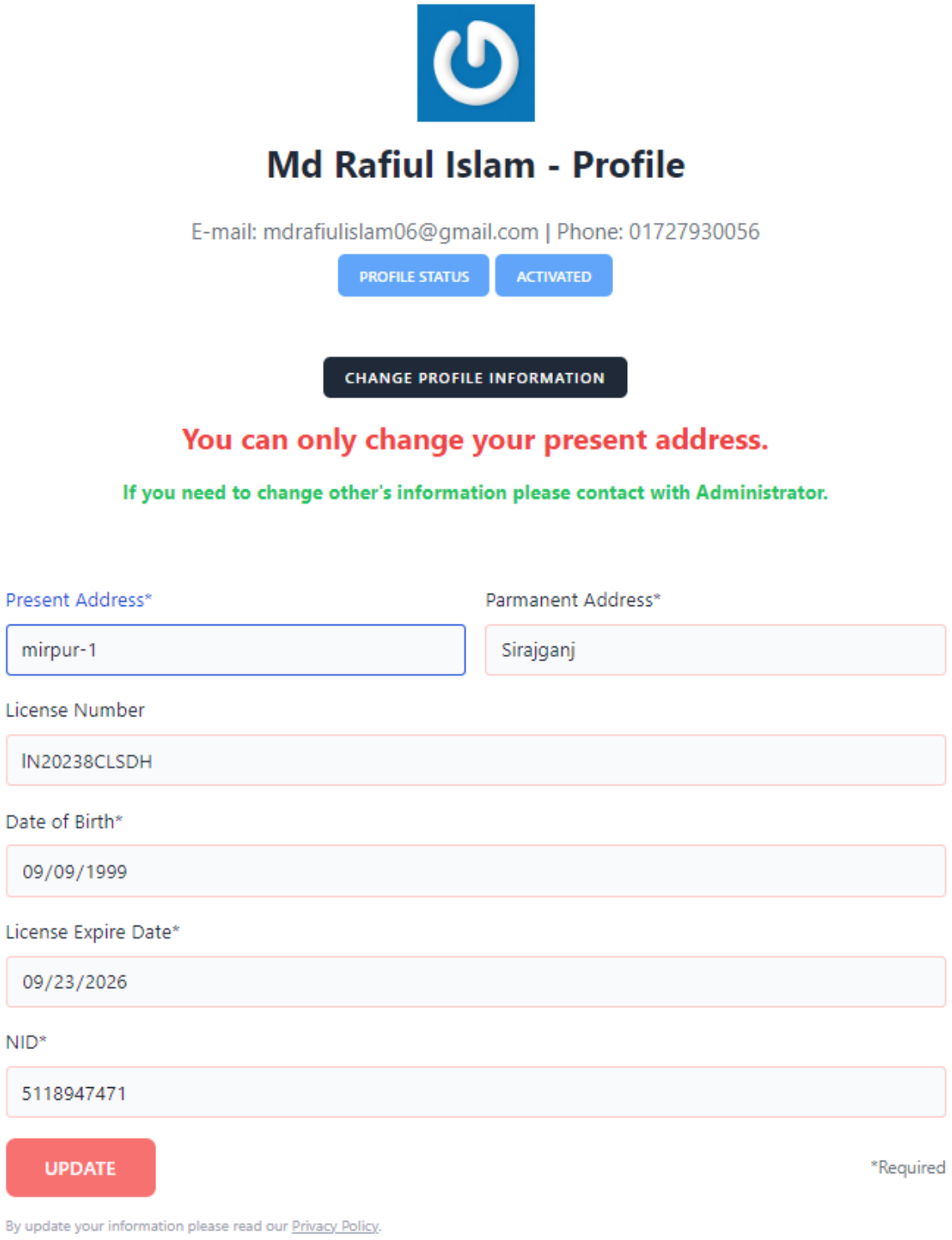


Fig 38: Driver another profile (Not updatable all data)

5.8 User/Drivers Operations

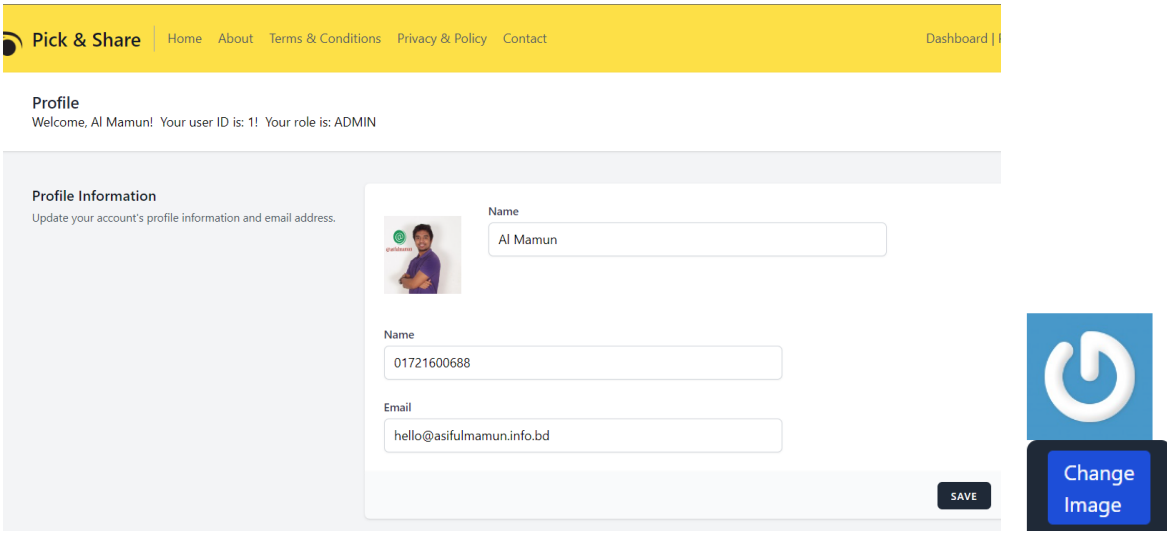


Fig 39: Profile also profile picture changeable (3rd party gravatar)

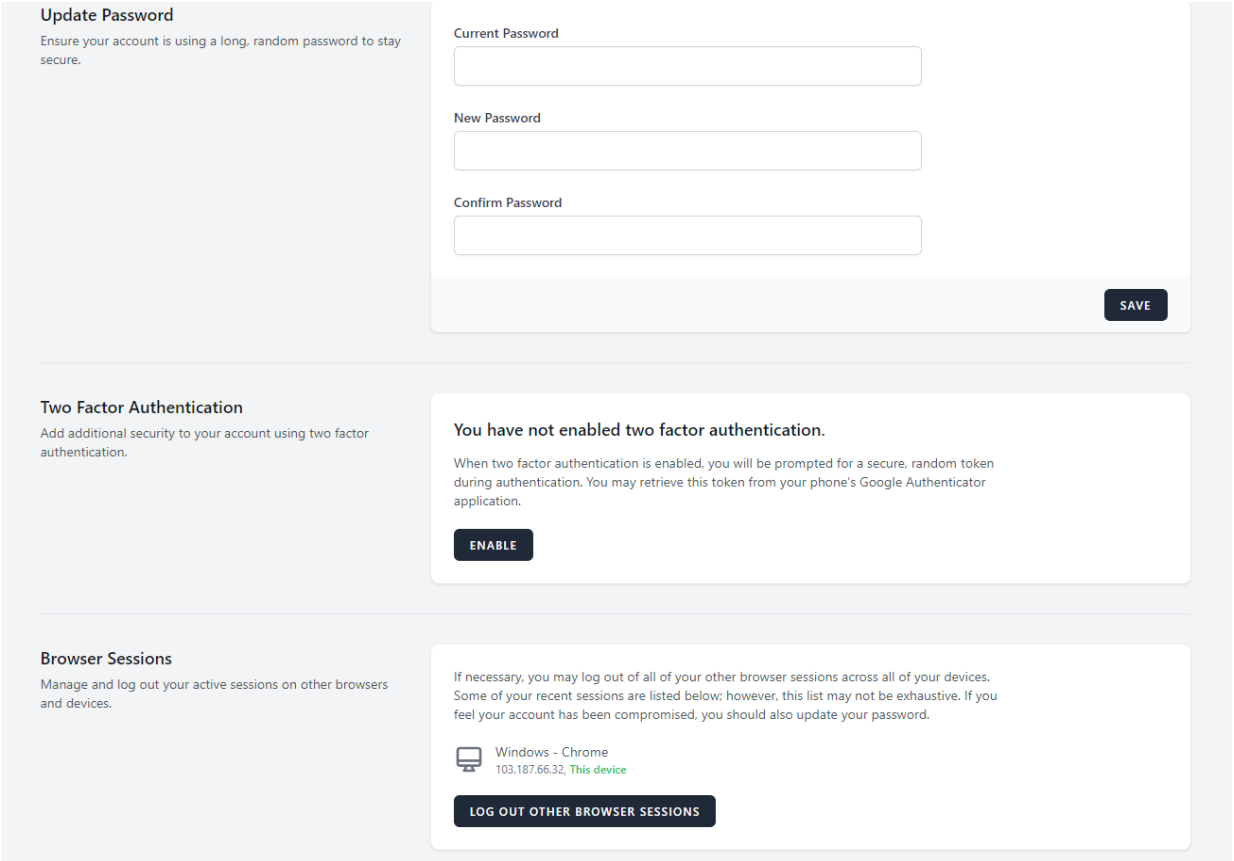


Fig 40: Password change, 2FA & Session (device) manager

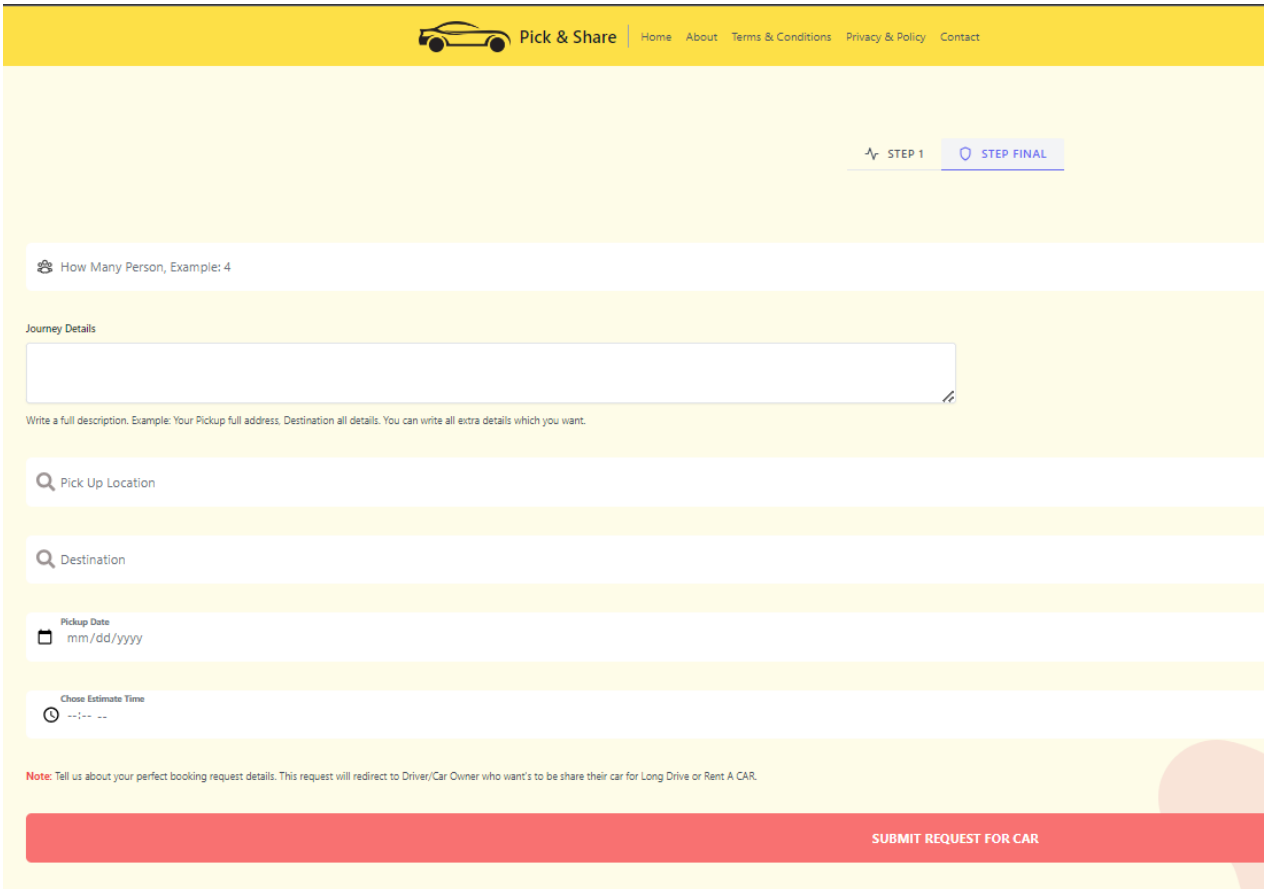


Fig 41: Booking Request Page

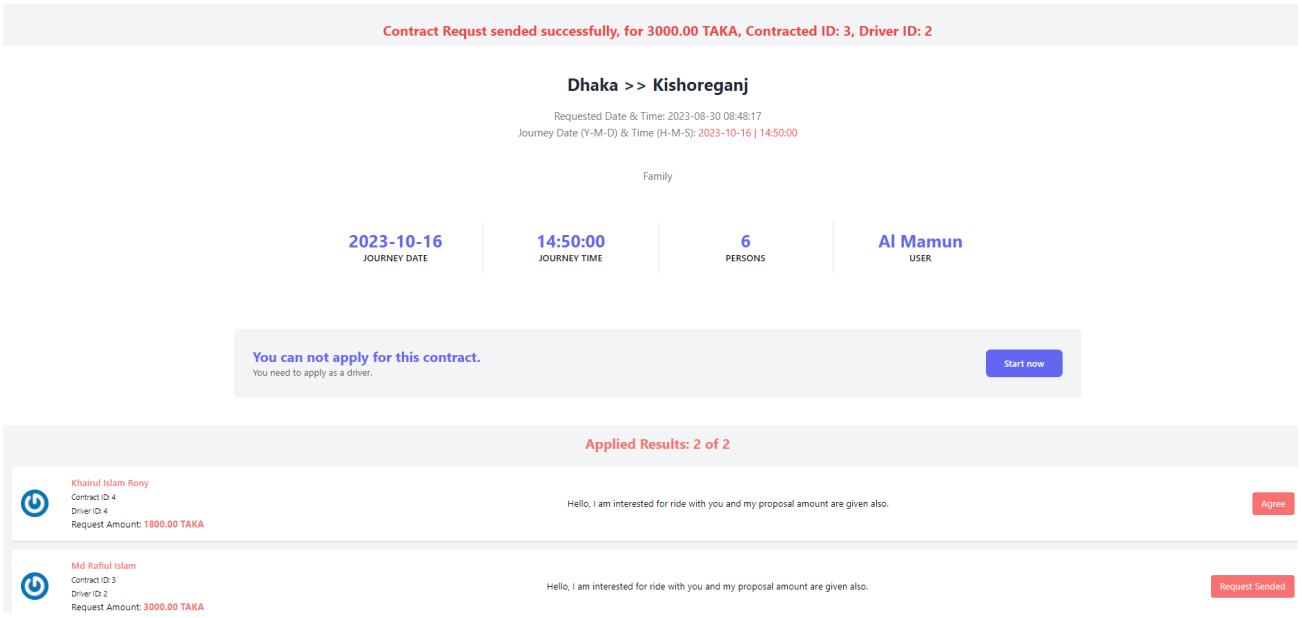


Fig 42: Drivers Response from Requested user (logged user)

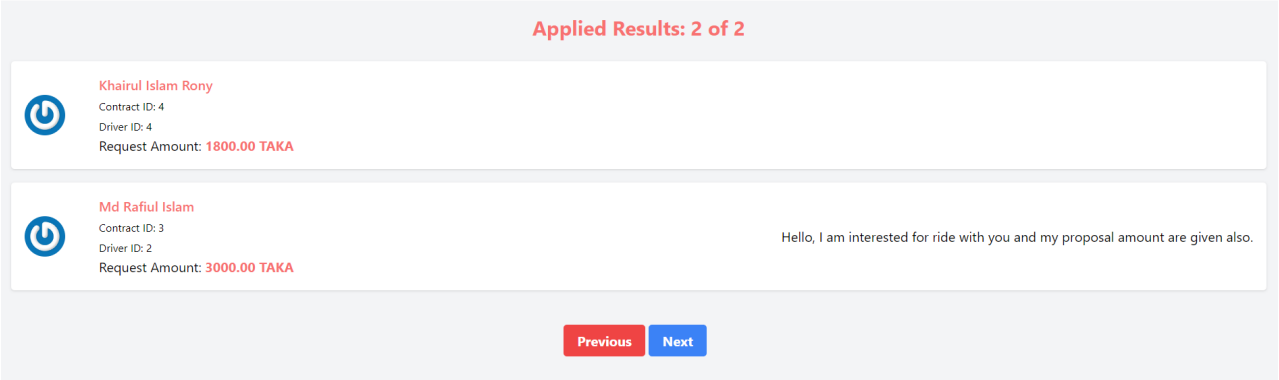


Fig 43: As a driver can see only own request details

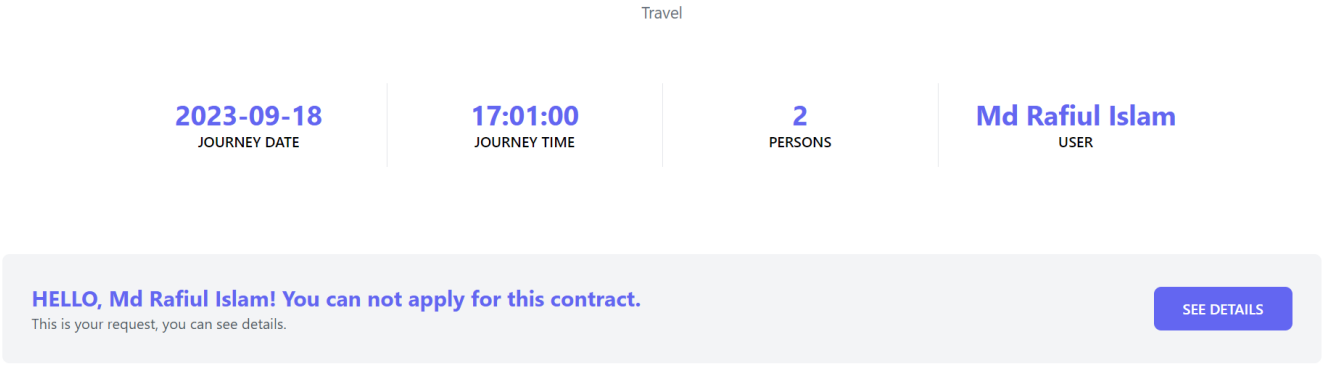


Fig 44: As a driver can not response own booking request

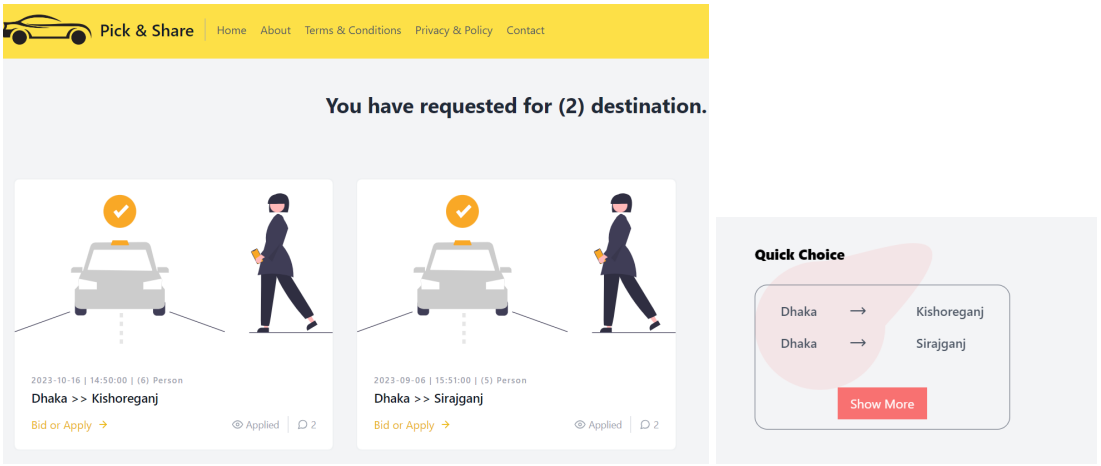


Fig 45: Booking Requests also quick list

Chapter 6

Testing and Quality Assurance

6.0 Introduction

Testing and Quality Assurance (QA) are crucial for ensuring the reliability and functionality of a ride-sharing website. It involves a series of processes to identify and rectify issues before the website is launched.

Functional Testing:

Ensuring that all the features of the ride-sharing website work as intended. This involves testing things like account registration, booking rides, fare calculation, payment processing, and driver-passenger interactions.

Compatibility Testing

Verifying that the website works correctly on different web browsers, devices (desktops, tablets, smartphones), and operating systems.

Performance Testing

Evaluating the website's responsiveness and load-handling capabilities, particularly during peak usage times, to prevent slowdowns or crashes.

Security Testing

Checking for vulnerabilities and ensuring that user data is protected. This includes testing against common threats like SQL injection, cross-site scripting, and data breaches.

Usability Testing

Ensuring that the website is user-friendly and easy to navigate. This involves testing the user interface, user experience, and overall design.

Localization Testing

Verifying that the website functions correctly in different languages and cultural contexts.

Accessibility Testing

Ensuring that the website is accessible to users with disabilities, following accessibility guidelines like WCAG.

Load Testing

Testing the website's ability to handle a large number of concurrent users to prevent crashes during periods of high demand.

User Acceptance Testing (UAT)

Involving actual users to test the website and provide feedback to ensure that it meets their needs and expectations.

Continuous Testing

Implementing automated testing processes that run continuously to catch issues early in the development cycle.

By performing thorough testing and QA, you can enhance the reliability, security, and overall user experience of your ride-sharing website, leading to increased user trust and satisfaction.