LEGACY GARAGE

A PROJECT REPORT

Submitted by

Tarak Manharbhai Nayka

210170116061

Inpartial fulfilment for the award of the degree

Of

BECHELOR OF ENGINEERING

In

Information & Technology

Vishwakarma Government Engineering College,
Chandkheda,Ahmedabad,382424

Gujarat Technology University, Ahmedabad

[April,2025]

Vishwakarma Government Engineering College, Chandkheda

Ahmedabad, 382424

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **LEGACY GARAGE** (**Store OLD Car or buy/sale Car**) has been carried out by **Nayka Tarak Manharbhai** under my guidance in partial fulfillment for the degree of Bachelor of Engineering in Information Technology, 8th Semester of Gujarat Technological University, Ahmadabad during the academic year 2024-2025.

Prof. Dipak.C.patel	Prof.Vibha.D.Patel
Internal Guide	Head of the Department

Company Certificate

GROWNITED

Date: - 19/04/2025

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Tarakkumar Manharbhai Nayka**, a student of Vishwakarma Government Engineering College has successfully completed his/her internship in the field of MERN STACK from 20th January 2025 to 19th April 2025 12 Weeks under the guidance of **Rahul Kirpekar**.

His internship activities include Project Understanding & Requirement Analysis, Technical Documentation, Learning New Technologies, Development Activities, Testing & Quality Assurance, Soft Skills & Reporting.

During the internship period, he/she was exposed to various processes and was found to be diligent, hardworking, and inquisitive.

We wish him/her all the best in future endeavors.

For, Grownited Private Limited.

Rahul Kirpekar

(Authorised Signature)

301-305, 3rd Floor, Surbhi Complex, Nr. Municipal Market, Chimanlal Girdharlal Road, Navrangpura,Ahmedabad Gujarat- 380009, 7874014621, hr@grownited.com

Vishwakarma Government Engineering College, Chandkheda

Ahmedabad, 382424

DECLARATION

We hereby declare that the Internship report submitted along with the Internship entitle **LEGACY GARAGE** (**Store OLD Car or buy/sale Car**) submitted in partial fulfilment for the degree of Bachelor of Engineering in Information Technology to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me at **Royal Technosoft** under the supervision of Rahul Kirpekar and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of The Student Sign of Student

1. **TARAK NAYKA**

ACKNOWLEDGEMENT

This Internship has been the most practical and exciting part of my learning experience, which would be an asset for me and also for my future carrier. With a deep sense of gratitude and respect, I wish to express my heartfelt appreciation to all those who have contributed to this project, both explicitly and implicitly, without the co-operation of whom, it would not have been possible to complete this. No system is created entirely by individual. Many people have helped to create this system and each of their contribution has been valuable. Proper organization of concept and analysis of system is due to keen interest and helping hand of our teachers and colleagues.

I would like to thank my **Head of Department Prof. V. D. Patel**, who was a constant source of inspiration. My most sincere thanks to my Internal guide **Prof. Patel Dipak sir** for his kind co-operation and who has always been guiding, encouraging and motivating us throughout the project. I am grateful to my college **Vishwakarma Government Engineering College** for providing me all the required resources and a good working environment.

My most sincere thanks to my External guide **Mr. Rahul Kripekar**, for his kind co- operation and who has always been guiding, encouraging and motivating us throughout the project. I am grateful to my company Grownited Pvt Ltd. for providing me all the required resources and a good working environment.

Thanking you,

TarakKumar Manharbhai Nayka

ABSTRACT

Legacy Garage is an online platform designed for people who love old and vintage cars. It allows users to explore and buy classic cars, but only the admin has the authority to add new car listings to the site. Each listing includes detailed information, photos, and the history of the vehicle. Users can create accounts to view cars, save favorites , and make purchases. The platform also features a store section where users can buy old car parts, accessories, and collectibles. An admin dashboard provides full control over listings, users, and store items. The website is fully responsive, working well on both desktop and mobile devices. Legacy Garage is built using modern technologies like React, Node.js, Express.js, and MongoDB to ensure speed, security, and smooth performance. The project aims to bring together vintage car lovers and keep the classic car culture alive online.

LIST OF FIGURE

LIST OF TABLE

LIST OF ABSERVATION

Following is the list of abbreviations used in the report:

Abbreviation	Full Form
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
JS	Javascript
API	Application Program Interface
DB	Database
JSON	Javascript Object Notation
QA	Quality Analysis
JSX	Javascript XML
UI	User Interface
MERN	MongoDB, Express, React.js, Nodejs
DOM	Data Object Model

TABLE OF CONTENT

CHAPTER 1: OVERVIEW OF THE COMPANY

1.1 ABOUT COMPANY

Figure 1.1 Grownited Logo

Company Name: Grownited.

Address: 3rd floor, Surbhi Complex, Navrangpura , Ahmedabad, 380009

Contact No.: 7874014621

E-mail: contact.grownited@gmail.com

Grownited is a IT solutions provider and technology consulting firm dedicated to empowering businesses through innovative and reliable digital services. As a trusted partner to organizations of all sizes—from growing startups to Fortune 100 enterprises—Grownited offers end-to-end IT consulting, system integration, and managed service solutions across industries.

With a strong focus on customer satisfaction and continuous innovation, Grownited is committed to improving lives through technology. The company specializes in delivering high- quality, secure, and scalable applications across multiple platforms, including Android, iOS, Windows, Web, and enterprise-level solutions.

Driven by a passion for excellence, Grownited combines deep technical expertise with a client- first approach to ensure every solution not only meets business goals but also adds lasting value. Whether it's mobile development, cloud integration, or digital transformation, Grownited is your reliable partner in shaping the future of technology.

1.2 OVERVIEW OF PRODUCT & WORK

The product developed during the internship is a web-based application named Legacy Garage, which serves as a digital platform for showcasing and selling old and vintage cars. The project is focused on creating a user-friendly system that allows car enthusiasts and buyers to explore detailed listings of classic cars, while ensuring that only the admin can upload and manage these listings. This approach maintains quality control and builds trust among users.

The platform includes two major modules: the Car Listing System and the Auto Parts Store. The Car Listing System displays various vintage cars with complete specifications, high-resolution images, pricing, and ownership history. The Auto Parts Store allows users to browse and purchase accessories or spare parts related to old cars, further enhancing the user experience.

Work carried out included full-stack development using the MERN stack (MongoDB, Express.js, React.js, Node.js). Tasks involved setting up the database schema, designing frontend UI components, creating secure login systems, and developing admin-specific functionality to add, update, and remove car listings and store items.

This project reflects real-world practices in building e-commerce and catalog-style platforms and helped enhance technical, design, and problem-solving skills during the internship.

CHAPTER 2: PROCESS BEING CARRIED OUT IN COMPUNY

2.1 WORK BEING CARRIDE OUT AT COMPANY

During the internship, the main work involved the development of a web application named Legacy Garage, which is a platform for showcasing and selling old and vintage cars. The work was focused on the backend and frontend development, database design, admin panel features, and user interface design.The internship provided hands-on experience with real-time project development using the MERN (MongoDB, Express.js, React.js, Node.js) technology stack. Tasks included creating RESTful APIs, designing responsive UI components, implementing secure login systems, and managing data storage and retrieval processes.Additionally, the work involved regular meetings with the supervisor or mentor to discuss project progress, get feedback, and implement changes accordingly. Version control tools like Git were used to track changes and manage the codebase efficiently. The overall focus was on delivering a clean, functional, and scalable system that meets user needs and business goals.This work experience gave valuable insights into full-stack development and helped improve technical and teamwork skills in a professional environment.

2.2 STAGE EXPLAINATION

The development of the Legacy Garage project was carried out in several key stages to ensure a smooth and structured workflow. Each stage contributed to the successful completion of the platform, from planning to deployment. The main stages are explained below:

1. Requirement Gathering:
   In this stage, the basic idea of the project was discussed and finalized. The features and functions of the system were defined, including the focus on admin-only car listing, user registration, and a parts store.
2. System Design:
   This stage involved designing the architecture of the system, including database schema, user interface layouts, and system flow diagrams such as Use Case, ER Diagram, and Class Diagram.
3. Frontend Development:
   Using React.js, the user interface was created with a focus on simplicity, responsiveness, and smooth user experience. Components like car listing cards, login forms, and product views were developed.

4. Backend Development:
   The server-side logic was implemented using Node.js and Express.js. APIs were developed for user management, admin control, product handling, and data interactions with MongoDB.
5. Testing:
   Manual testing was done to check the system's functionality, usability, and performance. Bugs were fixed, and improvements were made based on feedback.
6. Deployment and Review:
   The final project was deployed in a live environment or demo server. Project review sessions were conducted to analyze outcomes and areas of improvement.

Each stage was handled step-by-step to ensure a professional approach and quality output. This structured workflow helped in achieving the project goals effectively within the internship period.

CHAPTER 3: INTRODUCTION

3.1 INTERNSHIP SUMMARY

I completed my internship at Grownited, a growing IT company focused on web and software development. During this internship, I worked on a live project named Legacy Garage – Old Car Shop and Store. The platform is designed for users to explore and buy vintage and classic cars. A key feature of the system is that only the admin can add and manage car listings. The platform also includes a store for old car parts and accessories. The entire project was developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js). I was responsible for designing interfaces, building backend APIs, and testing features. This internship helped me understand real-world development practices. I gained hands-on experience with full-stack technologies and improved my problem-solving skills. Overall, the internship at Grownited was a great learning experience that prepared me for future challenges.

3.2 PURPOSE

The purpose of the **Legacy Garage** project is to create a reliable and user-friendly online platform for showcasing and selling vintage and old cars. It aims to bring together car enthusiasts, collectors, and buyers who are interested in classic automobiles. The system is designed in such a way that only the **admin** can add or manage car listings, ensuring the authenticity and quality of the content. In addition to cars, the platform also includes a store for vintage auto parts and accessories. By developing this system, the goal is to preserve the legacy of classic vehicles while using modern technology to simplify the buying and browsing experience for users.

3.3 OBJECTIVE

- To develop a web-based platform for showcasing and selling old and vintage cars.
- To ensure that only the admin can add, update, or remove car listings for better control and authenticity.
- To create a user-friendly interface where users can view car details and purchase auto parts.
- To build a secure login and registration system for users.
- To implement a searchable and filterable listing system for easy browsing.
- To design a responsive layout that works on both desktop and mobile devices.

- To manage and store data efficiently using MongoDB.
- To use modern web technologies like React, Node.js, and Express for full-stack development.
- To provide an admin dashboard for managing users, cars, and products.
- To gain real-world experience in full project development and deployment.

3.4 SCOP

The scope of the Legacy Garage project is to develop a complete web-based system for displaying and managing vintage car listings and related auto parts. The platform is designed for users who are interested in viewing and purchasing old cars, while giving full control to the admin for managing all content. The project includes key modules like user registration and login, car browsing with filters, product listings, and an admin dashboard. It supports responsive design for both desktop and mobile devices. The system is scalable, allowing future enhancements such as payment integration, user reviews, and advanced search features. The project uses the MERN stack, making it suitable for real-time, dynamic applications. This platform can be used by businesses that sell classic cars and spare parts to reach a wider audience online.

3.5 TOOLS AND TECHNOLOGY

Text Editor : VS CODE

Programming Languages : JavaScript

TECHNOLOGY

The Legacy Garage platform was developed using the MERN stack (MongoDB, Express.js, React.js, Node.js), which provides a robust and scalable architecture for building modern web applications.

- MongoDB: A NoSQL database used to store and manage car listings, user data, and store items. Its flexibility and scalability make it a great choice for this project.
- Express.js: A web framework for Node.js that simplifies the creation of backend APIs and server-side logic. It provides a clean and structured way to handle routing and HTTP requests.
- React.js: A powerful frontend library used to build a dynamic and responsive user interface. React ensures a smooth user experience, with real-time updates and component-based architecture.

- Node.js: A JavaScript runtime used to build the backend of the application. Node.js allows for fast and scalable network applications, handling multiple requests efficiently.
- JWT (JSON Web Tokens): Used for secure user authentication and authorization, ensuring that only authorized users can access certain parts of the platform.
- CSS (Tailwind CSS): A utility-first CSS framework used for designing a responsive and visually appealing interface.

Together, these technologies ensure a fast, scalable, and secure application for both admins and users.

3.6 INTERNSHIP PLANNING

The internship at Grownited was divided into key phases to ensure structured development of the Legacy Garage project:

1. Initial Phase (Week 1-2):
   Understand project requirements and set up the development environment.
2. Design Phase (Week 3-4):
   Design UI/UX wireframes and system architecture.
3. Development Phase (Week 5-8):
   Implement core features like car listings, user registration, and admin dashboard.
4. Testing Phase (Week 9-10):
   Conduct testing, debug issues, and optimize performance.
5. Final Phase (Week 11-12):
   Deploy the project and prepare the final report and presentation.

The planning was flexible to ensure a comprehensive learning experience while developing the project.

3.6.1 Project Development Approach:

The development of the Legacy Garage project followed a phased and structured approach based on best practices of Agile methodology. This approach allowed for flexibility and iterative progress, enabling continuous feedback and improvement throughout the project lifecycle.

- **Requirement Analysis and Planning:**
  The project started with a comprehensive analysis of the core requirements. Key features such as car listings, user registration, product store for parts, and the admin dashboard were outlined and defined. This phase included

understanding user roles, data management needs, and overall functionality.

A detailed project plan was created, specifying timelines, development milestones, and tasks.

- **Design Phase:**

  During the design phase, UI/UX wireframes were created to define the look and feel of the platform. The goal was to design a simple, clean, and user-friendly interface. The system architecture was also planned, with a clear definition of the database schema (MongoDB), API endpoints (Express.js), and interaction flows between different modules.

  The design was focused on both the frontend and backend, ensuring that the platform was responsive, easy to navigate, and scalable for future updates.

- **Iterative Development and Implementation:**

  The development was broken down into short sprints, focusing on one module at a time. Each sprint consisted of both frontend and backend tasks. The frontend development focused on creating responsive React components for the car listings, registration forms, product pages, and an interactive admin panel. The backend development involved setting up the Express.js server, building RESTful APIs for CRUD operations (Create, Read, Update, Delete), and integrating MongoDB for data storage.

  User authentication was implemented using JWT (JSON Web Tokens) to ensure secure access for users and admins.

- **Testing and Quality Assurance:**

  After the completion of each sprint, thorough manual and functional testing was conducted. The goal was to identify bugs, ensure the system met requirements, and improve overall functionality.

  Regular feedback sessions with the mentor ensured that the project stayed aligned with the initial objectives. The testing phase also focused on performance optimization, debugging, and resolving any security concerns.

- **Deployment and Final Review:**

  Once the core features were implemented and tested, the system was deployed to a staging server for real-time use. This included testing the application under live conditions and ensuring that all features worked as expected.

  A final project review was conducted with the mentor and other team members to evaluate the platform's performance, scalability, and usability. Improvements were made based on feedback received during this phase.

3.6.2 ROLES AND RESPONSIBILITIES

During my internship at Grownited, I actively participated in the development of the Legacy Garage project. I was assigned multiple responsibilities across different phases of the project. These responsibilities helped me gain hands-on experience in full-stack development. My key roles and responsibilities included:

- Requirement Understanding:
  Participated in initial discussions to understand the client's needs and the overall scope of the Legacy Garage platform.
  Assisted in documenting the system requirements and feature list.
- UI/UX Design Contribution:
  Helped create wireframes and layout designs for user and admin interfaces using Figma.
  Focused on ensuring a clean, responsive, and user-friendly design.
- Frontend Development:
  Developed key components of the user interface using React.js, including the homepage, car listings, product pages, and user login/register pages.
  Ensured responsiveness and cross-browser compatibility.
- Backend Development:
  Built and integrated RESTful APIs using Node.js and Express.js.
  Handled backend logic for user authentication (JWT), data management, and admin access controls.
- Database Management:
  Designed and maintained MongoDB collections for storing user data, car listings, and store products.
  Performed CRUD operations and ensured efficient data handling.
- Testing and Debugging:
  Conducted manual testing of each module.
  Identified and resolved bugs to ensure smooth functionality across the application.
- Version Control and Collaboration:
  Used Git and GitHub for source code management and collaboration.
  Worked closely with team members and mentor to stay aligned with development goals.
- Deployment and Final Review:
  Helped deploy the project on a local/staging server for testing.
  Participated in the final review and presentation of the project to the mentor.

CHAPTER 4: SYSTEM ANALYSIS

4.1 STUDY OF CURRENT SYSTEM

1. Frontend – React.js

React.js is the primary technology I use for building user interfaces. It is based on a component-driven architecture and follows a declarative programming model. With React, I am learning to create reusable UI components, manage state using Hooks (useState, useEffect), and efficiently update the DOM using the Virtual DOM mechanism. I also implement routing through React Router, enabling smooth navigation within single-page applications.

2. Backend – Node.js and Express.js

On the server side, I am working with Node.js, a runtime environment that executes JavaScript on the server, and Express.js, a lightweight web framework that simplifies the creation of server-side APIs. I am learning how to set up routes, handle HTTP requests, and use middleware to manage data flow and user authentication.

3. Database – MongoDB

For the database layer, I use MongoDB, a NoSQL database that stores data in flexible, JSON- like documents. Through Mongoose, an ODM (Object Data Modeling) library, I'm learning to define schemas, perform CRUD operations, and connect the backend with the database to store and retrieve application data.

4. API Communication

React communicates with the backend through RESTful APIs using tools like fetch or Axios. I am understanding how to send and receive data asynchronously, handle errors, and show appropriate UI responses.

5. Version Control & Collaboration

I am also learning to use Git for version control and GitHub for code collaboration. I regularly push my code, create branches, and follow basic pull request practices to collaborate with team members and mentors.

6. Development Workflow

•Setup of development environment using tools like VS Code.

•Running frontend and backend servers concurrently using concurrently

•Writing clean, modular, and maintainable code.

4.2 PROBLEM AND WEAKNESS OF CURRENT SYSTEM

Before developing the Legacy Garage platform, an analysis of the existing systems used for buying and selling old or vintage cars revealed several problems and limitations. These issues affect both user experience and system efficiency. Some of the key weaknesses identified are:

Lack of Specialization:

Existing platforms like OLX and other classified sites cater to all types of vehicles, which creates clutter and makes it difficult for users interested specifically in old or vintage cars to find relevant listings.

Unverified Listings:

In most current systems, anyone can upload a listing without proper verification. This leads to fake, duplicate, or low-quality postings that reduce the trustworthiness of the platform.

No Admin Control:

Most platforms lack a strong admin control system, which results in inconsistent content, inappropriate posts, and lack of standardization in vehicle details.

Poor User Interface for Classic Car Enthusiasts:

Existing platforms are not designed with classic car lovers in mind. They often lack detailed filters, categories, and designs focused on vintage car aesthetics.

Limited Product Support:

Spare systematically. Buyers often have to look elsewhere, which breaks the user flow parts or related accessories for old cars are rarely available or listed

Security Concerns:

Weak authentication mechanisms lead to spam, scams, and security issues for users. There is no strong system to ensure data integrity and user privacy.

Lack of Personalization:

Users don’t receive personalized recommendations or a tailored browsing experience, which can reduce engagement and satisfaction.

Slow and Unresponsive Systems:

Older platforms are often not optimized for mobile devices and can be slow to load or hard to navigate.

4.3 REQUIREMENTS OF NEW SYSTEM

The new system, Legacy Garage, is designed to address the drawbacks of existing platforms and provide a specialized space for vintage car buyers and enthusiasts. One of the main requirements is that only the admin has permission to add, update, or remove car listings and products, ensuring that all entries are verified and trustworthy. The system must also include secure user registration and login features, allowing users to browse cars and products while protecting their personal information through token-based authentication (JWT). To enhance usability, the platform should offer powerful search and filter options, enabling users to find vehicles and parts based on specific criteria like model, price, and year. From a technical perspective, the system must be responsive, delivering a smooth experience on both desktop and mobile devices. Non-functional requirements include ensuring the system is secure, scalable, and reliable, with performance optimized for fast loading times. The backend and frontend should be well-structured and maintainable, allowing easy updates and future enhancements. Altogether, these requirements form the foundation for a clean, secure, and efficient system tailored for classic car commerce.

4.4 SYSTEM FEASIBILYTY

The Legacy Garage system is highly feasible in terms of technical, operational, and economic aspects. Technically, the project uses the MERN stack (MongoDB, Express.js, React.js, Node.js), which is well-suited for developing scalable and responsive web applications. These technologies are open-source, well-documented, and widely used in the industry, making them a practical choice. Operationally, the system is easy to use for both admin and users. Admins have full control over content, while users enjoy a clean interface with features like search and filters to enhance their browsing experience. The restricted admin-only access model increases trust and quality of listings. Economically, the development and deployment costs are low due to the use of free tools, cloud-based services, and modern frameworks. Additionally, the project requires minimal hardware, making it cost-effective for small businesses or startups. Given these points, the system is both viable and sustainable in the long term.

4.5 FEATURES OF NEW SYSTEM

The Legacy Garage platform is designed with key features that enhance both user and admin experience. Only the admin can add and manage car listings, ensuring quality and authenticity. Users can securely register and browse cars and spare parts, with advanced search and filtering options for easy navigation. The platform is optimized for mobile devices and offers a responsive.

CHAPTER 5: SYSTEM DESIGN

5.1 SYSTEM DESIGN AND METHODLOGY

The design and development of the Legacy Garage system follow a structured and scalable approach, focusing on maintainability, performance, and user experience. The project adopts the Agile development methodology, allowing for iterative progress, regular feedback, and flexibility to adapt to changes during the development cycle. This methodology is ideal for projects like Legacy Garage, where requirements may evolve based on testing and real-time usage. The system is built using the MERN stack—MongoDB for database management, Express.js and Node.js for the backend server and APIs, and React.js for the frontend interface. The entire architecture is component-based and modular, allowing developers to isolate and manage each part of the system such as user authentication, admin panel, product management, and car listings. The system is designed with a layered structure where the frontend communicates with backend APIs via secure HTTP requests, and the backend interacts with the database to fetch or store information. During the design phase, several diagrams were created including Use Case Diagrams, ER Diagrams, and Class Diagrams to visualize the flow of data and user interactions. Special focus was placed on developing a responsive UI, ensuring compatibility across devices. The backend APIs were designed to be RESTful, allowing for efficient and secure data communication. Overall, the system design and methodology not only meet the current requirements but also provide a solid foundation for future enhancements like payment integration, chat support, and customer reviews.

5.2 DIAGRAMS :-

5.2.1 CLASS DIAGRAMS

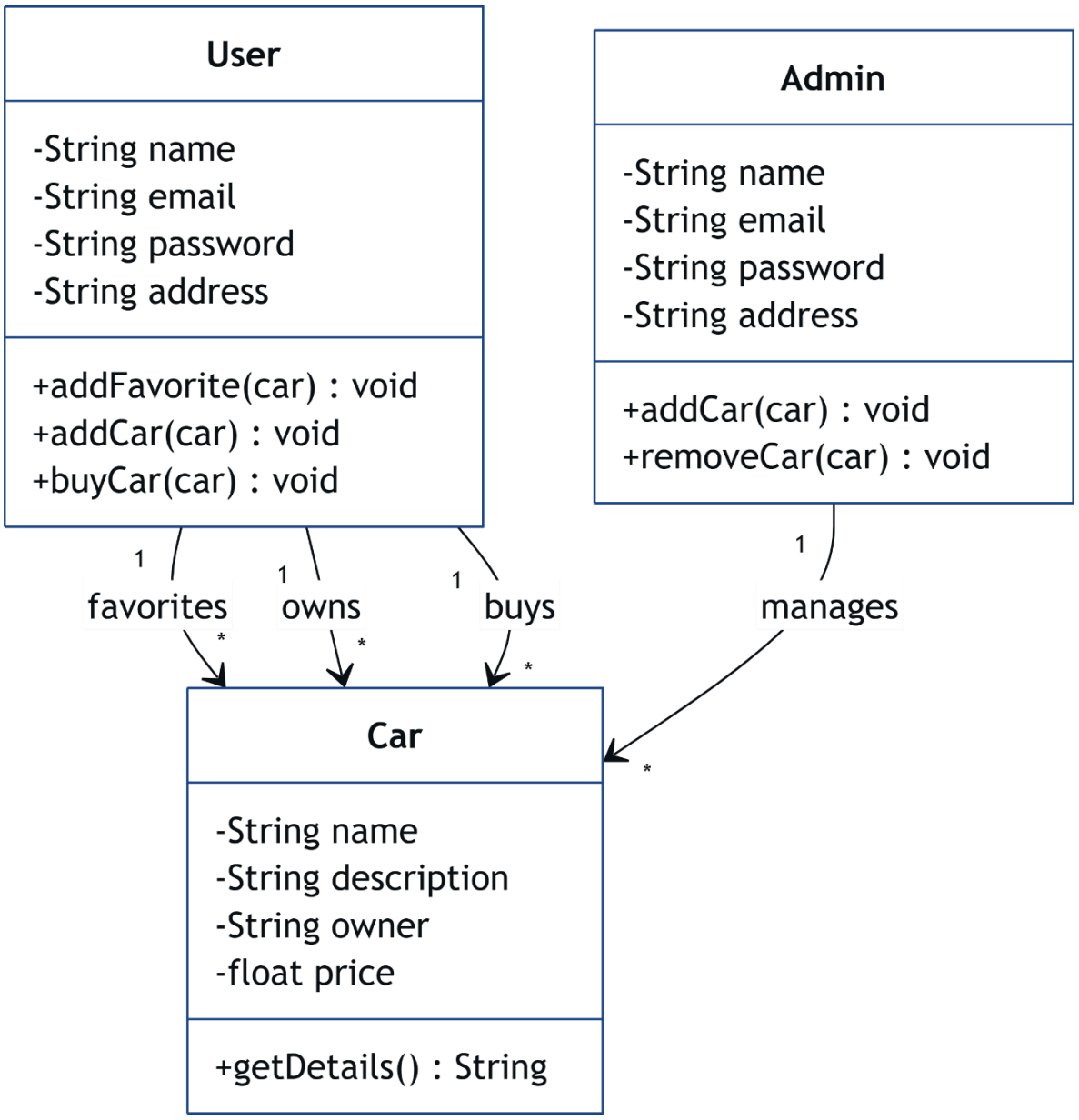

Figure 5.2.1 Class Diagram

5.2.2 E R DIAGRAMS :-

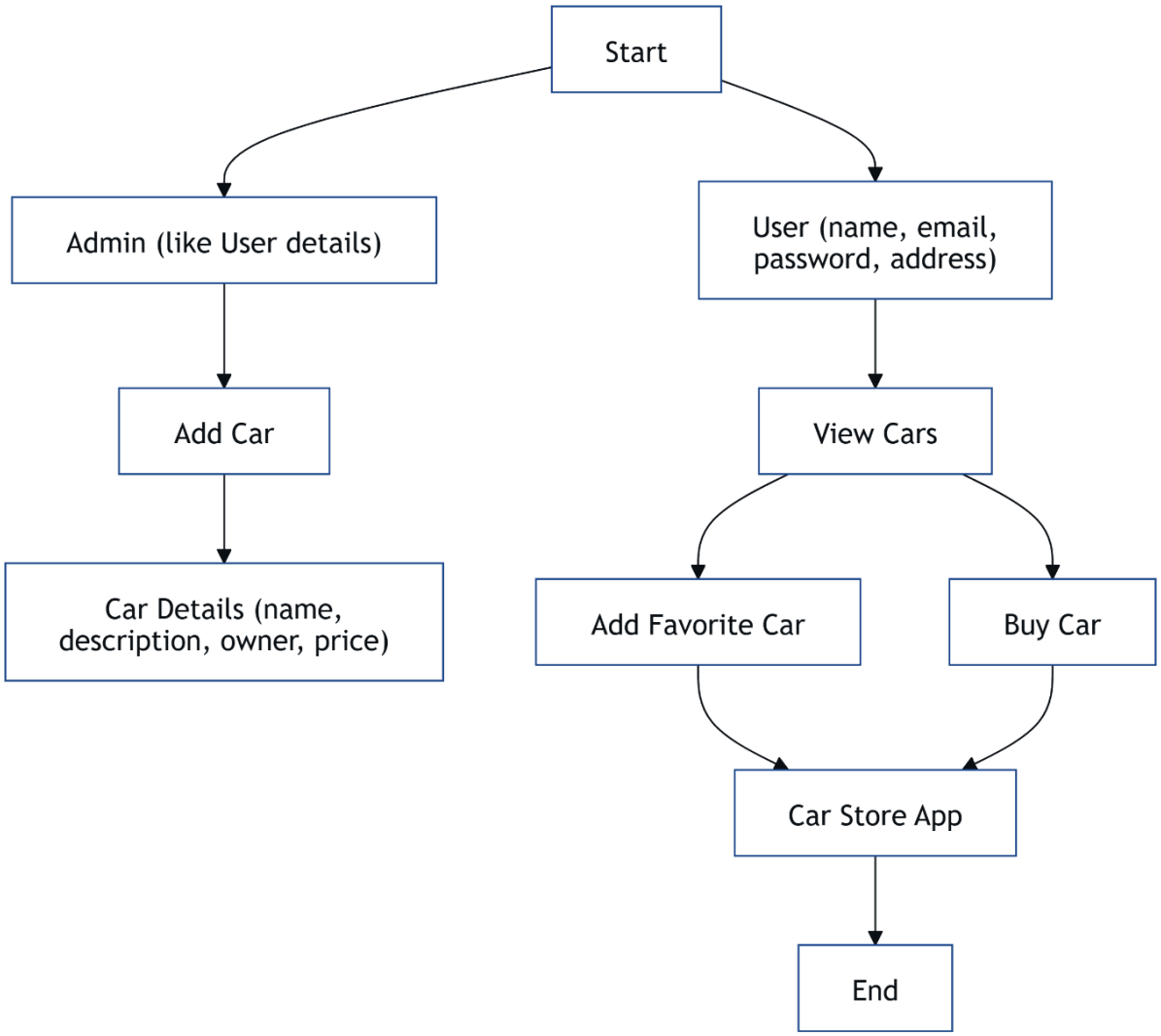

Figure 5.2.2 ER Diagram

5.2.3 STATE DIAGRAMS :-

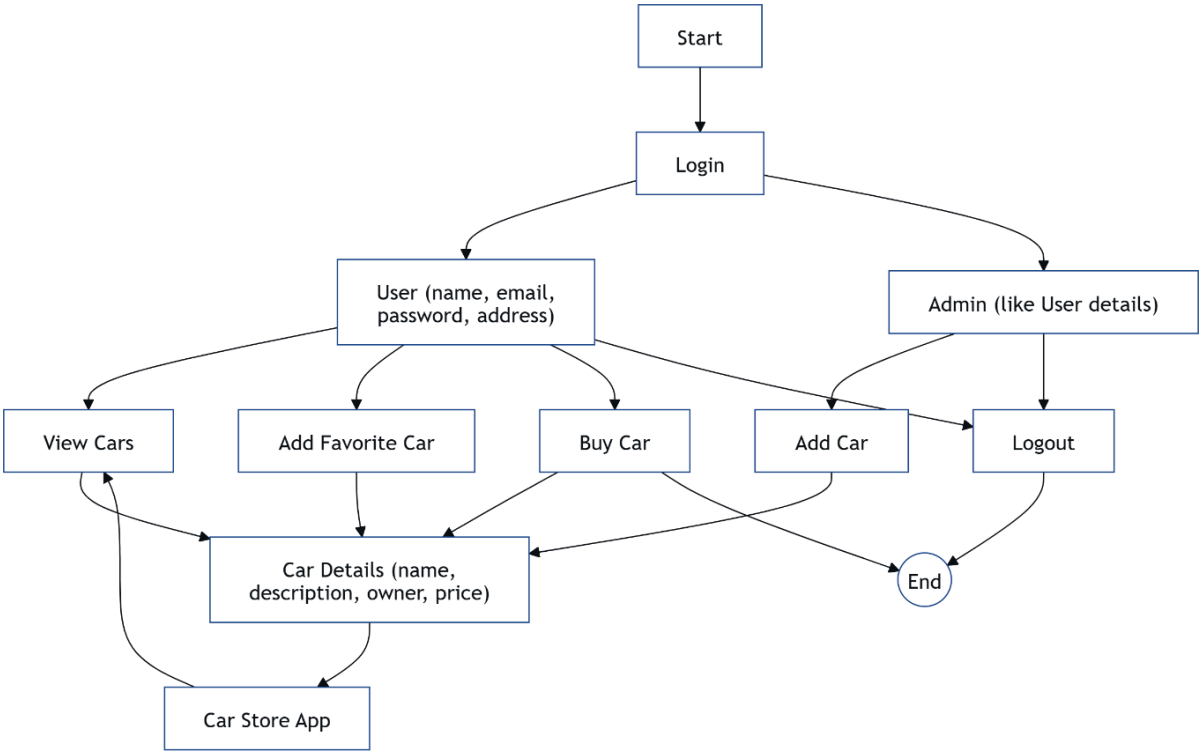

Figure 5.2.3 State Diagram

5.2.4 FLOW DIAGRAMS :-

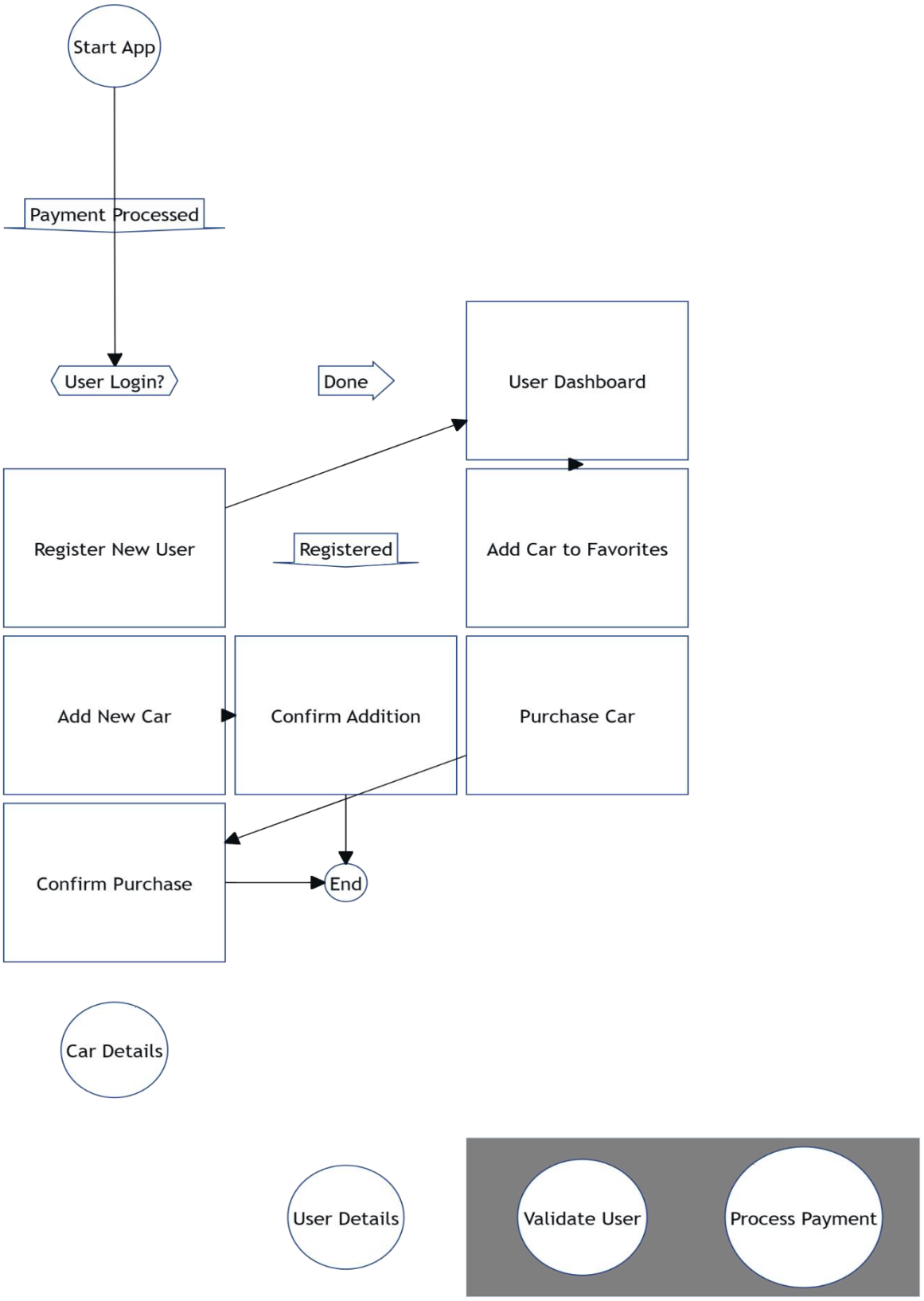

Figure 5.2.4 Flow Diagram

5.2.5 KEYS DIAGRAMS :-

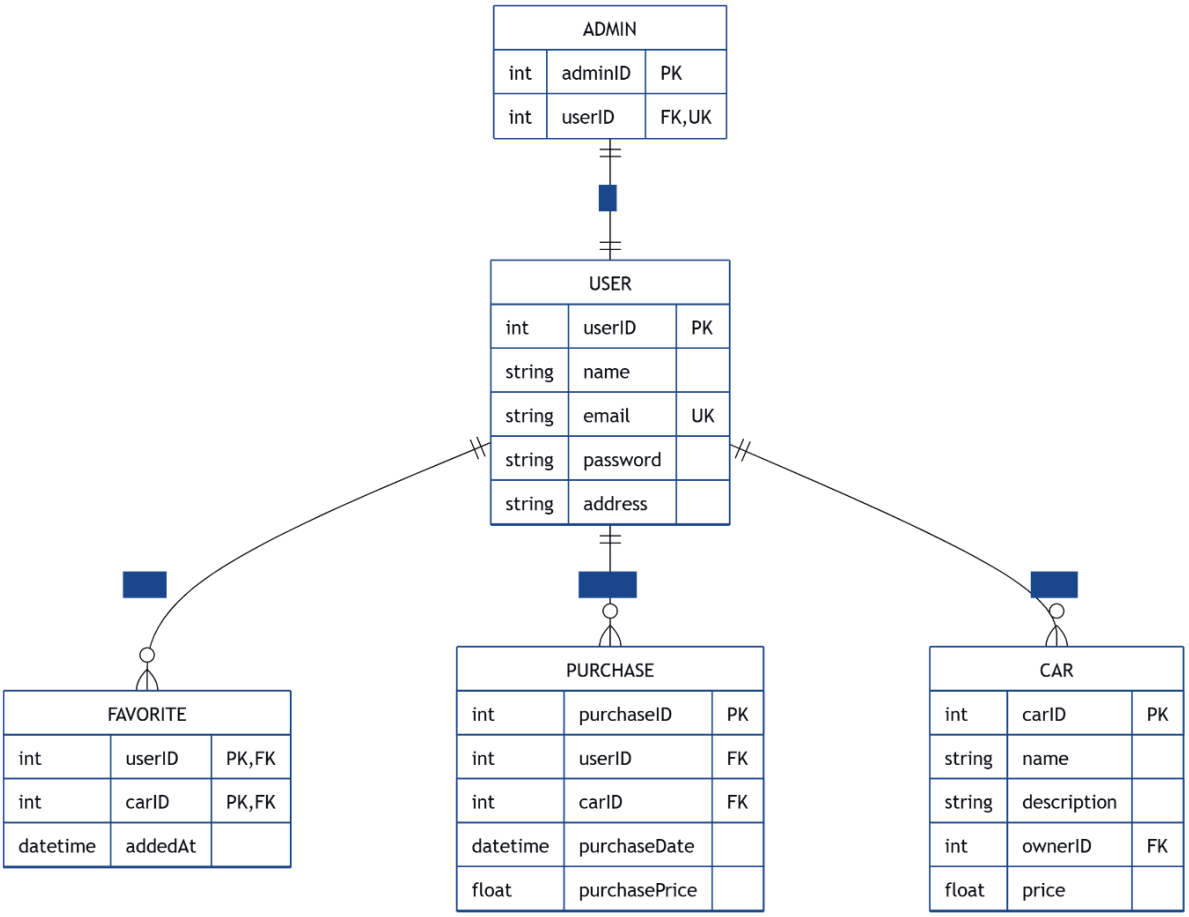

Figure 5.2.5 Key Diagram

5.2.6 TREE MAP :-

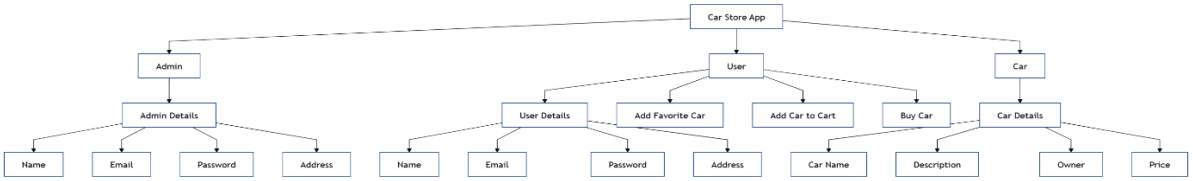

Figure 5.2.6 Tree Map

5.2.6 MIND MAP :-

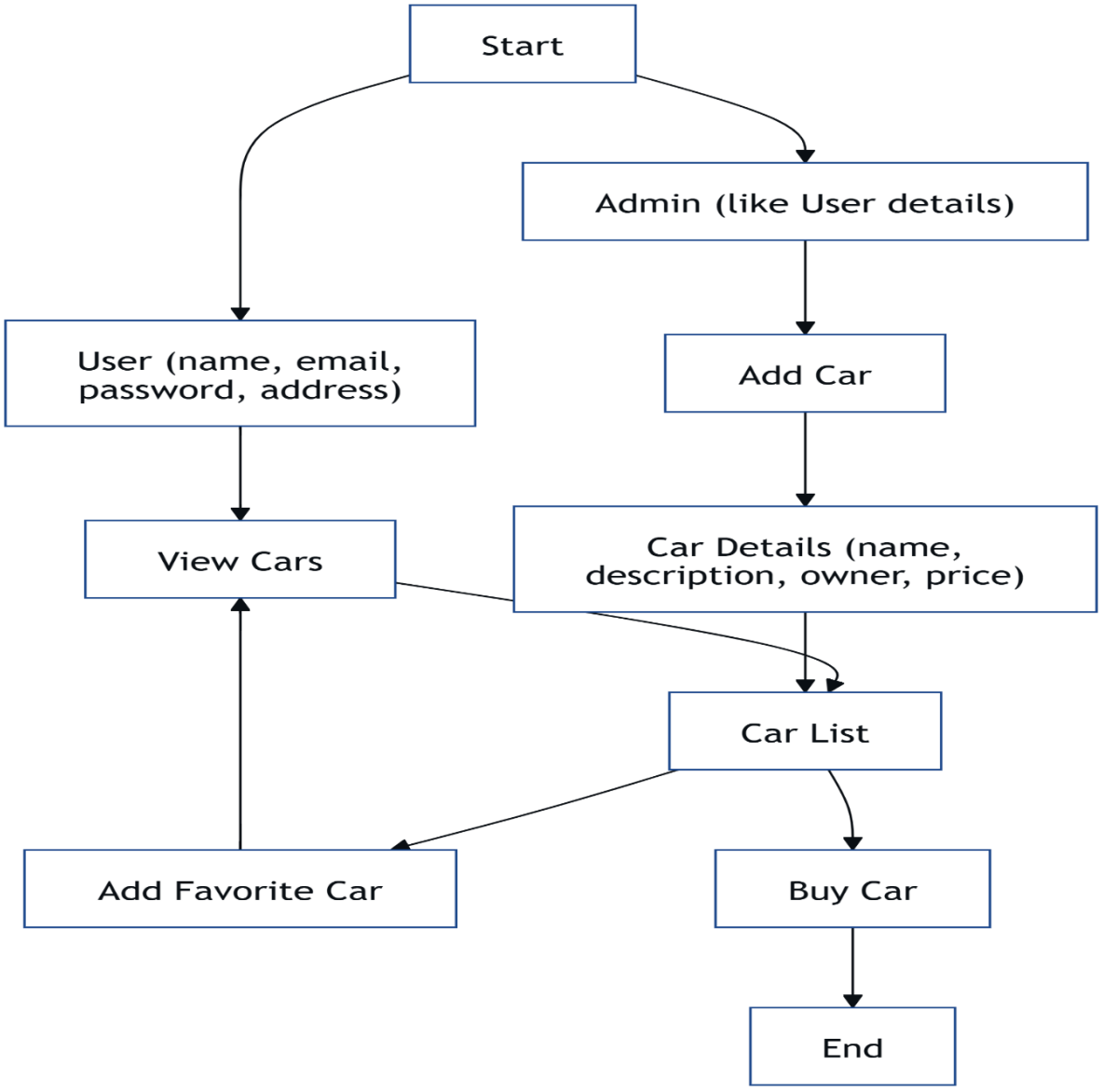

Figure 5.2.7 Mind Map

5.2.6 REQUIREMENT DIAGRAM :-

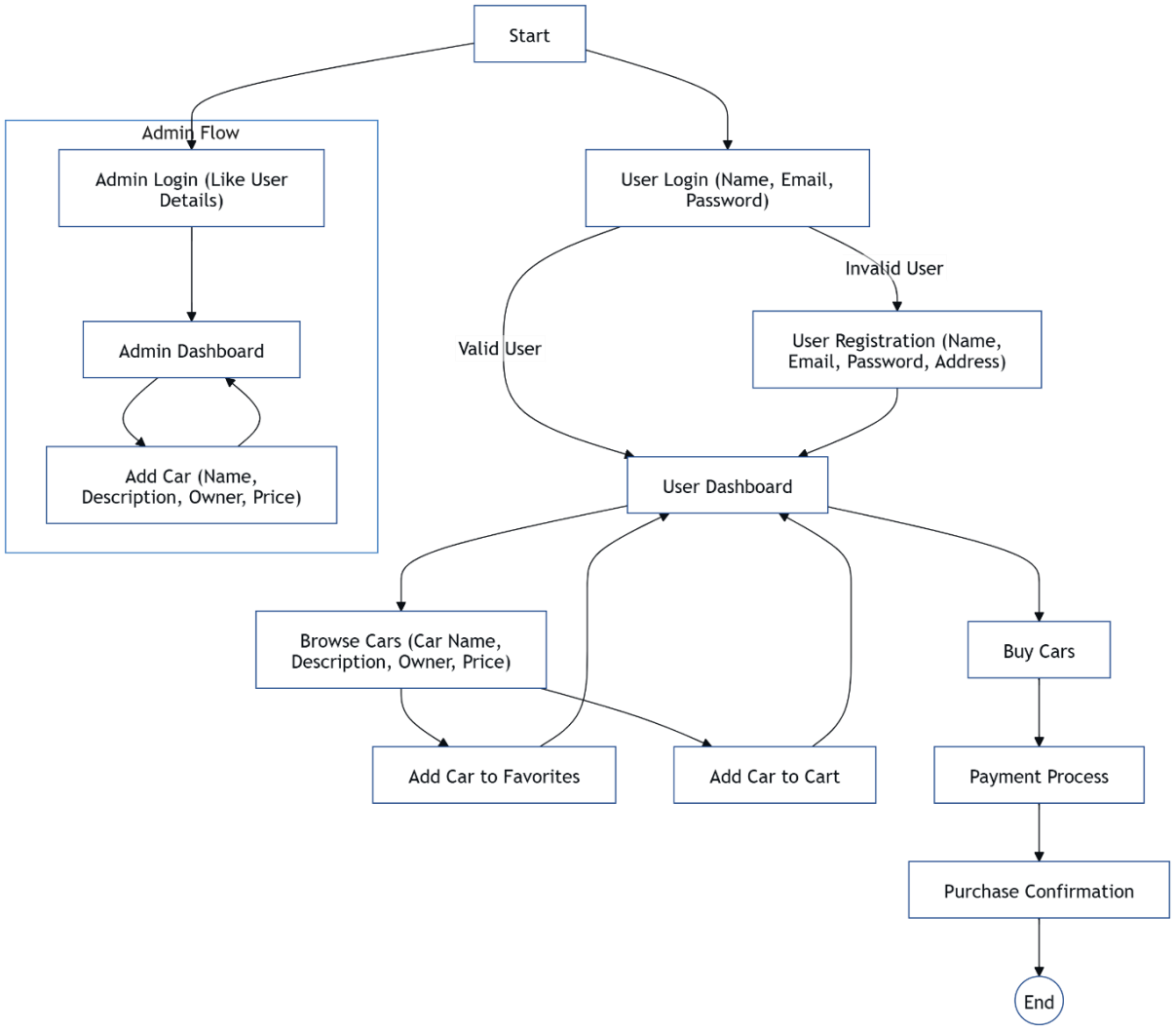

Figure 5.2.8 Requirement Diagram

CHAPTER 6: IMPLEMENTATION

6.1 Implementation Platform

The Legacy Garage project is implemented on the powerful and flexible MERN stack, which provides an end-to-end JavaScript-based solution for building modern web applications. This platform was chosen for its efficiency, scalability, and compatibility across the development environment. Below is a breakdown of the technologies and tools used:

- Frontend–React.js
  React.js is used for building the user interface. Its component-based structure allows for the creation of reusable and modular UI elements. The use of React Router ensures smooth navigation between different pages in the single-page application (SPA).
- Backend–Node.js,Express.js
  The backend of the application is built with Node.js, which allows JavaScript to run on the server side. Express.js is used as a lightweight framework to handle routing, middleware, and HTTP requests efficiently.
- Database–MongoDB
  MongoDB, a NoSQL document database, is used to store user data and car listings. It allows for flexible schema design, making it easy to adapt to changing data structures. Integration is done via Mongoose, which acts as an Object Data Modeling (ODM) library.
- API-Communication
  The frontend communicates with the backend via RESTful APIs. This separation of concerns makes the application more modular and easier to debug and maintain. Tools like Axios or native fetch are used to send HTTP requests.
- Development Tools
  - Visual Studio Code – Used as the primary code editor, with support for JavaScript, Node, and React.
  - Git & GitHub – For version control and collaboration. Branches, commits, and pull requests help maintain clean code history.
  - Concurrently – Used to run the frontend and backend servers simultaneously during development.
  - Postman – For testing API endpoints and verifying backend logic.

- Platform-Compatibility
  The application is designed to be responsive and works on desktops, tablets, and mobile devices, ensuring a consistent user experience across platforms.

This implementation platform supports a robust architecture, efficient data handling, and a user-friendly interface, making Legacy Garage a reliable system for vintage car showcasing and browsing.

6.2 RESULT

6.2.1. SIGNUP

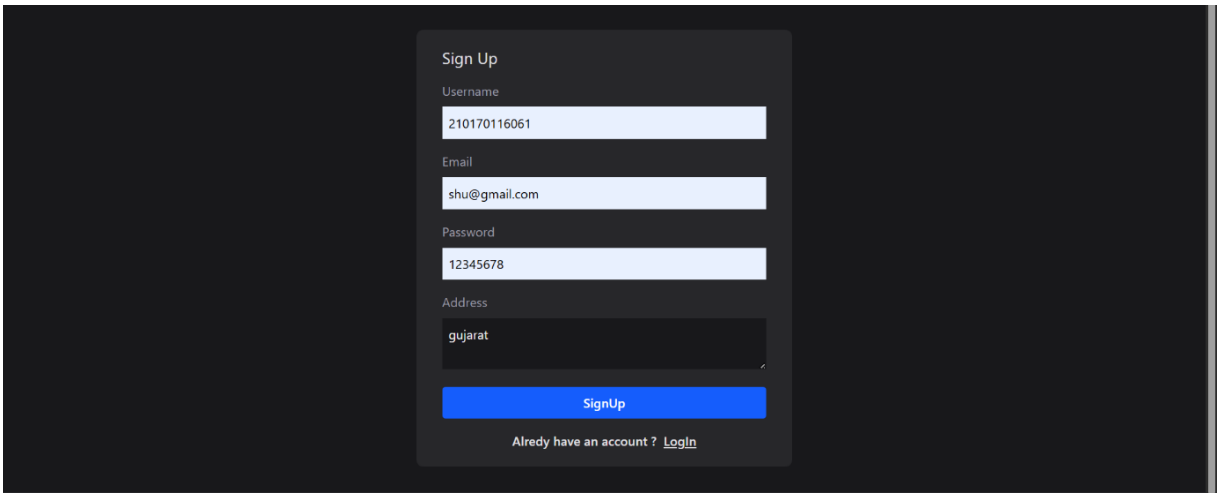

Figure 6.2.1 SignUp

6.2.2 LOGIN

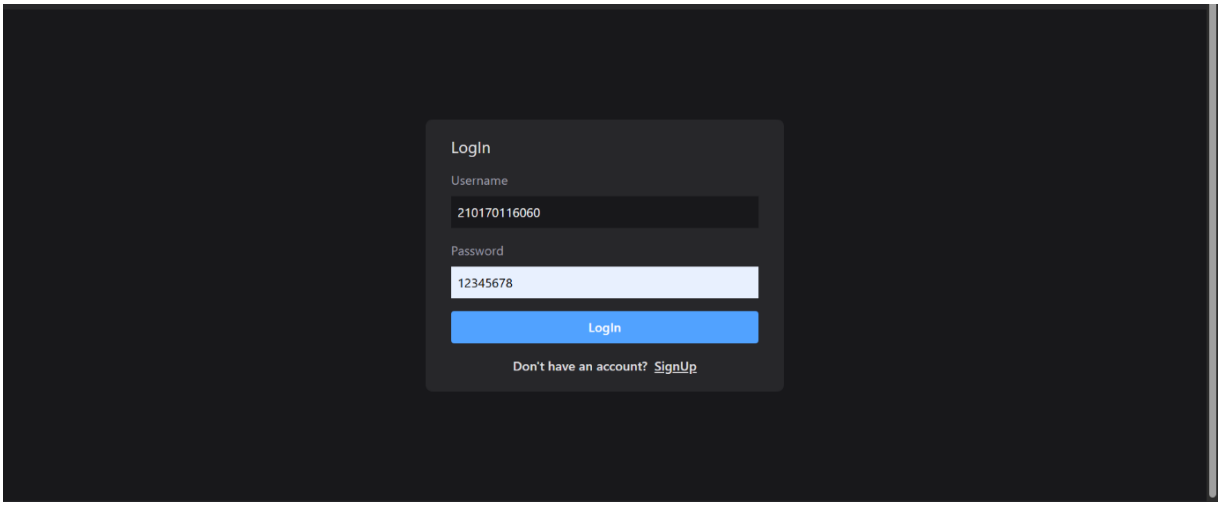

Figure 6.2.2 LogIn

6.2.3 FIRST PAGE(BEFOR SIGNUP)

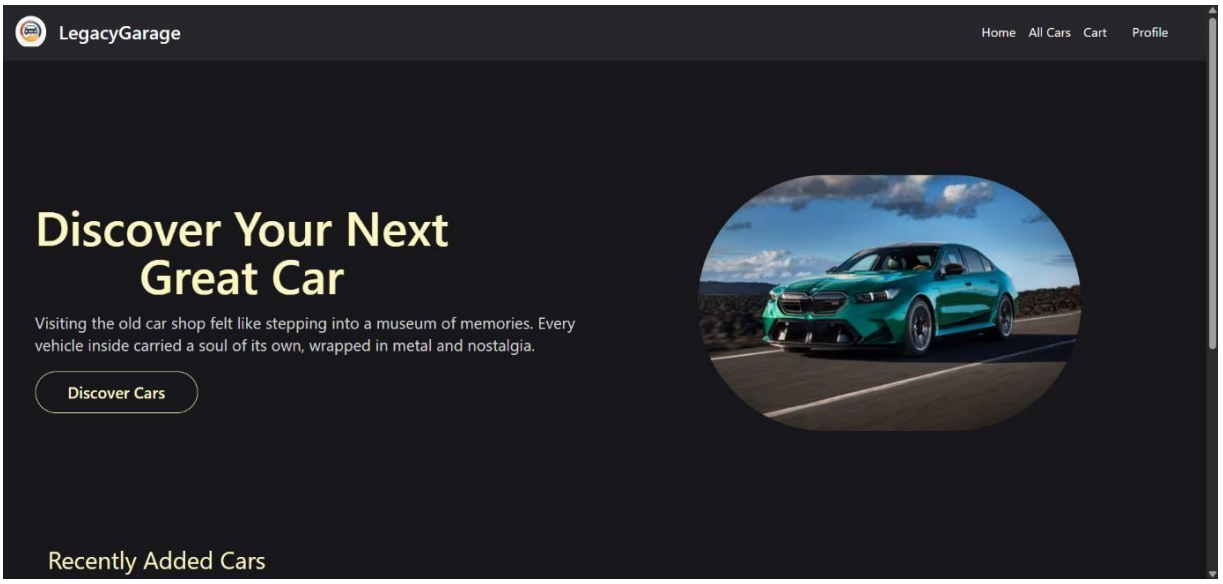

Figure 6.2.3 First Page(BF)

6.2.4 FIRST PAGE(AFTER SIGNUP)

Figure 6.2.4 First Page (AF)

6.2.5 RECENTLY ADD CAR

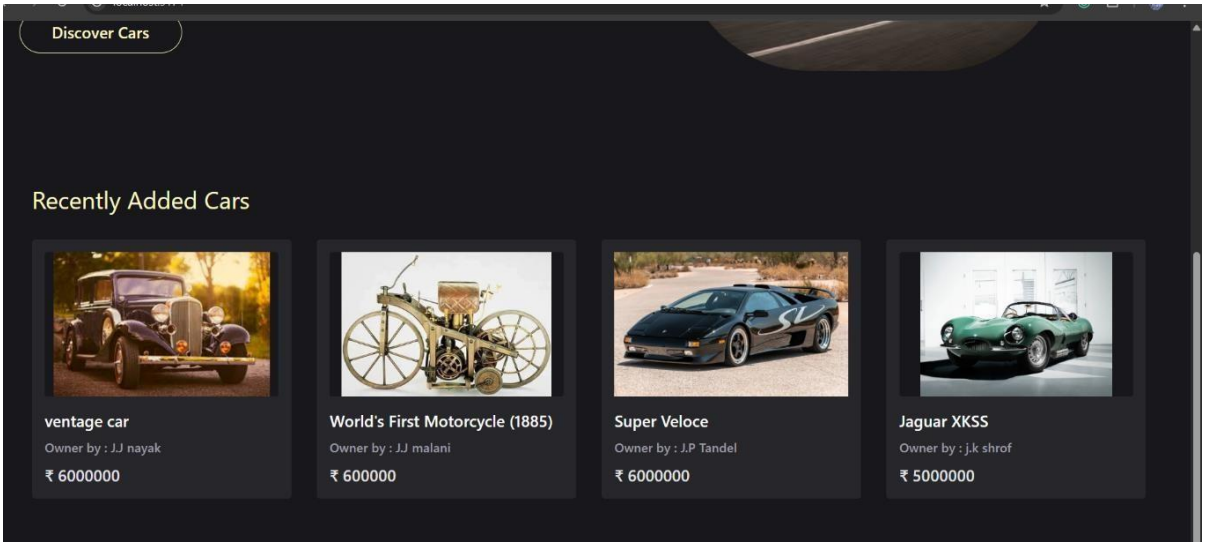

Figure 6.2.5 Recently add car

6.2.6 ALL CARS

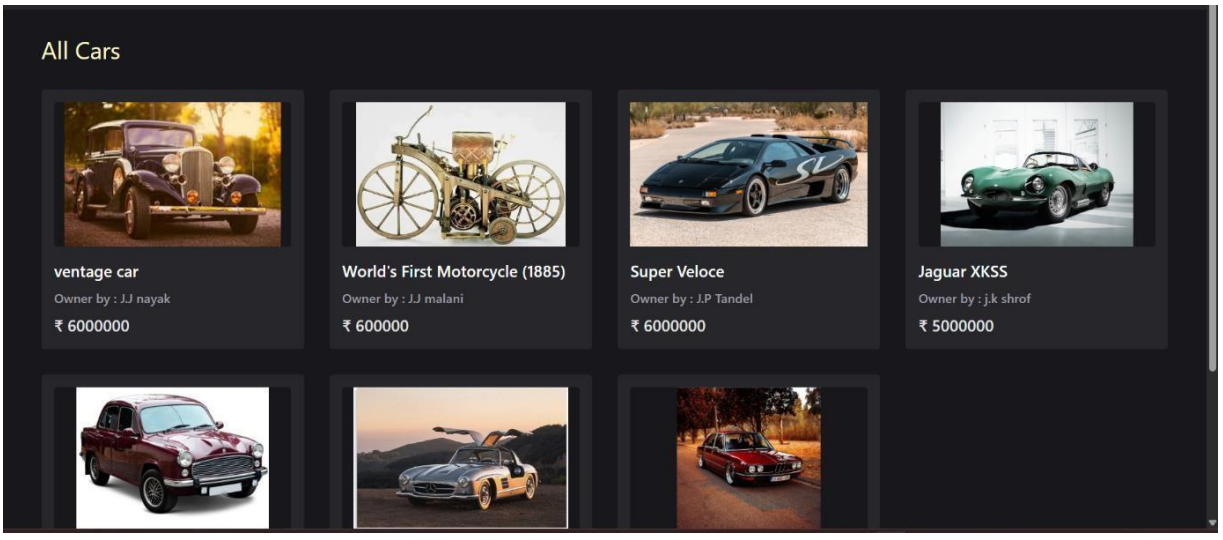

Figure 6.2.6 All Cars

6.2.7 CAR DETAILS

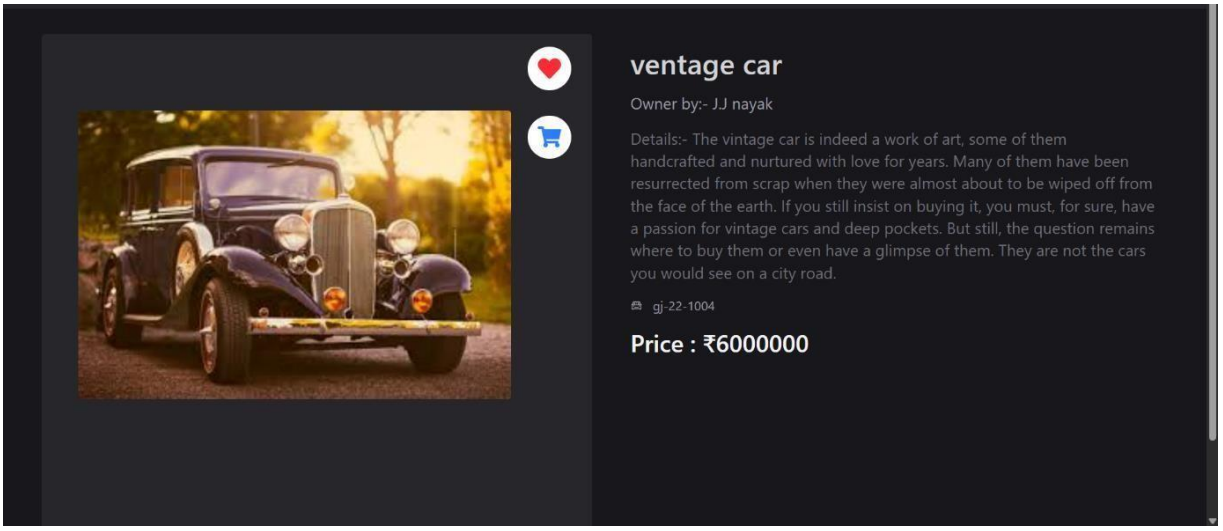

Figure 6.2.7 Cars Details

6.2.8 ADD TO FAVOURITES

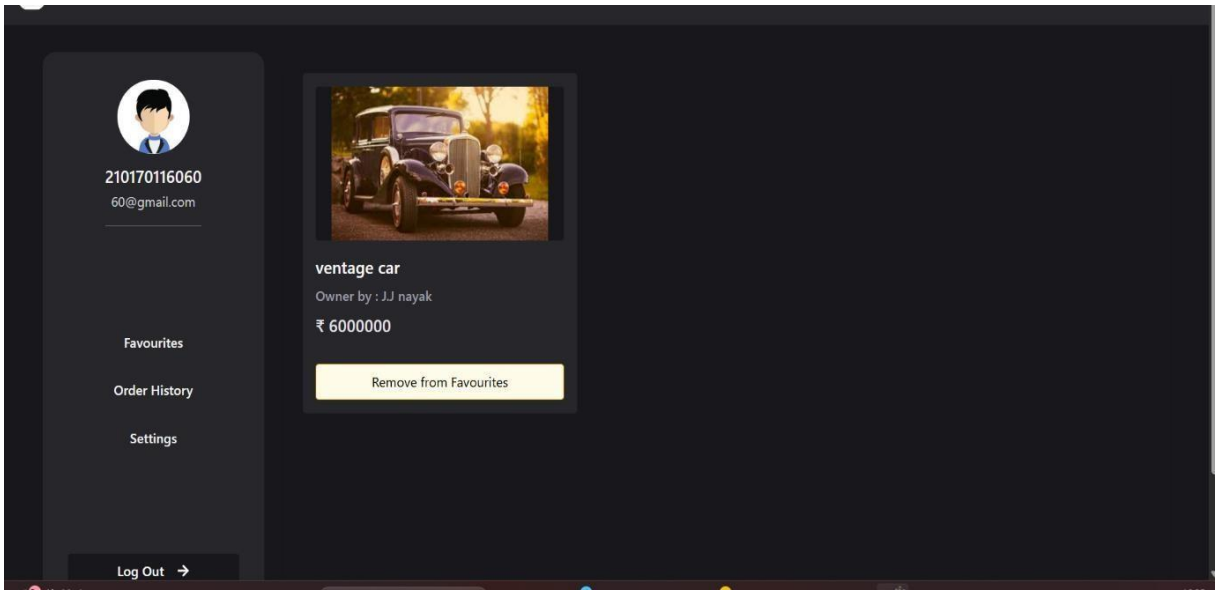

Figure 6.2.8 Add to Favourites

6.2.9 ADD TO CART

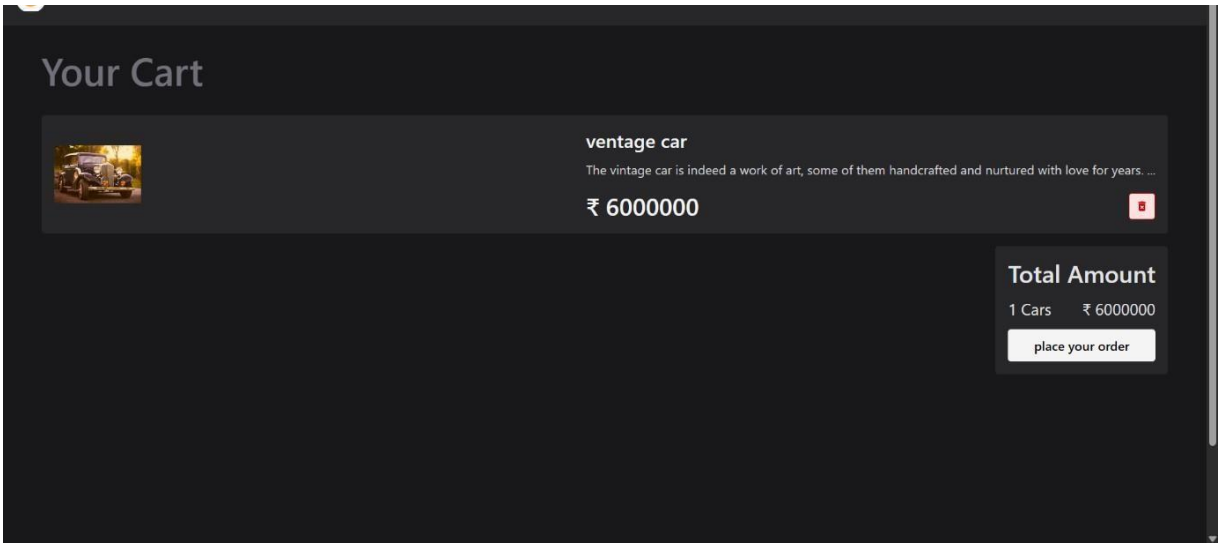

Figure 6.2.9 Add to CART

6.2.10 EMPTY CART

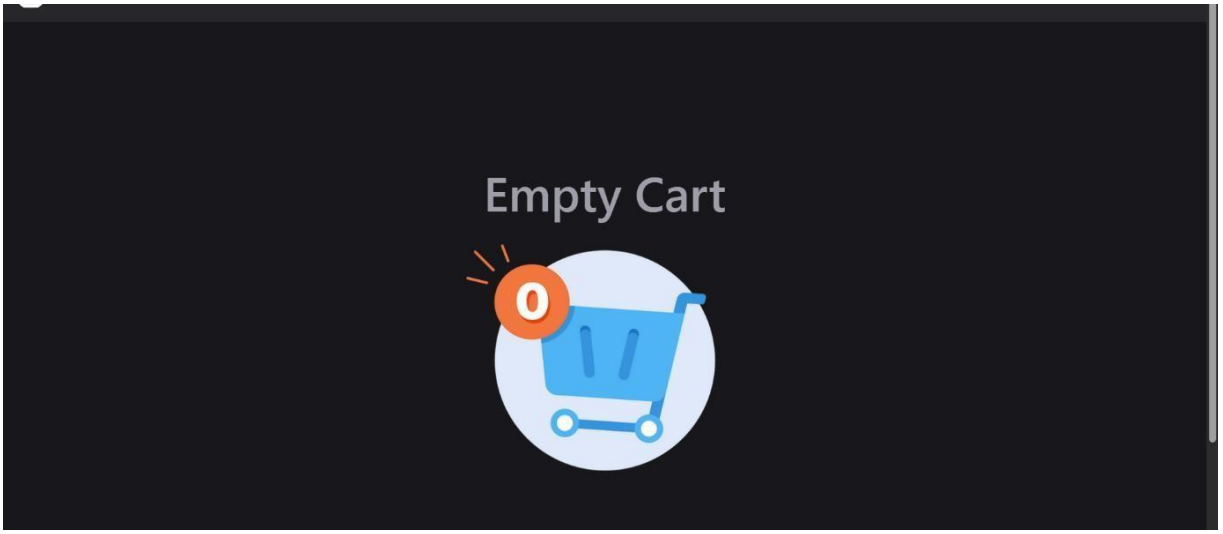

Figure 6.2.10 EMPTY CART

6.2.11 PLACE ORDER HISTORY

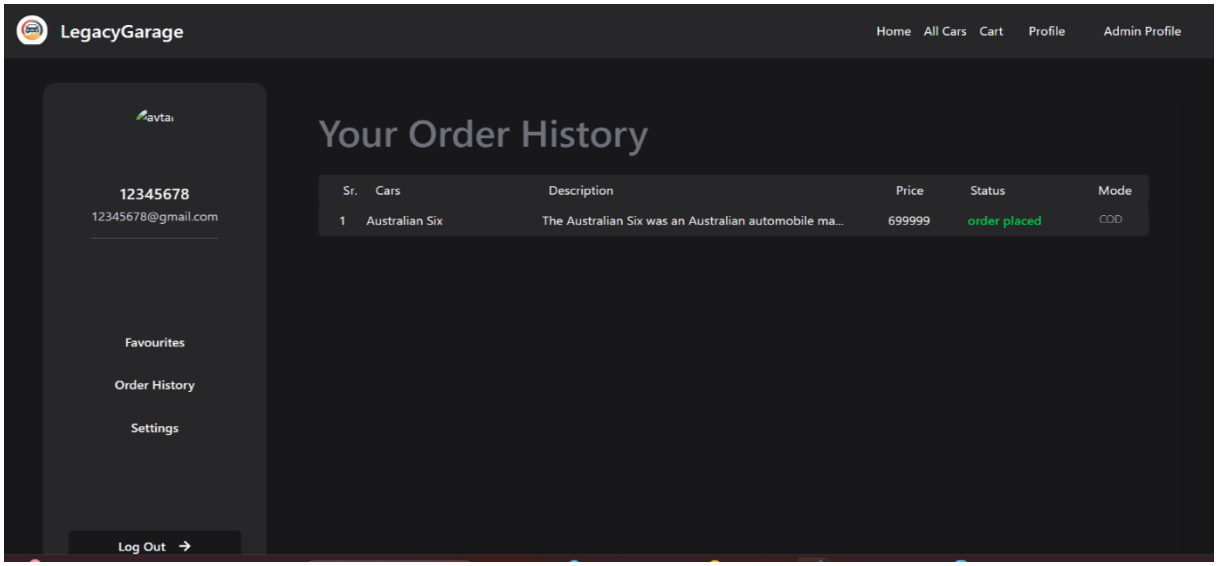

Figure 6.2.11 order placed history

6.2.12 USER LOGOUT

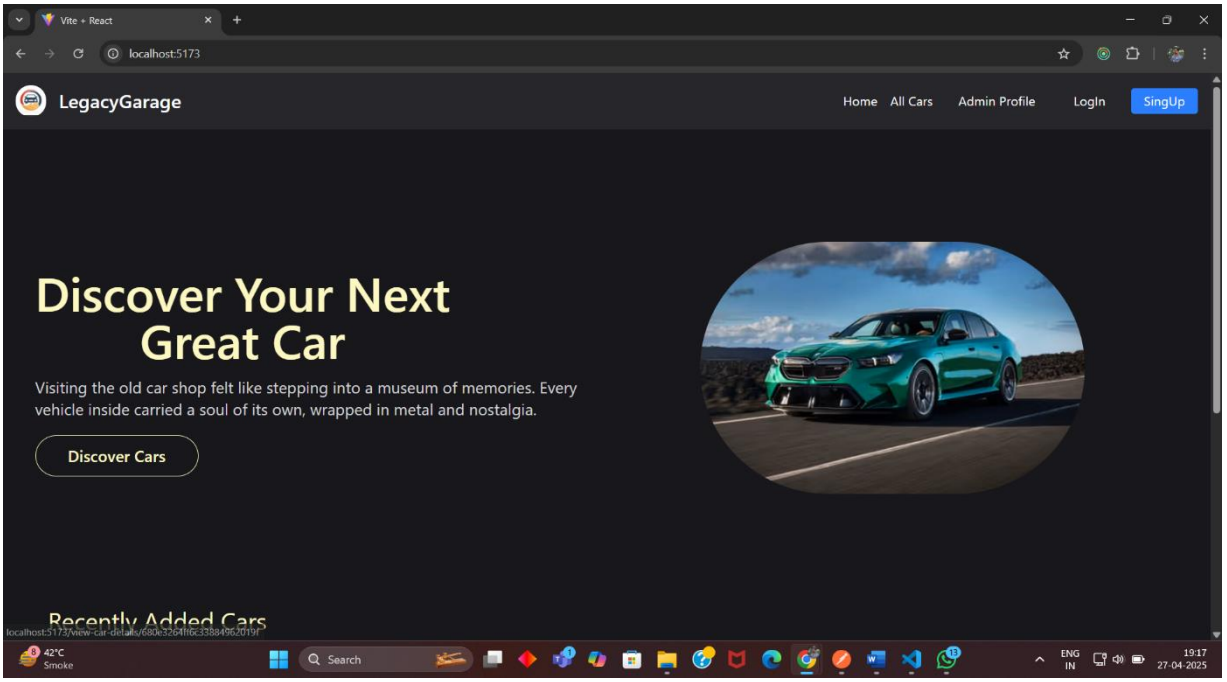

Figure 6.2.12 user Logout

6.2.13 ADMIN LOGIN

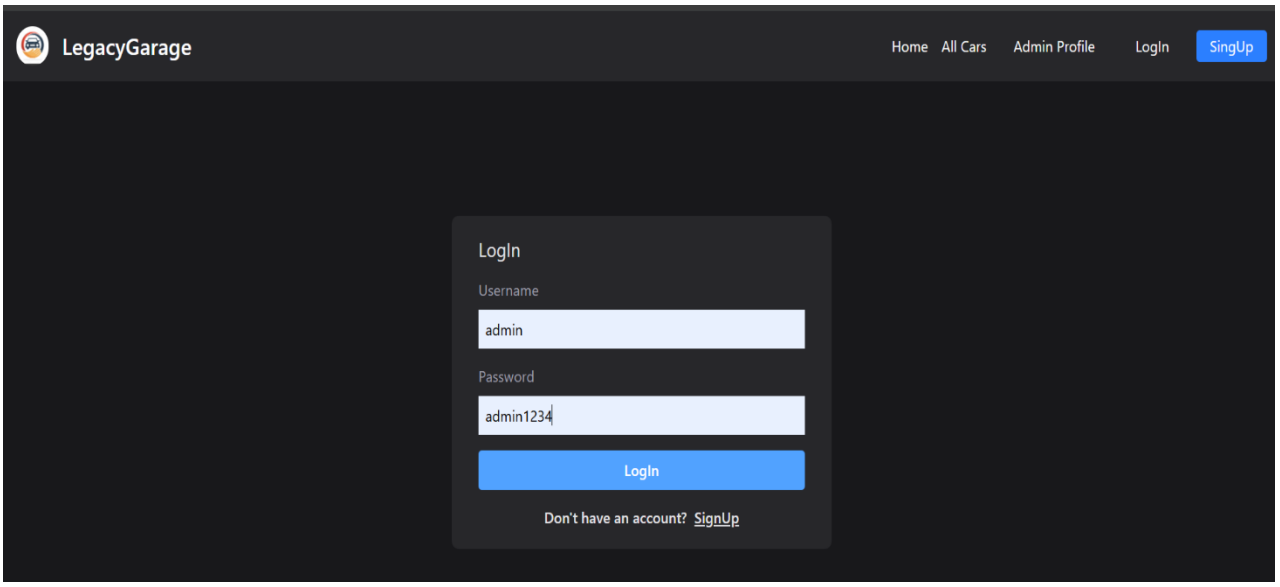

Figure 6.2.13 admin login

6.2.14 FIRST PAGE ORDER HISTORY (ADMIN PAGE)

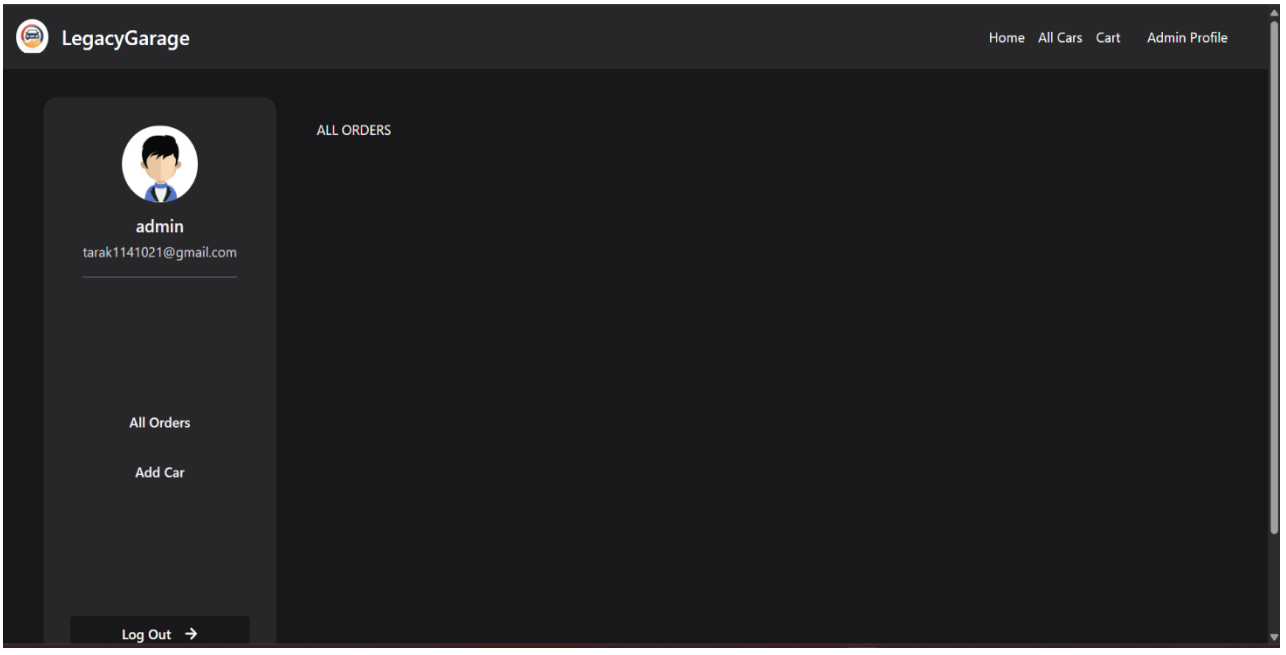

Figure 6.2.14 admin First page

6.2.15 UPDATE and DELETE CARS (ADMIN)

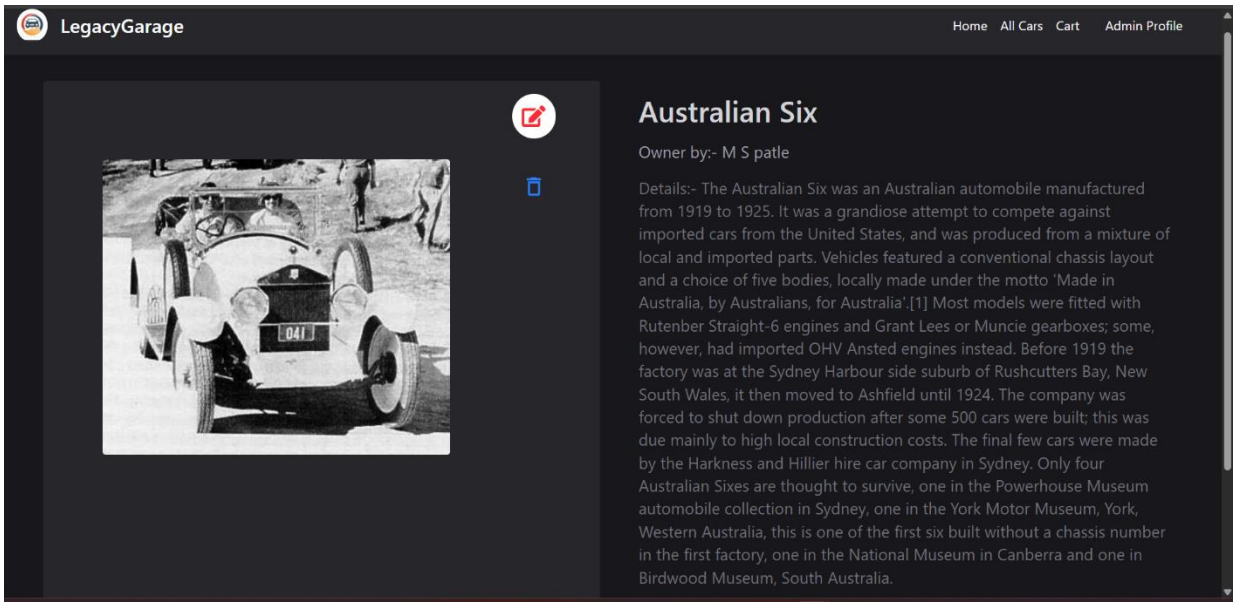

Figure 6.2.15 update and Delete cars

6.2.16 UPDATE CAR (ADMIN)

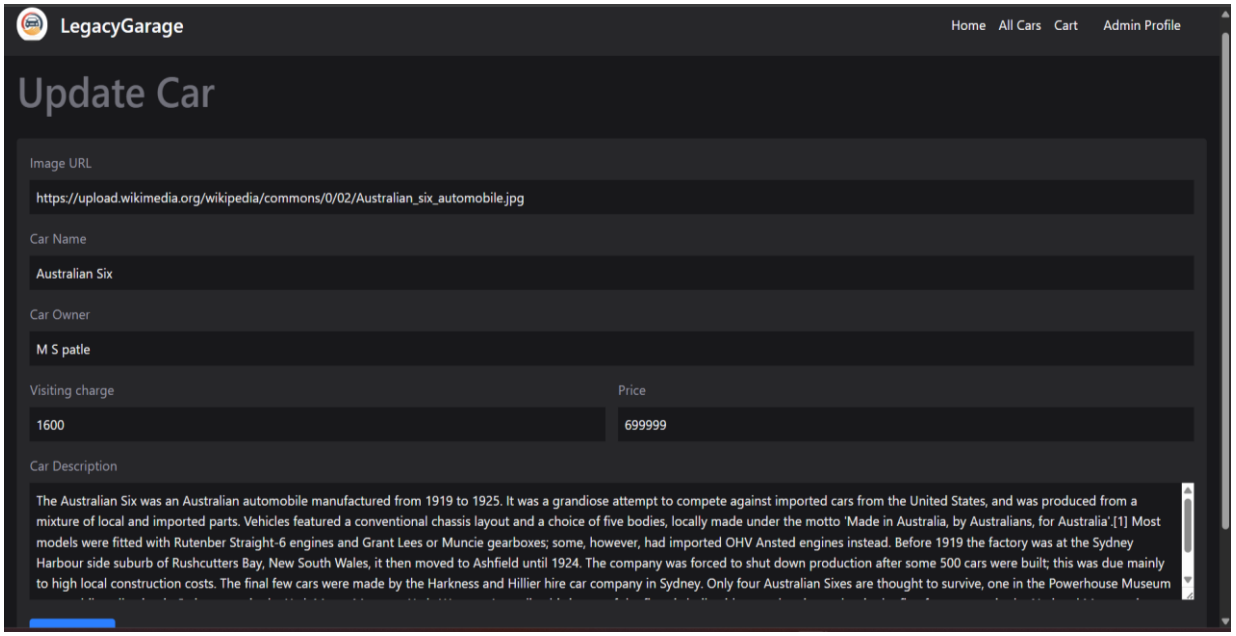

Figure 6.2.16 update cars

6.2.17 ADD CAR (ADMIN)

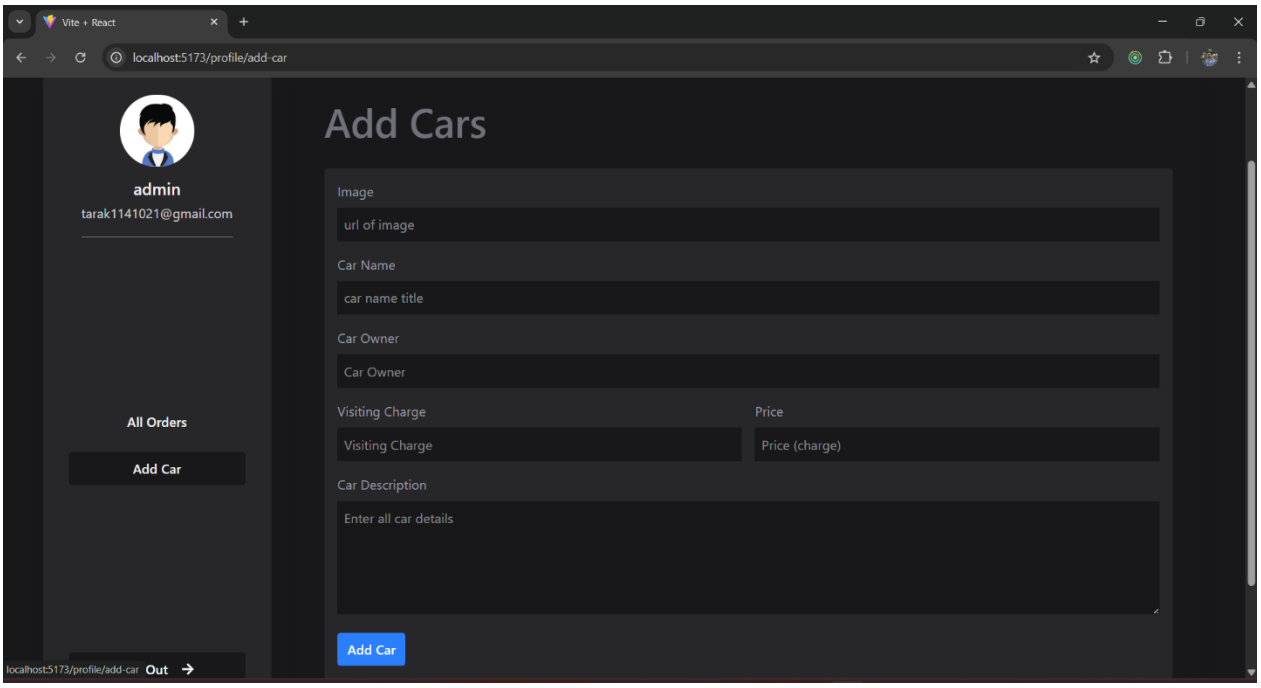

Figure 6.2.16 add cars

6.2.18 ALL CAR (ADMIN)

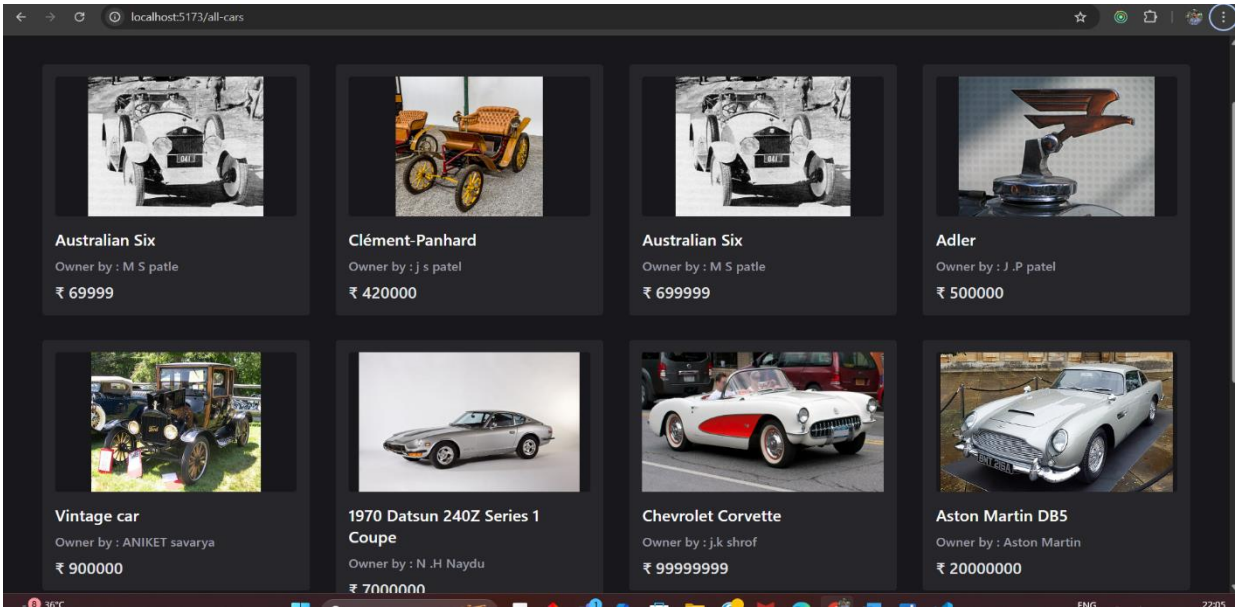

Figure 6.2.16 all cars

6.3 API'S RESULT:-

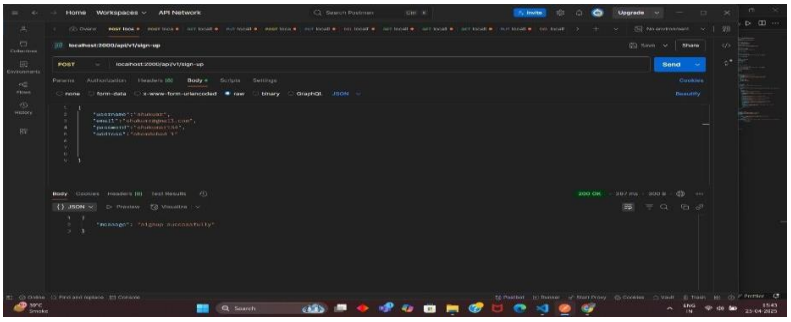

Figure 6.2.14 admin First page

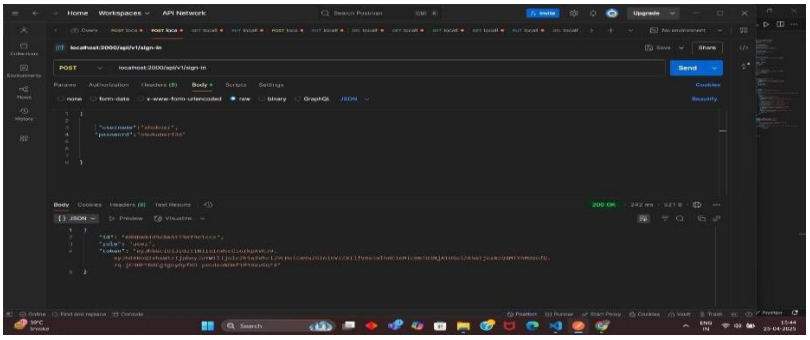

Figure 6.2.14 admin First page

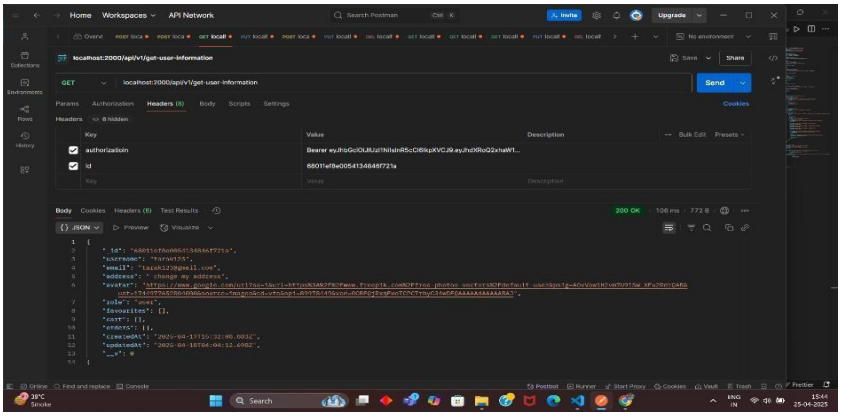

Figure 6.2.14 admin First page

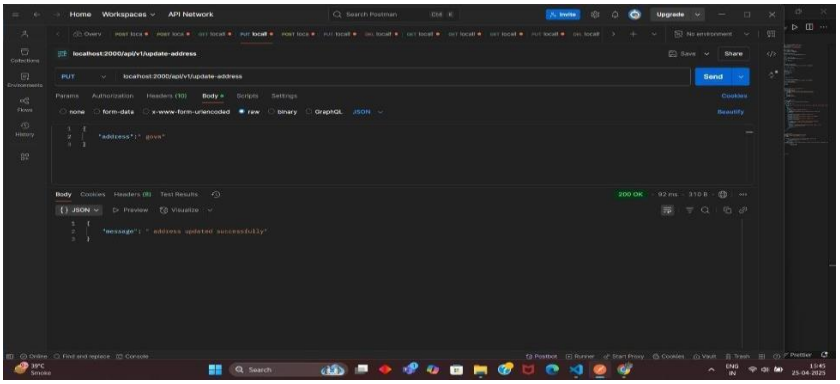

Figure 6.2.14 admin First page

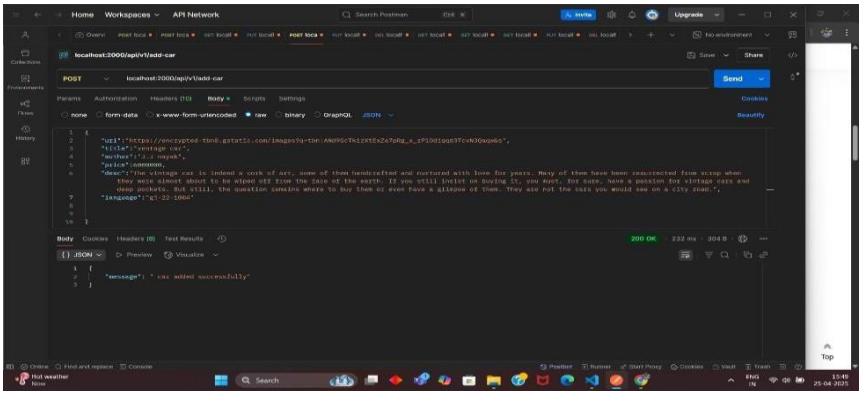

Figure 6.2.14 admin First page

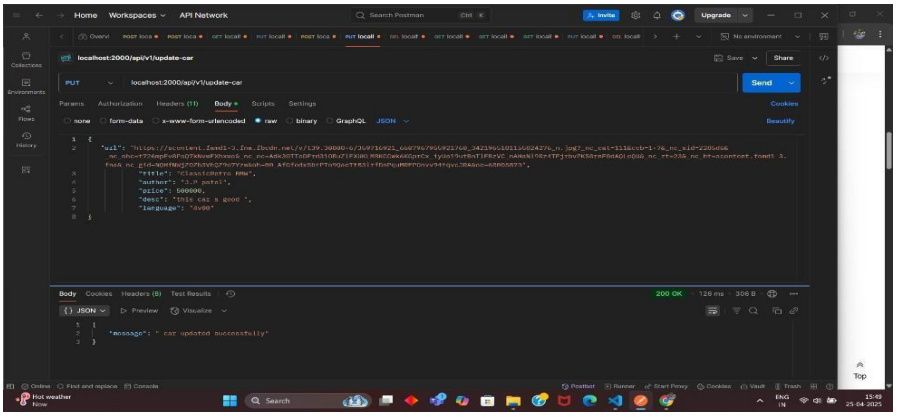

Figure 6.2.14 admin First page

CHAPTER 7: TESTING

7.1 TESTING PLANIG

To ensure the Legacy Garage system functions smoothly and meets all the required standards, a structured testing plan was implemented throughout the development lifecycle. The goal of the testing phase is to identify bugs, verify features, and ensure usability, reliability, and performance. The testing plan includes the following types of testing:

1. Unit Testing

- Each component and function was tested individually.
- Examples include testing form validations, car data handling, and user authentication logic.
- Tools like Jest were used for frontend unit testing in React.

2. Integration Testing

- Ensured that modules like the frontend and backend interact correctly.
- Tested scenarios such as car listing display after admin adds a new car, or order data retrieval by the backend.
- Used Postman and Supertest to simulate API calls and check data flow.

3. Functional Testing

- Verified that the system behaves according to the requirements.
- Examples: Admin can add cars, users can view cars, login and registration processes work correctly.

4. User Interface (UI) Testing

- Checked whether the interface is user-friendly and responsive across devices.
- Ensured buttons, forms, and navigation worked as intended in different screen sizes.

5. Security Testing

- Focused on login and authentication mechanisms.
- Tested protection against unauthorized access, such as users trying to access admin-only features.

7.2 TESTING RESULT

Test Case	Expected Result	Actual Result	Status
Admin Login	Admin is able to log in with correct details	Admin successfully logs in	Passed
Admin Adds Car	Car details are saved and listed	Car appears correctly in car listings	Passed
User Registration	New user gets registered	User data saved and redirected to login	Passed
User Login	User is able to log in	User logs in and can view car listings	Passed
Car List View (User Side)	All cars should be visible to the user	Cars displayed with correct details	Passed
Unauthorized Access (Admin Panel)	Non-admins blocked from admin actions	Access denied with message	Passed
Input Validation (Forms)	Errors shown for empty/invalid inputs	Validations triggered as expected	Passed
Page Responsiveness (Mobile View)	UI adapts to mobile screens	Layout adjusts and remains usable	Passed
API Data Fetching	Correct data received from backend	Data loads with no error	Passed
Error Handling (Invalid Login)	Error message displayed	Correct error shown for wrong credentials	Passed

TABLE 7.1

CHAPTER 8 : LIMITATION AND FUTURE ENHANCEMENTS

8.1 LIMITATION

Despite the overall success of the Legacy Garage system, there are certain limitations that were identified during development and testing:

- No User-to-User Selling: Only the admin can add car listings. Regular users cannot post or sell their own cars.
- Basic Search and Filter: The current search and filtering system is limited and does not include advanced filters like car brand, year, or mileage.
- No Payment Integration: The system does not support online payment or transaction features for car purchases.
- Limited Admin Dashboard: Admin has basic functionalities, but detailed analytics or reporting tools are not yet implemented.
- Authentication Simplicity: Security features like OTP, two-factor authentication, or captcha are not integrated.

8.2 FUTURE ENHANCEMENTS

To improve the functionality and user experience of the Legacy Garage platform, the following enhancements are planned:

- User Car Listing Feature: Enable verified users to upload and manage their own car listings.
- Advanced Filter & Search Options: Add filtering based on car brand, year, price range, condition, and location.
- Payment Gateway Integration: Include secure payment options to support transactions between buyers and sellers.
- Enhanced Admin Panel: Add analytics, sales charts, and user activity tracking to help admin manage the platform more efficiently.
- Mobile App Version: Develop a mobile-friendly version or native mobile app to increase accessibility.
- Notification System: Implement email or SMS alerts for updates, new car listings, or important user actions.
- Chat Feature: Add real-time chat between users and admin for inquiries and negotiation.

CHAPTER 9: CONCLUSIONS AND DISCUSSION

The Legacy Garage project has successfully met its primary objective of providing a digital platform to showcase and manage vintage and second-hand cars through a user-friendly web interface. The system was designed using the MERN stack (MongoDB, Express.js, React.js, Node.js), enabling a smooth and efficient full-stack development experience. Admin users can securely log in, add car listings, and manage the system, while regular users can view available cars and explore detailed car information.

During development, key web development concepts were practiced, including component-based frontend design, RESTful API integration, state management, form handling, and secure user authentication. The project also provided practical experience in version control using Git and collaborative development through GitHub.

Through testing and feedback, several limitations were identified, such as the absence of user-uploaded car listings and payment processing. However, these have been documented and are planned as part of future enhancements.In conclusion, Legacy Garage serves as a strong foundation for a scalable car-selling platform, and with additional features and refinements, it has the potential to evolve into a comprehensive online marketplace for classic and used cars.

CHAPTER 10: REFERENCES

- React.js – https://reactjs.org/docs/getting-started.html
- Node.js – https://nodejs.org/en/docs/
- Express.js – https://expressjs.com/en/starter/guide.html
- MongoDB – https://www.mongodb.com/docs/
- JWT (JSON Web Token) – https://jwt.io/introduction
- Git – https://git-scm.com/doc
- YouTube - https://youtu.be/nRWBCa3sr3k?si=F5uZARyWchr_rhZS