AUTOMATE HUB
Commercial Vehicle Sales and Service System

Project Report

Submitted by

KEERTHI U

Reg. No.: AJC22MCA-2055

In Partial Fulfillment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS

(MCA TWO YEAR)

(Accredited by NBA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2022-2024

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**AUTOMATE HUB"** is the bonafide work of **KEERTHI U(Regno: AJC22MCA-2055)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2023-24.

Ms. Rini Kurian	**Ms. Meera Rose Mathew**
Internal Guide	**Coordinator**
Rev. Fr. Dr. Rubin Thottupurathu Jose	
Head of the Department	**External Examiner**

// CERTIFICATE ON PLAGIARISM CHECK

DECLARATION

I hereby declare that the project report **“AUTOMATE HUB”** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2023-2024.

Date:	**KEERTHI U**
KANJIRAPPALLY	**Reg: AJC22MCA-2055**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Director(Administration) **Rev. Fr. Dr. Roy Abraham Pazhayaparampil** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Rini Kurian** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

KEERTHI U

ABSTRACT

The proposed project aims to revolutionize the commercial vehicle industry by offering a comprehensive solution that addresses the procurement, maintenance, and protection needs of commercial vehicle owners and operators. By consolidating various services such as new vehicle sales, maintenance, spare parts delivery, service booking, vehicle booking, insurance provisions, and insurance renewals into a single, user-friendly platform, the project seeks to streamline operations and enhance efficiency for customers. Through the deployment of an innovative online platform, the project endeavors to redefine industry standards, providing users with a seamless experience for navigating, selecting, and procuring spare parts. Key features of the platform include real-time inventory tracking, predictive maintenance scheduling, personalized insurance options, flexible service booking, and convenient vehicle procurement options. By fostering collaboration and partnerships within the industry, the platform aims to offer unparalleled choice and flexibility to its users. Leveraging data-driven insights and cutting-edge technologies, the platform will continuously refine its offerings to better meet the evolving needs of commercial vehicle owners and operators. Overall, the envisioned website holds the potential to significantly transform the automotive sector by offering unparalleled convenience and accessibility in commercial vehicle management.

.

CONTENT

List of Abbreviations

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language.

CSS - Cascading Style Sheet

SQLite - Structured Query Language

UML - Unified Modeling Language

AJAX - Asynchronous JavaScript and XML

JS - Java Script

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The project aims to provide a comprehensive solution for the commercial vehicle industry, addressing various aspects such as procurement, maintenance, and protection needs of commercial vehicle owners and operators. It seeks to streamline operations and enhance efficiency for customers by consolidating services like new vehicle sales, parts delivery, maintenance, and insurance provisions into a single, user-friendly platform. One of the primary focuses of the project is to optimize parts delivery services, which is crucial for reducing vehicle downtime and maintenance costs while simultaneously improving overall productivity and profitability for businesses. By leveraging innovative online platforms, the project aims to redefine industry standards, offering users a seamless experience for navigating, selecting, and procuring spare parts.Overall, the envisioned website holds the potential to significantly transform the automotive sector by providing unparalleled convenience and accessibility in commercial vehicle management.

1.2 PROJECT SPECIFICATION

The project specification delineates a multifaceted approach to revolutionizing the commercial vehicle industry. With a clear focus on addressing the diverse needs of commercial vehicle owners and operators, the project endeavors to provide a holistic solution encompassing procurement, maintenance, and protection aspects. Central to this endeavor is the consolidation of various services onto a unified platform, facilitating seamless access and operation for customers. A paramount objective of the project is the optimization of parts delivery services. By streamlining the process and enhancing efficiency, the project aims to mitigate vehicle downtime and maintenance costs while bolstering overall productivity and profitability for businesses. This optimization is envisaged to be supported by a user-friendly interface, ensuring effortless navigation and procurement of spare parts for customers.

Admin Module Functionalities:

The admin module acts as the overseer, responsible for monitoring and managing user profiles, service branch information, and staff details. This includes the ability to add new branches and new vehicle, manage associated staff members.

User Module Functionalities:

Users, on the other hand, engage with the platform by registering, browsing spare parts and services, managing their profiles, and facilitating transactions for spare parts purchase and vehicle service bookings, vehicle new bookings, booking history view

Service Branch Module Functionalities:

Service branches are equipped with functionalities to service bookings, vehicle booking view, vehicle listing providing a dashboard to view all appointments and the ability to confirm or reject bookings based on various criteria like availability or service specifics.

Insurance Module Functionalities:

The New Insurance function simplifies policy initiation, letting users provide personal details and customize coverage. Similarly, Insurance Renewal streamlines extending policies, notifying users of deadlines and allowing updates for comprehensive protection. Administrators expand offerings with Add Insurance Package, tailoring packages to coverage types, terms, and conditions. Update Package Details empowers administrators to refine existing packages, ensuring accuracy and relevance for users. These features enhance insurance management, prioritizing convenience and coverage.

Delivery:

The spare parts delivery module simplifies and optimizes the delivery process efficiently. Administrators can input and organize crucial information such as part details, quantity, destination, and recipient data using the Manage Spare Parts Delivery Details feature, ensuring accessibility and thorough management. With the Update Delivery Status functionality, users can track deliveries in real-time, fostering transparency and keeping users informed from dispatch to delivery. Additionally, seamless coordination between the delivery team and users is facilitated through proactive communication channels, including notifications and direct interaction, ensuring timely delivery and enhancing overall customer satisfaction.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

The proposed project aims to revolutionize the commercial vehicle industry by offering a comprehensive solution that addresses the procurement, maintenance, and protection needs of commercial vehicle owners and operators. By consolidating various services such as new vehicle sales, maintenance, and insurance provisions into a single, user-friendly platform, the project seeks to streamline operations and enhance efficiency for customers. Through the deployment of an innovative online platform, the project endeavors to redefine industry standards, providing users with a seamless experience for navigating, selecting, and procuring spare parts. In conclusion, the envisioned website holds the potential to significantly transform the automotive sector by offering unparalleled convenience and accessibility in commercial vehicle management.

2.2 EXISTING SYSTEM

The current system for commercial vehicle spare parts procurement and service booking relies heavily on physical visits to brick-and-mortar locations of the company. Customers are obliged to personally attend these locations to purchase spare parts and schedule vehicle maintenance. This traditional method poses inconvenience and limited accessibility, thereby hindering customers seeking efficient solutions. The system lacks the convenience of remote access and online transactions, potentially causing time constraints and geographical limitations for customers.

2.2.1 NATURAL SYSTEM STUDIED

The natural system studied encompasses a broader analysis of the environment and factors influencing the automotive industry and its customers. This includes an examination of consumer behavior, market trends, technological advancements, and the increasing shift towards online transactions and remote services. Recognizing customers' growing preference for digital solutions, the analysis considers the need for convenience and the competitive landscape within the automotive industry. It takes into account evolving customer expectations regarding accessibility, ease of use, and efficiency in procuring spare parts and scheduling vehicle services.

2.2.2 DESIGNED SYSTEM STUDIED

The proposed designed system aims to revolutionize the current process by introducing an online platform dedicated to commercial vehicle spare parts procurement and service booking. The system's design prioritizes a user-friendly interface accessible via a website. This platform will offer customers the convenience of remotely browsing, selecting, and purchasing required spare parts hassle-free. Additionally, the platform will facilitate seamless booking of vehicle services, providing users with the ability to schedule maintenance appointments efficiently.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Customers are constrained by the necessity of physically visiting company locations, which can be time-consuming and inconvenient
- The system restricts access to spare parts and services solely through physical outlets, creating barriers for customers who prefer or require remote or online solutions.
- The current method lacks efficiency due to manual processes involved in purchasing spare parts and scheduling vehicle maintenance, leading to potential delays and errors.
- Customers located in distant areas might face difficulties accessing the company's physical locations, causing inconvenience and delays in procuring necessary parts and services.
- The system's rigid structure does not adapt to the changing needs and preferences of modern customers who increasingly seek convenient, on-demand services.
- The inconvenience of the current system might result in decreased customer satisfaction and retention due to the inability to meet their expectations for easy and accessible solutions.
- In an evolving market where competitors offer online solutions and enhanced customer experiences, the existing system's limitations might place the company at a disadvantage

2.4 PROPOSED SYSTEM

The current system for procuring commercial vehicle spare parts and booking service appointments involves customers directly visiting the company's physical location to purchase parts and schedule vehicle maintenance. This process lacks convenience and accessibility for customers seeking an efficient solution.This project aims to revolutionize the automotive industry by introducing an online platform designed to facilitate the seamless purchase of spare parts and the hassle-free booking of vehicle services. The website will provide a user-friendly interface that allows customers to effortlessly browse, select, and purchase the required spare parts with ease.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Enhanced Accessibility
- Convenience for Customers
- Time Efficiency
- Improved Customer Satisfaction
- Adaptability and Modernization
- Competitive Edge
- Geographical Flexibility

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Planning, organizing, and managing resources to ensure the achievement of particular project goals and objectives is the process of project management. A feasibility study is a preliminary examination of a prospective project or end to determine its merits and viability. A feasibility study aims to provide an objective assessment of the technical, economic, financial, legal, and environmental elements of a proposed project. The information can then be used by decisionmakers to decide whether to proceed with the project or not. The findings of the feasibility study can also be used to develop a practical project plan and budget. It cannot be simple to determine whether or not a proposed project is worthwhile pursuing without a feasibility study.
The document provides the feasibility of the project that is being designed and lists. Various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibility. The following are its features:

3.1.1 Economical Feasibility

The proposed project aims to revolutionize the commercial vehicle industry by offering a comprehensive solution that integrates various services into a single platform, thereby reducing operational costs associated with multiple systems and processes. By streamlining procurement, maintenance, spare parts delivery, service booking, vehicle booking, insurance provisions, and insurance renewals, the project seeks to optimize resource allocation and improve cost-efficiency for commercial vehicle owners and operators. Additionally, by fostering collaboration and partnerships within the industry, the platform can leverage economies of scale and negotiate favorable deals with suppliers and service providers, further enhancing its economic viability.

3.1.2 Technical Feasibility

The project demonstrates feasibility through its deployment of an innovative online platform that integrates multiple functionalities seamlessly. Leveraging cutting-edge technologies such as real-time inventory tracking, predictive maintenance scheduling, and data-driven insights, the platform demonstrates its technical capabilities to meet the complex needs of commercial vehicle management. Additionally, the use of scalable architecture ensures that the platform can accommodate growth and handle increasing user traffic without compromising performance or reliability.

3.1.3 Behavioral Feasibility

Behavioral feasibility is evident in the project's focus on providing a user-friendly experience for commercial vehicle owners and operators. By consolidating various services into a single platform and offering features such as personalized insurance options, flexible service booking, and convenient vehicle procurement, the project aims to simplify and streamline tasks, thereby encouraging user adoption and satisfaction. Moreover, continuous refinement based on user feedback and data-driven insights ensures that the platform evolves to better meet the changing needs and preferences of its users, fostering long-term engagement and loyalty within the commercial vehicle industry.

3.1.4 Feasibility Study Questionnaire

1. Project Overview: The project aims to introduce an online platform revolutionizing commercial vehicle spare parts procurement and service booking. This web-based system targets enhancing accessibility and convenience by allowing users to purchase spare parts and schedule vehicle services effortlessly.

2. To what extent the system is proposed for?

The system is proposed to cater to the entire process of commercial vehicle spare parts procurement and service booking. It aims to facilitate both purchasing spare parts and scheduling vehicle maintenance services.

3. Specify the Viewers/Public involved in the System?

The viewers/public involved in the system include customers/users seeking to procure spare parts and schedule vehicle services, service branch staff managing service bookings, parts managers handling inventory, and the system administrators overseeing the entire platform.

4. List the Modules included in your System?

Admin Module
User Module
Service Branch Module
Insurance Module
Delivery Module

5. Identify the users in your project?

Users encompass customers seeking spare parts and services, service branch staff managing bookings, parts managers overseeing inventory, and administrators responsible for system supervision.

6. Who owns the system?

The ownership of the system lies with the entity responsible for its development and maintenance. This could be the automotive company or the organization spearheading the project.

System is related to which firm/industry/organization?

The system is related to the automotive industry, particularly targeting commercial vehicle spare parts procurement and service booking. It might be associated with an automotive company, dealership, or service provider within this industry.

7. Details of person that you have contacted for data collection?

Kirankumar O (KUN Capital Ashok Leyland) Service Manager

8. How satisfied are you with the existing process of procuring commercial vehicle spare parts and booking service appointments?

Highly Dissatisfied

9. Would you prefer an online platform for purchasing spare parts and booking service appointments over the current in-person method?

Yes

10. How important is it for you to have an easy-to-use interface while browsing for spare parts and scheduling service appointments?

Extremely Important

11. What mode of communication would you prefer for updates regarding your spare parts orders or service appointments?

Email, SMS/Text Message

12. What specific features would you like to see in an online platform for spare parts procurement and service booking?

User-friendly interface, Detailed part descriptions, Service appointment scheduling, Order tracking, Secure payment options

13. How likely are you to use an online platform for purchasing spare parts and scheduling service appointments if it offers convenience and reliability?

An online platform provides accessibility from anywhere at any time. It eliminates the need to physically visit a location, saving time and effort. With a user-friendly interface, browsing and purchasing become seamless, catering to the user's schedule and preferences.

14. How frequently do you require spare parts for your commercial vehicles and schedule service appointments?

We typically require spare parts for our commercial vehicles on a monthly basis, and service appointments are scheduled quarterly for routine maintenance checks.

15. What challenges have you encountered in the past when procuring spare parts or booking service appointments for your commercial vehicles?

Some challenges we've faced include delays in receiving spare parts orders, difficulty in finding specific parts, and long waiting times to secure service appointments, especially during peak seasons.

16. What factors do you consider most important when choosing a supplier or service provider for spare parts and vehicle maintenance?

The factors we prioritize include reliability of the supplier or service provider, quality of parts or service offered, pricing competitiveness, and promptness in fulfilling orders or appointments.

17. How do you currently track the status of your spare parts orders or service appointments?

We primarily rely on phone calls or emails to follow up with suppliers or service providers to track the status of our orders or appointments.

18. Would you be willing to provide feedback or participate in user testing during the development of the online platform?

Yes, we would be more than willing to provide feedback and participate in user testing to ensure the online platform meets our needs and requirements.

19. How do you typically make payment for spare parts purchases or service appointments?

We usually make payments for spare parts purchases via invoice, while service appointments are paid for using a combination of cash and credit card.

20. What concerns, if any, do you have about transitioning from the current method of procuring spare parts and booking service appointments to an online platform?

Our main concern would be ensuring the security of our transactions and data when using the online platform. Additionally, we would need assurance that the platform is user-friendly and intuitive to use.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	-	intel core i3
RAM	-	4 G B
Hard disk	-	5 1 2 G B

3.2.2 Software Specification

Front End	-	HTML, CSS
Backend	-	PYTHON DJANGO
Client on PC	-	Windows 7 and above.

Database	-	SQLite
Technologies used	-	JS, HTML5, AJAX, J Query, Python Django, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PYTHON

Python is a high-level, versatile programming language renowned for its simplicity, readability, and versatility. Created by Guido van Rossum in the late 1980s, Python emphasizes code readability and ease of use, fostering a clean and concise syntax that facilitates rapid development.Known for its extensive standard library, Python offers a rich set of modules and packages, empowering developers to accomplish diverse tasks, from web development and data analysis to artificial intelligence and scientific computing.

Python's interpreted nature allows for interactive and dynamic coding, making it an ideal choice for beginners and seasoned developers alike. Its object-oriented approach promotes code reusability and modularity. Python's community-driven ethos has led to a vast ecosystem of third- party libraries and frameworks, such as Django, Flask, NumPy, and TensorFlow, contributing to its widespread adoption across industries. Its cross-platform compatibility and support for multiple programming paradigms, including procedural, functional, and object-oriented programming, underscore Python's status as a go-to language for various applications.

3.3.2 DJANGO

Django, a leading open-source web framework, epitomizes the pinnacle of modern web development with its efficiency and versatility. Built on Python, Django follows the Model-View-Controller architecture, prioritizing simplicity and flexibility. Noteworthy features include its Object-Relational Mapping system, streamlined URL routing, and a user-friendly template engine.The built-in admin interface simplifies data management, while robust security measures and middleware support enhance application integrity. Django scales effortlessly, accommodating growing datasets and traffic demands. Its REST Framework extension further extends its utility to API development. Supported by an active community and extensive documentation, Django stands as a powerful toolkit, seamlessly shaping the development of dynamic and secure web applications for diverse purposes, from content management systems to Restful API. In essence, Django empowers developers with a comprehensive and adaptable framework for crafting sophisticated and salable web solutions.

3.3.3 SQLite

SQLite, a lightweight and server less relational database management system, is celebrated for its simplicity, portability, and efficiency. Operating as a self-contained, single-file database, SQLite requires minimal setup and eliminates the need for a separate server process. This design makes it a preferred choice for embedded systems, mobile applications, and small to medium-scale projects.With zero configuration and a server less architecture, SQLite streamlines the database management process. It supports ACID transactions, ensuring data integrity and reliability, even in the face of system interruptions. The dynamic typing feature allows flexible data storage,accommodating various data types within the same column.

Noteworthy is SQLite's cross-platform compatibility, making it versatile across different operating systems and environments. Its wide adoption is evident in applications ranging from mobile apps to web browsers, where it is utilized for local data storage. SQLite stands out as a go-to solution for projects requiring a lightweight, self-contained, and easily deployable database system,embodying simplicity without compromising on functionality and reliability.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design isa creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. UML stands for Unified Modeling Language. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with objectoriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram

• Collaboration diagram

• Activity diagram

• State chart diagram

• Deployment diagram

• Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML(Unified Modeling Language), a standard notation for the modeling of real-world objects and systems. System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases

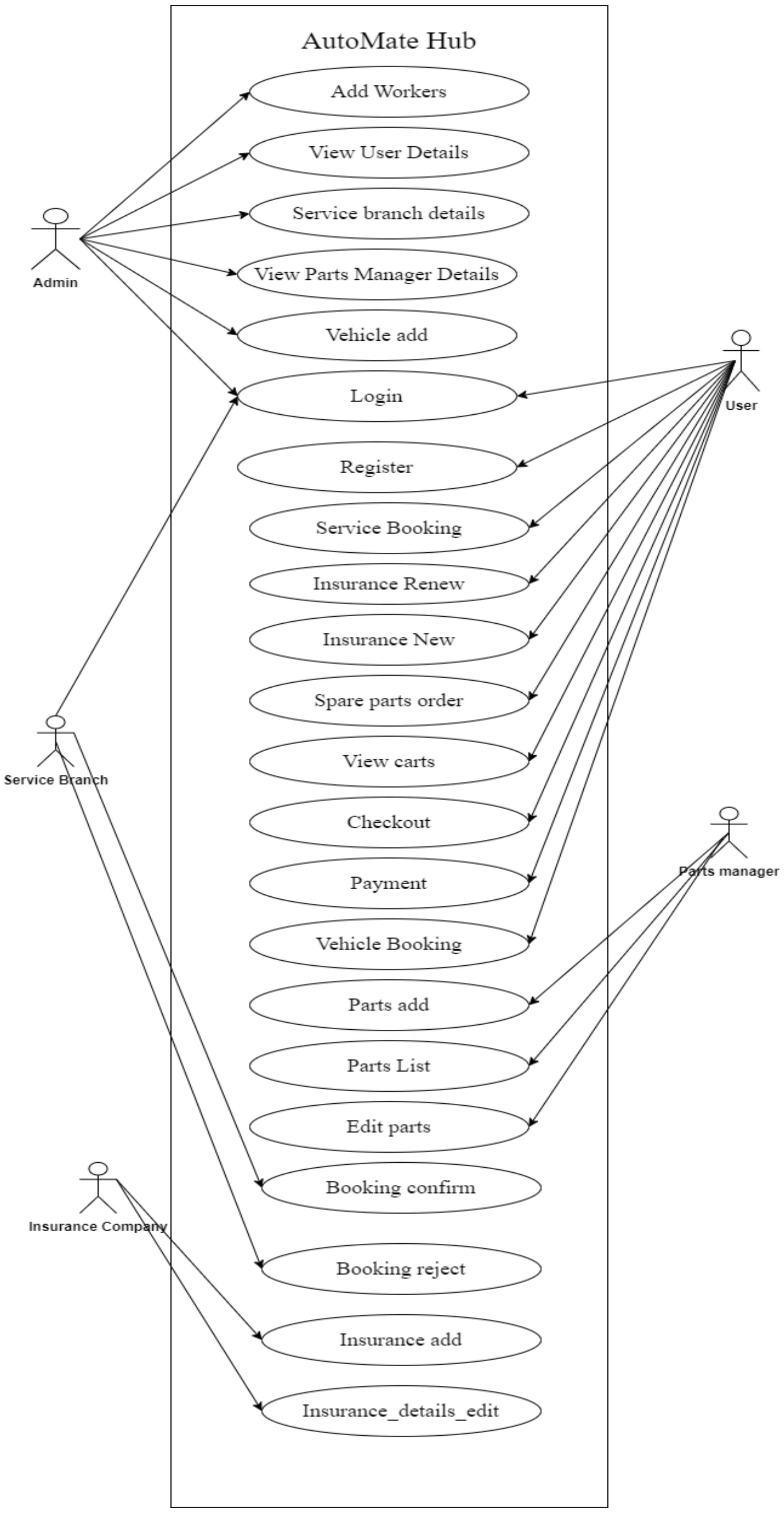


Fig. 4.2.1 Use case diagram for AutoMate Hub

4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.
Messages can be broadly classified into the following categories:

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

iv. **Guards –** To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or particular process

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

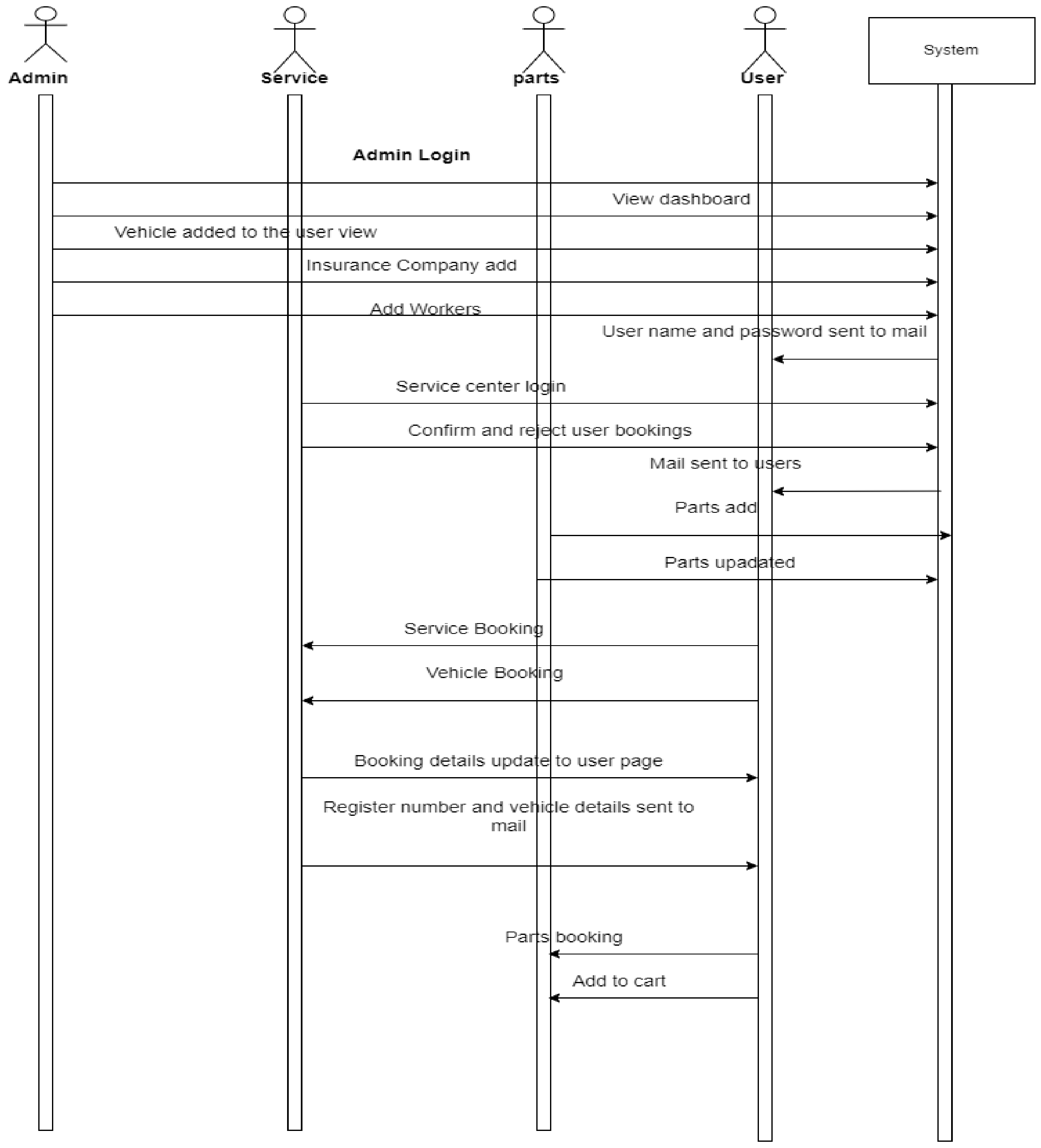


Fig. 4.2.2 sequence diagrams for AutoMate Hub

4.2.3 State Chart Diagram

State Chart Diagram A particular form of diagram used in computer science and related subjects toexplain how systems behave is called a state diagram. State diagrams call for the system being represented to consist of a finite number of states; occasionally, this is the case, and other times, it'sonly an acceptable abstraction. State diagrams come in a variety of shapes and sizes, each with a distinct meaning. State diagrams are there to provide a system's behaviour an abstract explanation.In order to evaluate and demonstrate this behaviour, a sequence of events that could occur in one or more hypothetical states is employed. "Each diagram typically depicts objects of a single class and monitor the different states of its objects across the system," according to this.

- **Initial State -** This state represents the starting point of the system or object and is denotedby a solid black circle.
- **State -** This element describes the current state of the system or object at a specific point intime and is represented by a rectangle with rounded corners.
- **Transition -** This element shows the movement of the system or object from one state toanother and is represented by an arrow.
- **Event and Action -** An event is a trigger that causes a transition to occur, and an action isthe behavior or effect of the transition.
- **Signal -** A message or trigger caused by an event that is sent to a state, causing a transitionto occur.
- **Final State -** The State Chart Diagram ends with a Final State element, which is representedby a solid black circle with a dot inside. It indicates that the behavior of the system or objecthas completed

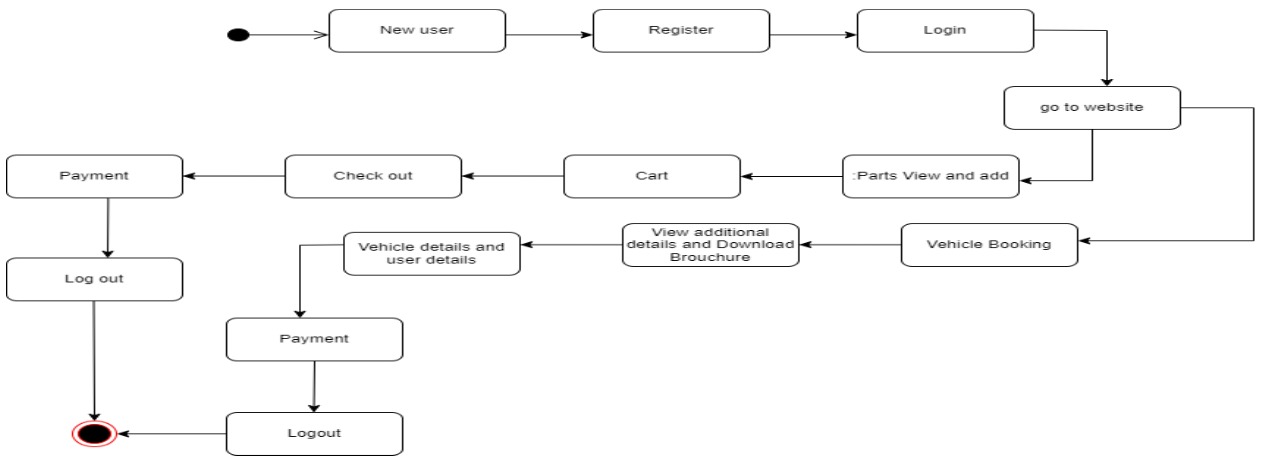


Fig. 4.2.3 State chart diagram for AutoMate Hub

4.2.4 Activity Diagram

Activity diagrams depict how different levels of abstraction of activities are linked to provide a service. Typically, an event should be completed by some activities, particularly when the activity is intended to do multiple separate goals that need coordination. Another typical requirement is howthe events in a single use case interact with one another, particularly in use cases where operationsmay overlap and require coordination. It may also be used to show how a collection of interrelateduse cases interacts to reflect business operations.

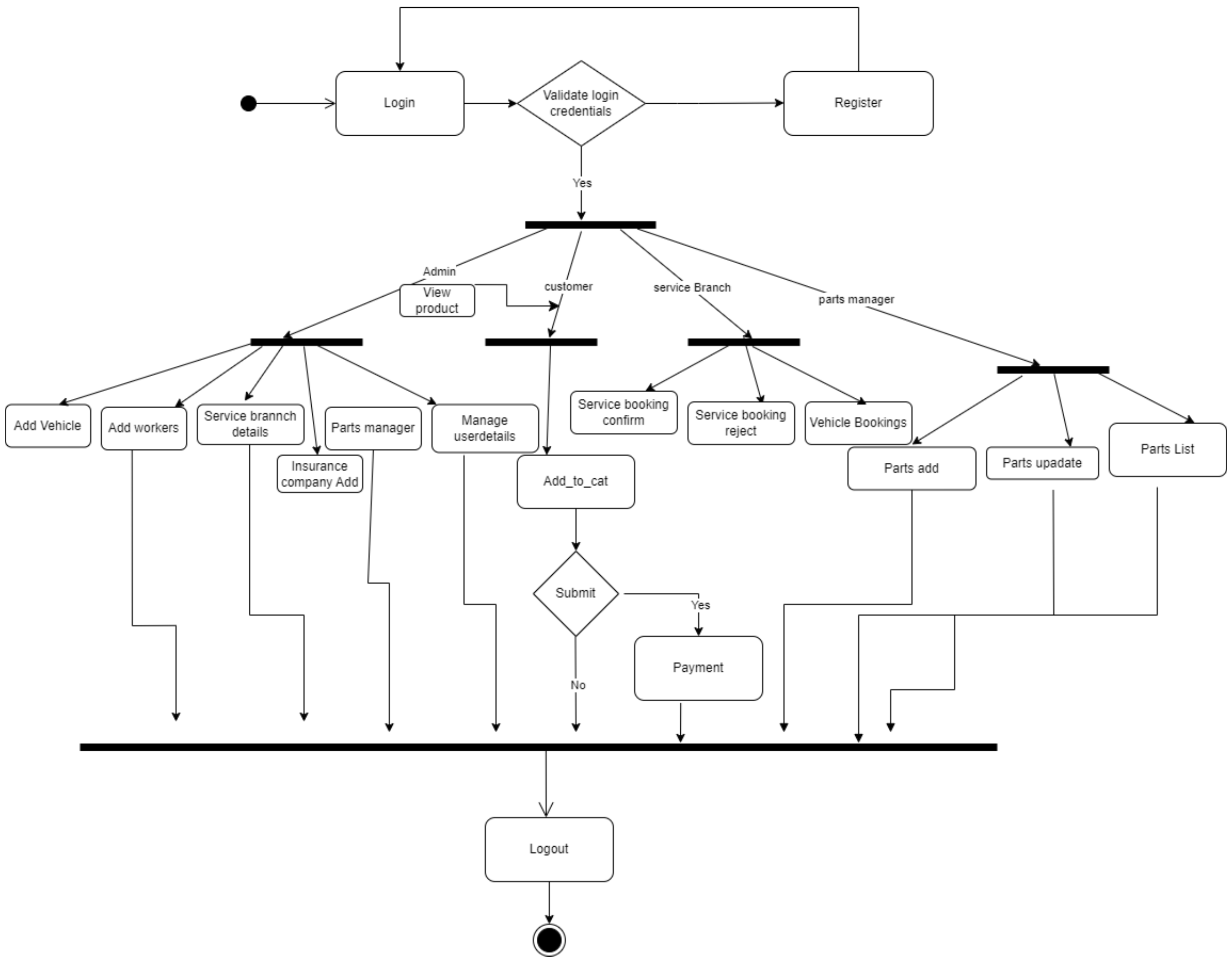


Fig. 4.2.4 Activity diagram for AutoMate Hub

4.2.5 Class Diagram

Class diagram is a static diagram. It represents the static view of the application. Class diagrams are useful for visualising, describing, and documenting various system components as well as for writing executable code for software applications. A class diagram describes the constraints imposed on the system together with the properties and operations of a class. The only UML diagrams that can be directly converted into objectoriented languages are class diagrams, which are extensively utilised in the designing of object-oriented systems. An assortment of classes, interfaces, affiliations, partnerships, and limitations are displayed in a class diagram. It also goes by the name "structural diagram."

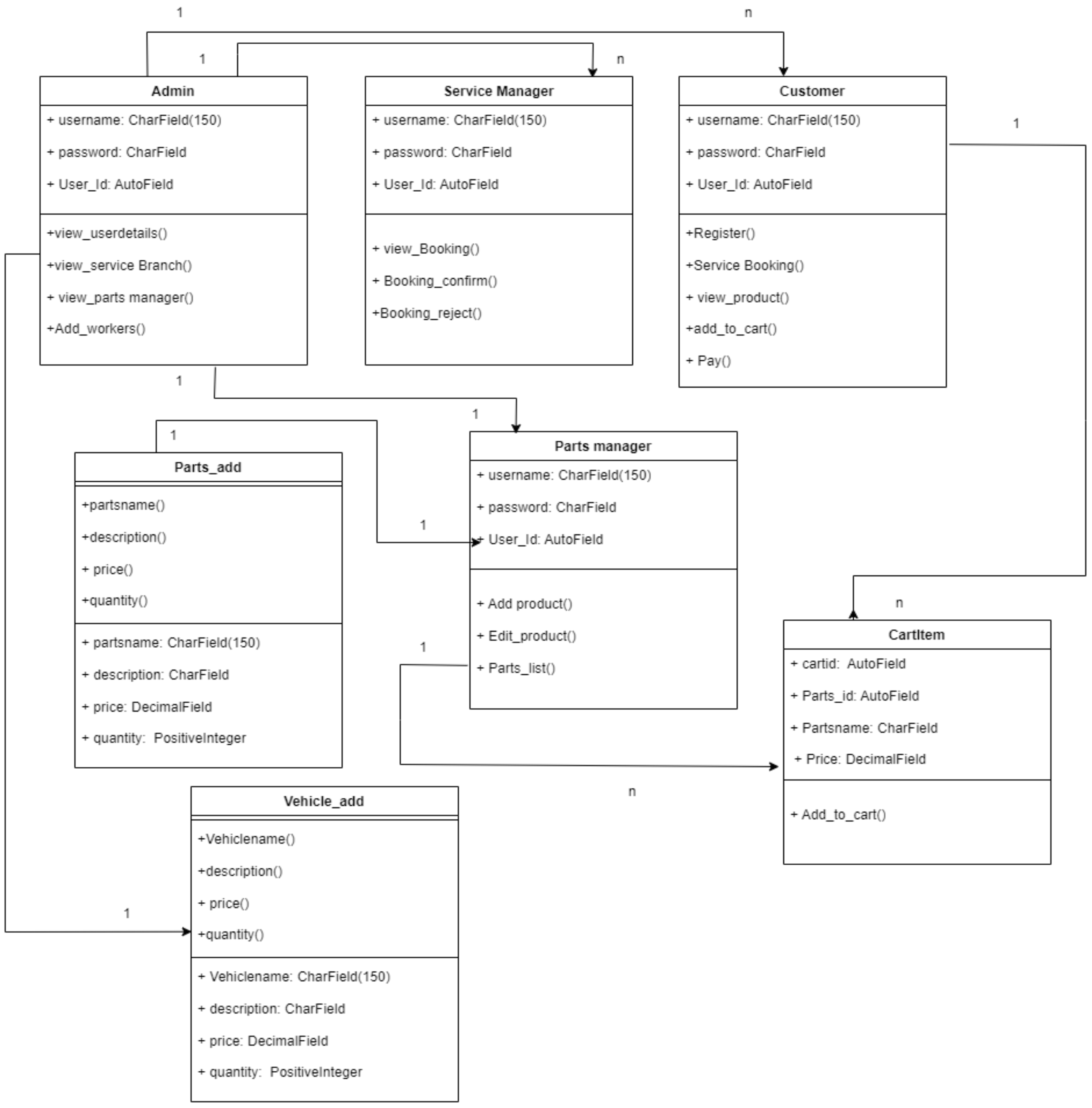


Fig. 4.2.5 Class diagram for AutoMate Hub

4.2.6 Object Diagram

Class diagrams are necessary before object diagrams can be created since they are the ancestor of object diagrams. An object diagram represents a particular instance of a class diagram. The underlying concepts used in class and object diagrams are the same. Object diagrams may also describe the static view of a system, although this static view only depicts a current state of the system. Object diagrams are used to show links between a set of things.

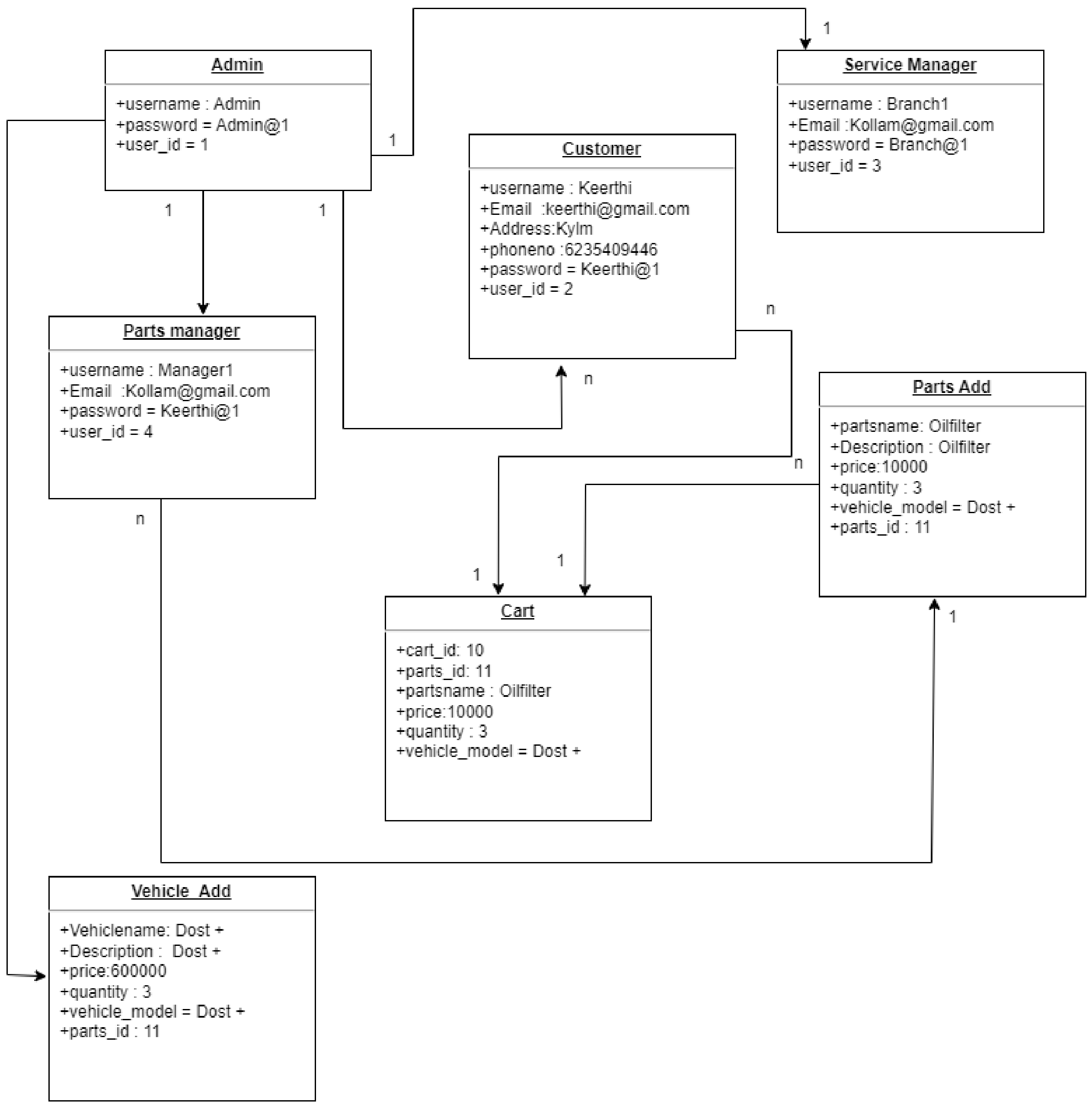


Fig. 4.2.6 Object diagram for AutoMate Hub

4.2.7 Component Diagram

Component diagrams have different behaviours and personalities. The physical parts of the system are represented using component diagrams. Executables, libraries, files, documents, and other items that are physically present in a node are just a few examples. Component diagrams are used to show how the components of a system are connected and arranged. These diagrams may also be used to construct systems that can be run

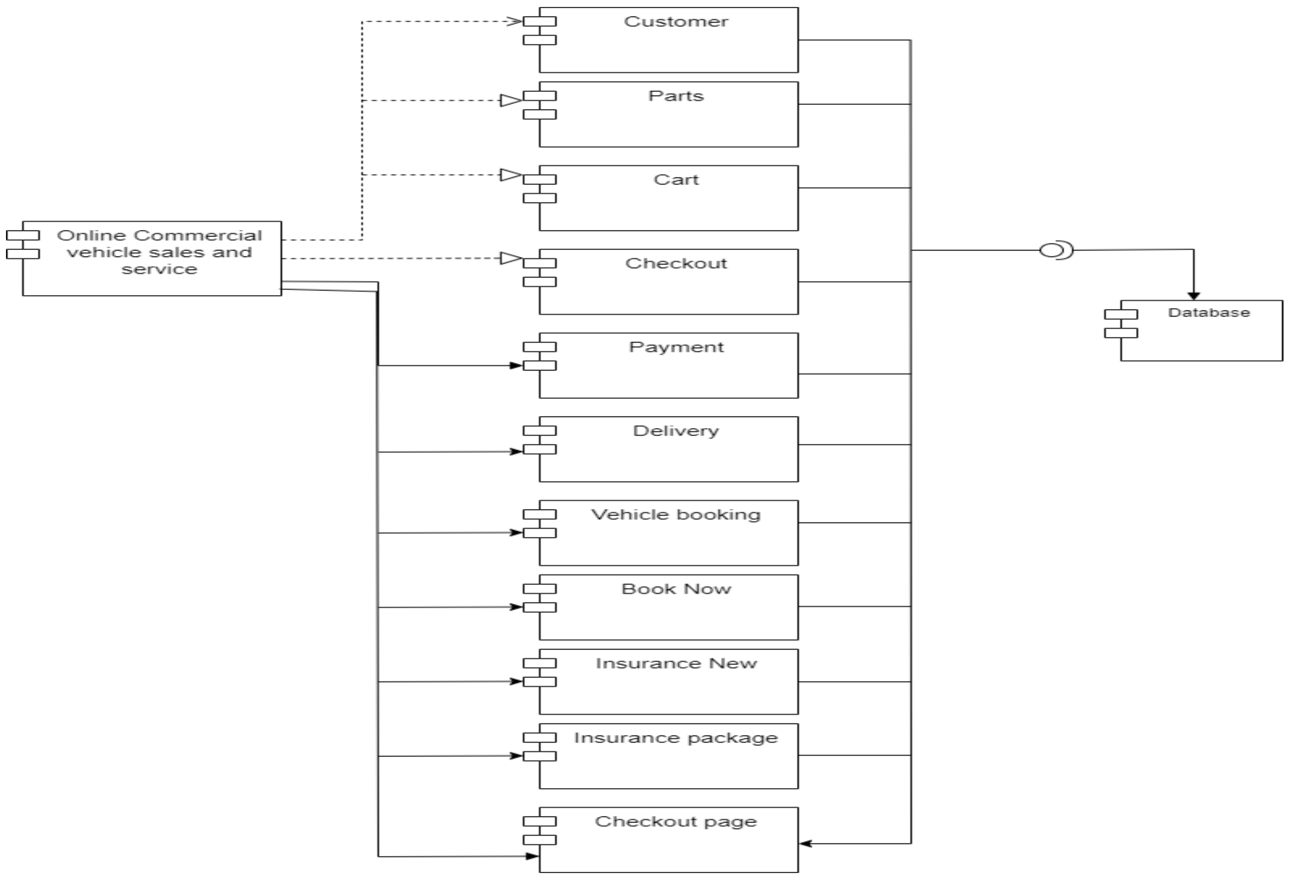
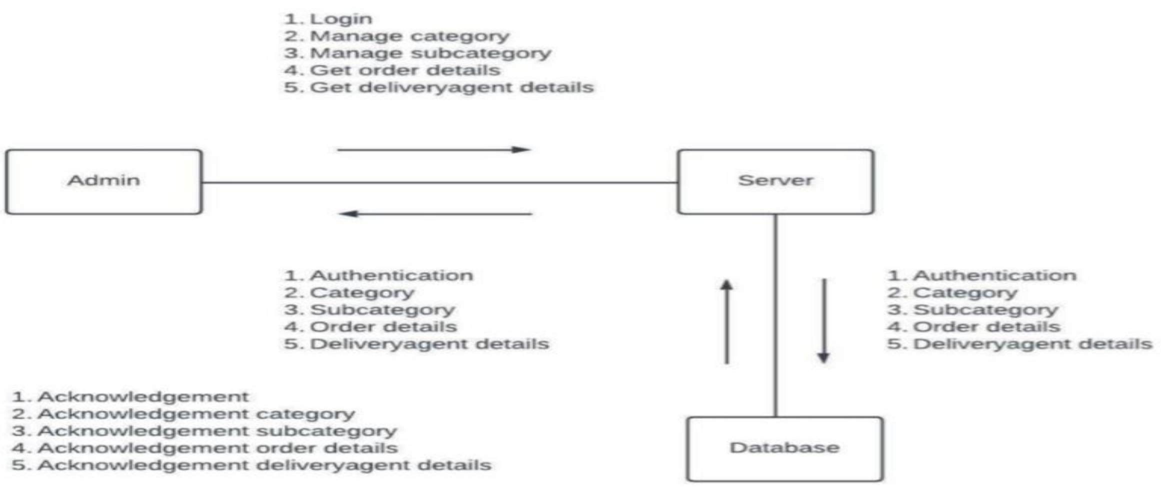


Fig. 4.2.7 Component diagram for AutoMate Hub

4.2.8 Collaboration Diagram

A collaboration diagram is a diagram that is used to represent the relationships between objects in a system. It is so similar to a sequence diagram in that it represents the same information, but it does so in a different way. Instead of showing message flow between objects, it depicts the structure of the objects in the system. This is because collaboration diagrams are based on object-oriented programming, where objects have various attributes and are connected to each other. Thus, collaboration diagrams are a visual representation of the objects architecture on a system.



4.2.8 Deployment Diagram

An execution architecture of a system, containing nodes like hardware or software execution environments, and the middleware linking them, is shown in a deployment diagram, a form of UML diagram. Typically, deployment diagrams are used to represent the actual hardware and software of a system. By using it, you can comprehend how the hardware will physically deliver the system. In contrast to other UML diagram types, which primarily depict the logical components of a system, deployment diagrams assist describe the hardware structure of a system.

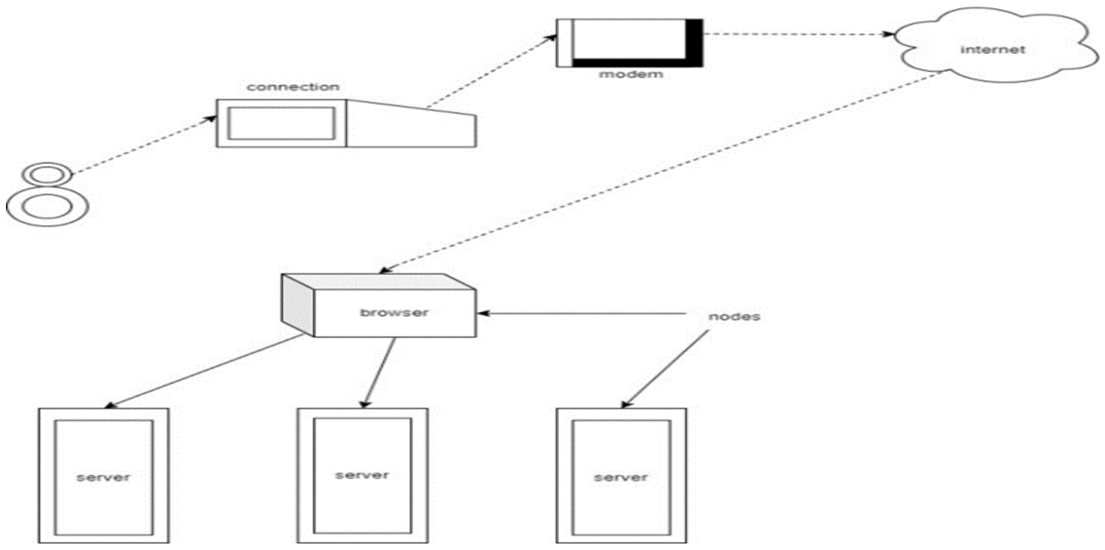
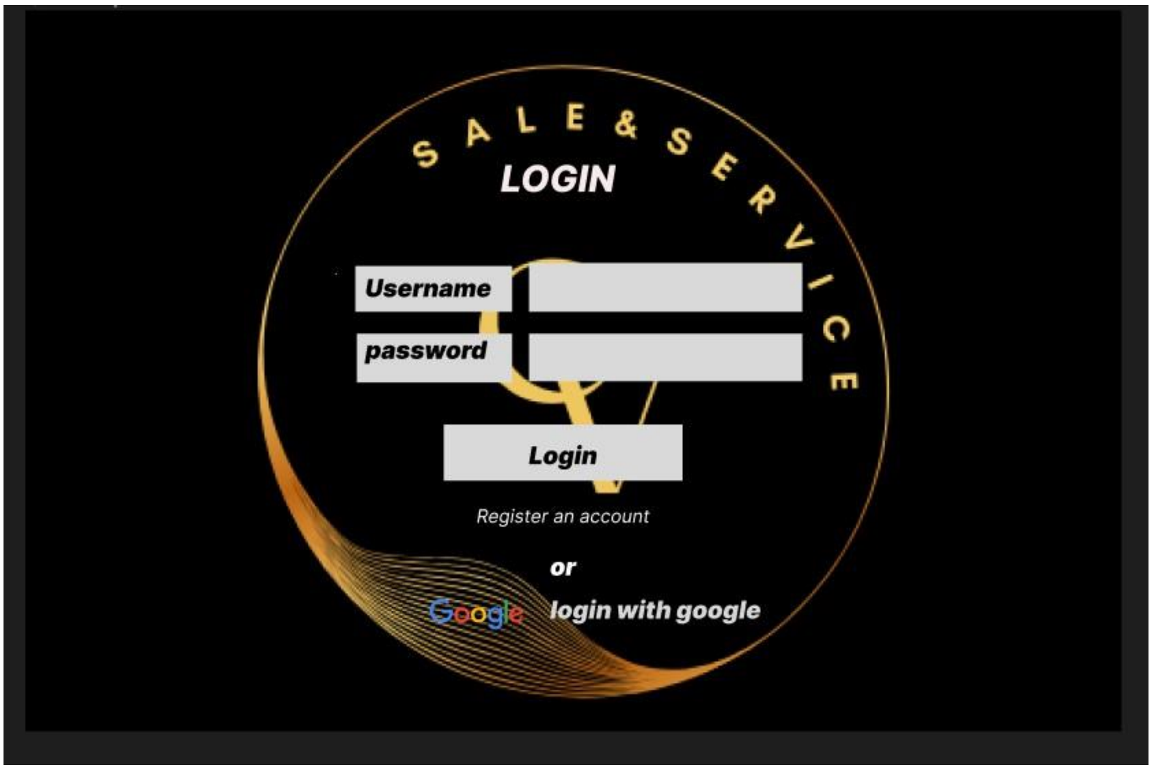


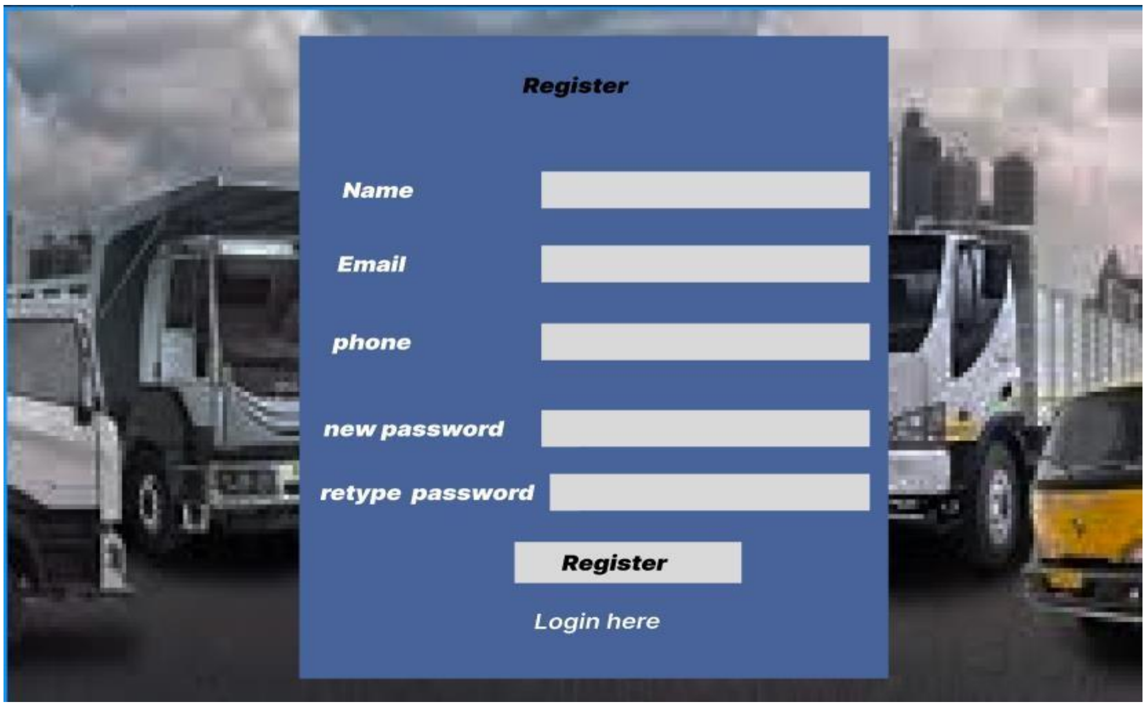
Fig. 4.2.8 Deployment diagram for AutoMate Hub

4.3 USER INTERFACE DESIGN USING FIGMA

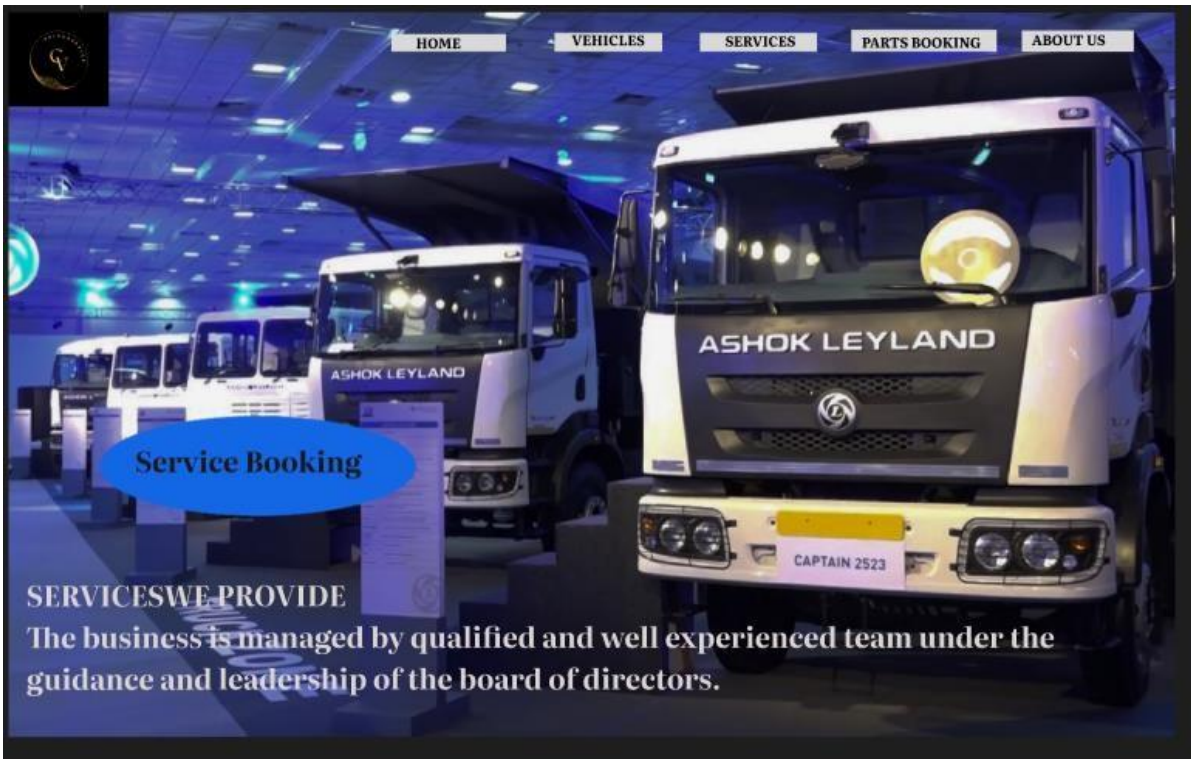
Form Name: Login page



Form Name: Signup page

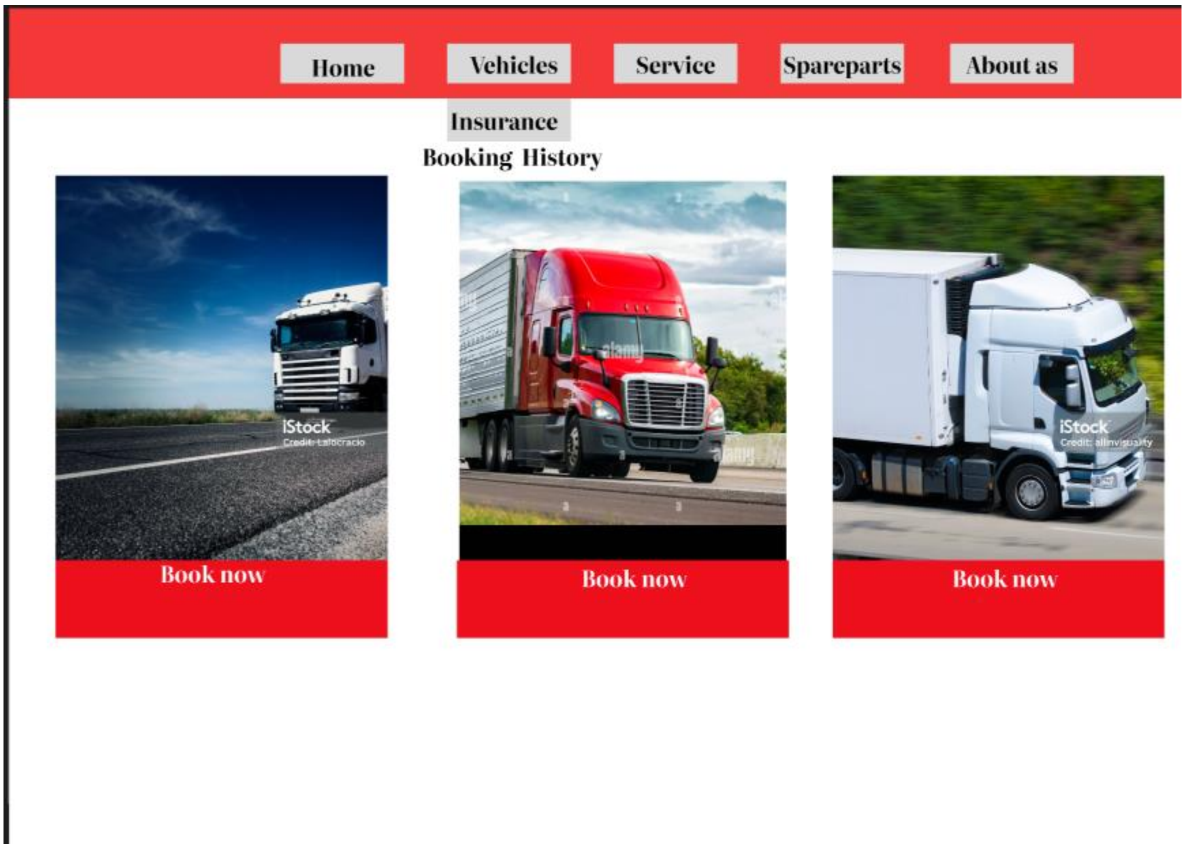


Form Name: Home page

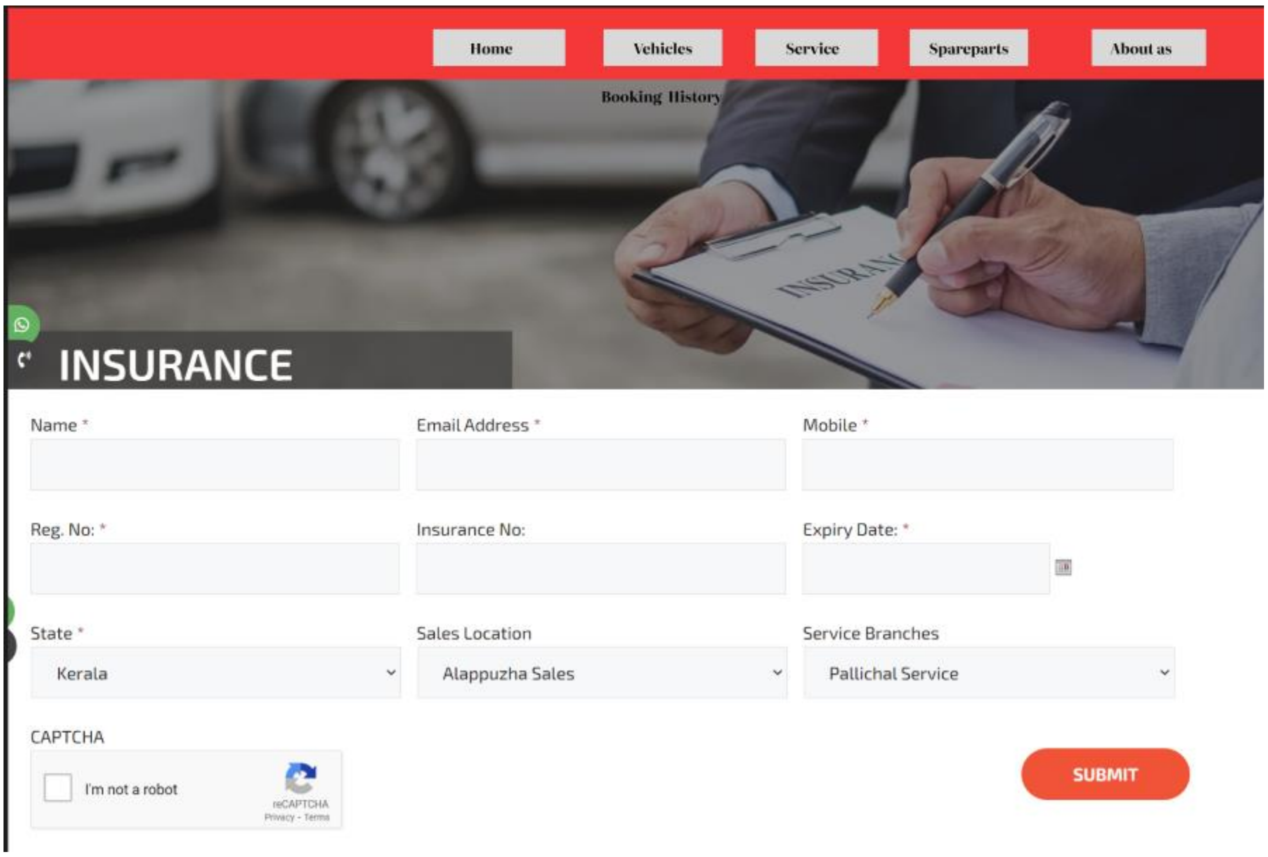


SERVICESWE PROVIDE
The business is managed by qualified and well experienced team under the guidance and leadership of the board of directors.

Form Name: Booking page



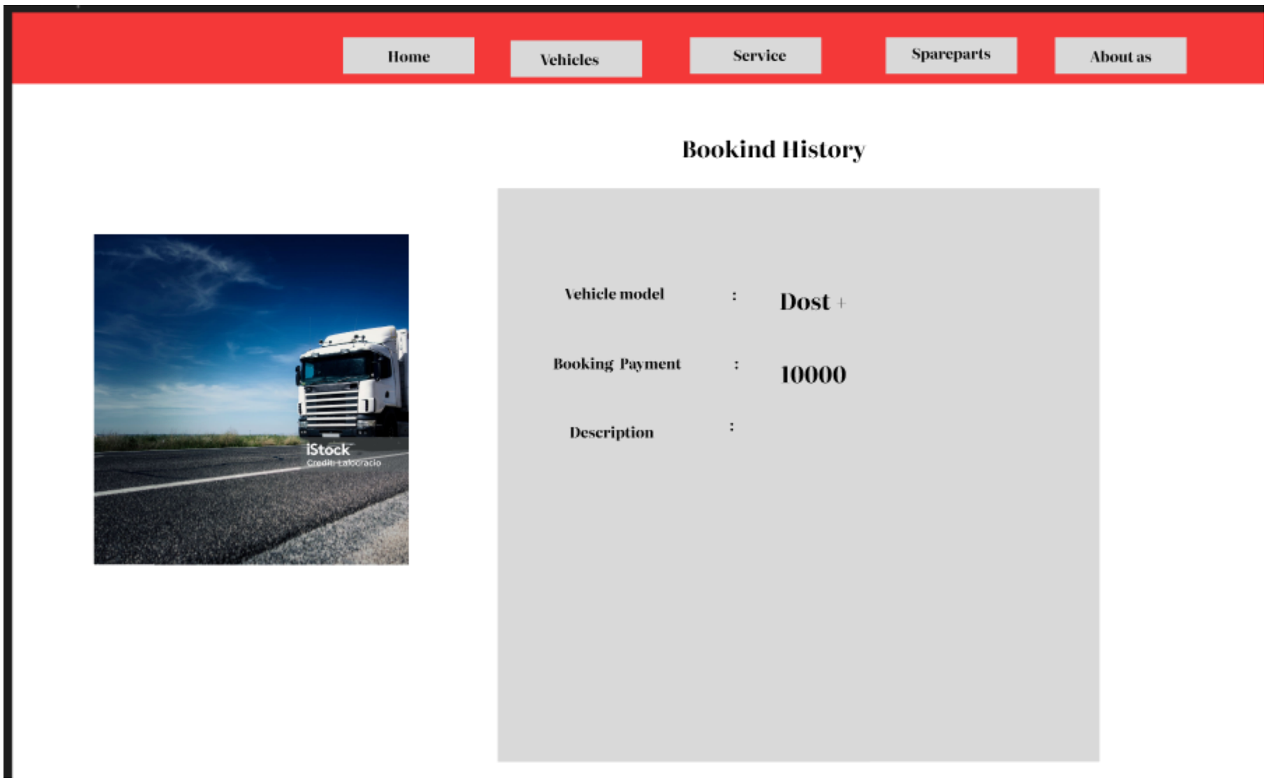
Form Name: Insurance page



Name *	Email Address *	Mobile *
Reg. No: *	Insurance No:	Expiry Date: *
State *	Sales Location	Service Branches
Kerala	Alappuzha Sales	Pallichal Service

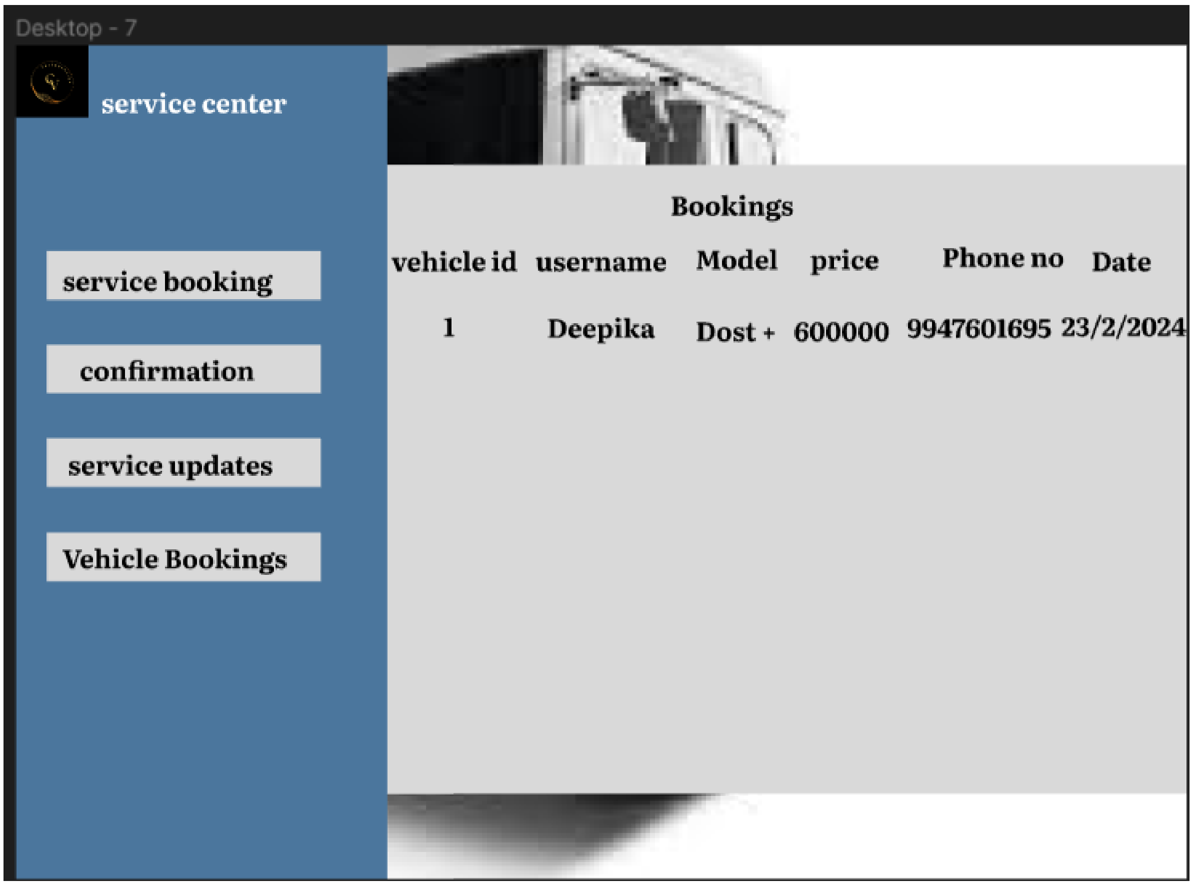
CAPTCHA

Form Name: Booking history page



Vehicle model	:	Dost +
Booking Payment	:	10000
Description	:	

Form Name: Bookings view page



Bookings

vehicle id	username	Model	price	Phone no	Date
1	Deepika	Dost +	600000	9947601695	23/2/2024

4.2 DATABASE DESIGN

A database is a collection of data that has been organized to make it simple to manage and update.Information security could be one of the main aims of any database. The database design process is divided into two steps. The goal of the first step is to gather user requirements to create a databasethat as clearly as possible satisfies user needs. It is known as information-level design and is carriedout without the aid of any DBMS. An information-level design is changed to a specific DBMS design that will be used to develop the system in the following stage. The physical-level design phase is where the characteristics of the specific DBMS are considered.

4.4.1 Relational Database Management System (RDBMS)

A Relational Database Management System (RDBMS) is a critical software application for organizing and managing data in a structured manner. It stores data in tables with rows and columns,where each row represents a record, and each column is a specific attribute or field. RDBMS systems like MySQL, Oracle, or Microsoft SQL Server ensure data integrity, enforce relationshipsbetween tables, and allow for efficient data retrieval and manipulation through Structured Query Language (SQL). These systems are widely used in various applications, such as e-commerce websites, financial systems, and inventory management, due to their robustness, scalability, and ability to handle complex data structures, making them essential for modern data-driven environments. A Relational Database Management System (RDBMS) is a cornerstone of modern data management. It structures data into tables, where each row represents a unique entry, and eachcolumn corresponds to a specific attribute, ensuring a logical and organized storage system. RDBMS systems employ complex algorithms for data retrieval and manipulation, guaranteeing data consistency and adherence to predefined relationships between tables. Popular RDBMS software, including PostgreSQL, MySQL, and Microsoft SQL Server, provides robust features fortransactions, data security, and scalability. These systems are integral to a myriad of applications, from healthcare records and customer databases to inventory control and financial platforms, supporting the efficient storage, retrieval, and analysis of vast datasets, making them essential toolsin today's data-driven world.

4.4.2 Normalization

Normalization is a database design technique used in relational database management systems (RDBMS) to eliminate data redundancy and improve data integrity. It involves organizing data in a way that reduces data duplication and ensures that relationships between tables are defined and maintained.

The primary goals of normalization are

Minimizing Data Redundancy: By breaking down data into separate tables and ensuring that each piece of data is stored only once, normalization helps reduce the chances of data inconsistencies orerrors.

Ensuring Data Integrity: Normalization enforces the rules of referential integrity, which means that relationships between tables are well-defined and maintained. This ensures that data remains accurate and consistent.

Normalization typically involves dividing a database into multiple related tables and using primarykeys and foreign keys to establish relationships between these tables. There are several normal forms, from First Normal Form (1NF) to Fifth Normal Form (5NF), each with specific rules and requirements. The level of normalization achieved depends on the specific needs of the database and the trade-off between data redundancy and query performance.

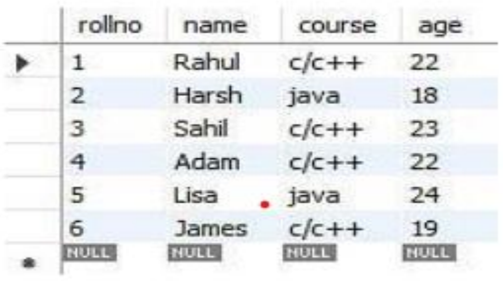
By applying normalization principles, designers can create efficient and reliable database structuresthat support data integrity and ease data maintenance and manipulation.

There are several normal forms (NF) that define specific rules and requirements for achieving progressively higher levels of normalization. Here are the most common normal forms, from FirstNormal Form (1NF) to Fifth Normal Form (5NF)

First Normal Form (1NF)

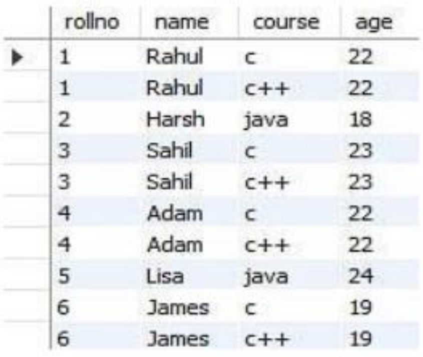
A table is considered to be in the First Normal Form if it satisfies the requirement of atomicity,which means that the table's values cannot be divided into smaller, more granular parts, and each attribute or column contains only a single value. In other words, each column should hold only one piece of information, and there should be no repeating groups or arrays of values within a single row. The concept of atomicity in this context denotes that a cell within a database cannot contain more than one value, and must exclusively store a single-valued attribute. The first normal form in database normalization places limitations on attributes that have multiple values, are composed of multiple sub-attributes, or contain a combination of both. Therefore, in order to satisfy the conditions of the first normal form., these attributes need to be modified or removed, a relation must not have multi-valued or composite attributes, and any such attributes must be split into individual attributes to form atomic values.

E.g. The table contains information pertaining to students, including their roll number, name, course of study, and age.



rollno	name	course	age
1	Rahul	c/c++	22
2	Harsh	java	18
3	Sahil	c/c++	23
4	Adam	c/c++	22
5	Lisa	java	24
6	James	c/c++	19
NULL	NULL	NULL	NULL

The table containing the records of students displays a violation of the First Normal Form due to the presence of two distinct values in the course column. To ensure compliance with the First Normal Form, the table has been modified resulting in the formation of a new table.

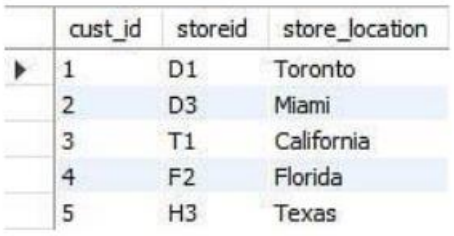


rollno	name	course	age
1	Rahul	c	22
1	Rahul	c++	22
2	Harsh	java	18
3	Sahil	c	23
3	Sahil	c++	23
4	Adam	c	22
4	Adam	c++	22
5	Lisa	java	24
6	James	c	19
6	James	c++	19

Second Normal Form (2NF):

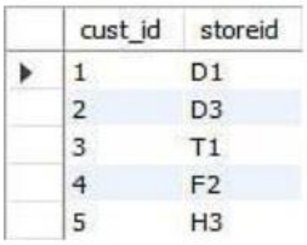
To meet requirements of Second Normal Form, a table must satisfy the criteria of First Normal Form as a prerequisite. Furthermore, the table must not exhibit partial dependency, which refers to a scenario where a non-prime attribute is dependent on a proper subset of the candidate key. In other words, the table should have no attributes that are determined by only a portion of the primary key.Now understand the Second Normal Form with the help of an example.

Consider the table Location:

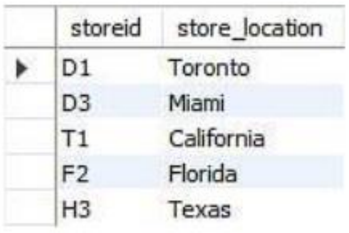


cust_id	storeid	store_location
1	D1	Toronto
2	D3	Miami
3	T1	California
4	F2	Florida
5	H3	Texas

The Location table currently has a composite primary key consisting of cust id and storied, and its non-key attribute is store location. However, the store location attribute is directly determined by the storied attribute, which is part of the primary key. As a result, this table does not meet the requirements of second normal form. To address this issue and ensure second normal form is met, it is necessary to separate the Location table into two separate tables. This will result in the creation of two distinct tables that accurately represent the relevant data and relationships: one for customer IDs and store IDs, and another for store IDs and their respective locations.:



cust_id	storeid
1	D1
2	D3
3	T1
4	F2
5	H3

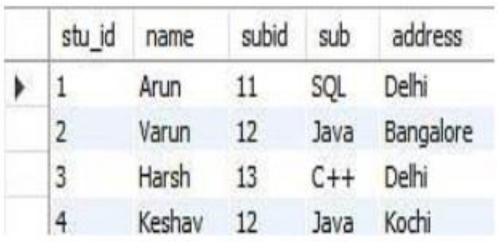


storeid	store_location
D1	Toronto
D3	Miami
T1	California
F2	Florida
H3	Texas

Third Normal Form (3NF):

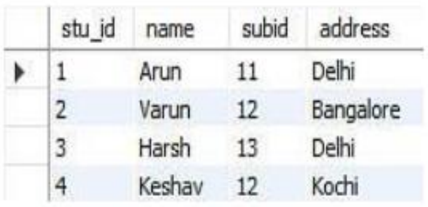
A table must meet the requirements of Second Normal Form in order to be considered in Third Normal Form, and also fulfill two additional conditions. The second condition states that non-prime attributes should not rely on non-prime characteristics that are not a part of the candidate key within the same table, thus avoiding transitive dependencies. A transitive dependency arises when A → C indirectly, due to A → B and B → C, where B is not functionally dependent on A. The main objective of achieving Third Normal Form is to reduce data redundancy and guarantee data integrity.

For instance, let's consider a student table that includes columns like student ID, student name, subject ID, subject name, and address of the student. To comply with the requirements for Third Normal Form, this table must first meet the standards of Second Normal Form and then ensure that there are no transitive dependencies between non-prime attributes.

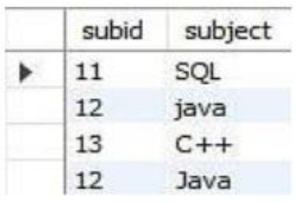


stu_id	name	subid	sub	address
1	Arun	11	SQL	Delhi
2	Varun	12	Java	Bangalore
3	Harsh	13	C++	Delhi
4	Keshav	12	Java	Kochi

Now to change the table to the third normal form, you need to divide the table as shown below: Based on the given student table, it can be observed that the stu_id attribute determines the sub_id attribute, and the sub_id attribute determines the subject (sub). This implies that there is a transitive functional dependency between stu_id and sub. As a result, the table does not satisfy the criteria for the third normal form. To adhere to the third normal form, the table must be divided into separate tables where each table represents a unique entity or relationship. In this case, the table can be divided into two tables: one for the student-subject relationship (stu_id and sub_id), and another for the subject information (sub_id and sub):



stu_id	name	subid	address
1	Arun	11	Delhi
2	Varun	12	Bangalore
3	Harsh	13	Delhi
4	Keshav	12	Kochi



subid	subject
11	SQL
12	java
13	C++
12	Java

The two tables illustrate that the non-key attributes are completely determined by and reliant on the primary key. In the first table, the columns for name, sub_id, and addresses are all exclusively linked to the stu_id. Likewise, the second table demonstrates that the sub column is entirely dependent on the sub_id.

4.4.3 Sanitization

In Python Django, sanitization refers to the process of cleaning and validating data to ensure that it is safe and free from malicious content before it is used or stored in a database. Sanitization is a crucial security practice to prevent various forms of attacks, such as cross-site scripting (XSS) and SQL injection, which can compromise the security and integrity of a web application.

4.4.4 Indexing

Indexing in Django is a fundamental database optimization technique. It involves creating data structures, or indexes, on specific fields to expedite data retrieval. These indexes act as signposts that enable the database to swiftly locate and fetch relevant data, especially in tables with substantialamounts of information. Django offers automatic index creation for primary keys, unique fields, and foreign keys. Additionally, developers can define custom indexes for fields frequently used infiltering, sorting, or searching. The choice of proper indexing plays a pivotal role in improving query performance, resulting in more responsive and scalable web applications. It's essential to consider application-specific query patterns and create indexes accordingly, as well as to understandthe capabilities and limitations of the chosen database backend. Effective indexing is a cornerstone of efficient database operations in Django, contributing to enhanced application performance.

4.3 TABLE DESIGN

1.tbl_Registration

Primary Key: User_id

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	User_id	AutoField	Primary Key	Id for Registration table
2	custmname	CharField(150)	Not Null	User’s customername
3	Email	EmailField	Unique	User’s email
4	Address	TextField	Not Null	User’s address
5	Phone Number	CharField(12)	Not Null	User’s Phone Number
6	Password	CharField	Not Null	User’s Password
7	Role	CharField(100)	Default:	User’s role type

2. tbl_Userprofile

Foreign Key : **User_id** references table **tbl_Registration**
Primary Key : **Profile_Id**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Profile_Id	AutoField	Primary Key	Unique Id for Userprofile
2	First_name	CharField(255)	Not null	User firstname
3	Last_name	CharField(255)	Not Null	Users last name
4	Phone_number	CharField(12)	Not Null	User phoneno
5	Address	TextField	Not Null	User Address
6	Pincode	CharField(10)	Not Null	User pincode
7	City	CharField(50)	Not Null	User city
8	State	CharField(50)	Not Null	User state
9	User_id	OneToOneField	Not Null	Unique identification

3. tbl_Booking

Foreign Key : **User_id** references table **tbl_Registration**
Primary Key : **Booking_Id**

No:	**Field name**	**Datatype (Size)**	**Key Constraints**	**Description of the field**
1	Booking_Id	BigAutoField	Primary key	Unique identifier
2	Driver_number	CharField(15)	Not Null	Vehicle driver number
3	Vehicle_number	CharField(50)	Not Null	User vehicle number
4	Service_branch	CharField(50)	Not Null	Service branch
5	Vehicle_model	CharField(50)	Not Null	Vehicle model
6	Service_type	CharField(10)	Not Null	Service type
7	Service_date	DateField	Not Null	Service date
8	Email	EmailField	Not Null	User email
9	User_id	OneToOneField	Foreign key	Reference to register page

4. tbl_Parts

Primary Key : **Parts_Id**

No:	**Field name**	**Datatype (Size)**	**Key Constraints**	**Description of the field**
1	Parts_Id	AutoField	Primary Key	Unique identity
2	Partsname	CharField(100)	Not Null	Parts name
3	Description	TextField	Not Null	Parts description
4	Price	DecimalField	Not Null	Parts price
5	Parts_image	ImageField	Not Null	Parts image
6	Quantity	PositiveInteger	Not Null	Parts quantity
6	Products	CharField	Not Null	Parts model

5. tbl_Cart

Primary Key : **Cart_id**
Foreign key : **User_id** references table **tbl_Registration**
Foreign key : **Parts_id** references table **tbl_Parts**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Cart_id	AutoField	Primary Key	Unique identity
2	User_id	OneToOneField	Foreign key	Reference to registration page
3	Parts_id	AutoField	Foreign key	Reference to Parts table

6. tbl_Order

Primary Key : **Order_id**
Foreign key : **User_id** references table **tbl_Registration**
Foreign key : **Parts_id** references table **tbl_Parts**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Order_id	BigAutoField	Primary key	Unique identification
2	Parts_id	PositiveIntegerField	Foreign key	Reference to partstable
3	Total_amount	DecimalField	Not Null	Total amount
4	Payment_status	BooleanField	Not Null	Not Null
5	Created_at	DateTimeField	Not Null	Not Null
6	User_id	OneToOneField	Foreign key	Reference to registertable

7. tbl_Vehicle_add

Primary Key : **Vehicle_id**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Vehicle_id	BigAutoField	Primary key	Unique identification
2	Vehiclename	CharField(255)	Not Null	Vehicle Name
3	Description	TextField()	Not Null	Vehicle description
4	Price	DecimalField(10)	Not Null	Vehicle Price
5	Vehicle_model	CharField(20)	Not Null	Vehicle model
6	Vehicle_usage	CharField(20)	Not Null	Vehicle usage
7	Vehicle_application	CharField(20)	Not Null	Vehicle Application
8	Fuel_type	CharField(20)	Not Null	Vehicle Fuel_type
9	Transmission_type	CharField(20)	Not Null	Transmission type
10	Engine_size	IntegerField()	Not Null	Vehicle Engine_size
11	Mileage	FloatField()	Not Null	Vehicle Mileage
12	Warranty	CharField(100)	Not Null	Vehicle Warranty
13	Seating_capacity	IntegerField()	Not Null	Vehicle Seating capacity
14	Fuel_tank_capacity	FloatField()	Not Null	Vehicle Fuel tank capacity
15	Stock	IntegerField()	Not Null	Vehicle Stock
16	Brouchure	FileField()	Not Null	Vehicle Brouchure
17	Booking Amount	DecimalField(10)	Not Null	Vehicle Booking Amount

8. tbl_Vehicle_booking

Primary key: **Vehicle_book_id**
Foreign key: **Vehicle_id** references table **tbl_Vehicle_add**
Foreign key: **User_id** refrences table **tbl_Registration**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Vehicle_book_id	BigAutoField	Primary key	Unique identification
2	Vehicle_id	BigAutoField	Foreign key	Reference to Vehicle_add page
3	User_id	OneToOneField	Foreign key	Reference to register page
4	Booking_amount	DecimalField	Not Null	Total amount
5	Booking_time	CharField(100)	Not Null	Not Null
6	Status	BooleanField	Not Null	Not Null

9. tbl_Insurance_add

Primary key: **Insurance_id**
Foreign key: **Vehicle_id** references table **tbl_Vehicle_add**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Insurance_id	BigAutoField	Primary key	Unique identification
2	Vehicle_id	BigAutoField	Foreign key	Reference to vehicle add page
3	Insurance_type	CharField(20)	Not Null	Insurance type
4	Insurance_price	DecimalField(10)	Not Null	Insurance price
5	Renew_price	DecimalField(10)	Not Null	Insurance renewal price

10. tbl_InsuranceNew

Primary key: **Insurance_New_id**
Foreign key: **User_id** references table **tbl_Registration**
Foreign key: **Register_id** references table **tbl_ VehicleRegistration**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	User_id	AutoField()	Foreign key	Reference to registration page
2	Insurance_New_id	AutoField()	Primary key	Unique identification

3	Register_id	AutoField()	Foreign key	Reference to Vehicle registration no
4	State	CharField()	Not Null	Vehicle register state
5	Id_proof	FileField()	Not Null	Vehicle Id_proof

11. tbl_PaymentRecord

Primary key: **Payments_id**
Foreign key: **Insurance_id** references table **tbl_Insurance_add**
Foreign key: **User_id** references table **tbl_Registration**
Foreign key: **Insurance_New_id** references table **tbl_ InsuranceNew**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	User_id	AutoField()	Foreign key	Reference to register page
2	Insurance_id	AutoField()	Foreign key	Reference to Insurance page
3	Amount_paid	DecimalField(10)	Not Null	Insurance new paid amount
4	Insurance_New_id	AutoField()	Foreign key	Insurance new id reference to insurance new page
5	Payment_datetime	DateTimeField()	Not Null	Payment date and time
6	Payments_id	AutoField	Primary key	Unique identification

12. tbl_Policy

Primary key: **Policy_Number**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Policy_Number	AutoField()	Primary key	Unique identification
2	Issued_date	DateField()	Not Null	Policy issude date
3	Expiry_date	DateField()	Not Null	Policy expire date

13. tbl_VehicleRegistration

Primary key: **Register_id**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Register_id	AutoField()	Primary key	Unique identification
2	Register_number	CharField()	Not Null	Vehicle Register Number

14. tbl_ConfirmedVehicle

Primary key: **Conf_Vehicle_id**
Foreign key: Register_id reference table tbl_VehicleRegistration
Foreign key: **User_id** reference to **tbl_registration**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Conf_Vehicle_id	CharField(100)	Primary Key	Confirmed vehicle id
2	User_id	AutoField()	Foreign key	Reference to register page
3	Register_id	AutoField()	Foreign key	Reference to VehicleRegistrat ion page
4	Booking_amount	DecimalField()	Not Null	Vehicle booking Ammount
5	Booking_time	DateTimeField()	Not Null	Vehicle booking time

CHAPTER 5
SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer thequestion – Does the software behave as specified? Software testing is often used in association withthe terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is justone kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted. Other activities which are often associated with software testing are static analysisand dynamic analysis. Static analysis investigates the source code of software, looking for problemsand gathering metrics without actually executing the code. Dynamic analysis looks at the behaviorof software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information. Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the systemtesting objectives, there are several rules that can serve as testing objectives. They are: Testing is aprocess of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error. If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met. Thereare three ways to test program.
- For correctness
- For implementation efficiency
- For computational complexity. Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan. The levels of testing include: Unit testing Integration Testing Data validation Testing Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is whitebox oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested. Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. More over differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests. Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points: Input Screen Designs, Output Screen Designs, The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.2.5 Automation Testing

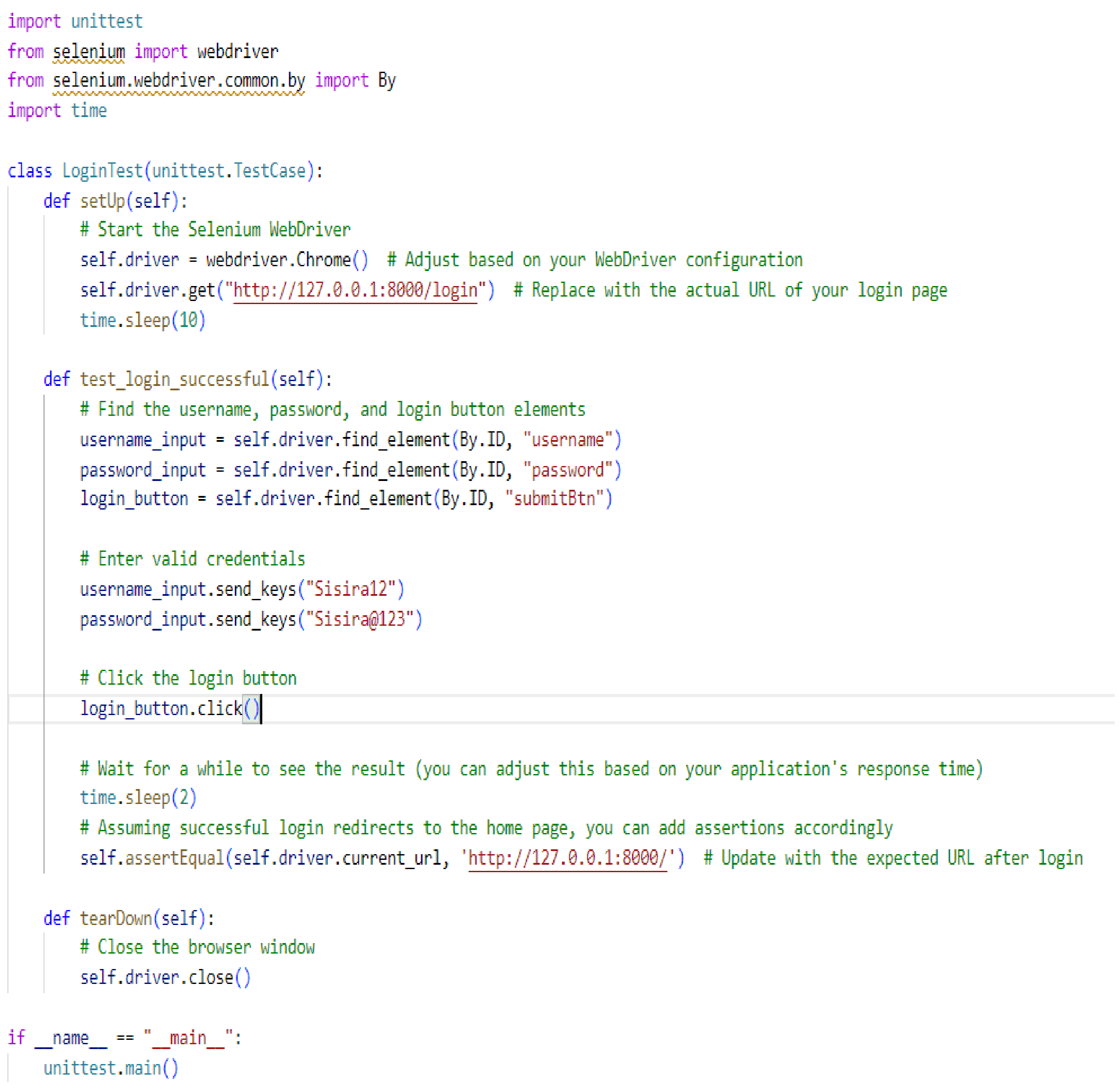
Automated testing is a process that validates if software is functioning appropriately and meeting requirements before it is released into production. This software testing method uses scripted sequences that are executed by testing tools. UI automation testing is a technique where these testingprocesses are performed using an automation tool. Instead of having testers click through the application to verify data and action flows visually, test scripts are written for each test case. A series of steps to follow when the verifying data is then added. Automatic testing is required when you want to run the same test cases across multiple machines at the same time.

5.2.6 Selenium Testing

Selenium is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python, etc to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as Selenium Testing. Selenium Integrated Development Environment (IDE) is the simplest framework in the Selenium suite and is the easiest one to learn. It is a Chrome and Firefox plugin that you can install as easily as you can with other plugins. However, because of its simplicity, Selenium IDE should only be used as a prototyping tool. If you want to create more advanced test cases, you will need to use either Selenium RC or WebDriver. Selenium RC was the flagship testing framework of the whole Selenium project for a long time. This is the first automated web testing tool that allows users to use a programming language they prefer. RC can support the following programming languages JavaC#, PHP, Python, Perl, Ruby. Selenium can be used to automate functional tests and can be integrated with automation test tools such as Maven, Jenkins, & Docker to achieve continuous testing. It can also be integrated with tools such as TestNG, & JUnit for managing test cases and generating reports. For example, Selenium is used to test the login functionality of the patient portal by automating the entry of login credentials and verifying that the correct patient dashboard is displayed after successful authentication. Selenium can also be used to automate the testing of appointment scheduling by programmatically entering various input scenarios and verifying that the resulting output matches the expected behavior.

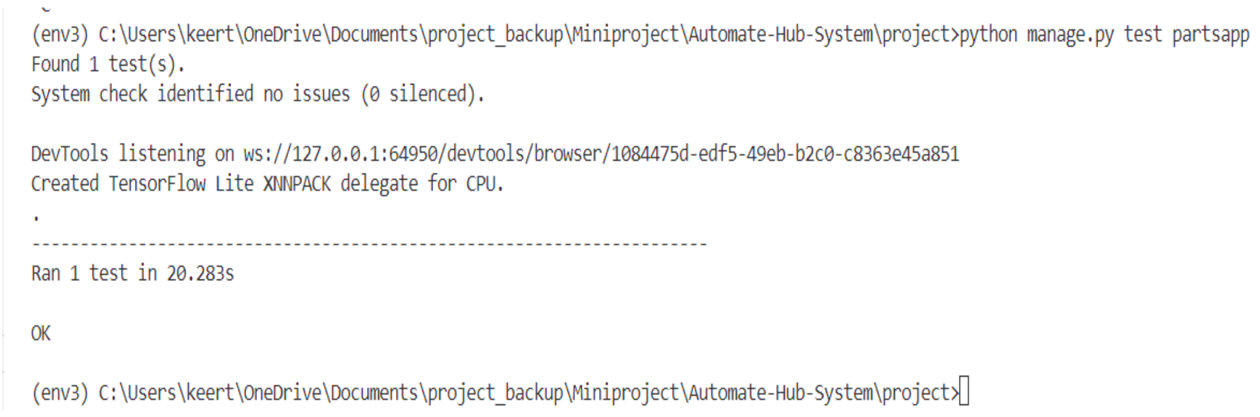
Test Case 1

Code



```
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

class LoginTest(unittest.TestCase):
    def setUp(self):
        # Start the Selenium WebDriver
        self.driver = webdriver.Chrome()  # Adjust based on your WebDriver configuration
        self.driver.get("http://127.0.0.1:8000/login")  # Replace with the actual URL of your login page
        time.sleep(10)

    def test_login_successful(self):
        # Find the username, password, and login button elements
        username_input = self.driver.find_element(By.ID, "username")
        password_input = self.driver.find_element(By.ID, "password")
        login_button = self.driver.find_element(By.ID, "submitBtn")

        # Enter valid credentials
        username_input.send_keys("Sisira12")
        password_input.send_keys("Sisira@123")

        # Click the login button
        login_button.click()

        # Wait for a while to see the result (you can adjust this based on your application's response time)
        time.sleep(2)
        # Assuming successful login redirects to the home page, you can add assertions accordingly
        self.assertEqual(self.driver.current_url, 'http://127.0.0.1:8000/')  # Update with the expected URL after login

    def tearDown(self):
        # Close the browser window
        self.driver.close()

if __name__ == "__main__":
    unittest.main()
```

Screenshot



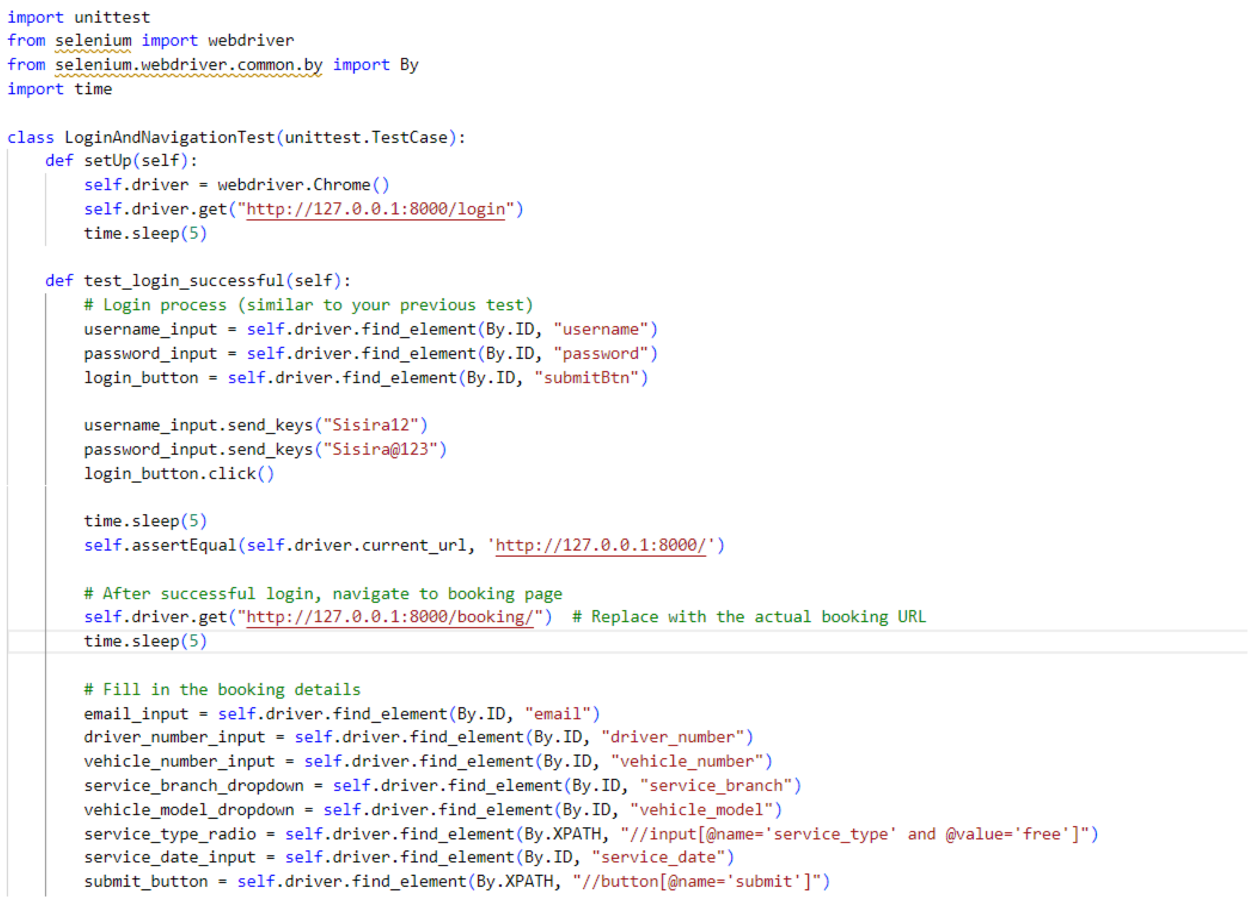
```
(env3) C:\Users\keert\OneDrive\Documents\project_backup\Miniproject\Automate-Hub-System\project>python manage.py test partsapp
Found 1 test(s).
System check identified no issues (0 silenced).

DevTools listening on ws://127.0.0.1:64950/devtools/browser/1084475d-edf5-49eb-b2c0-c8363e45a851
Created TensorFlow Lite XNNPACK delegate for CPU.
.
----------------------------------------------------------------------
Ran 1 test in 20.283s

OK

(env3) C:\Users\keert\OneDrive\Documents\project_backup\Miniproject\Automate-Hub-System\project>
```

Test Report

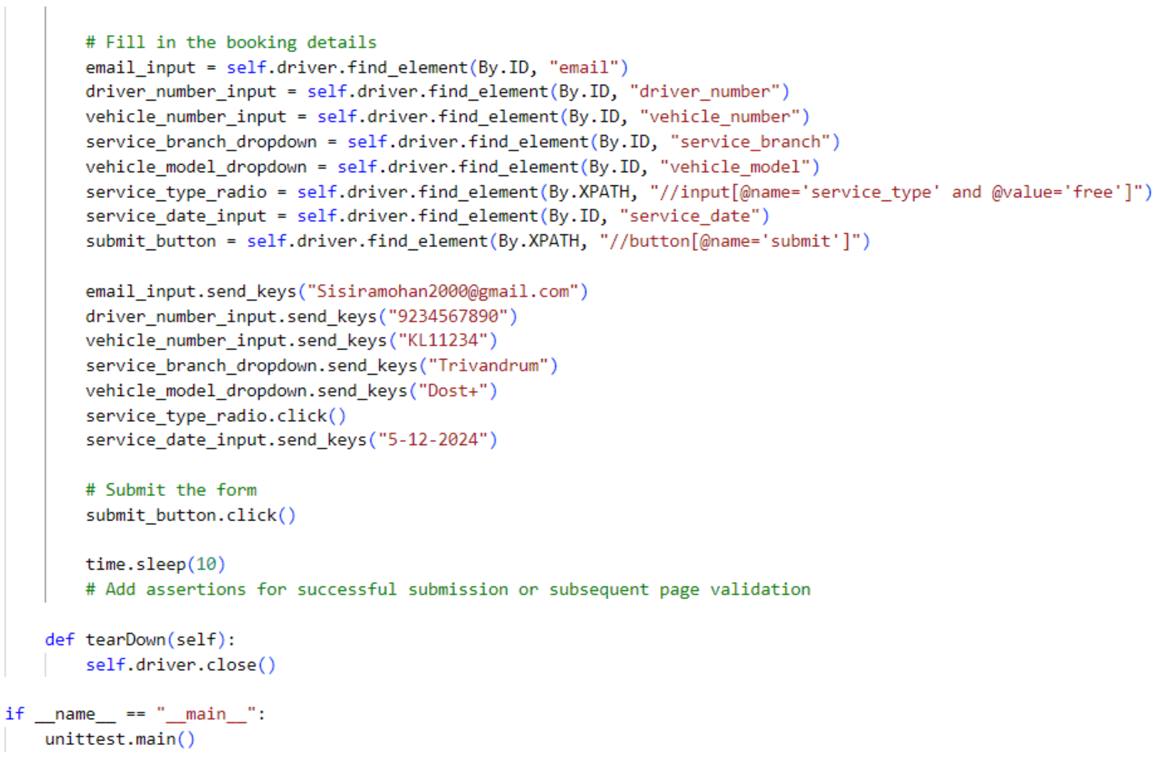
Test Case 1

Project Name: Commercial vehicle service and Spare parts Management					
Login Test Case					
Test Case ID: Test_1			**Test Designed By:** Keerthi U		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 12/4/2024		
Module Name: Login page			**Test Executed By :** Rini Kurian		
Test Title : Verify login with email and password			**Test Execution Date: 1**5/4/2024		
Description: Testing the login page					
Pre-Condition : User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass /Fail)**
1	Navigation to Login page		Login Page should be displayed	Login Page displayed	Pass
2	Provide valid username	Username: Keerthi	User should be able to login	User logged in and navigated to dashboard	Pass
3	Provide valid password	Keerthi@1			
4	Click on Login Button				
5	Provide Invalid username or password	Username: keerthi1 Password: keerthi123	User should not be able in Login	Message for enter valid email id or password displayed	Pass
6	Provide null username or password	username:null Password: null			
7	Click on Login In button				
Post-Condition: Post-Condition: User is validated with database and login to the website. The account session details are logged in database					

Test Case 2:

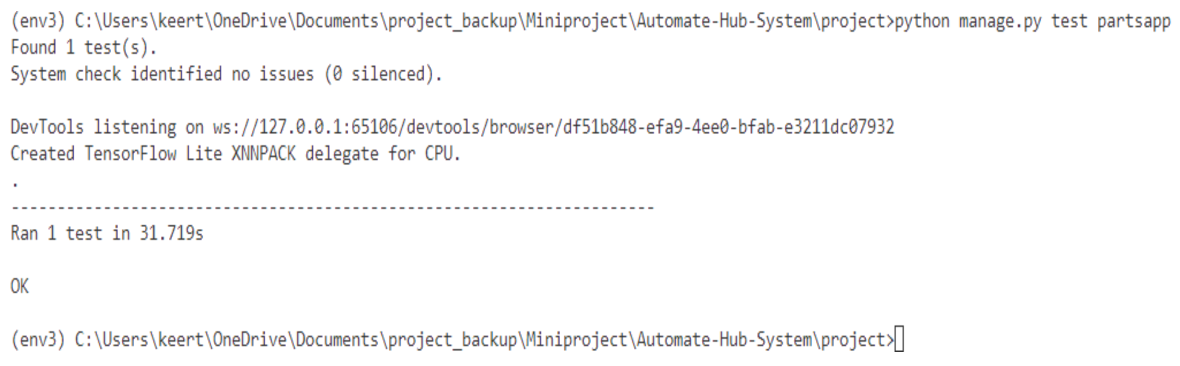


```python
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

class LoginAndNavigationTest(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.get("http://127.0.0.1:8000/login")
        time.sleep(5)

    def test_login_successful(self):
        # Login process (similar to your previous test)
        username_input = self.driver.find_element(By.ID, "username")
        password_input = self.driver.find_element(By.ID, "password")
        login_button = self.driver.find_element(By.ID, "submitBtn")

        username_input.send_keys("Sisira12")
        password_input.send_keys("Sisira@123")
        login_button.click()

        time.sleep(5)
        self.assertEqual(self.driver.current_url, 'http://127.0.0.1:8000/')

        # After successful login, navigate to booking page
        self.driver.get("http://127.0.0.1:8000/booking/")  # Replace with the actual booking URL
        time.sleep(5)

        # Fill in the booking details
        email_input = self.driver.find_element(By.ID, "email")
        driver_number_input = self.driver.find_element(By.ID, "driver_number")
        vehicle_number_input = self.driver.find_element(By.ID, "vehicle_number")
        service_branch_dropdown = self.driver.find_element(By.ID, "service_branch")
        vehicle_model_dropdown = self.driver.find_element(By.ID, "vehicle_model")
        service_type_radio = self.driver.find_element(By.XPATH, "//input[@name='service_type' and @value='free']")
        service_date_input = self.driver.find_element(By.ID, "service_date")
        submit_button = self.driver.find_element(By.XPATH, "//button[@name='submit']")
```



```python
        # Fill in the booking details
        email_input = self.driver.find_element(By.ID, "email")
        driver_number_input = self.driver.find_element(By.ID, "driver_number")
        vehicle_number_input = self.driver.find_element(By.ID, "vehicle_number")
        service_branch_dropdown = self.driver.find_element(By.ID, "service_branch")
        vehicle_model_dropdown = self.driver.find_element(By.ID, "vehicle_model")
        service_type_radio = self.driver.find_element(By.XPATH, "//input[@name='service_type' and @value='free']")
        service_date_input = self.driver.find_element(By.ID, "service_date")
        submit_button = self.driver.find_element(By.XPATH, "//button[@name='submit']")

        email_input.send_keys("Sisiramohan2000@gmail.com")
        driver_number_input.send_keys("9234567890")
        vehicle_number_input.send_keys("KL11234")
        service_branch_dropdown.send_keys("Trivandrum")
        vehicle_model_dropdown.send_keys("Dost+")
        service_type_radio.click()
        service_date_input.send_keys("5-12-2024")

        # Submit the form
        submit_button.click()

        time.sleep(10)
        # Add assertions for successful submission or subsequent page validation

    def tearDown(self):
        self.driver.close()

if __name__ == "__main__":
    unittest.main()
```

Screenshot


```
(env3) C:\Users\keert\OneDrive\Documents\project_backup\Miniproject\Automate-Hub-System\project>python manage.py test partsapp
Found 1 test(s).
System check identified no issues (0 silenced).

DevTools listening on ws://127.0.0.1:65106/devtools/browser/df51b848-efa9-4ee0-bfab-e3211dc07932
Created TensorFlow Lite XNNPACK delegate for CPU.
.
----------------------------------------------------------------------
Ran 1 test in 31.719s

OK

(env3) C:\Users\keert\OneDrive\Documents\project_backup\Miniproject\Automate-Hub-System\project>
```

Test report

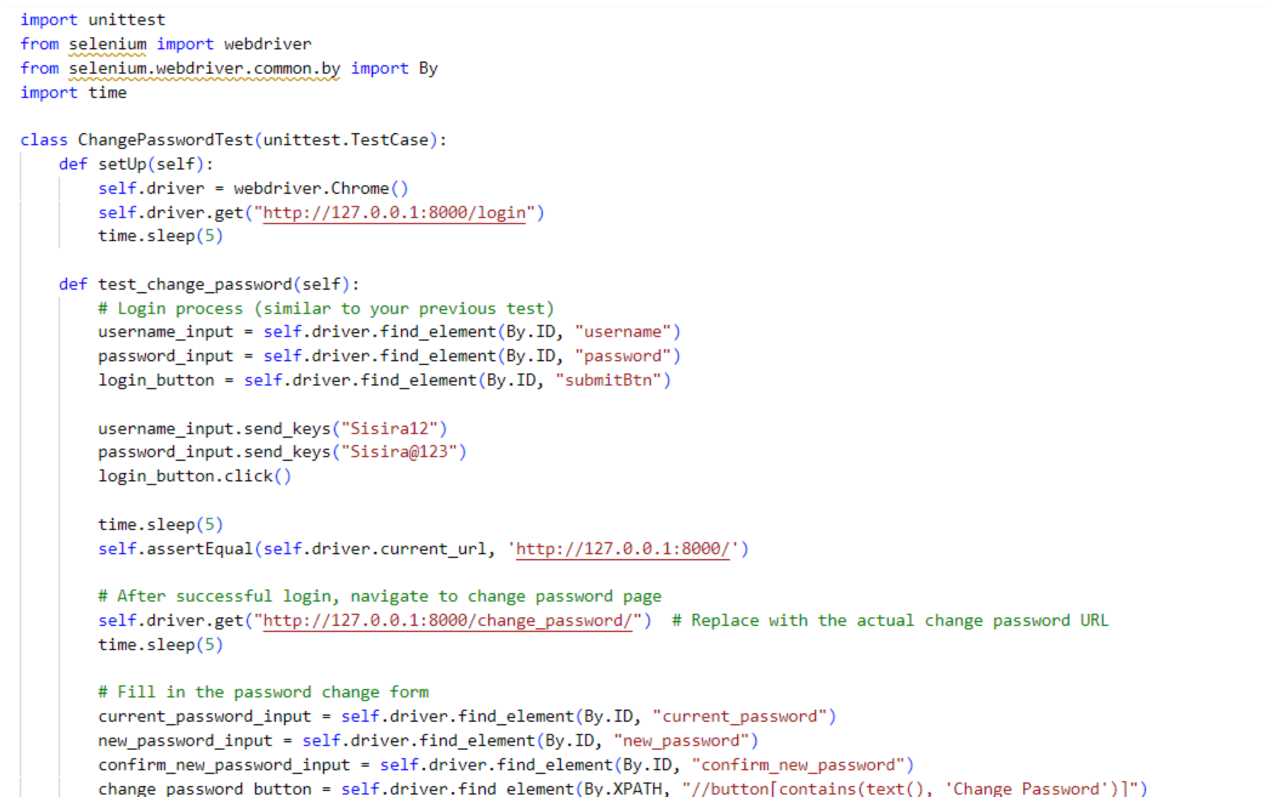
Test Case 2

Project Name: Commercial vehicle service and Spare parts Management					
Booking Test Case					
Test Case ID: Test_2			**Test Designed By:** Keerthi U		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 13/4/2024		
Module Name: booking page			**Test Executed By :** Rini Kurian		
Test Title : Verify booking details			**Test Execution Date:** 15/4/2024		
Description: Testing the booking					
Pre-Condition : Booking completed					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigation to Login page		Login Page should be displayed	Login Page displayed	Pass
2	Provide valid username	Username: Keerthi	User should be able to login	User logged in and navigated to dashboard	Pass
3	Provide valid password	Keerthi@1			
4	Click on Login Button				

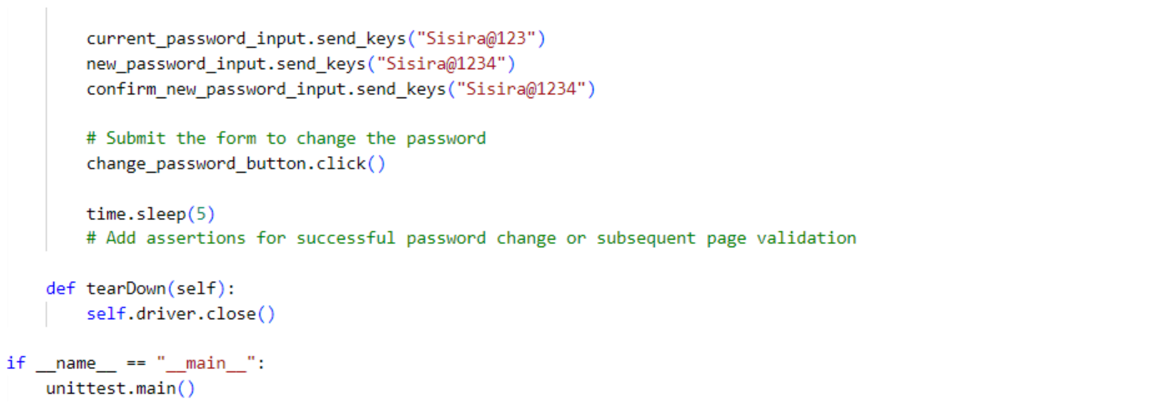
5	Navigation to Booking page	Username:keerthi Email: email Vehicleno:kl2345 Driverno:62354094 4 Service_type:free Vehicle_type:Dost+ Service_date: 4/4/2023 Submit	User should not be able in submit booking	Message for Booking confirm or reject	Pass
6	Provide null Booking details	Username:null Email: null Vehicleno:null Driverno:null Service_type:null Vehicle_type:null Service_date: null			
7	Click on submit In button	Submit			
Post-Condition: Booking successfully completed					

Test Case 3

Code

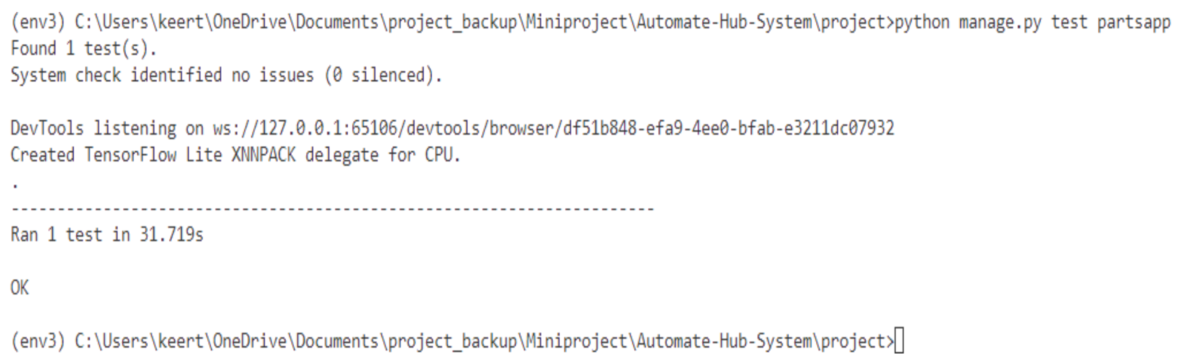


```
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

class ChangePasswordTest(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.get("http://127.0.0.1:8000/login")
        time.sleep(5)

    def test_change_password(self):
        # Login process (similar to your previous test)
        username_input = self.driver.find_element(By.ID, "username")
        password_input = self.driver.find_element(By.ID, "password")
        login_button = self.driver.find_element(By.ID, "submitBtn")

        username_input.send_keys("Sisira12")
        password_input.send_keys("Sisira@123")
        login_button.click()

        time.sleep(5)
        self.assertEqual(self.driver.current_url, 'http://127.0.0.1:8000/')

        # After successful login, navigate to change password page
        self.driver.get("http://127.0.0.1:8000/change_password/")  # Replace with the actual change password URL
        time.sleep(5)

        # Fill in the password change form
        current_password_input = self.driver.find_element(By.ID, "current_password")
        new_password_input = self.driver.find_element(By.ID, "new_password")
        confirm_new_password_input = self.driver.find_element(By.ID, "confirm_new_password")
        change_password_button = self.driver.find_element(By.XPATH, "//button[contains(text(), 'Change Password')]")
```



```
        current_password_input.send_keys("Sisira@123")
        new_password_input.send_keys("Sisira@1234")
        confirm_new_password_input.send_keys("Sisira@1234")

        # Submit the form to change the password
        change_password_button.click()

        time.sleep(5)
        # Add assertions for successful password change or subsequent page validation

    def tearDown(self):
        self.driver.close()

if __name__ == "__main__":
    unittest.main()
```

Screenshot



```
(env3) C:\Users\keert\OneDrive\Documents\project_backup\Miniproject\Automate-Hub-System\project>python manage.py test partsapp
Found 1 test(s).
System check identified no issues (0 silenced).

DevTools listening on ws://127.0.0.1:65106/devtools/browser/df51b848-efa9-4ee0-bfab-e3211dc07932
Created TensorFlow Lite XNNPACK delegate for CPU.
.
----------------------------------------------------------------------
Ran 1 test in 31.719s

OK

(env3) C:\Users\keert\OneDrive\Documents\project_backup\Miniproject\Automate-Hub-System\project>
```

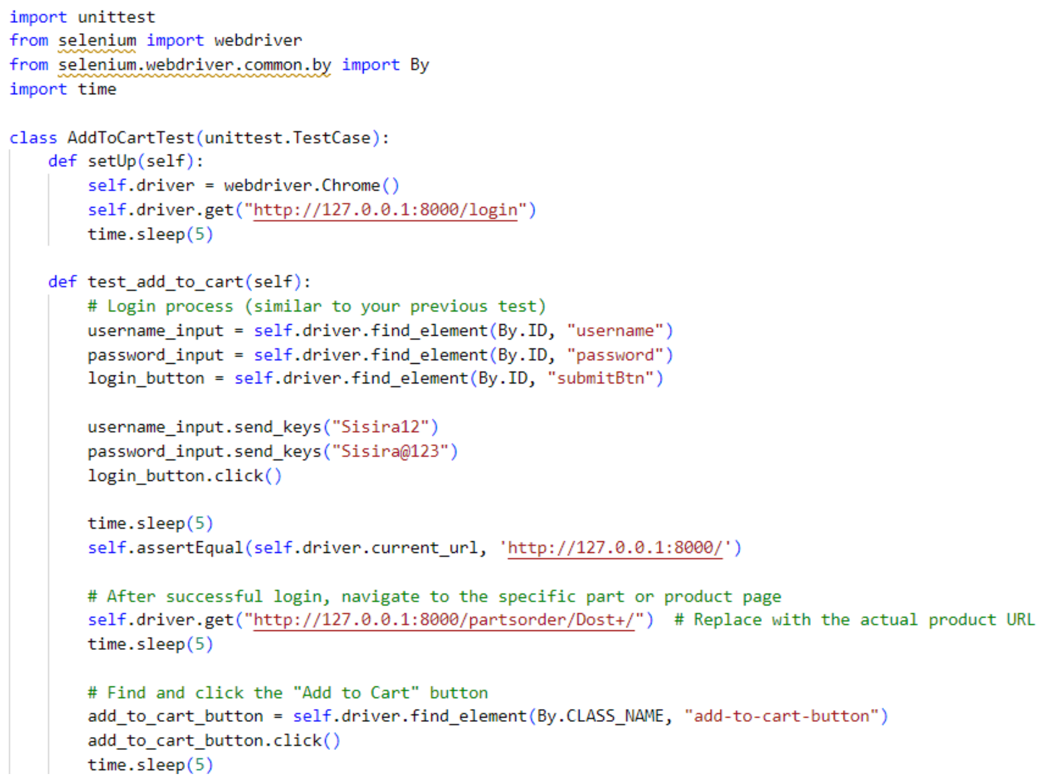
Test report

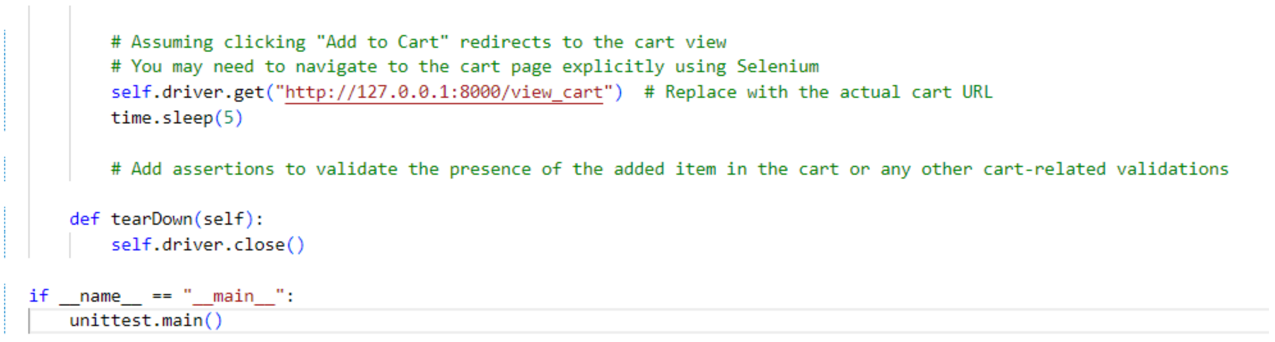
Test Case 3

Project Name: Commercial vehicle service and Spare parts Management					
Change password Test Case					
Test Case ID: Test_3			**Test Designed By:** Keerthi u		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 12/4/2024		
Module Name: Change password page			**Test Executed By :** RINI KURIAN		
Test Title : Verify change password			**Test Execution Date:** 15/4/2024		
Description: Testing the change to password					
Pre-Condition : User has current password and new password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass / Fail)**
1	Navigation to Login page		Login Page should be displayed	Login Page displayed	Pass
2	Provide valid username	Username: Keerthi	User should be able to login	User logged in and navigated to dashboard	Pass
3	Provide valid password	Keerthi@1			
4	Click on Login Button				
5	Navigation to change password page	Current password :Keerthi@12 New password: Keerthi@123 Confirmpassword: Keerthi@123	User should be able in submit change password	Message for password	Pass
6	Provide null password details	Current password New password:keerthi @123 Confirmpassword: Keerthi@1234			
7	Click on submit In button	Submit			
Post-Condition: Password changed successfully					

Test Case 4:

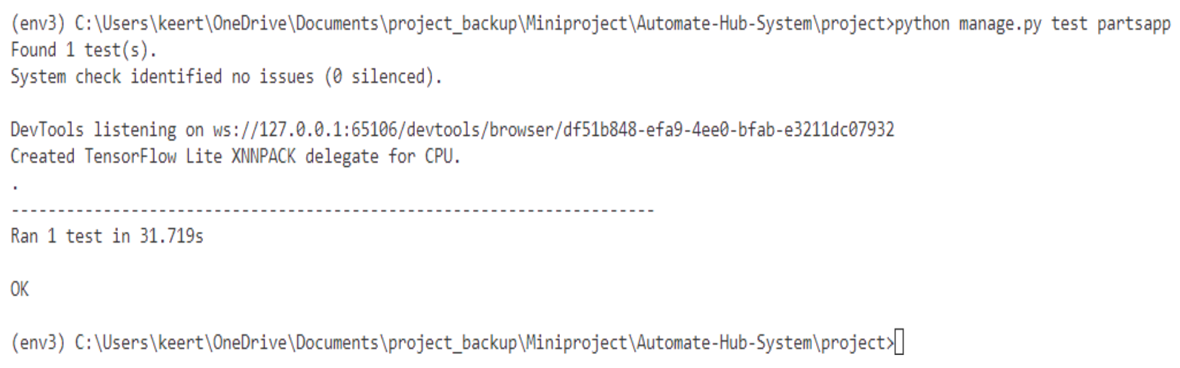
Code





```python
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

class AddToCartTest(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.get("http://127.0.0.1:8000/login")
        time.sleep(5)

    def test_add_to_cart(self):
        # Login process (similar to your previous test)
        username_input = self.driver.find_element(By.ID, "username")
        password_input = self.driver.find_element(By.ID, "password")
        login_button = self.driver.find_element(By.ID, "submitBtn")

        username_input.send_keys("Sisira12")
        password_input.send_keys("Sisira@123")
        login_button.click()

        time.sleep(5)
        self.assertEqual(self.driver.current_url, 'http://127.0.0.1:8000/')

        # After successful login, navigate to the specific part or product page
        self.driver.get("http://127.0.0.1:8000/partsorder/Dost+/")  # Replace with the actual product URL
        time.sleep(5)

        # Find and click the "Add to Cart" button
        add_to_cart_button = self.driver.find_element(By.CLASS_NAME, "add-to-cart-button")
        add_to_cart_button.click()
        time.sleep(5)

        # Assuming clicking "Add to Cart" redirects to the cart view
        # You may need to navigate to the cart page explicitly using Selenium
        self.driver.get("http://127.0.0.1:8000/view_cart")  # Replace with the actual cart URL
        time.sleep(5)

        # Add assertions to validate the presence of the added item in the cart or any other cart-related validations

    def tearDown(self):
        self.driver.close()

if __name__ == "__main__":
    unittest.main()
```

Screenshot



```
(env3) C:\Users\keert\OneDrive\Documents\project_backup\Miniproject\Automate-Hub-System\project>python manage.py test partsapp
Found 1 test(s).
System check identified no issues (0 silenced).

DevTools listening on ws://127.0.0.1:65106/devtools/browser/df51b848-efa9-4ee0-bfab-e3211dc07932
Created TensorFlow Lite XNNPACK delegate for CPU.
.
----------------------------------------------------------------------
Ran 1 test in 31.719s

OK

(env3) C:\Users\keert\OneDrive\Documents\project_backup\Miniproject\Automate-Hub-System\project>
```

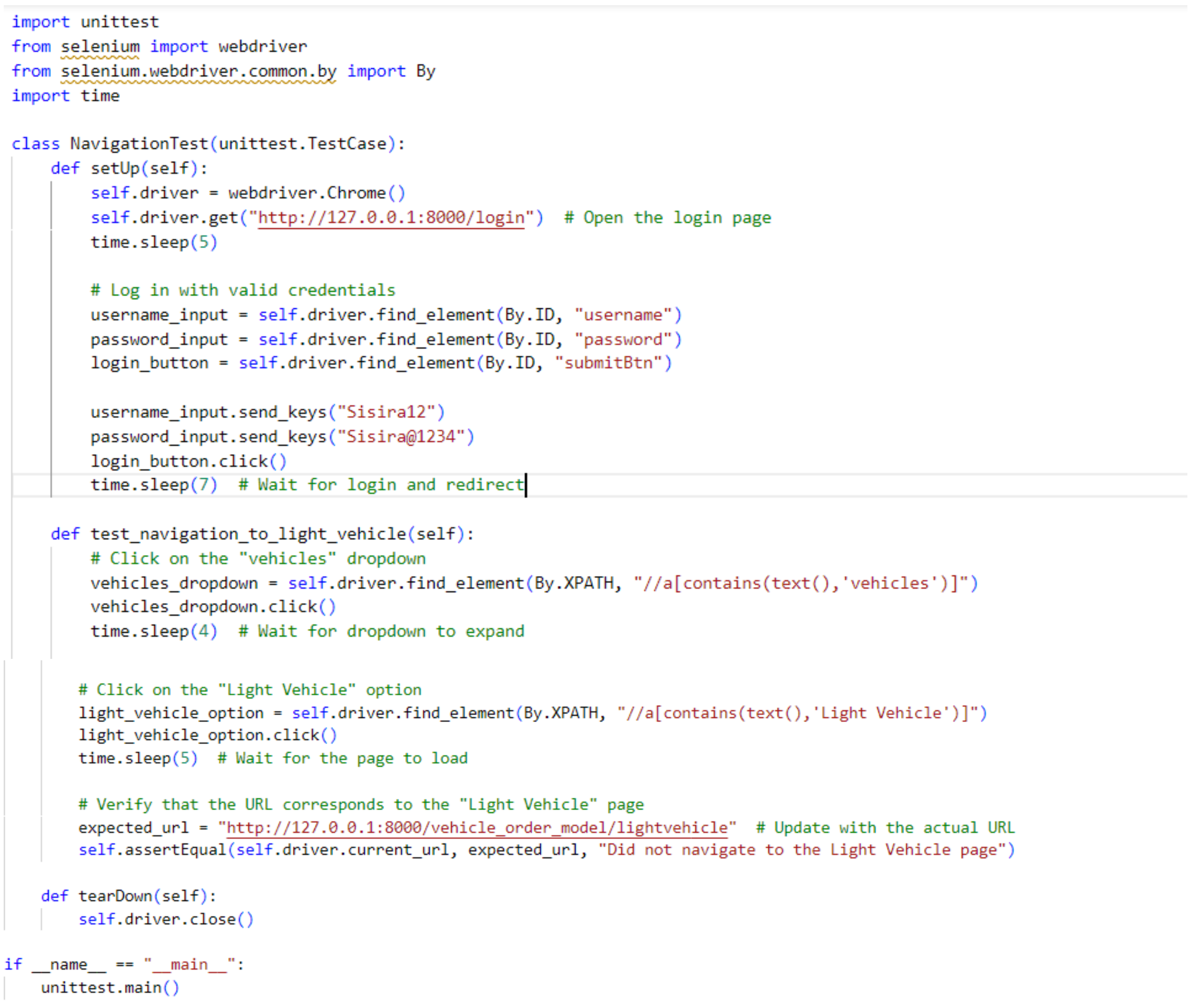
Test report

Test Case 4

Project Name: Commercial vehicle service and Spare parts Management					
Cart Test Case					
Test Case ID: Test_4			**Test Designed By: Keerthi U**		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 13/4/2024		
Module Name: Parts page			**Test Executed By :** Rini Kurian		
Test Title : Verify parts add to cart			**Test Execution Date:** 15/4/2024		
Description: Testing the Cart page					
Pre-Condition : Parts added to cart					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass / Fail)**
1	Navigation to Login page		Login Page should be displayed	Login Page displayed	Pass
2	Provide valid username	Username: Keerthi	User should be able to login	User logged in and navigated to dashboard	Pass
3	Provide valid password	Keerthi@1			
4	Click on Login Button				
5	Navigation to partsadd page	Product add to cart	User should be able to click the product	Message for Parts add to cart	Pass
6	Not select the product	Product not add			
7	Click on submit In button	Submit			
Post-Condition: Parts add_to_cart Successfully					

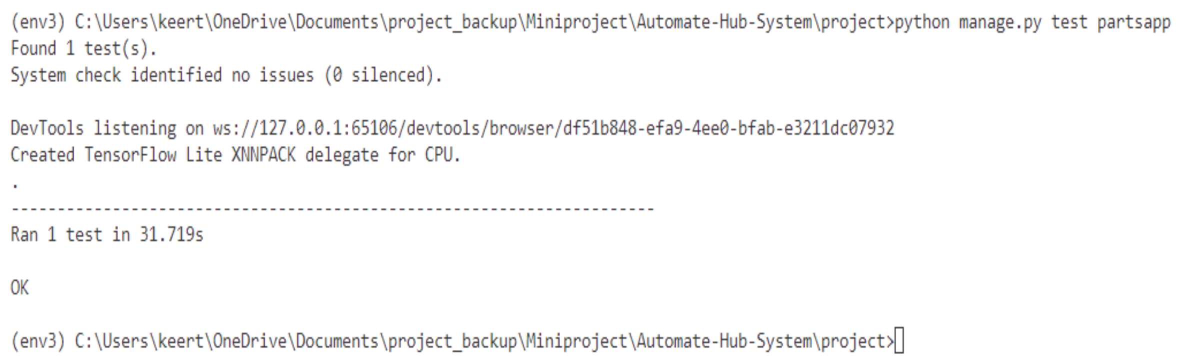
Test Case 5:

Code



```
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

class NavigationTest(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.get("http://127.0.0.1:8000/login")  # Open the login page
        time.sleep(5)

        # Log in with valid credentials
        username_input = self.driver.find_element(By.ID, "username")
        password_input = self.driver.find_element(By.ID, "password")
        login_button = self.driver.find_element(By.ID, "submitBtn")

        username_input.send_keys("Sisira12")
        password_input.send_keys("Sisira@1234")
        login_button.click()
        time.sleep(7)  # Wait for login and redirect

    def test_navigation_to_light_vehicle(self):
        # Click on the "vehicles" dropdown
        vehicles_dropdown = self.driver.find_element(By.XPATH, "//a[contains(text(),'vehicles')]")
        vehicles_dropdown.click()
        time.sleep(4)  # Wait for dropdown to expand

        # Click on the "Light Vehicle" option
        light_vehicle_option = self.driver.find_element(By.XPATH, "//a[contains(text(),'Light Vehicle')]")
        light_vehicle_option.click()
        time.sleep(5)  # Wait for the page to load

        # Verify that the URL corresponds to the "Light Vehicle" page
        expected_url = "http://127.0.0.1:8000/vehicle_order_model/lightvehicle"  # Update with the actual URL
        self.assertEqual(self.driver.current_url, expected_url, "Did not navigate to the Light Vehicle page")

    def tearDown(self):
        self.driver.close()

if __name__ == "__main__":
    unittest.main()
```

Screenshot



```
(env3) C:\Users\keert\OneDrive\Documents\project_backup\Miniproject\Automate-Hub-System\project>python manage.py test partsapp
Found 1 test(s).
System check identified no issues (0 silenced).

DevTools listening on ws://127.0.0.1:65106/devtools/browser/df51b848-efa9-4ee0-bfab-e3211dc07932
Created TensorFlow Lite XNNPACK delegate for CPU.
.
----------------------------------------------------------------------
Ran 1 test in 31.719s

OK

(env3) C:\Users\keert\OneDrive\Documents\project_backup\Miniproject\Automate-Hub-System\project>
```

Test report

Test Case 5

Project Name: Commercial vehicle service and Spare parts Management	
Vehicle Test Case	
Test Case ID: Test_5	**Test Designed By: Keerthi U**
Test Priority(Low/Medium/High): High	**Test Designed Date:** 13/4/2024
Module Name: Vehicle page	**Test Executed By :** Rini Kurian
Test Title : Verify vehicle add to Booking	**Test Execution Date:** 15/4/2024
Description: Testing the vehicle page	
Pre-Condition :vehicle added to booking page	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass / Fail)**
1	Navigation to Login page		Login Page should be displayed	Login Page displayed	Pass
2	Provide valid username	Username: Keerthi	User should be able to login	User logged in and navigated to dashboard	Pass
3	Provide valid password	Keerthi@1			
4	Click on Login Button				
5	Navigation to vehicle add page	Vehicle add to booking	User should be able to click the vehicle	Message for vehicle add to booking	Pass
6	Not select the vehicle	Vehicle not add			
7	Click on submit In button	Submit			

Post-Condition: vehicle add_to_booking page Successfully

CHAPTER 6
IMPLEMENTATION

6.1INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It isprimarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one. At this stage the main work load, the greatest upheaval and the major impact onthe existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion. Implementation includes all those activities that takeplace to convert from the existing system to the new system. The new system may be a totally new,replacing an existing manual or automated system or it may be a modification to an existing system.Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The morecomplex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover. The implementation state involves the following tasks: Careful planning. Investigationof system and constraints. Design of methods to achieve the changeover

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, tothe satisfaction of the intended uses and the operation of the system. In many organizations someonewho will not be operating it, will commission the software development project. In the initial stagepeople doubt about the software but we have to ensure that the resistance does not build up, as onehas to make sure that: The active user must be aware of the benefits of using the new system. Theirconfidence in the software is built up. Proper guidance is imparted to the user so that he is comfortable in using the application. Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use ofthe new system such as the screen flow, screen design type of help on the screen, type of errorswhile entering the data, the corresponding validation check at each entry and the ways to correctthe date entered. It should thencover information needed by the specific user/ group to use thesystem or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes"

6.2.4 Hosting

Hosting a website entails making it accessible to users on the internet by storing its files and data on a server. This process involves several steps, beginning with choosing a suitable hosting provider that meets your website's requirements for storage, bandwidth, uptime, and support. Once a hosting provider is selected, you typically register a domain name for your website, which serves as its unique address.After domain registration, you set up a hosting account by selecting a plan, providing payment information, and creating an account with the hosting provider. Subsequently, you upload your website files to the server using FTP or a web-based file manager. Organizing these files properly, including HTML, CSS, JavaScript, images, and other assets, ensures smooth functioning. Configuring domain settings to point to the hosting server via DNS records is the next step. Thorough testing of the website for issues such as broken links or formatting problems is crucial before making it live. Once everything is set, you launch the website by updating hosting account settings or additional configurations as needed.Post-launch, regular monitoring of performance, security, and uptime is essential. Many hosting providers offer tools and analytics to help track website metrics and address any arising issues. Consistently updating website content and software ensures security and relevance to the audience, ensuring a seamless online presence.

6.2.4.1 AMAZON ELASTIC COMPUTE CLOUD (EC2)

Amazon Elastic Compute Cloud (EC2) is a core component of Amazon Web Services (AWS), offering users the ability to rent virtual servers, or instances, in the cloud. It allows for flexible scaling of computing resources based on demand, with a variety of instance types optimized for different workloads. EC2 provides full control over server instances, supports various operating systems, and offers features for security, scalability, and cost-effectiveness. Users can easily manage their instances through the AWS Management Console or APIs. Overall, AWS EC2 enables businesses to run applications and services in a reliable, scalable, and cost-efficient manner in the cloud.

Procedure for hosting a website on Amazon EC2:

- **Step1: Create an AWS Account**: Start by registering for an AWS account if you haven't already done so. Once logged in to the AWS Management Console, proceed to the EC2 dashboard.
- **Step2: Launch an EC2 Instance:** Click on the "Launch Instance" button to initiate a new EC2 instance. Select the Ubuntu AMI (Amazon Machine Image) and choose an appropriate instance type according to your project's specifications.
- **Step3: Configure Instance Settings:** Customize instance parameters such as quantity, networking configurations, and storage preferences. Include storage volumes to securely store your project files and data.
- **Step4: Set Up Security Group:** Establish a new security group or utilize an existing one to define firewall regulations. Ensure that ports 22 (SSH) for remote access and 8000 (or any other required port for your application) for web traffic are accessible.
- **Step5: Launch Instance:** Review the instance setup and commence the EC2 instance launch process. Opt for or generate a key pair for SSH access.
- **Step6: Connect to Your Instance:** Upon instance launch, establish an SSH connection.

Utilize the public IP address or DNS name of your instance along with the key pair to establish a secure connection.

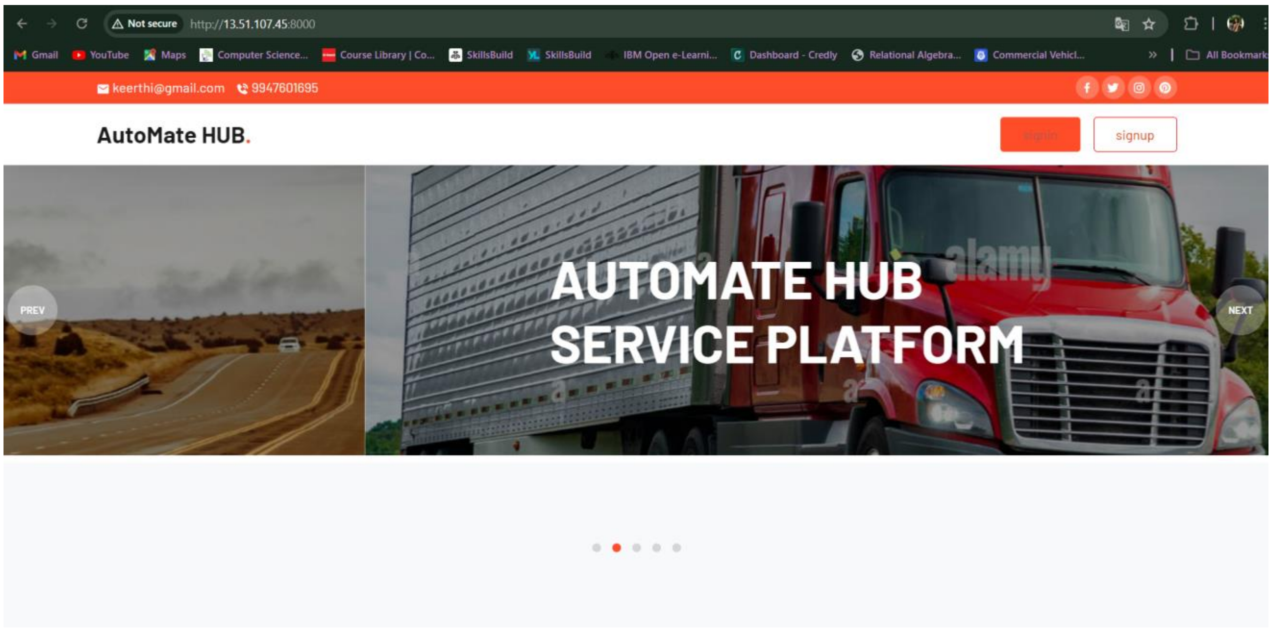
- **Step7: Clone Project Repository:** Install Git on the EC2 instance and clone the EduSphere Fusion project repository from your Git repository using the git clone command.
- **Step8: Install Python and Django:** Deploy Python and Django on the EC2 instance. Utilize the apt package manager for Python installation and pip to install Django along with any other necessary Python packages.
- **Step9: Install Dependencies:** Install additional packages and dependencies essential for running the website, including database drivers and Django extensions.
- **Step10: Configure Django Settings:** Update the Django settings file (settings.py) with the required database configuration, static file settings, and other project-specific configurations.
- **Step11: Run Django Server:** Initiate the Django development server by executing the command python manage.py runserver 0.0.0.0:8000. This command operates the server on port 8000, accessible on all network interfaces.
- **Step12: Test Your Website:** Access a web browser and navigate to the public IP address or DNS name of your EC2 instance followed by the designated port number (<public_ip>:8000). Confirm that your website is accessible and operates correctly.
- **Step13: Domain Name Configuration (Optional):** If you own a domain name, configure the DNS settings to point to the public IP address of your EC2 instance.
- **Step14: SSL/TLS Certificate Setup (Optional):** Enable HTTPS for your website by setting up an SSL/TLS certificate using AWS Certificate Manager or a third-party provider.
- **Step15: Monitor Your EC2 Instance:** Employ AWS CloudWatch or other monitoring utilities to oversee the performance, security, and uptime of your EC2 instance. Regularly update your

Hosted Website: Amazon EC2

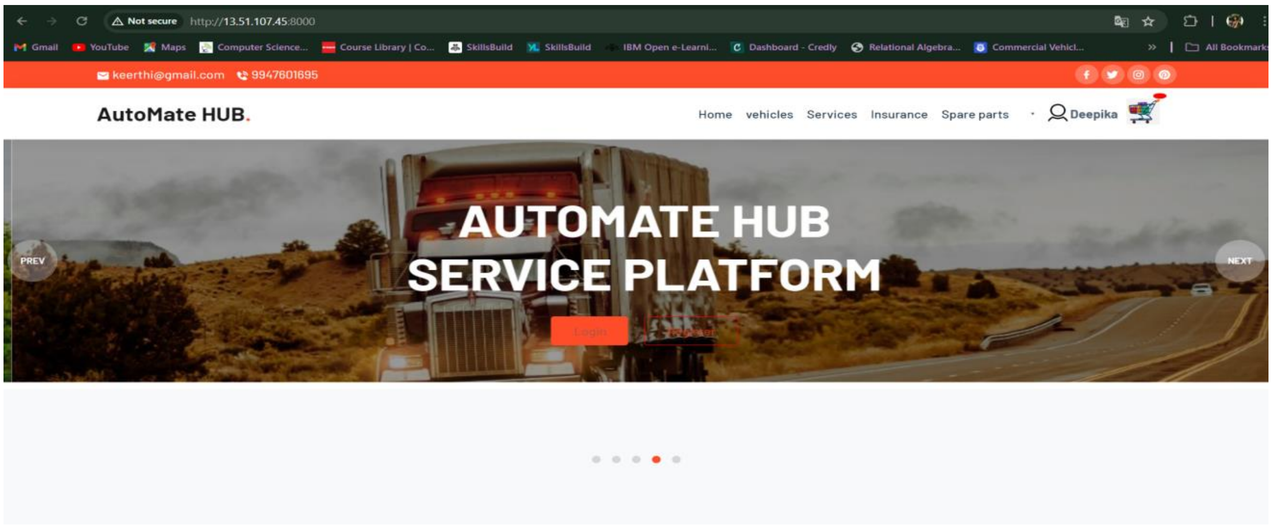
Hosted Link: http://13.51.107.45:8000/

Screenshot

Login page


Not secure http://13.51.107.45:8000
keerthi@gmail.com 9947601695
AutoMate HUB.
signup
PREV
NEXT
AUTOMATE HUB
SERVICE PLATFORM

Dashboard


Not secure http://13.51.107.45:8000
keerthi@gmail.com 9947601695
AutoMate HUB.
Home vehicles Services Insurance Spare parts
Deepika
PREV
NEXT
AUTOMATE HUB
SERVICE PLATFORM

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In the realm of commercial vehicle sales and services, our project stands as a testament to innovation, efficiency, and customer-centricity. Through meticulous planning, relentless dedication, and collaborative efforts, we have successfully developed a comprehensive platform that addresses the diverse needs of commercial vehicle owners and operators.Our journey has been marked by significant achievements, from the seamless integration of vehicle sales, servicing, spare parts procurement, and insurance management to the implementation of user-friendly interfaces and efficient delivery systems. By prioritizing customer satisfaction and operational excellence, we have garnered trust, loyalty, and recognition within the industry. As we conclude this phase of our project, we do so with a sense of pride in our accomplishments and gratitude for the support of our team members, partners, and stakeholders. However, our work is far from complete. The future beckons with opportunities for further growth, innovation, and impact.Looking ahead, we remain committed to pushing the boundaries of possibility, embracing emerging technologies, diversifying our services, expanding our reach, and championing sustainability. Our vision extends beyond mere transactional interactions; it encompasses a holistic approach to enhancing the lives and livelihoods of our customers while contributing to the advancement of the commercial vehicle industry as a whole.

7.2 FUTURE SCOPE

While this project has achieved significant milestones, there are numerous opportunities for further exploration and enhancement in the future. One avenue for future development lies in the integration of emerging technologies. By incorporating advancements such as artificial intelligence, machine learning, and blockchain, we can enhance the capabilities and efficiency of our platform. This includes implementing predictive analytics for personalized recommendations, leveraging AI-powered chatbots for customer support, and exploring blockchain for secure transactions and data management.Additionally, there is potential for geographic expansion and market diversification. Scaling the platform to new regions or targeting specific niche markets can broaden our reach and tap into previously untapped customer segments. Moreover, partnerships and collaborations with industry stakeholders, including manufacturers, suppliers, and service providers, offer opportunities for mutual growth and innovation. By forging strategic alliances, we can leverage complementary strengths and resources to create value-added solutions for our customers.Furthermore, ongoing research and development will be crucial for staying ahead of evolving trends and customer needs. This includes conducting market analyses, gathering customer feedback, and staying abreast of technological advancements. By continuously refining our platform and offerings based on insights gained from research, we can ensure that we remain competitive and relevant in the dynamic landscape of commercial vehicle sales and services.

In conclusion, the future is ripe with possibilities for further innovation, expansion, and impact. By embracing emerging technologies, exploring new markets, fostering strategic partnerships, and committing to ongoing research and development, we can position ourselves at the forefront of the industry and continue to deliver value to our customers in the years to come.

CHAPTER 8
BIBLIOGRAPHY

REFERENCES:

- PankajJalote, “Software engineering: a precise approach”
- Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009
- Ken Schwaber, Mike Beedle, Agile Software Development with Scrum , Pearson(2008)
- Roger S Pressman, “Software Engineering”
- IEEE Std 1016 Recommended Practice for Software Design Descriptions

WEBSITES:

- https://w3schools.com
- https://geeksforgeeks.com
- https://jquery.com

https://chat.openai.com/chat

CHAPTER 9
APPENDIX

9.1 Sample Code

Login.html

```
{% load static %}
{% load socialaccount %}
<!DOCTYPE html>
<html lang="en">
<head>
<link rel="icon" type="image/x-icon" href="favicon.ico">

<!-- Bootstrap CSS -->
<link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
<link rel="stylesheet" href="{% static 'css/owl.carousel.min.css' %}">
<link rel="stylesheet" href="{% static 'css/owl.theme.default.min.css' %}">
<link href='https://unpkg.com/boxicons@2.1.1/css/boxicons.min.css' rel='stylesheet'>
<link rel="stylesheet" href="{% static 'css/style.css' %}">
<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
<!-- Design by foolishdeveloper.com -->
{% comment %} <title>Glassmorphism login Form Tutorial in html css</title>

<link rel="preconnect" href="https://fonts.gstatic.com">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css">
<link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;600&display=swap"
rel="stylesheet">
<!--Stylesheet--> {% endcomment %}
<style>
body {
/* Set background properties for the entire page */
background-image: url({% static 'img/Untitled_design.jpg' %}); /* Replace
'path_to_your_image.jpg' with the actual path to your image */
background-size: cover; /* Adjust the background size */
/* Add other necessary properties */
}

/* Style for the form */
#myForm {
/* Adjust form position and other properties */
/* Example styles */
width: 30%;
margin: 0 auto;
background-color: rgba(255, 255, 255, 0.8); /* Background color for better readability */
padding: 20px;
border-radius: 10px;
}
form {
margin: 20px;
border: 2px solid #007bff; /* Add a blue border around the form */
padding: 20px;
border-radius: 10px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.2); /* Add a slight shadow */
width: 40%; /* Reduce the width to 60% of the parent container */
margin: 0 auto;
{% comment %} background-image: url({% static 'img/bg_banner1.jpg' %}); /* Add your image
path here */ {% endcomment %}
background-size: cover; /* Make the background cover the entire form */
background-repeat: no-repeat;
}

label {
font-weight: bold;
}

input[type="text"],
input[type="email"],
input[type="password"],
input[type="tel"],
select {
width: 100%;
padding: 5px;
margin: 5px 0;
border: 1px solid #ccc;
border-radius: 5px;
}

button {
background-color: #007bff;
color: white;
padding: 10px 20px;
border: none;
border-radius: 5px;
cursor: pointer;

}

/* Add styles for error messages or other custom styles */
span {
color: red;
}
</style>

</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-white sticky-top">
<div class="container">
<a class="navbar-brand" href="#">Sparx motors<span class="dot">.</span></a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
```

```
<div class="collapse navbar-collapse" id="navbarNav">
<ul class="navbar-nav ms-auto">
{% if user.is_authenticated %}
<li class="nav-item">
<a class="nav-link" href="#home">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#about"></a>
</li>
<li class="nav-item">
<a class="nav-link" href="#services">vehicles</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#portfolio">Services</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#team">Spare parts</a>
</li>
{% else %}
<li class="nav-item">
<a class="nav-link" href="{% url 'login' %}">signin</a>
</li>

<li class="nav-item">
<a class="btn btn-brand" href="{% url 'signup' %}">signup</a>
</li>
{% endif %}
</ul>

</div>
</div>
</nav>
{% if messages %}
<div class="container">
```

```
{% for message in messages %}
<div class="alert alert-{{ message.tags }} alert-dismissible fade show" role="alert">
{{ message }}
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
{% endfor %}
</div>
{% endif %}
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<form action="" method="POST">
{% csrf_token %}
<h2 align="center"><b>Login</b></h2>

<label for="username">Username</label>
<input type="text" name="username" placeholder="Username" id="username" required>
<span id="usernameError" style="color: red;"></span>

<label for="password">Password</label>
<input type="password" name="password" placeholder="Password" id="password" required>
<span id="passwordError" style="color: red;"></span>

<button type="submit" id="submitBtn" style="background-color: #007bff; color:
white;">Login</button>
{% if error_message %}
<div style="color: red;">
{{ error_message }}
</div>
{% endif %}

<a href="{% provider_login_url 'google' %}" class="google-login-button"><i class="fab fa-
google"></i> Google</a>
```

```
<p>Don't have an account? <a href="{% url 'signup' %}">Sign Up</a></p>
<p class="forgot-password"><a href="{% url 'password_reset' %}">Forgot Password?</a></p>
</div>

</form>

</body>
</html>
```

Vehicleorder.html

```
<!DOCTYPE html>
  <html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product Booking Dashboard</title>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.0/css/all.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css">
    <style>
      /* Reset some default styles */
      body, h1, h2, p, ul, li {
        margin: 0;
        padding: 0;
      }

      body {
        font-family: 'Arial', sans-serif;
         display: flex;
        background-color: #f2f2f2;
        margin: 0;
      }
```

```
.sidebar {
   width: 500px;
   height: 200vh;
   background-color: #543c4b;
   padding-top: 100px;
   color: white;
   box-shadow: 2px 0 5px rgba(0, 0, 0, 0.1);
}

.sidebar ul {
   list-style-type: none;
   padding: 0;
}

.sidebar li {
   padding: 15px;
   text-align: center;
   transition: background-color 0.3s;
}

.sidebar a {
   text-decoration: none;
   color: white;
   display: block;
}

.sidebar a:hover {
   background-color: #fc3503;
}

{% comment %} .main-content {
   flex: 1;
   padding: 20px;
```

```
        }
    {% endcomment %}
        /* Vehicle card styles */
        .vehicle-list {
            display: flex;
            flex-wrap: wrap;
            justify-content: space-around;
            margin-top: 20px;
        }

        .vehicle-card {
            width: 300px;
            margin: 20px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            border-radius: 8px;
            overflow: hidden;
            transition: transform 0.3s;
            background-color: #fff;
        }

        .vehicle-card:hover {
            transform: scale(1.05);
        }

        .vehicle-image {
            position: relative;
            overflow: hidden;
        }

        .vehicle-image img {
            width: 100%;
            height: auto;
            transition: transform 0.3s;
        }
```

```
.vehicle-card:hover .vehicle-image img {
    transform: scale(1.1);
}

.out-of-stock-overlay {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(255, 0, 0, 0.5);
    color: white;
    display: flex;
    align-items: center;
    justify-content: center;
    font-weight: bold;
}

.vehicle-info {
    padding: 15px;
}

.vehicle-info h2 {
    font-size: 1.2em;
    margin-bottom: 10px;
    color: #333;
}

.vehicle-info p {
    margin-bottom: 8px;
    color: #555;
}
```

```
.add-to-cart-form {
   margin-top: 10px;
}

.add-to-cart-button {
   padding: 8px 16px;
   background-color: #3498db;
   color: white;
   border: none;
   cursor: pointer;
   transition: background-color 0.3s;
}

.add-to-cart-button:hover {
   background-color: #2980b9;
}

.view-more-button {
   margin-top: 10px;
}

.view-more-button a {
   text-decoration: none;
   color: #fff;
   padding: 8px 16px;
   border: 1px solid #fff;
   border-radius: 4px;
   transition: background-color 0.3s, color 0.3s;
}

.view-more-button a:hover {
   background-color: #fff;
   color: #333;
}
```

```
 {% comment %} .main-content {
    flex: 1;
    padding: 80px 20px 20px; /* Adjust top padding to match header height */
} {% endcomment %}
.filter-group {
    margin-bottom: 15px;
}

label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
}

select,
input {
    width: 90%;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-sizing: border-box;
    margin-bottom: 10px;
}

button {
    background-color: #4caf50;
    color: #fff;
    padding: 10px 15px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}
.wishlist-button {
    background: none;
    border: none;
    cursor: pointer;
}



.vehicle-card {
    width: 280px;
    margin: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    border-radius: 10px;
    overflow: hidden;
    transition: transform 0.3s;
    background-color: #fff;
    cursor: pointer;
}

.vehicle-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.1);
}

.vehicle-image {
    overflow: hidden;
    height: 200px;
}

.vehicle-image img {
    width: 100%;
    height: 100%;
    object-fit: cover;
    transition: transform 0.3s;
}

.vehicle-card:hover .vehicle-image img {
    transform: scale(1.1);
}

.vehicle-info {
    padding: 15px;
}

.vehicle-info h2 {
    font-size: 1.2em;
    margin-bottom: 10px;
    color: #333;
}

.vehicle-info p {
    margin-bottom: 8px;
    color: #555;
}

/* Button styles */
.action-button {
    padding: 10px 20px;
    background-color: #3498db;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s;
    text-decoration: none;
}
```

```
        .action-button:hover {
            background-color: #2980b9;
        }
        .vehicle-info p i {
            margin-right: 5px; /* Add spacing between the icon and text */
            color: #3498db; /* Change the color of the icon */
            font-size: 16px; /* Adjust the size of the icon */
        }
        .wishlist-icon {
            color: red; /* Change the color to pink */
            font-size: 24px; /* Increase the size of the icon */
        }
    </style>
    </style>
</head>
<body>
    <div class="sidebar">
        <form action="" method="GET">
            <h3>Filter Options</h3>

            <div class="filter-group">
                <label for="vehicle_usage">Vehicle Usage:</label>
                <select name="vehicle_usage" id="vehicle_usage">
                    <option value="goodscarriers">Goods Carriers</option>
                    <option value="passenger">Passenger</option>
                </select>
            </div>

            <div class="filter-group">
                <label for="vehicle_application">Vehicle Application:</label>
                <select name="vehicle_application" id="vehicle_application">
                    <option value="buildingmaterial">Building Material</option>
                    <option value="constructionhardware">Construction and Hardware</option>
                    <option value="fishery">Fishery</option>
                    <option value="fruits_vegetables">Fruits and Vegetables</option>
                    <option value="industrial_goods">Industrial Goods</option>
                    <option value="lpg_distribution">LPG Distribution</option>
                    <option value="school_transport">School Transport</option>
                    <option value="tours_travels">Tours and Travels</option>
                </select>
            </div>

            <div class="filter-group">
                <label for="fuel_type">Fuel Type:</label>
                <select name="fuel_type" id="fuel_type">
                    <option value="petrol">Petrol</option>
                    <option value="diesel">Diesel</option>
                    <option value="cng">CNG</option>
                    <option value="electric">Electric</option>
                </select>
            </div>

            <div class="filter-group">
                <label for="min_price">Min Price:</label>
                <input type="number" name="min_price" id="min_price" value="{{
        request.GET.min_price }}" placeholder="Min Price">
            </div>

            <div class="filter-group">
                <label for="max_price">Max Price:</label>
                <input type="number" name="max_price" id="max_price" value="{{
        request.GET.max_price }}" placeholder="Max Price">
            </div>



            <!-- Apply Filters button and closing form tag -->
            <button type="submit">Apply Filters</button>
```

```
            </form>

        </div>

        <div class="main-content">
            <div class="filter-group">
                <label for="image-upload-input">Upload Image:</label>
                <input type="file" id="image-upload-input" accept="image/*">
                <button id="image-search-button">Search</button>
                <img src="#" alt="Uploaded Image" id="uploaded-image" style="display: none;">
            </div>

            <section class="vehicle-list" id="vehicle-list">
                {% for vehicle in vehicles %}
                <div class="vehicle-card" data-vehicle-id="{{ vehicle.id }}">
                    <i class="far fa-heart wishlist-icon" data-vehicle-id="{{ vehicle.id }}"></i>
                    <div class="vehicle-image">
                        <img src="{{ vehicle.images.first.image.url }}" alt="{{ vehicle.name }} Image">
                        {% if vehicle.stock == 0 %}
                        <div class="out-of-stock-overlay">Out of Stock</div>
                        {% endif %}
                    </div>
                    <div class="vehicle-info">
                        <h2>{{ vehicle.name }}</h2>
                        <p><i class="fas fa-money-bill-wave"></i> Booking Amount: {{
vehicle.booking_amount }}</p>
                        <p><i class="fas fa-dollar-sign"></i> Price: {{ vehicle.price }}</p>
                        <p><i class="fas fa-gas-pump"></i> Fuel Type: {{ vehicle.fuel_type }}</p>
                        <p><i class="fas fa-cog"></i> Transmission: {{ vehicle.transmission_type }}</p>
                        <a href="{% url 'book_now' vehicle.id %}" class="action-button">Book Now</a>
                        <a href="{% url 'vehicle_details' vehicle.id %}" class="action-button">View
Details</a>
                    </div>
                </div>
                {% empty %}
                <p>No vehicles available.</p>
                {% endfor %}
            </section>

        </div>



                <script>
                    function viewDetails(vehicleId) {
                        // Redirect to the vehicle details page with the selected vehicle's ID
                        window.location.href = `/vehicle_details/${vehicleId}/`;
                    }
                </script>
            </section>
        </div>
        <!-- Include jQuery library -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script>
        $(document).ready(function () {
        // Intercept form submission
        $('#filter-form').submit(function (e) {
            // Prevent default form submission
            e.preventDefault();

            // Make an AJAX request to update the content
            $.ajax({
                type: 'GET',
                url: $(this).attr('action'),
                data: $(this).serialize(),
                success: function (data) {
                    // Update the content on success
                    // For simplicity, replace the entire content with the updated data
                    $('#vehicle-list').html(data);
                },
                error: function (error) {
                    console.log('Error:', error);
                }
            });
        });
        });
</script>

        <script>
            function searchVehicles() {
                var searchTerm = document.getElementById("search-bar").value.toLowerCase();
                var vehicleCards = document.getElementsByClassName("vehicle-card");

                for (var i = 0; i < vehicleCards.length; i++) {
                    var name = vehicleCards[i].getAttribute("data-name").toLowerCase();
                    var description = vehicleCards[i].getAttribute("data-description").toLowerCase();
                    var fuelType = vehicleCards[i].getAttribute("data-fuel").toLowerCase();
                    var transmissionType = vehicleCards[i].getAttribute("data-
transmission").toLowerCase();

                    if (name.includes(searchTerm) || description.includes(searchTerm) ||
fuelType.includes(searchTerm) || transmissionType.includes(searchTerm)) {
                        vehicleCards[i].style.display = "block";
                    } else {
                        vehicleCards[i].style.display = "none";
                    }
                }
            }

            document.getElementById("search-bar").addEventListener("input", function() {
                searchVehicles();
            });
```

```
</script>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
  $(document).ready(function () {
    // Function to handle toggling wishlist items
    $('.wishlist-icon').click(function(e) {
      e.preventDefault();
      var vehicleId = $(this).closest('.vehicle-card').data('vehicle-id');
      var icon = $(this);

      // Make AJAX request to toggle wishlist
      $.ajax({
        type: 'POST',
        url: '/wishlist/toggle/',
        data: {'vehicle_id': vehicleId},
        success: function(data) {
          // Check the status returned by the server
          if (data.status === 'added') {
            // Item added to wishlist, change icon to filled heart
            icon.removeClass('far').addClass('fas');
          } else if (data.status === 'removed') {
            // Item removed from wishlist, change icon to empty heart
            icon.removeClass('fas').addClass('far');
          }
        },
        error: function(error) {
          console.log('Error:', error);
        }
      });
    });
  });
</script>
<script>
  $(document).ready(function () {
    $('.wishlist-icon').click(function(e) {
        e.preventDefault();
        var vehicleId = $(this).data('vehicle-id');
        var icon = $(this);

        $.ajax({
            type: 'GET',
            url: `/wishlist/toggle/${vehicleId}/`,
            success: function(data) {
                if (data.status === 'added') {
                    icon.removeClass('far').addClass('fas');
                } else if (data.status === 'removed') {
                    icon.removeClass('fas').addClass('far');
                }
            },
            error: function(error) {
                console.log('Error:', error);
            }
        });
    });
});
</script>
<script>
    $(document).ready(function () {
        // Function to handle image upload and search
        $('#image-search-button').click(function() {
            var formData = new FormData();
            var fileInput = document.getElementById('image-upload-input');
            var uploadedImage = document.getElementById('uploaded-image');

            // Check if a file is selected
            if (fileInput.files.length > 0) {
                var file = fileInput.files[0];
                formData.append('image', file);
                // Display the uploaded image
                var reader = new FileReader();
                reader.onload = function(e) {
                    uploadedImage.src = e.target.result;
                    uploadedImage.style.display = 'block';
                };
                reader.readAsDataURL(file);

                // Make AJAX request to search for related items
                $.ajax({
                    type: 'POST',
                    url: '/search_related_items/',  // Replace with your view URL
                    data: formData,
                    processData: false,
                    contentType: false,
                    success: function(response) {
                        // Display related items based on search results
                        var relatedItemsDiv = $('#related-items');
                        relatedItemsDiv.empty();
                        response.forEach(function(item) {
                            relatedItemsDiv.append(`<div>${item.name}</div>`);
                            // You can customize how related items are displayed here
                        });
                    },
                    error: function(error) {
                        console.log('Error:', error);
                    }
                });
            } else {
                alert('Please select an image.');
            }
        });
    });
```

```
        </script>
    </body>
    </html>
```

Insurance_New.html

```
</html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Insurance New</title>
    <style>
        {% comment %} body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
        }
        .container {
            max-width: 500px;
            margin: 50px auto;
            padding: 20px;
            background-color: #fff;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        } {% endcomment %}
        h2 {
            color: #333;
            text-align: center;
            margin-bottom: 20px;
        }
        .form-group {
            margin-bottom: 20px;
        }
        label {
            display: block;
            font-weight: bold;
            margin-bottom: 5px;
        }
        input[type="text"],
        input[type="file"] {
            width: 100%;
            padding: 8px;
            border: 1px solid #ccc;
            border-radius: 4px;
            box-sizing: border-box;
        }
        button[type="submit"] {
            background-color: #007bff;
            color: #fff;
            padding: 10px 20px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }
        button[type="submit"]:hover {
            background-color: #0056b3;
        }
        #alert-message {
            color: red;
            margin-top: 10px;
            text-align: center;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Insurance New</h2>
        <form method="POST" action="{% url 'insurance_new' %}" onsubmit="return
validateForm()" enctype="multipart/form-data">
            {% csrf_token %}
            {% if error_message %}
            <div class="alert alert-danger">
                {{ error_message }}
            </div>
            {% endif %}
            <div id="alert-message" style="display: none;"></div>
            <!-- User Information -->
            <div class="form-group">
                <label for="full_name">Username:</label>
                <input type="text" id="full_name" name="full_name" value="{{ request.user.username
}}" readonly>
            </div>
            <div class="form-group">
                <label for="email">Email:</label>
                <input type="text" id="email" name="email" value="{{ request.user.email }}"
readonly>
            </div>
            <div class="form-group">
                <label for="address">Address:</label>
                <input type="text" id="address" name="address" value="{{ request.user.address }}"
readonly>
            </div>

            <div class="form-group">
                <label for="phone_number">Phone Number:</label>
                <input type="text" id="phone_number" name="phone_number" value="{{
request.user.phone_number }}" readonly>
            </div>
            <div class="form-group">
```

```
                <label for="register_number">Register Number:</label>
                <input type="text" id="register_number" name="register_number" required>
            </div>
            <div class="form-group">
                <label for="state">State:</label>
                <input type="text" id="state" name="state" required>
            </div>
            <div class="form-group">
                <label for="id_proof">ID Proof (PNG, JPG, PDF):</label>
                <input type="file" id="id_proof" name="id_proof" accept=".png, .jpg, .pdf">
            </div>
            <button type="submit" name="submit">Submit</button>
        </form>
    </div>
    <script>
        function validateForm() {
            var fileInput = document.getElementById('id_proof');
            var filePath = fileInput.value;
            var allowedExtensions = /(\.png|\.jpg|\.jpeg|\.pdf)$/i;

            if (!allowedExtensions.exec(filePath)) {
                alert('Please upload file having extensions .png, .jpg, .jpeg, .pdf only.');
                fileInput.value = '';
                return false;
            } else {
                return true;
            }
        }
    </script>
</body>
</html>
```

Insurance_package.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Insurance Process</title>
    <style>
        {% comment %} .container {
            max-width: 800px;
            margin: 0 auto;
            padding: 20px;
            font-family: Arial, sans-serif;
            display: flex;
            flex-direction: column;
            align-items: flex-start;
        } {% endcomment %}
```

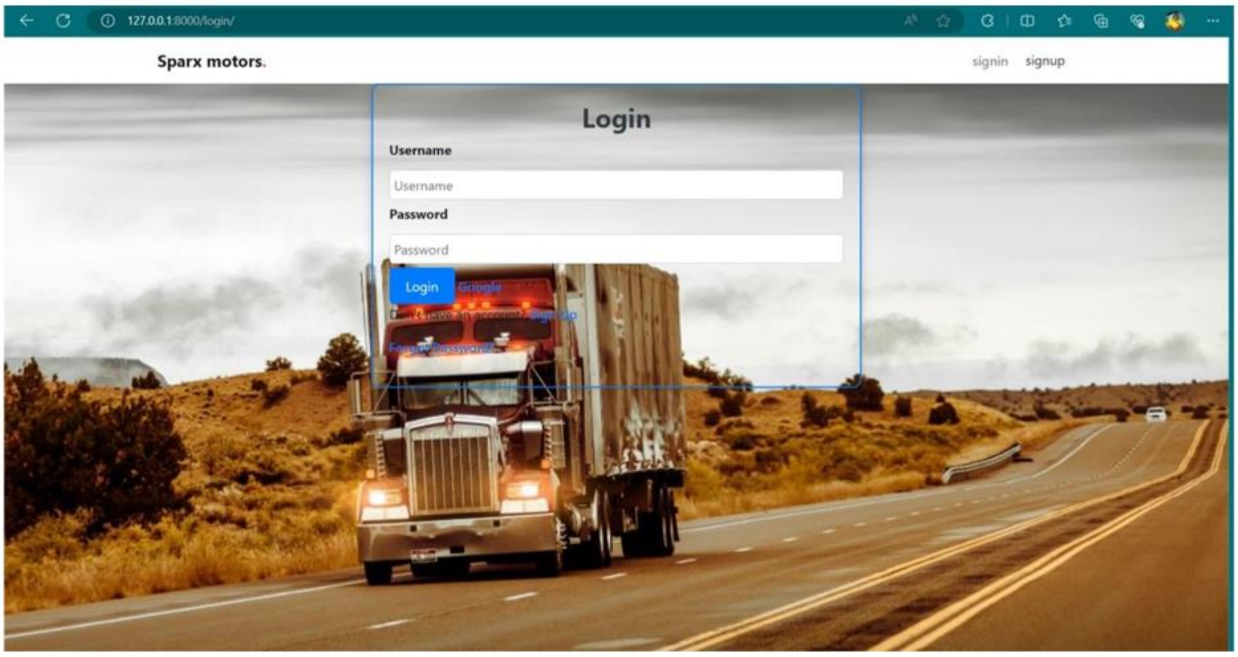
```
.insurance-list {
   display: flex;
   flex-wrap: wrap;
}

.insurance-item {
   margin: 10px;
}

/* Hide the default radio button */
.insurance-item input[type="radio"] {
    display: none;
}

/* Style the label as the custom radio button */
.insurance-item label {
   display: block;
   border: 2px solid #ccc;
   border-radius: 5px;
   padding: 10px;
   cursor: pointer;
   transition: background-color 0.3s ease;
   position: relative; /* Positioning for the custom radio button */
}

/* Change background color on hover */
.insurance-item label:hover {
   background-color: #f0f0f0;
}

/* Style for the checked state of the custom radio button */
.insurance-item input[type="radio"]:checked + label:before {
   content: '\2713'; /* Unicode character for checkmark */
   position: absolute;
   top: 50%;
   left: 50%;
   transform: translate(-50%, -50%);
   color: #007bff; /* Blue color */
   font-size: 20px;
}

/* Style for the unchecked state of the custom radio button */
.insurance-item input[type="radio"] + label:before {
   content: '';
   position: absolute;
   top: 50%;
   left: 50%;
   transform: translate(-50%, -50%);
   width: 20px;
   height: 20px;
   border: 2px solid #ccc;
```

```
            border-radius: 50%; /* Makes the border round to indicate radio button */
            background-color: #fff; /* Background color for the radio button */
        }

        /* Details styling */
        .insurance-details {
            font-size: 14px;
            color: #333;
        }

        /* Checkout button styling */
        #checkoutButton {
            margin-top: 20px;
        }
        h2 {
            text-align: center;
        }
        #checkoutButton {
            margin-top: 20px;
            background-color: #007bff; /* Blue background color */
            color: #fff; /* White text color */
            border: none;
            border-radius: 5px;
            padding: 10px 20px;
            cursor: pointer;
            font-size: 16px;
            transition: background-color 0.3s ease;
        }

        #checkoutButton:hover {
            background-color: #0056b3; /* Darker shade of blue on hover */
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Insurance Package</h2>
        <form method="POST" action="{% url 'package_details' %}">
        {% csrf_token %}
        <div class="insurance-list">
            {% for insurance in insurances %}
                <div class="insurance-item">
                    <input type="radio" id="insurance_{{ insurance.id }}" name="selected_insurance"
value="{{ insurance.id }}">
                    <label for="insurance_{{ insurance.id }}">
                        <div class="insurance-details">

                            <img src="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTd0z3iZAlJnF4biZrt-
MYHzi4Ltcy_PQo72tNg_W7RiLypzeEYRScKJfyaC0AwwfUeS9E&usqp=CAU" alt="Insurance
Company Logo">
```

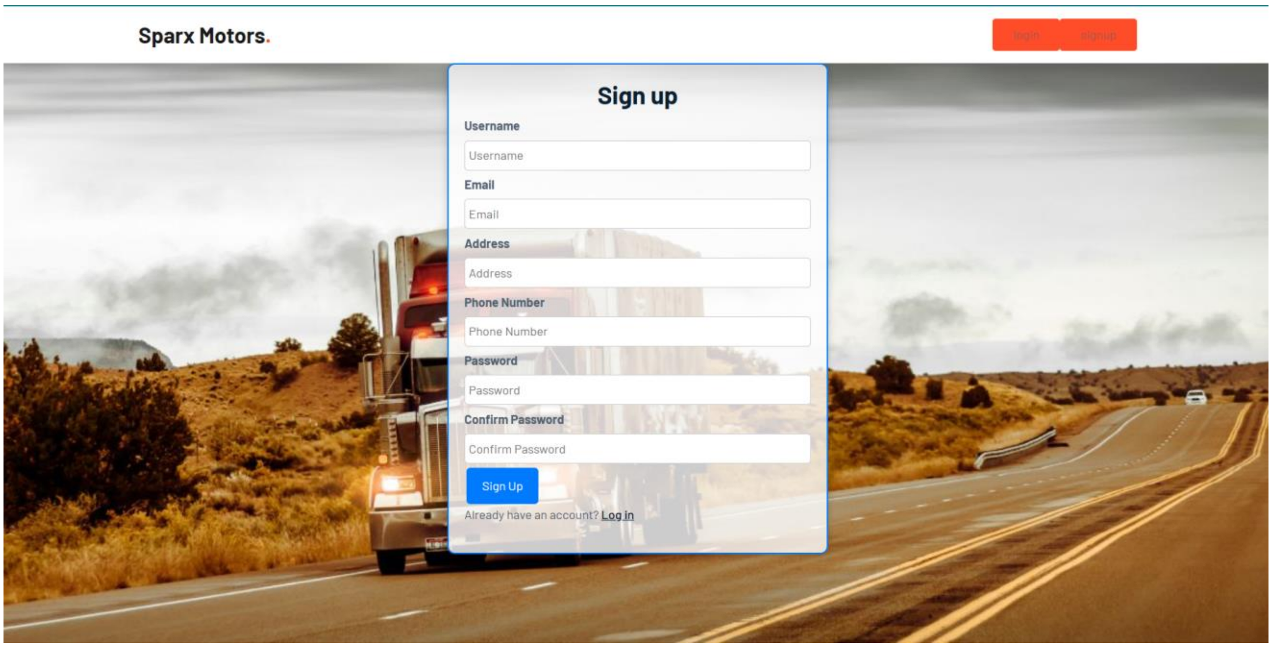
```
                              <p>{{ insurance.vehicle_model }}</p>
                              <p>{{ insurance.vehicle_usage }}</p>
                              <p>{{ insurance.fuel_type }}</p>
                              <p>{{ insurance.transmission_type }}</p>
                              <p>{{ insurance.insurance_type }}</p>

                              <p>Package: {{ insurance.price }}</p>
                          </div>
                      </label>
                  </div>
              {% endfor %}
          </div>
          <button type="submit" id="checkoutButton">Checkout</button>
      </form>
      </div>

</body>
</html>
```

9.2 Screen Shots

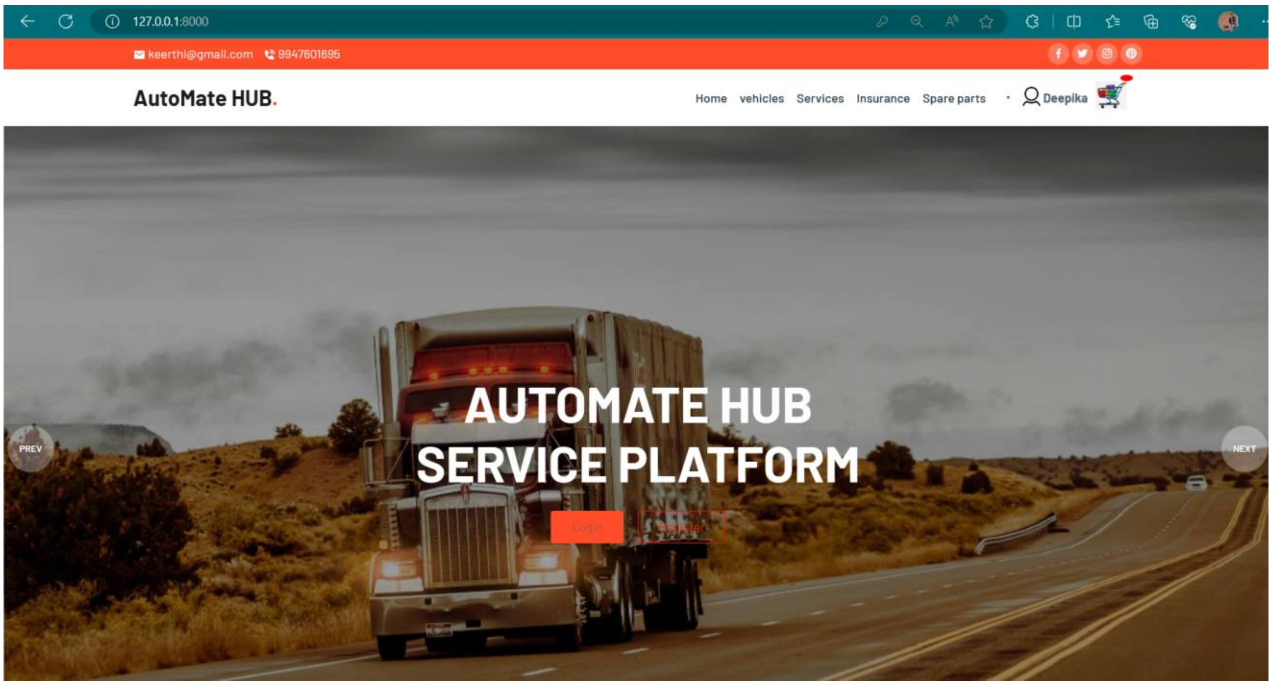
9.2.1 Login page



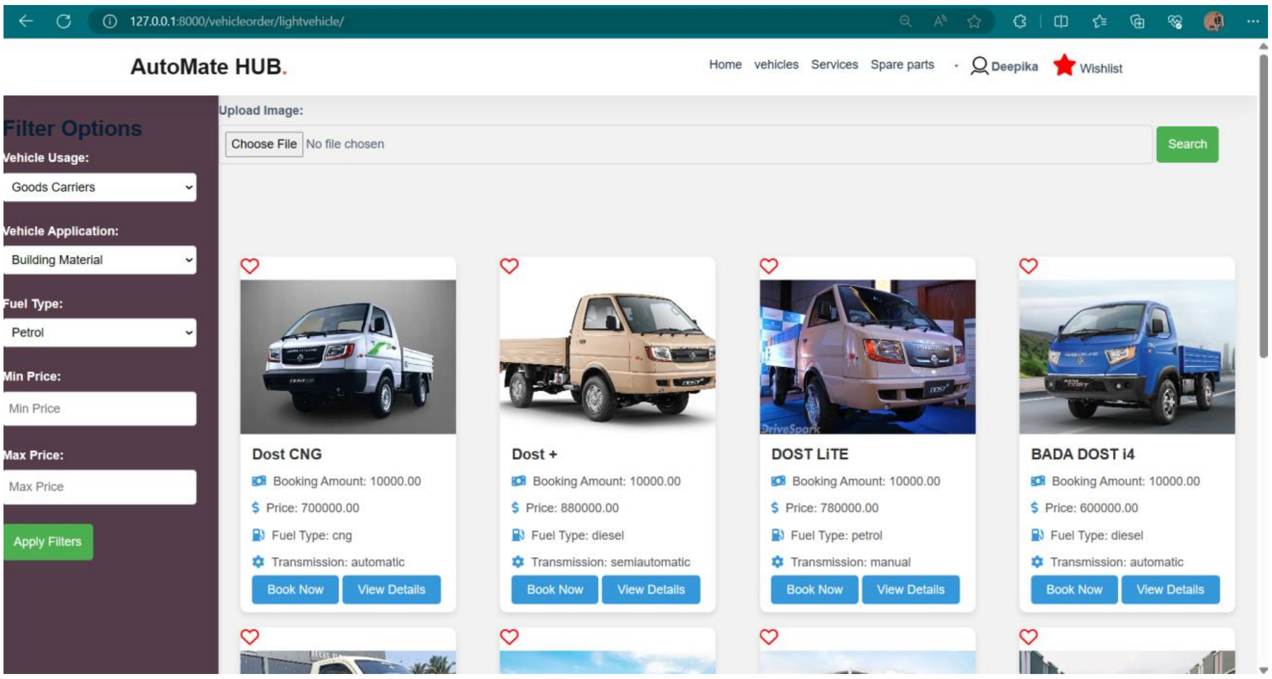
9.2.2 Signup page



9.2.3 Index page



9.2.4 Vehicle Booking page



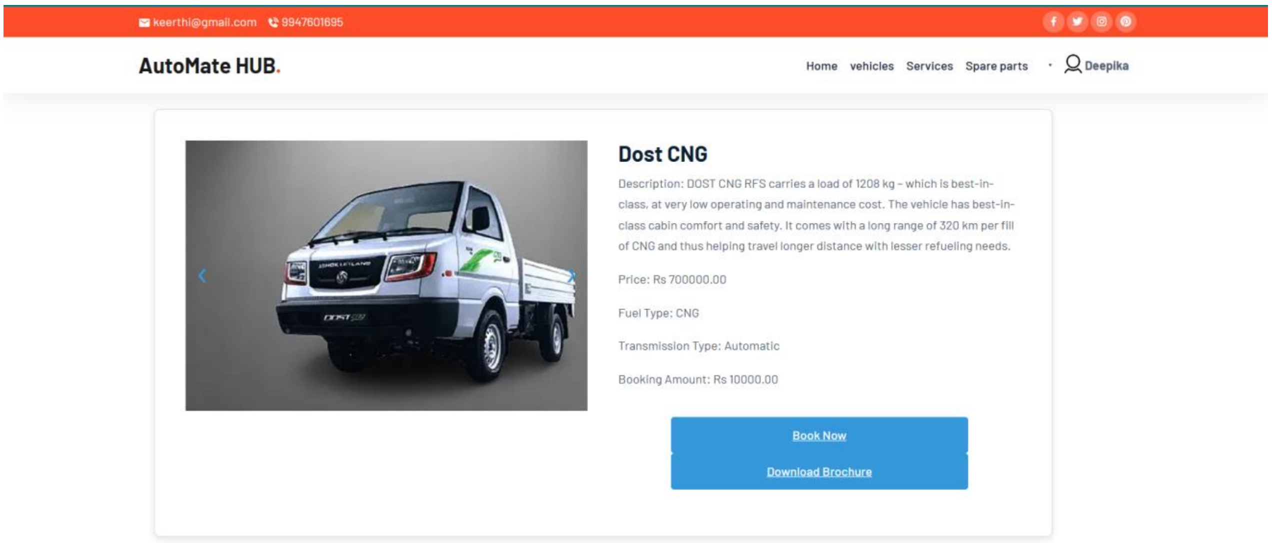
Dost CNG
Booking Amount: 10000.00
Price: 700000.00
Fuel Type: cng
Transmission: automatic

Dost +
Booking Amount: 10000.00
Price: 880000.00
Fuel Type: diesel
Transmission: semiautomatic

DOST LiTE
Booking Amount: 10000.00
Price: 780000.00
Fuel Type: petrol
Transmission: manual

BADA DOST i4
Booking Amount: 10000.00
Price: 600000.00
Fuel Type: diesel
Transmission: automatic

9.2.5 Book_now page



Dost CNG

Description: DOST CNG RFS carries a load of 1208 kg – which is best-in-class, at very low operating and maintenance cost. The vehicle has best-in-class cabin comfort and safety. It comes with a long range of 320 km per fill of CNG and thus helping travel longer distance with lesser refueling needs.

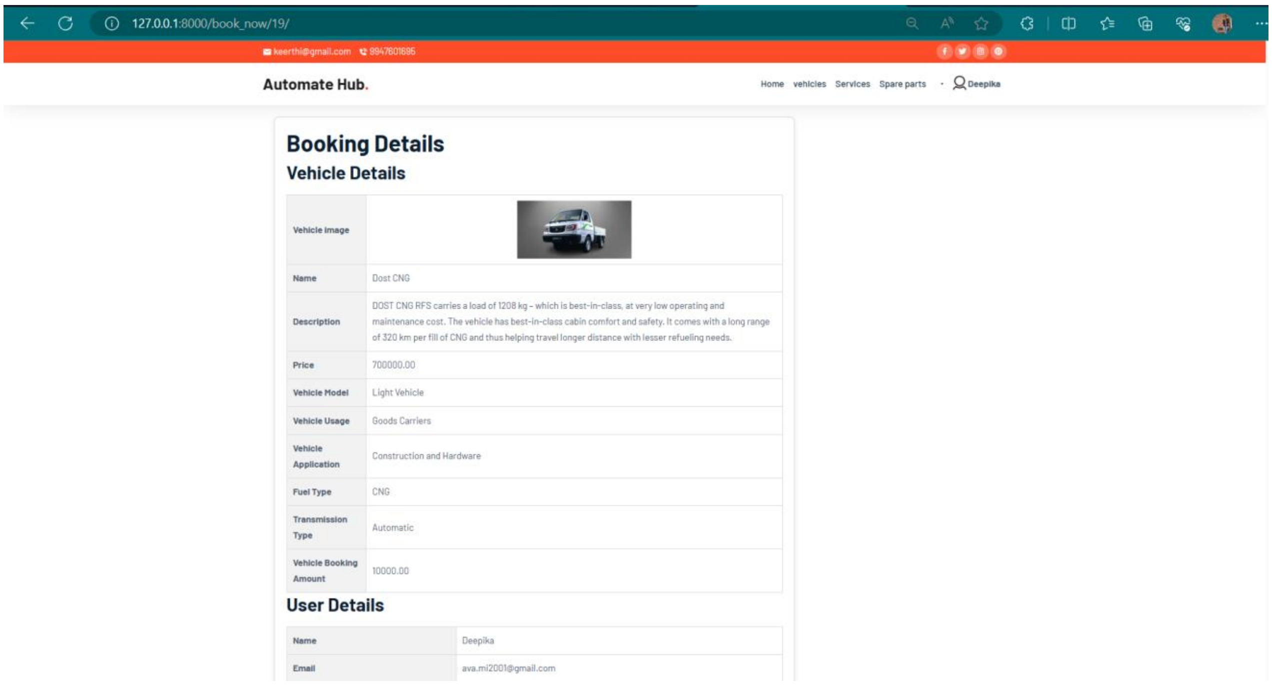
Price: Rs 700000.00

Fuel Type: CNG

Transmission Type: Automatic

Booking Amount: Rs 10000.00

9.2.6 Booking _details page



127.0.0.1:8000/book_now/19/

keerthi@gmail.com 9947601695

Automate Hub.

Home vehicles Services Spare parts Deepika

Booking Details

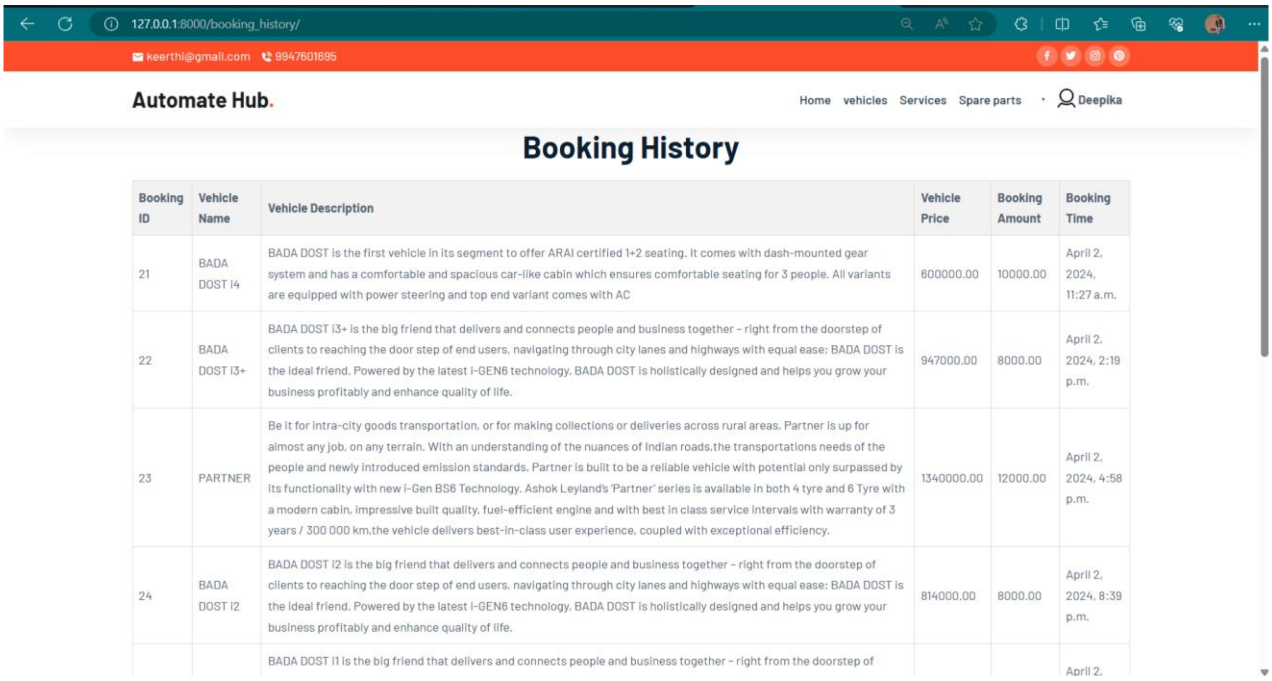
Vehicle Details

Vehicle image	
Name	Dost CNG
Description	DOST CNG RFS carries a load of 1208 kg – which is best-in-class, at very low operating and maintenance cost. The vehicle has best-in-class cabin comfort and safety. It comes with a long range of 320 km per fill of CNG and thus helping travel longer distance with lesser refueling needs.
Price	700000.00
Vehicle Model	Light Vehicle
Vehicle Usage	Goods Carriers
Vehicle Application	Construction and Hardware
Fuel Type	CNG
Transmission Type	Automatic
Vehicle Booking Amount	10000.00

User Details

Name	Deepika
Email	ava.mi2001@gmail.com

9.2.7 Booking history page



127.0.0.1:8000/booking_history/

keerthi@gmail.com 9947601695

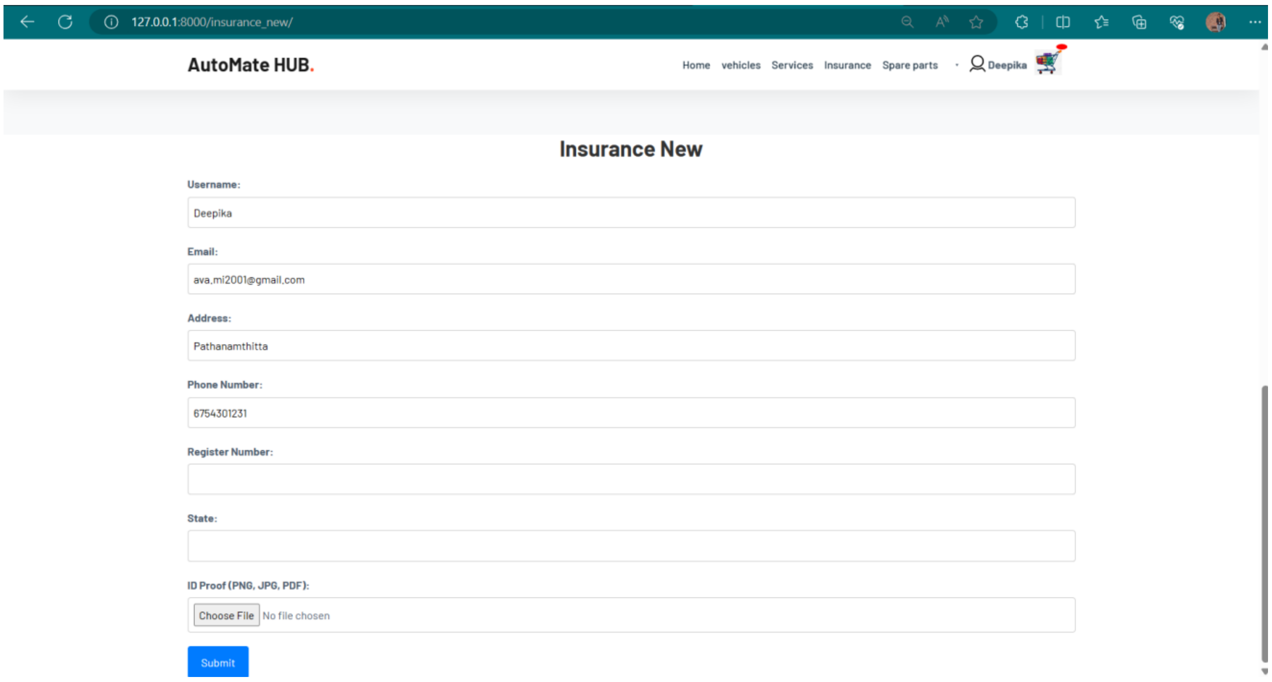
Automate Hub.

Home vehicles Services Spare parts Deepika

Booking History

Booking ID	Vehicle Name	Vehicle Description	Vehicle Price	Booking Amount	Booking Time
21	BADA DOST i4	BADA DOST is the first vehicle in its segment to offer ARAI certified 1+2 seating. It comes with dash-mounted gear system and has a comfortable and spacious car-like cabin which ensures comfortable seating for 3 people. All variants are equipped with power steering and top end variant comes with AC	600000.00	10000.00	April 2, 2024, 11:27 a.m.
22	BADA DOST i3+	BADA DOST i3+ is the big friend that delivers and connects people and business together – right from the doorstep of clients to reaching the door step of end users, navigating through city lanes and highways with equal ease; BADA DOST is the ideal friend. Powered by the latest i-GEN6 technology, BADA DOST is holistically designed and helps you grow your business profitably and enhance quality of life.	947000.00	8000.00	April 2, 2024, 2:19 p.m.
23	PARTNER	Be it for intra-city goods transportation, or for making collections or deliveries across rural areas, Partner is up for almost any job, on any terrain. With an understanding of the nuances of Indian roads,the transportations needs of the people and newly introduced emission standards, Partner is built to be a reliable vehicle with potential only surpassed by its functionality with new i-Gen BS6 Technology. Ashok Leyland's 'Partner' series is available in both 4 tyre and 6 Tyre with a modern cabin, impressive built quality, fuel-efficient engine and with best in class service intervals with warranty of 3 years / 300 000 km,the vehicle delivers best-in-class user experience, coupled with exceptional efficiency.	1340000.00	12000.00	April 2, 2024, 4:58 p.m.
24	BADA DOST i2	BADA DOST i2 is the big friend that delivers and connects people and business together – right from the doorstep of clients to reaching the door step of end users, navigating through city lanes and highways with equal ease; BADA DOST is the ideal friend. Powered by the latest i-GEN6 technology, BADA DOST is holistically designed and helps you grow your business profitably and enhance quality of life.	814000.00	8000.00	April 2, 2024, 8:39 p.m.
		BADA DOST i1 is the big friend that delivers and connects people and business together – right from the doorstep of			April 2,

9.2.8 Insurance _new page



127.0.0.1:8000/insurance_new/

AutoMate HUB.

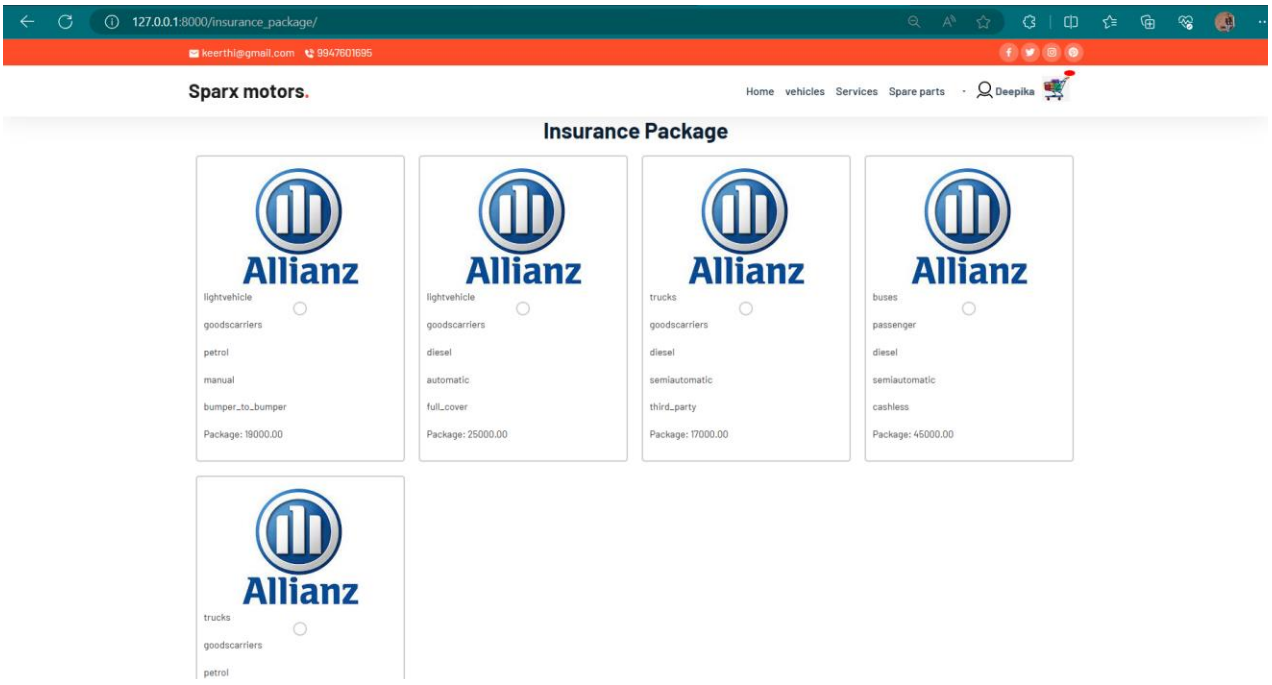
Home vehicles Services Insurance Spare parts Deepika

Insurance New

Username:	Deepika
Email:	ava.mi2001@gmail.com
Address:	Pathanamthitta
Phone Number:	6754301231
Register Number:	
State:	
ID Proof (PNG, JPG, PDF):	Choose File No file chosen

Submit

9.2.9 Insurance pakage



127.0.0.1:8000/insurance_package/

keerthi@gmail.com 9947601695

Sparx motors.

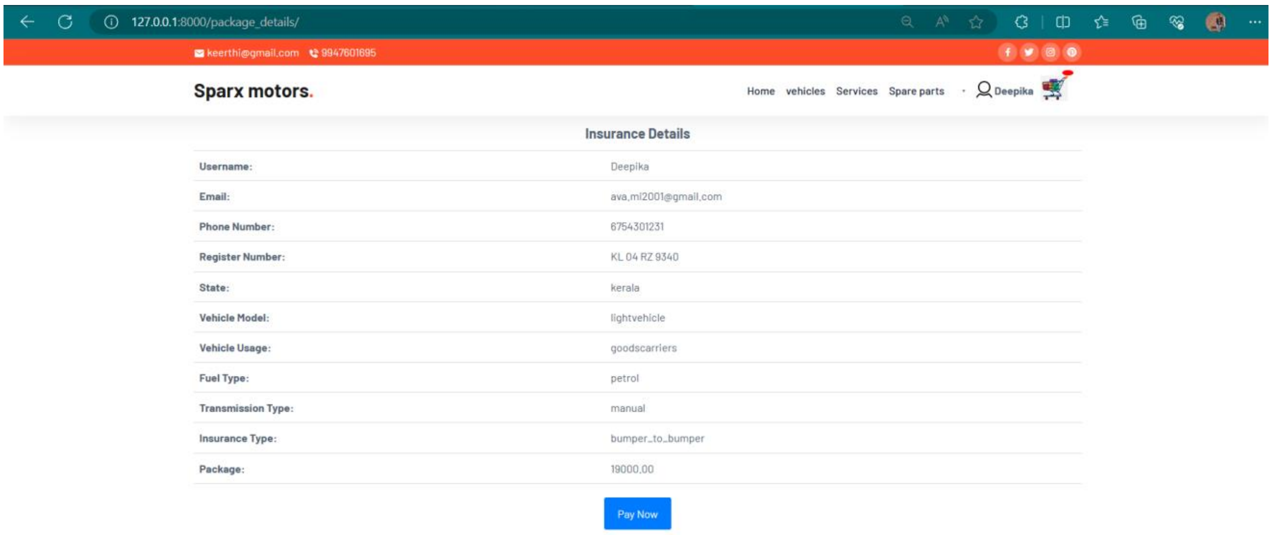
Home vehicles Services Spare parts Deepika

Insurance Package

Allianz	Allianz	Allianz	Allianz
lightvehicle	lightvehicle	trucks	buses
goodscarriers	goodscarriers	goodscarriers	passenger
petrol	diesel	diesel	diesel
manual	automatic	semiautomatic	semiautomatic
bumper_to_bumper	full_cover	third_party	cashless
Package: 19000.00	Package: 25000.00	Package: 17000.00	Package: 45000.00

Allianz
trucks
goodscarriers
petrol

9.2.10 Checkout page

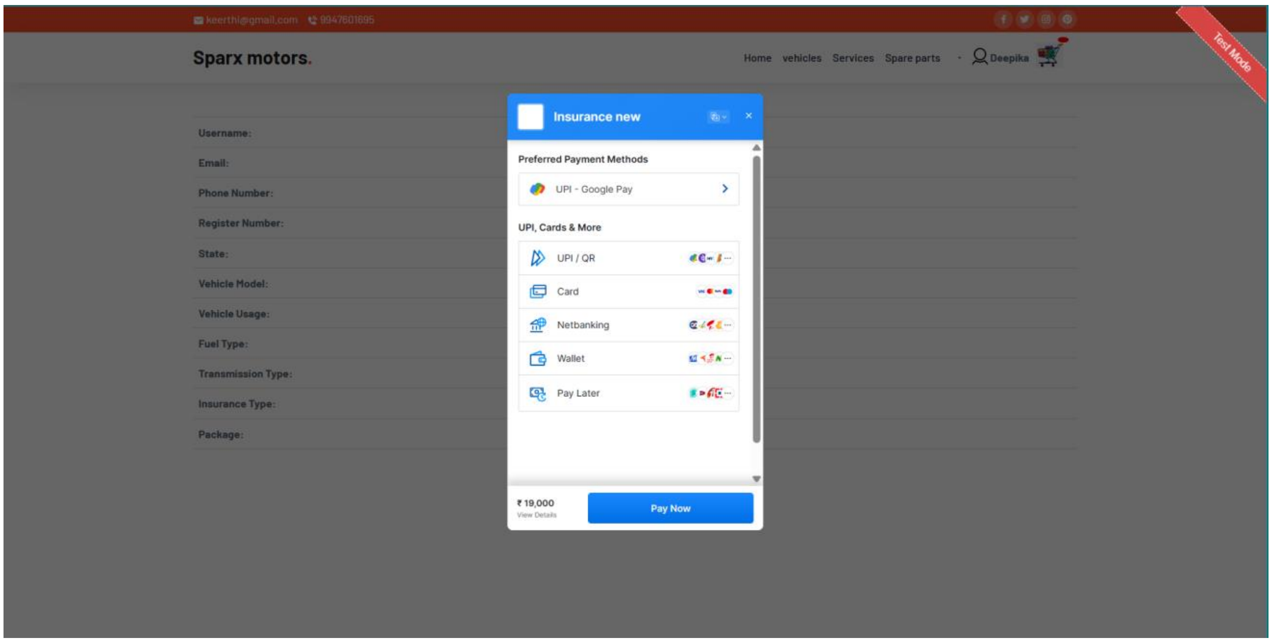

127.0.0.1:8000/package_details/
keerthi@gmail.com 9947601695
Sparx motors.
Home vehicles Services Spare parts Deepika

Insurance Details

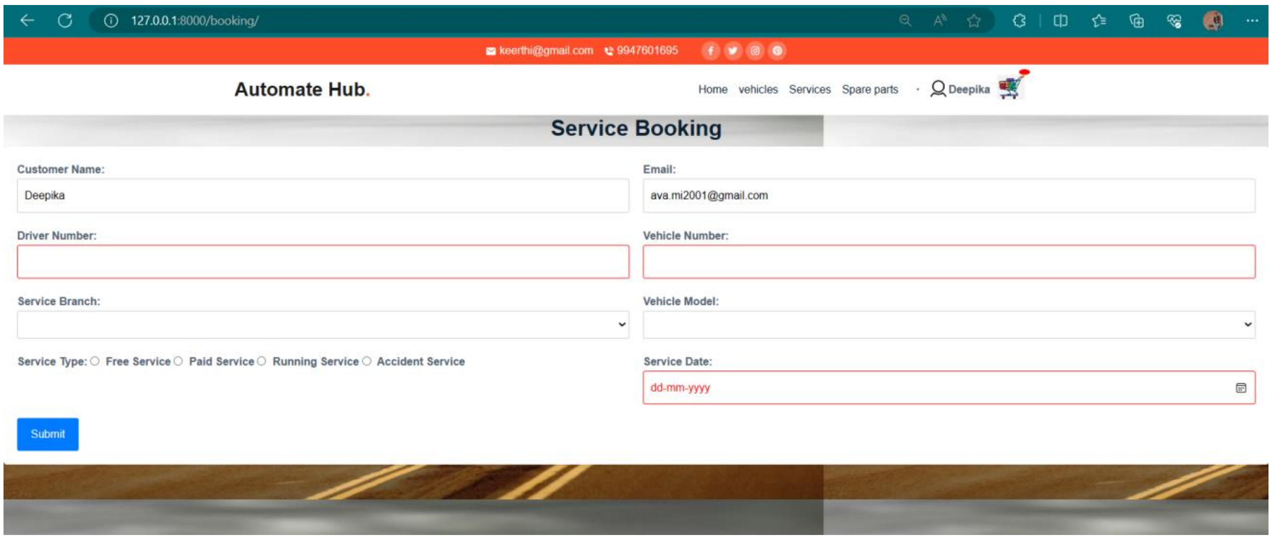
Username:	Deepika
Email:	ava.ml2001@gmail.com
Phone Number:	6754301231
Register Number:	KL 04 RZ 9340
State:	kerala
Vehicle Model:	lightvehicle
Vehicle Usage:	goodscarriers
Fuel Type:	petrol
Transmission Type:	manual
Insurance Type:	bumper_to_bumper
Package:	19000.00

Pay Now

9.2.11 Insurance payment



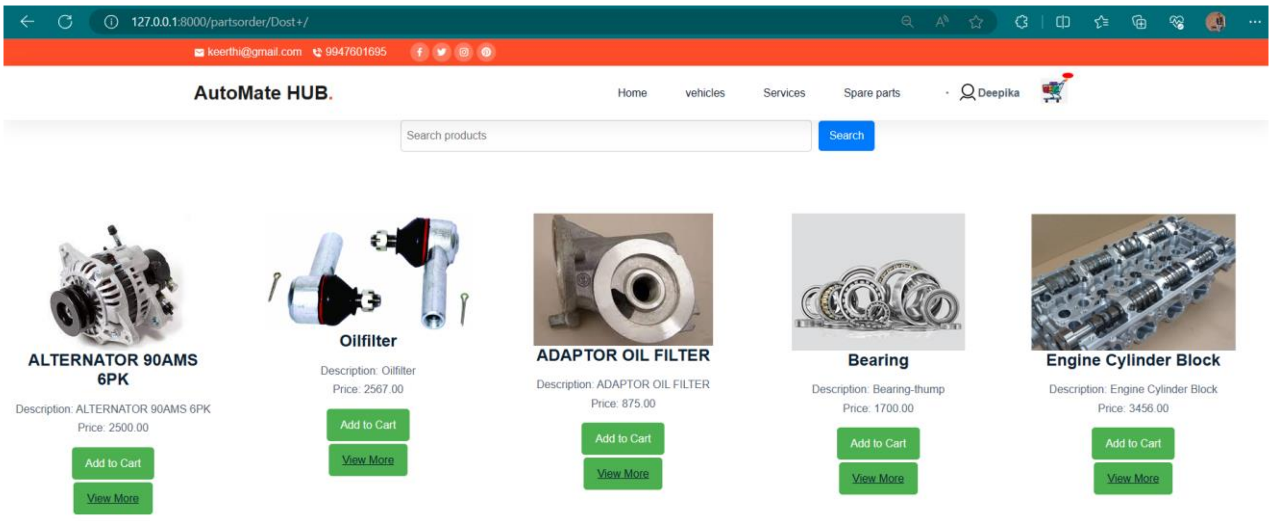
9.2.12 Vehicle_service booking


127.0.0.1:8000/booking/
keerthi@gmail.com 9947601695
Automate Hub.
Home vehicles Services Spare parts Deepika
Submit

Service Booking

Customer Name: Deepika	Email: ava.mi2001@gmail.com
Driver Number:	Vehicle Number:
Service Branch:	Vehicle Model:
Service Type: ○ Free Service ○ Paid Service ○ Running Service ○ Accident Service	Service Date: dd-mm-yyyy

9.2.13 Spare_parts order


127.0.0.1:8000/partsorder/Dost+/
keerthi@gmail.com 9947601695
AutoMate HUB.
Home vehicles Services Spare parts Deepika
Search products
Search
Add to Cart
View More

ALTERNATOR 90AMS 6PK
Description: ALTERNATOR 90AMS 6PK
Price: 2500.00

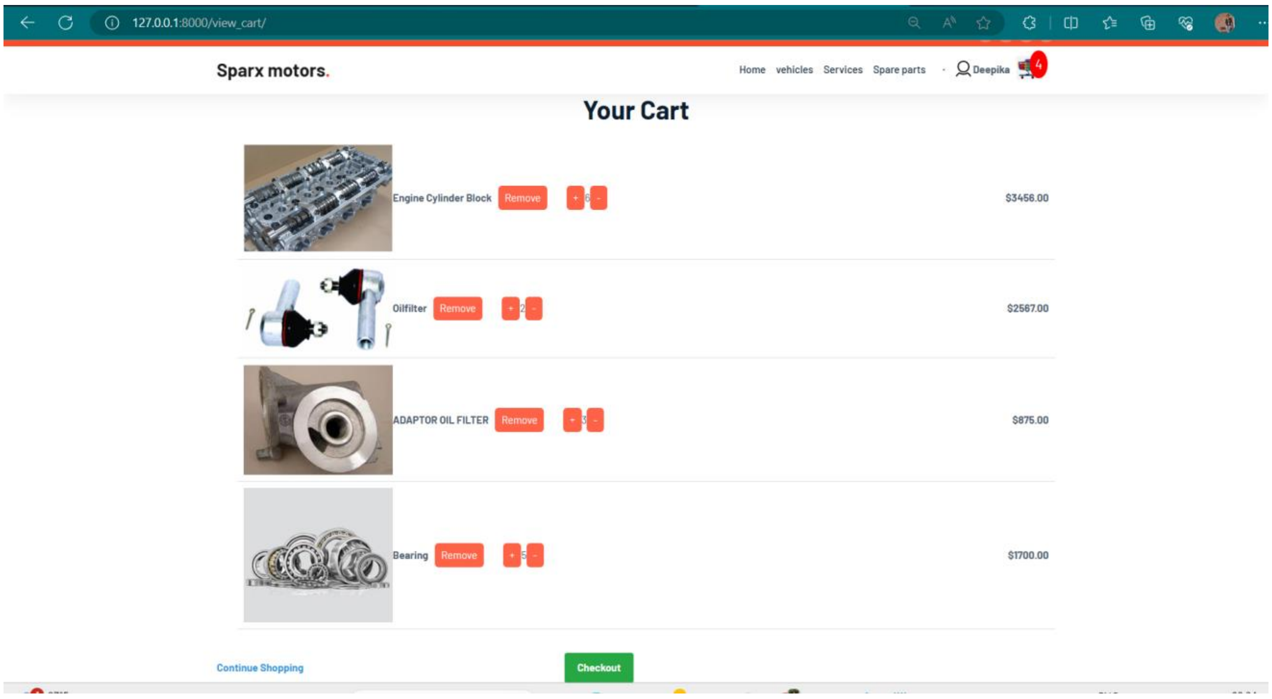
Oilfilter
Description: Oilfilter
Price: 2567.00

ADAPTOR OIL FILTER
Description: ADAPTOR OIL FILTER
Price: 875.00

Bearing
Description: Bearing-thump
Price: 1700.00

Engine Cylinder Block
Description: Engine Cylinder Block
Price: 3456.00

9.2.14 Checkout page


127.0.0.1:8000/view_cart/

Sparx motors.

Home vehicles Services Spare parts Deepika 4

Your Cart

Engine Cylinder Block	Remove	+ 6 -	$3456.00
Oilfilter	Remove	+ 2 -	$2567.00
ADAPTOR OIL FILTER	Remove	+ 3 -	$875.00
Bearing	Remove	+ 5 -	$1700.00

Continue Shopping Checkout

Attach Plagiarism Report