Al al-Bayt University

Prince Al Hussein bin Abdullah College of Information Technology

" Moonlight Motors "

Prepare:

Abdalrhman Janajrah 2100901259

Abdallah Al-mkhzoumi 2100901074

Supervisor:

Suhair Ahmad Mohammad

Committee Members	**Signatures**
.....	.....
.....	.....
.....	.....

This project was completed to fulfill the requirements for obtaining a bachelor's degree in computer information systems specialization from the College of Prince Al-Hussein Bin Abdullah II for Information Technology at Al al-Bayt University, First Semester 2024/202

Table of contents

List of figures

List of tables

Abstract

"Moonlight Motors" is an innovative web application developed as a graduation project by students of Al-Bayt University. The application aims to provide a comprehensive solution for the automotive market, addressing the needs of buyers, renters, and customers seeking maintenance services and spare parts. The platform offers services such as car purchasing and rental, booking maintenance appointments, and ordering spare parts directly from the company.

The application is designed to offer a seamless experience for users to explore cars available for sale or rent, book maintenance appointments, and order spare parts with ease. All products and services are provided directly by the company to ensure quality and reliability.

"Moonlight Motors" aims to deliver a centralized and integrated solution that meets the needs of customers in the automotive market in Jordan, ensuring high-quality services that simplify processes and save time and effort for customers

CHAPTER 1: Overview of the project

1.1 INTRODUCTION

In today's world, technology has become an integral part of our daily lives, making various commercial and service processes more accessible and efficient. One of the sectors that has experienced significant advancements is the automotive industry, where accessing services such as sales, maintenance, and rentals has never been easier.

Our project, "Moonlight Motors," is a comprehensive online platform designed to provide innovative and integrated services in the automotive field. The project aims to meet customer needs by offering a user-friendly interface that enables them to buy and rent cars, book maintenance appointments, and order spare parts with ease.

1.2 PROJECT PROBLEM

In the current market, automobile companies face various challenges in maintaining efficient operations and providing seamless services to their customers. Traditional methods of car sales, rentals, maintenance appointments, and spare parts sales often result in inefficiencies and customer dissatisfaction.

These challenges include:

1. **Inefficient Management:**
   Existing systems are either manual or fragmented, leading to delays in processing customer requests and poor coordination among departments.
2. **Limited Accessibility:**
   Customers struggle with accessing information about available cars for sale or rent, scheduling maintenance appointments, or purchasing spare parts without physically visiting the company.
3. **Customer Dissatisfaction:**
   A lack of a user-friendly online platform reduces customer engagement and satisfaction, as modern users expect digital convenience.

4. **Competitive Disadvantage:**
   Competitors offering comprehensive online platforms gain an edge in the market, leaving traditional systems lagging behind.

1.3 PROPOSED SOLUTION

The proposed solution to address the challenges faced by the car company is the development of an integrated online platform. This platform will streamline and enhance various operations, such as car sales, rentals, maintenance appointment scheduling, and spare parts purchasing. The solution will leverage modern web technologies to provide an efficient, user-friendly, and secure digital system.

Key features of the proposed solution include:

1. **Comprehensive Car Management System:**
   A centralized system to showcase available cars for sale or rent, including detailed specifications, pricing, and availability status.
2. **Online Maintenance Scheduling:**
   An intuitive interface for customers to book maintenance appointments, check available slots, and receive automated reminders.
3. **Spare Parts Purchasing Platform:**
   A dedicated section for browsing, selecting, and purchasing spare parts, with delivery or pickup options.
4. **User-Friendly Interface:**
   A well-designed, responsive interface that ensures a seamless experience across all devices, catering to customers' needs with minimal effort.
5. **Integrated Payment Gateway:**
   A secure payment system allowing customers to make online payments for services, purchases, and reservations.

1.4 PROJECT OBJECTIVES

This project aims to develop an integrated online platform to serve the car company by enhancing operational efficiency and improving the customer experience. The objectives of the project are as follows:

1. **Provide a comprehensive online platform:**
   Design and develop a digital system that supports the processes of car sales, rentals, maintenance appointment scheduling, and spare parts purchasing efficiently and effectively.
2. **Improve operational efficiency:**
   Reduce the time required to process customer requests and enhance coordination between different departments within the company through the digital system.
3. **Enhance customer experience:**
   Offer a user-friendly interface that enables customers to perform all required operations online, thereby increasing satisfaction and loyalty.
4. **Increase competitiveness:**
   Enable the company to compete more effectively in the market by offering innovative digital services that meet modern customer needs.

1.5 BENEFICIARIES OF THIS PROJECT

This project aims to bring value to various stakeholders by addressing their needs and enhancing their experience with streamlined, efficient, and modern solutions:

1. **The Car Company**
2. **Customers**
3. **Market Stakeholders**

CHAPTER 2: System analysis

2.1 INTRODUCTION

The next step is to adopt a methodology compatible with the workflow plan, capable of adapting to system changes, and effectively addressing any sudden modifications during the development process. This decision was made after thoroughly researching and understanding the project's requirements and organizing the ideas to be implemented in a structured manner.

Given its robust capability to manage dynamic business needs and facilitate continuous improvement, we chose to adopt **Agile methodologies**. The details and rationale behind selecting Agile will be discussed in subsequent sections.

2.2 DATA COLLECTION AND REQUIREMENT GATHERING

The data collection process was carried out in multiple stages, with each stage dedicated to gathering information related to the design and implementation of a specific part of the project. Below is a detailed explanation of the stages through which the data collection process was conducted:

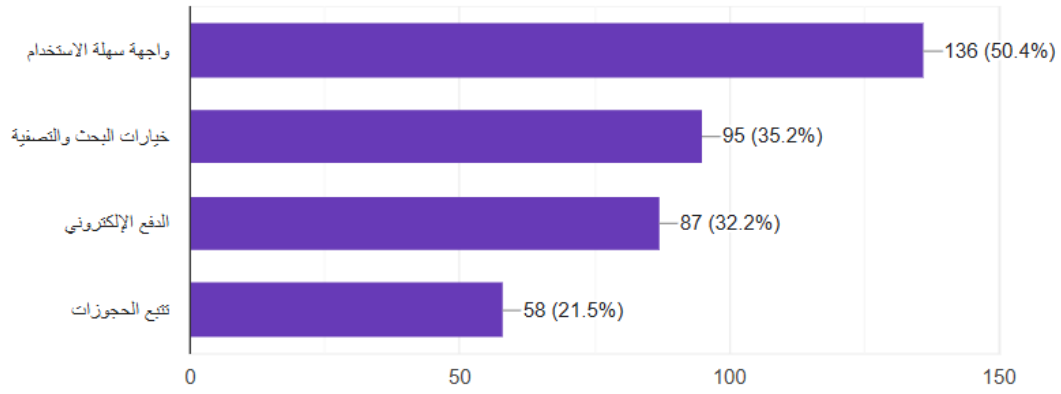
1. **Collecting Information from Websites:**
   Specialized websites were utilized to gather information related to the automotive sales and rental market. These websites provided valuable insights into how online platforms operate in this domain, particularly in terms of vehicle listings, categorization, and customer interactions.
2. **Questionnaires and Audience Feedback Collection:**
   An important part of the data collection process involved creating and distributing structured questionnaires to the target audience expected to use the system. These questionnaires were designed to identify the needs, preferences, and desired functionalities for the platform.

Here are some of the key findings from the questionnaire:

نسخ الرسم البياني

ما الذي تبحث عنه في نظام إلكتروني لإدارة خدمات السيارات؟

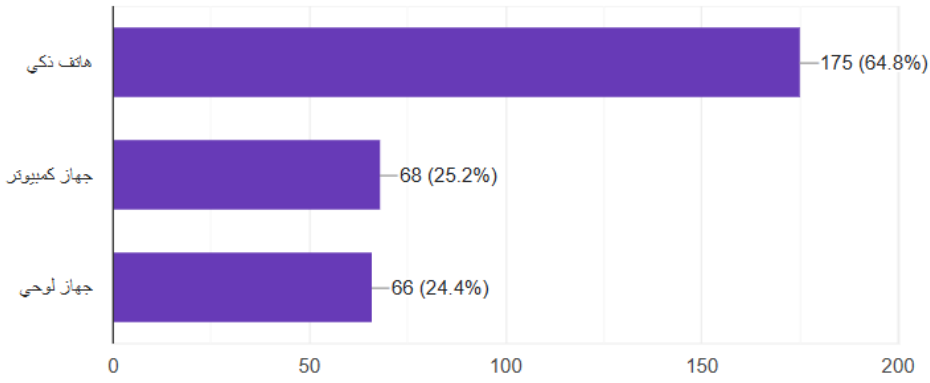
270 ردًا



نسخ الرسم البياني

ما هي الأجهزة التي تفضل استخدامها للوصول إلى النظام؟

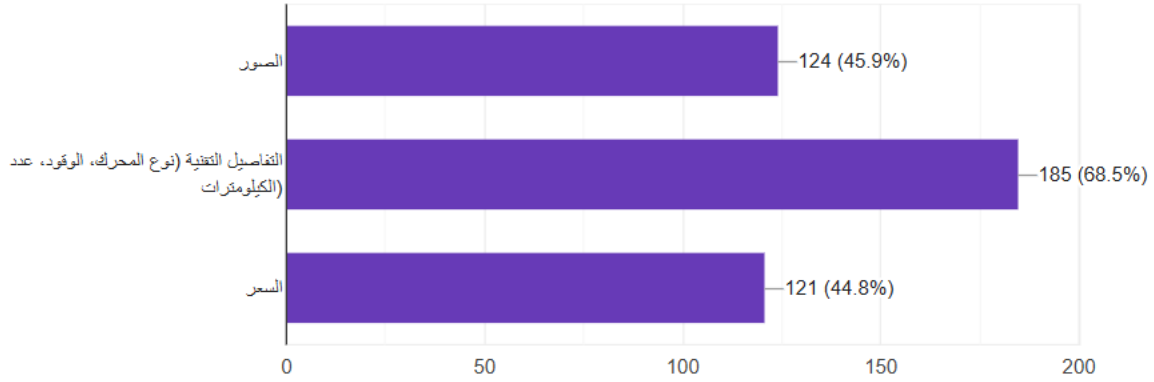
270 ردًا



نسخ الرسم البياني

عند تصفح السيارات (للتأجير أو البيع)، ما هي المعلومات التي تحتاجها؟

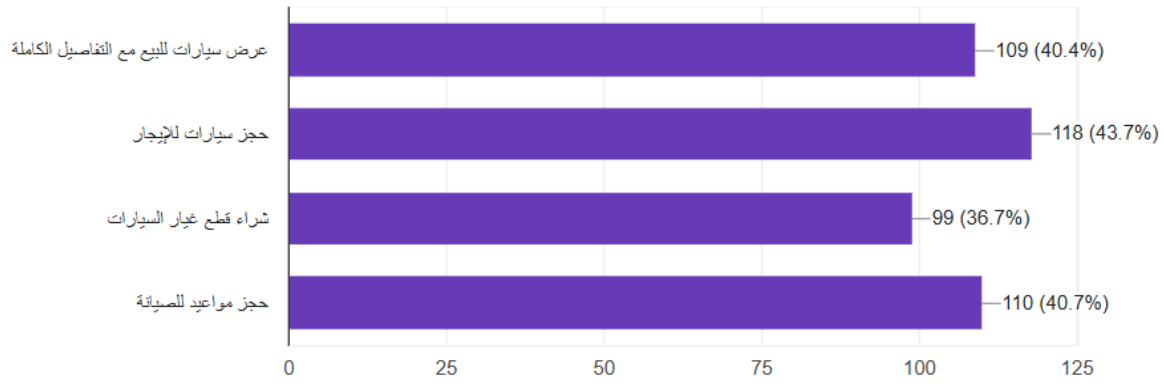
270 ردًا



نسخ الرسم البياني

ما هي الوظائف التي ترغب بأن يدعمها النظام؟ (يمكن اختيار أكثر من خيار)

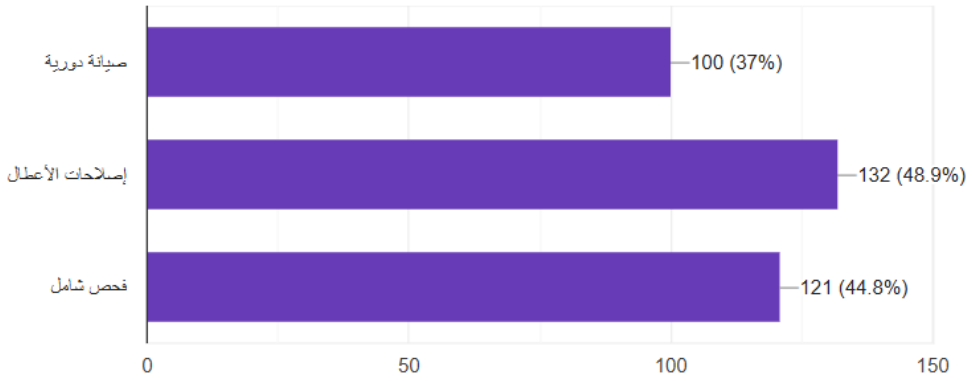
270 ردًا



نسخ الرسم البياني

ما هي أنواع الصيانة التي تحتاجها عادة؟

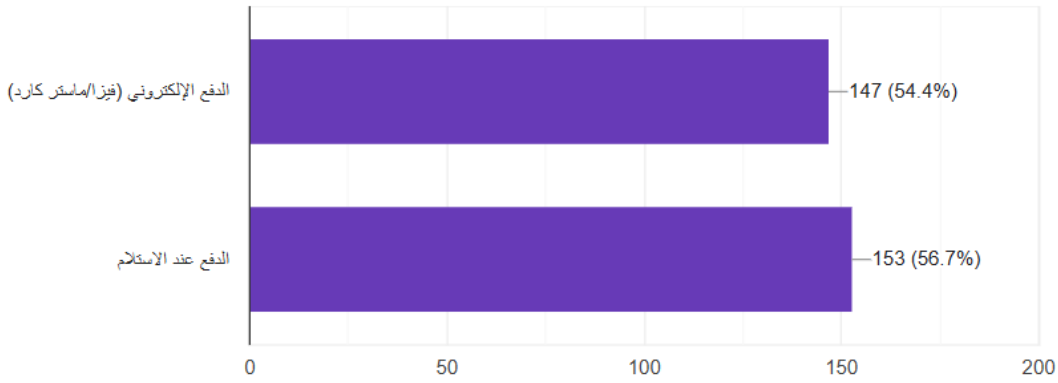
270 ردًا



نسخ الرسم البياني

ما هي طرق الدفع التي تفضل استخدامها؟

270 ردًا



2.3 SYSTEM DEVELOPMENT MODELS

When selecting a suitable system development model, it was essential to consider the nature of our project and its need for flexibility and efficient execution. After evaluating various models, we chose **Agile methodology** as the primary framework for system development, instead of the traditional **Waterfall model**.

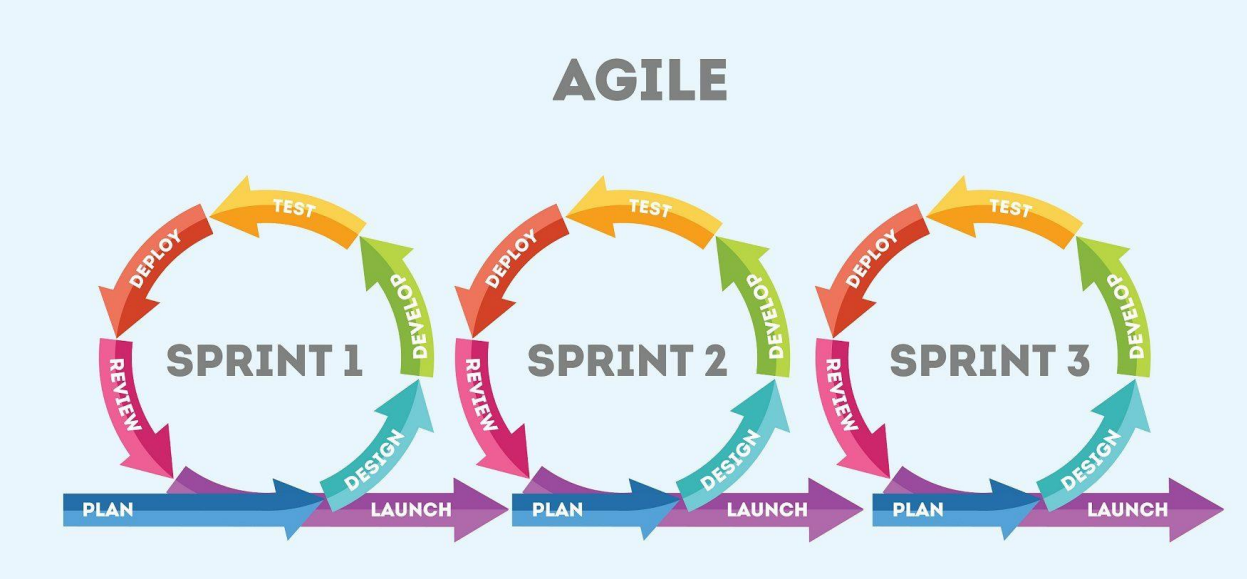


Figure 2-1:Agile methodology

2.3.1 Comparison between Agile and Waterfall:

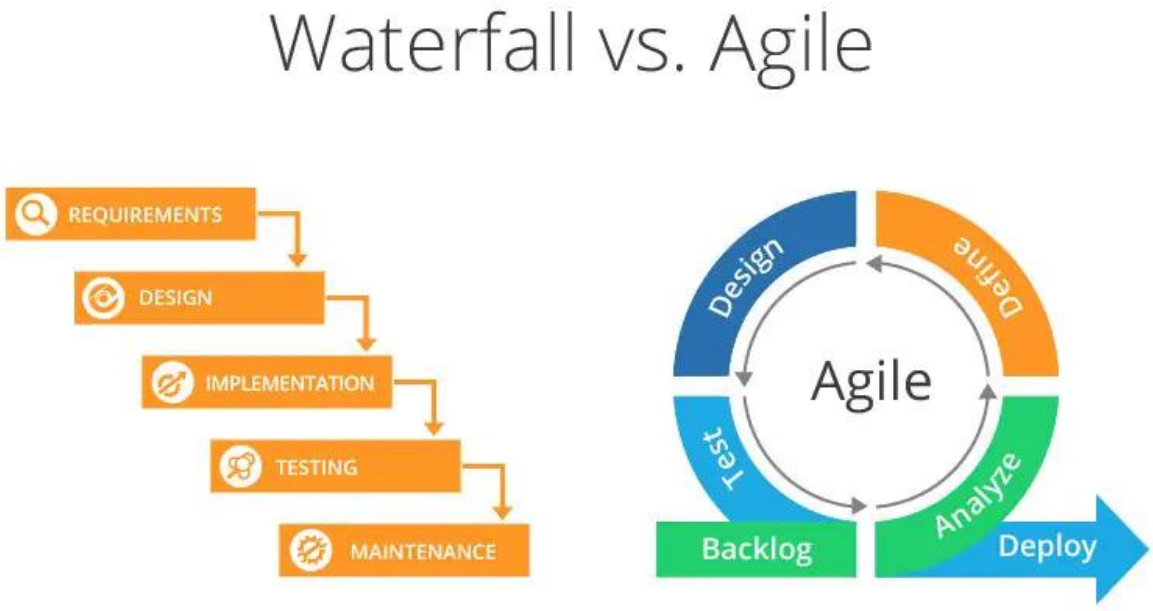


Figure 2-2: Comparison between Agile and Waterfall

1. **Flexibility and Adaptability to Changes:**
   a. The **Waterfall model** follows a rigid sequence of phases (requirements gathering, design, development, testing, and delivery), making it challenging to implement changes once a phase is completed.
   b. In contrast, **Agile** offers high flexibility by breaking the development process into smaller cycles (Sprints), enabling ongoing updates and adjustments.
2. **Risk Management:**
   a. With **Waterfall**, issues may remain undetected until the later stages, increasing the risk of project failure.
   b. Agile minimizes risks by delivering small, functional components of the system regularly and testing them continuously.
3. **Incremental Delivery:**
   a. The **Waterfall model** requires completing the entire system before deployment, which can delay the benefits.
   b. Agile allows incremental delivery, enabling stakeholders to gain value from usable components during the development process.

4. **Efficient Team Collaboration:**
    a. Agile emphasizes frequent team meetings and continuous review sessions to ensure effective collaboration and maintain project alignment, unlike Waterfall, where communication is more limited.

2.3.2 Reasons for Choosing Agile:

1. **Adaptability to Change:** Our project demands the ability to accommodate changes in requirements throughout the development process, making Agile an ideal choice.

2. **Organized Work Distribution :**Agile facilitates dividing the project into smaller tasks, making it easier to track progress and ensure consistent achievements.

3. **Early Value Delivery:** Agile allows delivering functional components early in the project timeline, ensuring gradual utilization of the system's features.

4. **Enhanced Team Efficiency:** Regular meetings and iterative feedback cycles in Agile contribute to faster issue resolution and improved quality of work.

2.3.3 Timeline

The project timeline is designed to ensure an organized and efficient workflow, aligning with the Agile methodology. Each phase of the project has been carefully planned with specific durations to allow for the smooth completion of tasks. By breaking the timeline into manageable sprints, the project team can focus on incremental progress while maintaining flexibility to adapt to changes or unforeseen challenges.

The Gantt chart below illustrates the planned timeline, showcasing the start and end dates for each sprint and the respective tasks associated with them. This structure ensures that all milestones are met within the defined project duration.

Tasks	Duration	Start	Finish
Sprint 1	14 days	01/09/2024	14/09/2024
Sprint 2	14 days	15/09/2024	28/09/2024
Sprint 3	21 days	29/09/2024	19/10/2024
Sprint 4	21 days	20/10/2024	09/11/2024
Sprint 5	21 days	10/11/2024	30/11/2024
Sprint 6	14 days	01/12/2024	14/12/2024
Sprint 7	15 days	15/12/2024	29/12/2024

Table 2-1:Timeline table

2.3.4 Gant chart

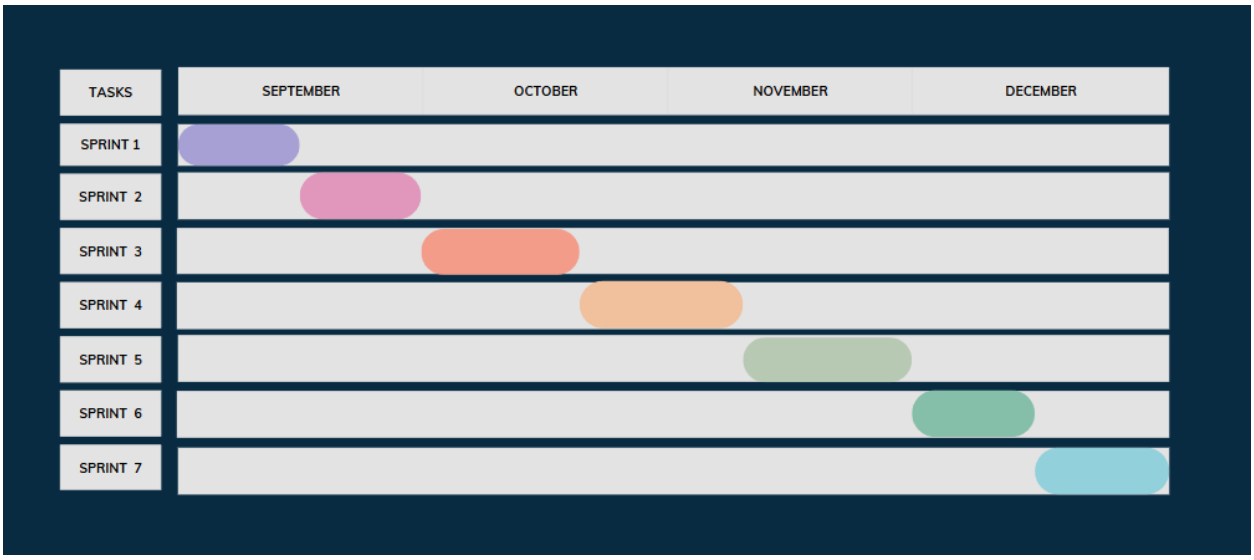


Figure 2-3: Gant chart

2.3.5 Project Development Plan (Work Sprints)

Sprint 1	Sprint 2	Sprint 3	Sprint 4
1.Gather and analyze requirements. 2.Define user stories and prioritize them. Create a project backlog	1.Design wireframes for key pages (Home, Login, Admin Panel). 2.Develop database schema for users, cars, and transactions. 3.Set up the development environment and version control. 4.Create the navigation structure 5.Start initial API design for backend services.	1.Implement user registration functionality. 2.Develop the login system with session management. 3.Build the Home page 4.Create the layout for the Admin Panel. 5.Test and validate initial functionalities	1.Develop car sales page with filters and search options 2.Create car rentals page 3.Build the maintenance services page with appointment booking. 4.Implement About Us page 5.Ensure responsive design across all pages
Sprint 5	**Sprint 6**	**Sprint 7**	
1.Implement Sales Management feature in the Admin Panel 2.Add Rental Management functionality for administrators. 3.Integrate payment gateway 4.Develop the car parts sales page	1.Create the Admin Car Parts Management page 2.Create and test the User Management page. 3.Conduct comprehensive testing of all features. 4.Fix bugs identified during testing. Develop the Cart page to allow users to manage their selections.	1.Ensure compliance with security standards 2.Finalize all project documentation 3.Conduct a final deployment test 4.Launch the system for end users	

Table 2-2:Work sprint table

Accompanying these seven sprints are recurring operations that are essential to ensuring the smooth progress of the project and the quality of the final system:

1. Conduct daily meetings with team members to discuss progress, address any obstacles, and ensure alignment with the planned sprint objectives.
2. Test the system after completing each sprint to verify that the implemented features work as intended and meet the defined requirements.
3. Regularly integrate code changes into the main repository and perform code reviews to maintain a quality
4. Update technical and user documentation during each sprint to reflect the progress and changes made to the system.

2.3.6 Functional Requirements

1. **User Management:**
   a. Allow users to create and manage their accounts.
   b. Implement login and logout functionalities with session management.
   c. Enable admins to manage user accounts, including adding, updating, and deactivating users
2. **Car Sales:**
   a. Display available cars for sale with detailed information
   b. Allow users to filter and search cars based on specific criteria
   c. Enable users to purchase cars directly through the platform.
3. **Car Rentals:**
   a. List cars available for rent.
   b. Allow users to book cars for specific time periods.
   c. Include rental history for users to track their bookings.
4. **Maintenance Services:**
   a. Allow users to book appointments for car maintenance services.
   b. Display available maintenance slots and service details.
   c. Car Parts Sales:
   d. Display car parts available for sale with details such as price and compatibility.
   e. Allow users to add items to a cart and proceed to checkout.
   f. Integrate the cart page to manage purchases.

5. **Cart Page:**
   a. Enable users to add, update, or remove items in the cart.
   b. Calculate and display the total cost dynamically.
   c. Allow users to proceed to payment from the cart.
6. **Admin Features:**
   a. Enable admins to manage car inventory, rentals, and sales.
   b. Include a dedicated page for managing car parts inventory (add, update, delete parts).
   c. Payment Integration:
   d. Ensure secure payment processing with proper encryption

2.3.7 Non-functional requirement

1. **Performance:**
   - Ensure the platform operates smoothly during local testing.
   - Maintain a response time of less than 2 seconds for key actions (e.g., searching, loading pages).
2. **Scalability:**
   - Design the system to support future deployment to a server without major architectural changes.
   - Ensure the system can be scaled to handle increased traffic and features when needed.
3. **Security:**
   - Use strong encryption for sensitive data, including passwords and transactions.
   - Implement secure authentication mechanisms, such as password hashing.
4. **Reliability:**

a. Ensure consistent functionality during local usage with minimal crashes or downtime.
b. Implement local backup mechanisms to prevent data loss during testing.

5. Usability:

a. Provide an intuitive and user-friendly interface for both end-users and admins.
b. Design the system to accommodate accessibility needs, including clear navigation and responsive layouts.

6. Compatibility:

a. Ensure compatibility with local browsers and devices during testing.
b. Test the system on multiple platforms (desktop and mobile) to verify responsiveness.

CHAPTER 3: System design

3.1 INTRODUCTION:

After covering the first stage, which is planning, and the second stage, which is analysis, we reached the stage of design, which was divided into more than one part, and the design was divided into several pages, each page with its own characteristics, interface and independent operations.

3.2 THE MOST IMPORTANT FORMS THAT EXPLAIN THE DESIGN FLOWCHART:

I. ER-Diagram.
II. Data Flow Diagram – DFD

3.2.1 ER-Diagram:

The ERD highlights the system's structure by showing entities such as UserInfo, SalesCars, SpareParts, RentCar, and their attributes, it defines the relationships between these entities, such as how a user interacts with purchases, rentals, spare parts, and payments, The diagram is essential for understanding the database design, ensuring data consistency, and supporting the system's overall functionality.

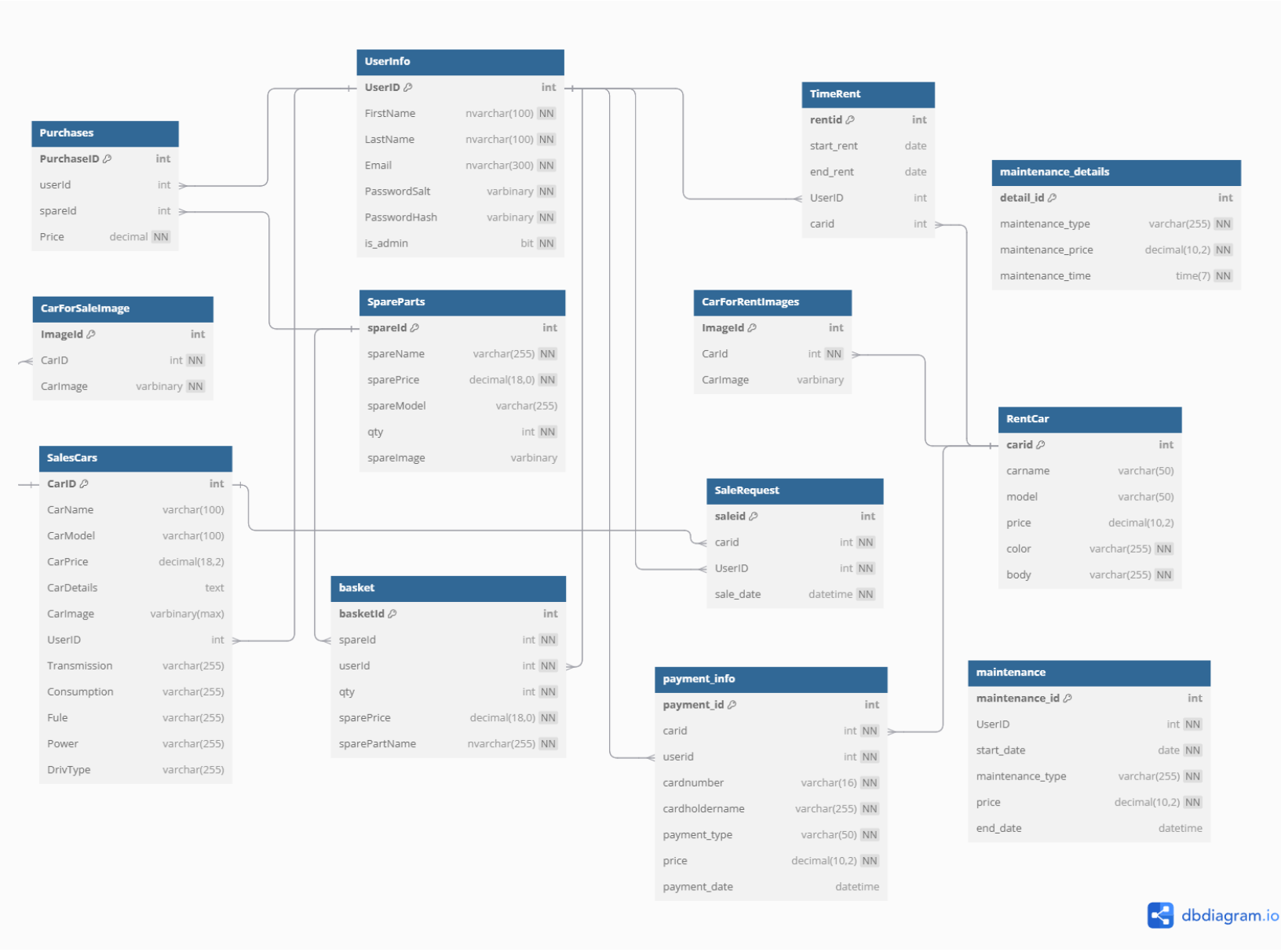


Figure 3-1:ER-diagram

3.2.2 Data flow diagram:

The following Data flow diagram illustrates the overall workflow and functionality of the system. The diagram demonstrates decision points and the flow of data between various modules to ensure a smooth user experience and efficient system operation.

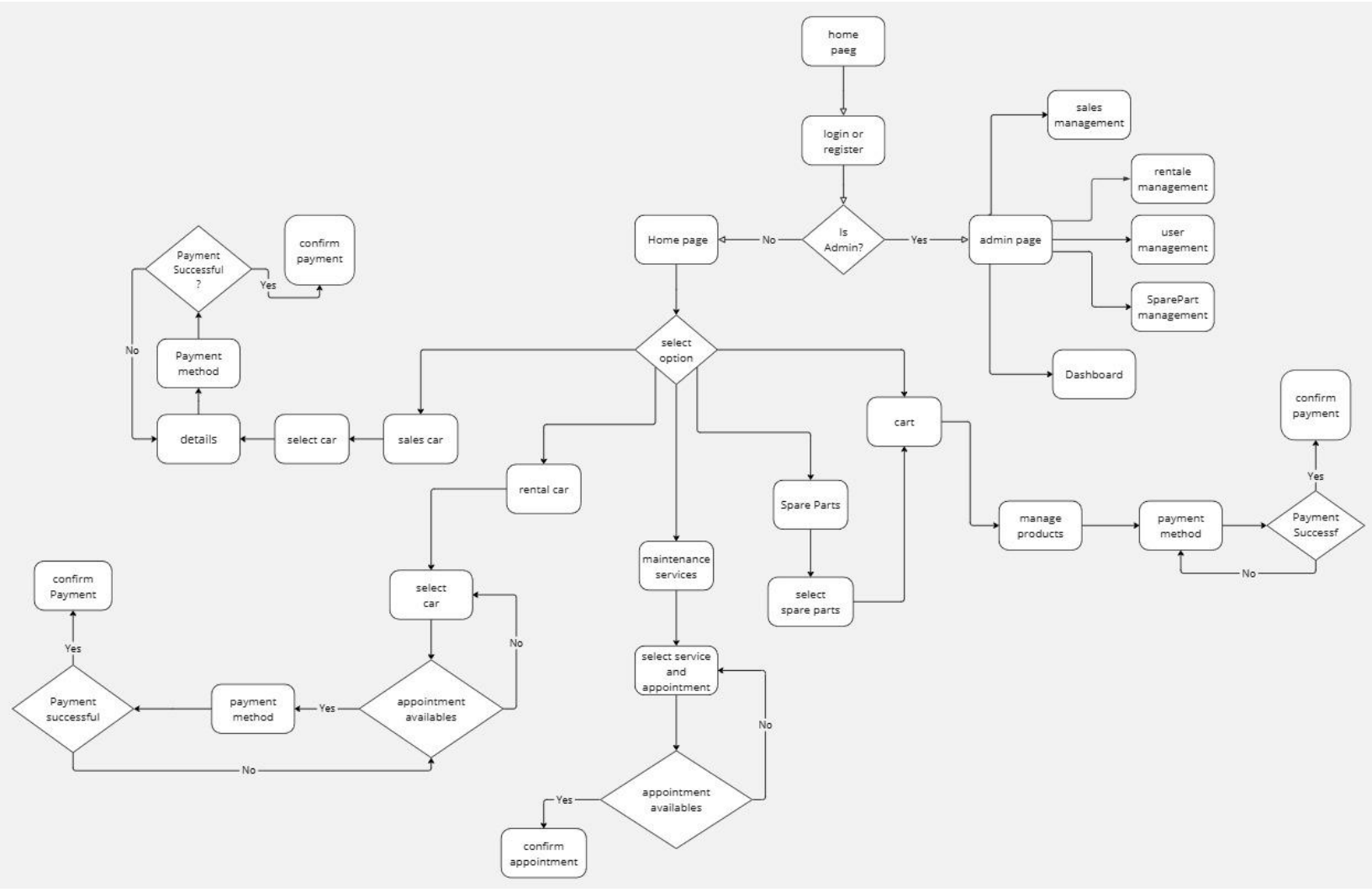


Figure 3-2:data flow diagram

CHAPTER 4: Implementation

4.1 INTRODUCTION

The implementation phase is the culmination of planning, analysis, and design efforts, where all concepts are transformed into a functional system. This chapter details the technologies, methodologies, and tools used in the development of the project, providing insights into the step-by-step execution and addressing challenges encountered along the way.

4.2 DEVELOPMENT ENVIRONMENT

4.2.1 Programming Languages and Frameworks:

1. **Backend**: ASP.NET Core (C#)
2. **Frontend**: HTML, CSS, JavaScript, and Bootstrap
3. **Database**: SQL Server
4. **Integration** Tools: APIs and Entity Framework

4.2.2 Development Tools:

1. Visual Studio code for coding and debugging
2. SQL Server Management Studio for database management
3. Azure for deployment

4.3 SYSTEM ARCHITECTURE

Overview:

The system was built on a client-server architecture to ensure scalability, maintainability, and efficiency. Key components include:

1. **Frontend**: A responsive web interface that allows users to interact with the system seamlessly.
2. **Backend**: Handles business logic, API integrations, and database interactions.
3. **Database**: Centralized storage for all data entities, optimized for retrieval and updates.

4.4 KEY MODULES

4.4.1 User Management

1. **Features:**
   a. User registration and authentication (role-based access control).
   b. Password encryption using salted hashing.
2. **Implementation:**
   a. C# for authentication logic, and Entity Framework for database interactions

4.4.2 Maintenance Scheduling

1. **Features**
   a. Users can schedule maintenance services.

2. **Implementation**
   a. Backend integration with notification services.
   b. Frontend forms for appointment booking.

4.4.3 Spare Parts Sales

1. **Features:**
   a. E-commerce functionality for spare parts.
   b. Payment gateway for transactions.
2. **Implementation:**
   a. APIs for order management and payment processing.
   b. Relational database design for inventory tracking.

4.5 DATABASE DESIGN

1. Entity-Relationship Model
2. Queries and Stored Procedures

4.6 INTERFACE DESIGN

The images below illustrate the key features and design of the user interface:

4.6.1 User interface:

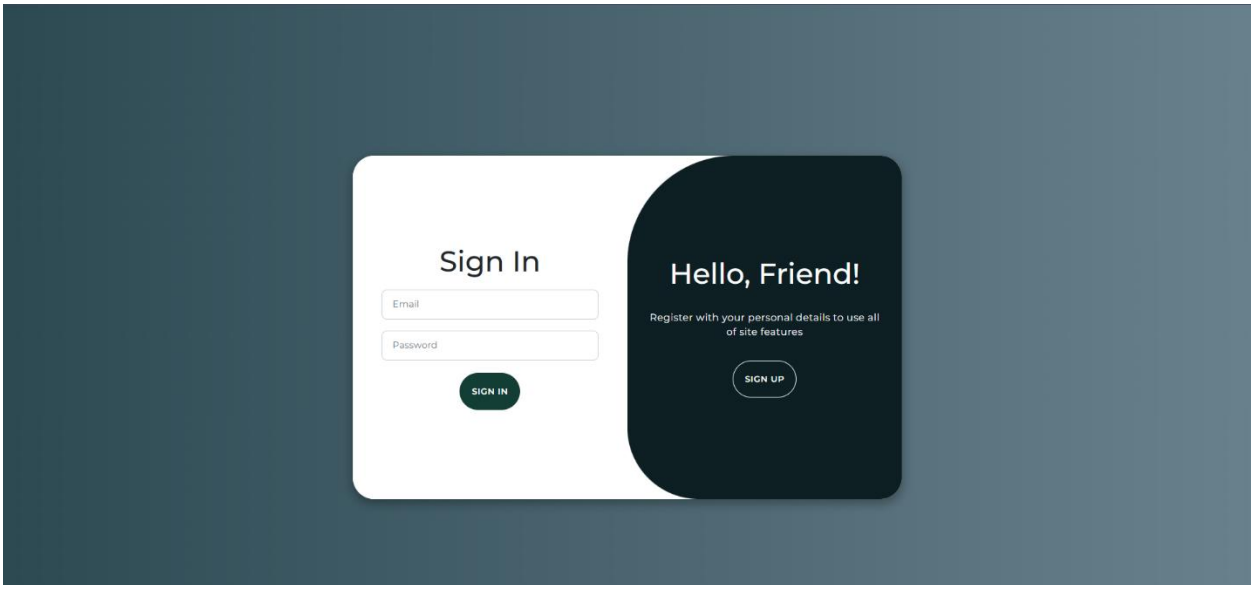


Figure 4-1 : This is a dynamic and interactive login and registration interface.

Figure 4-2:This homepage features a stunning parallax effect, where the background moves slower than the foreground content, creating a sense of depth and an engaging visual experience

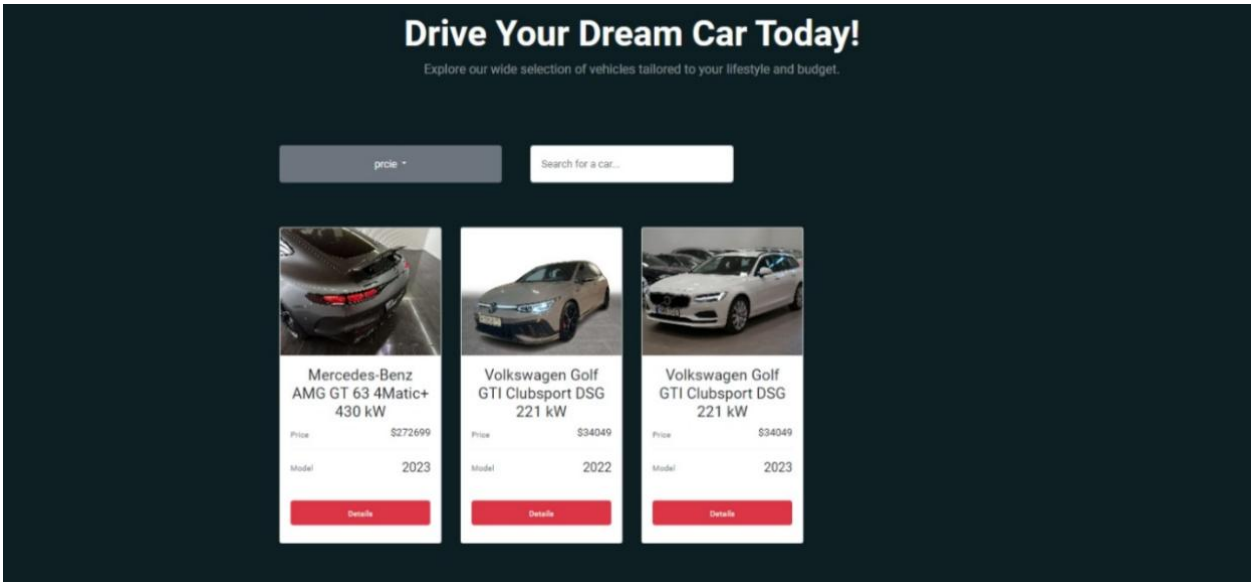


Figure 4-3:Sales car page

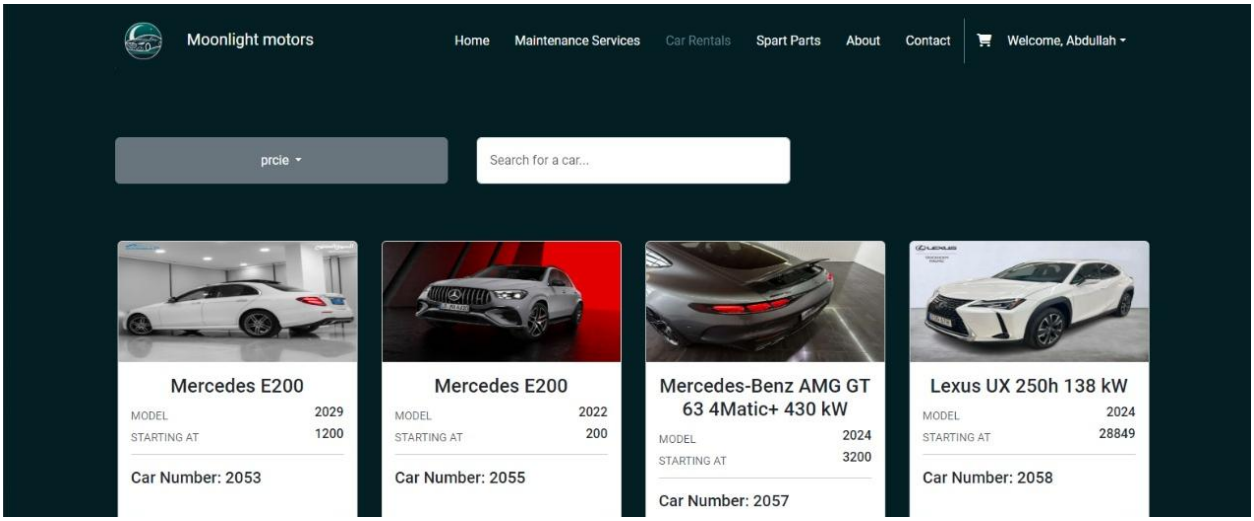


Figure 4-4: car rentals page

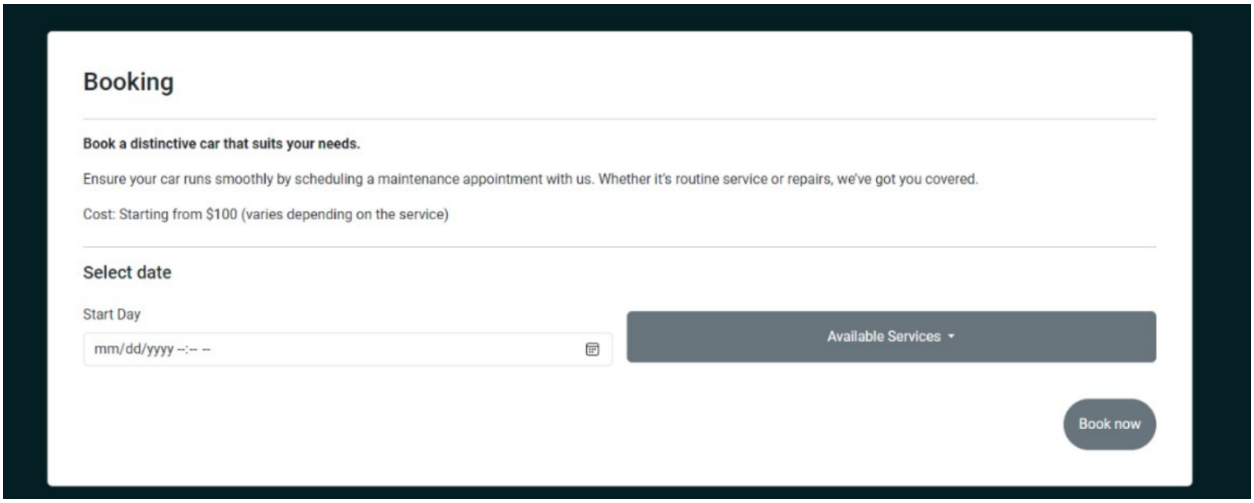


Figure 4-5Maintenance services

4.6.2 Admin interface:

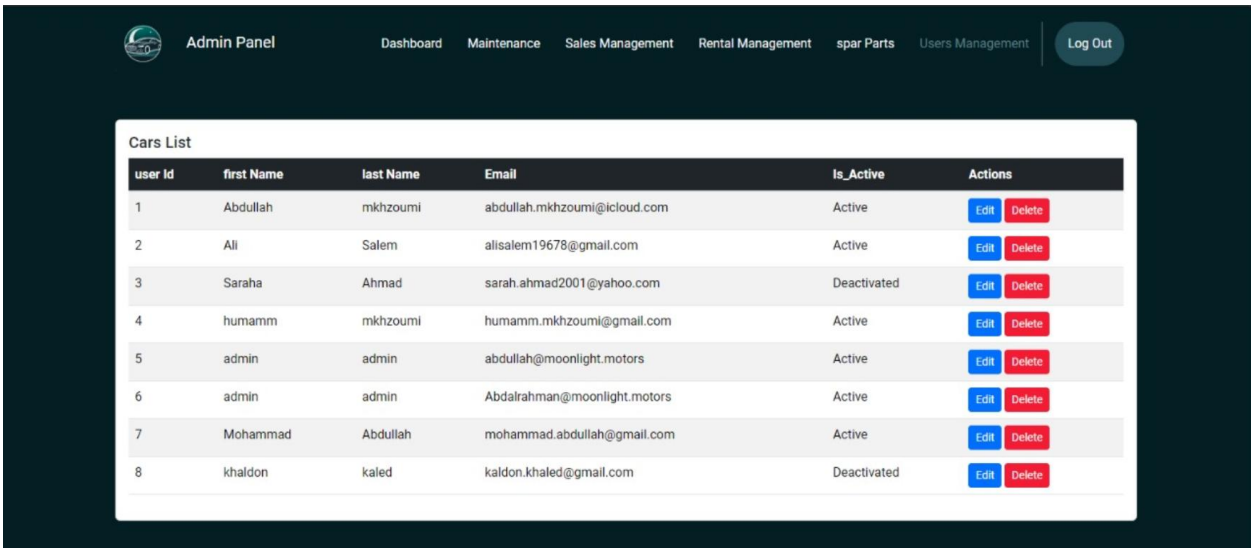


Figure 4-6:User management page

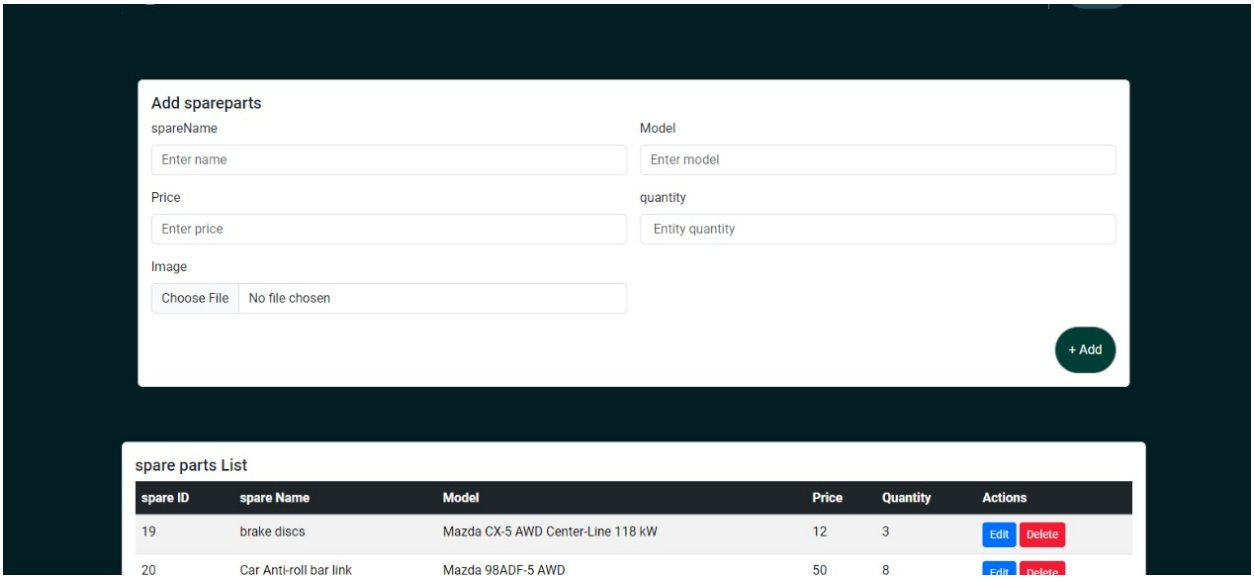


Figure 4-7: spare parts management

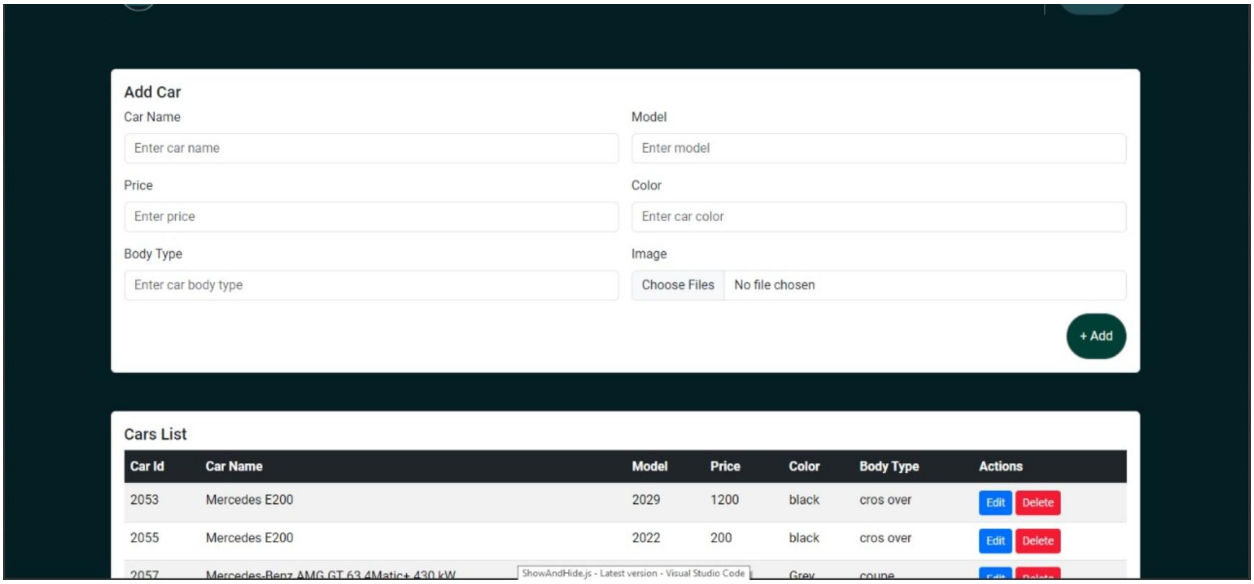


Figure 4-8:Rentals page management

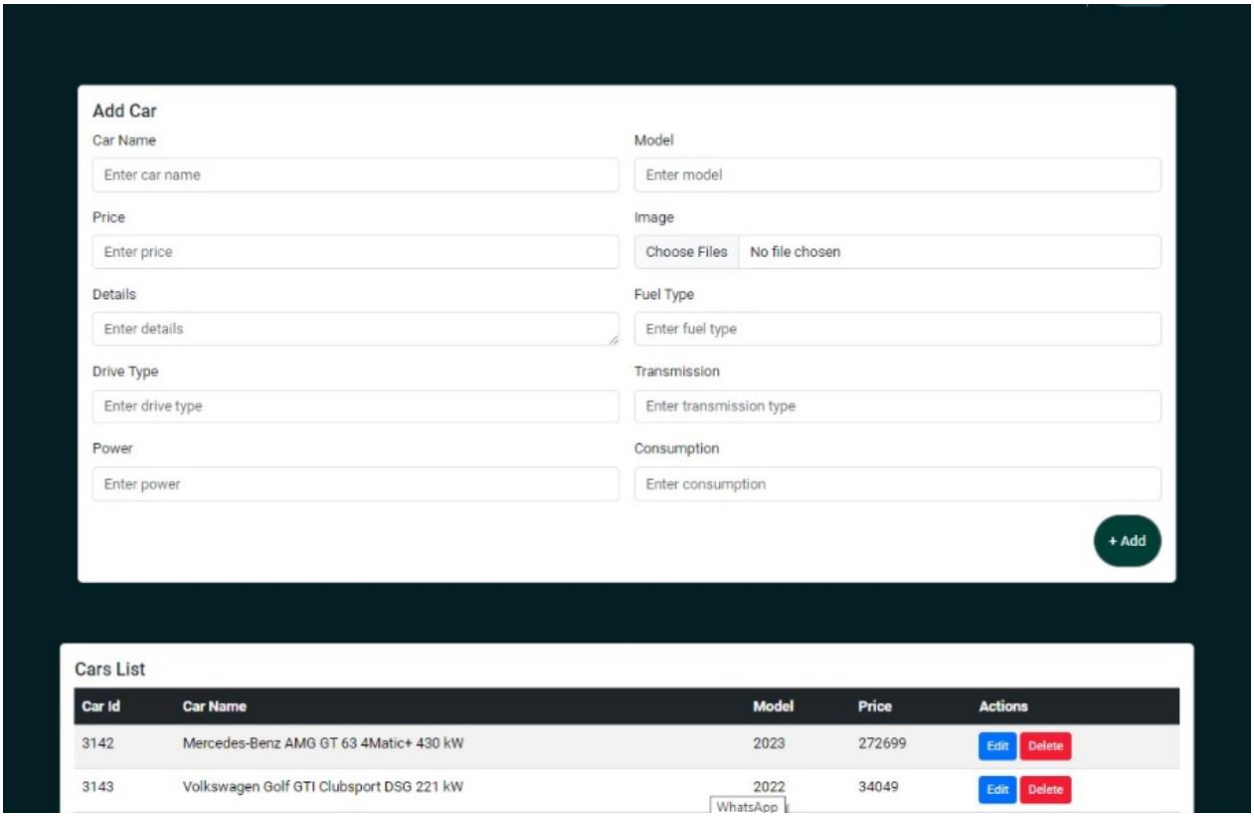


Figure 4-9:Sales page management

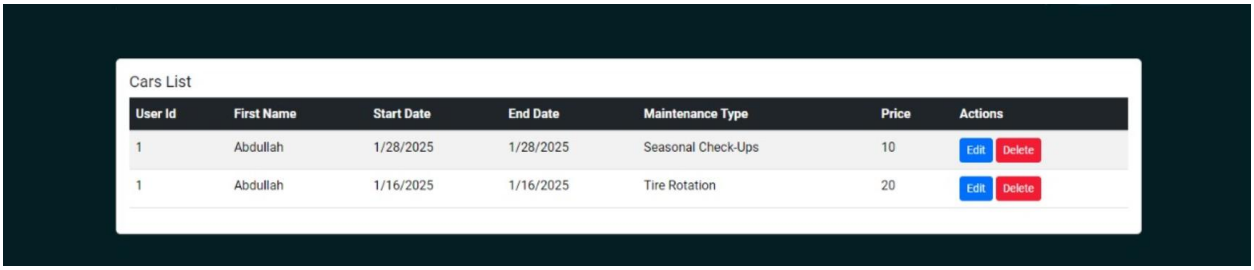


Figure 4-10:Maintenance managements

4.7 CHALLENGES AND SOLUTIONS

1. **Challenge:** Handling real-time data synchronization.
   Solution: Implemented caching mechanisms and database indexing.

2. **Challenge:** Ensuring user authentication and security.
   Solution: Used JWT tokens and HTTPS for secure communication.

3. **Challenge:** Integrating payment systems.
   Solution: Integrated a third-party payment gateway API.

4.8 TESTING AND DEBUGGING

1. **Unit Testing:** Ensured individual components function correctly.
2. **Integration Testing:** Validated interaction between modules.
3. **System Testing:** Confirmed that the entire system performs as intended.
4. **Tools Used:** Postman for API testing, Visual Studio debugging tools.

CHAPTER 5: Future work and References

5.1 FUTURE WORK

1. **Mobile Application Development:**

   Developing a dedicated mobile application for iOS and Android platforms to enhance user accessibility and convenience. The app will allow users to browse vehicles, book maintenance appointments, and manage their purchases on the go.

2. **Enhanced Customer Support:**

   Implementing an AI-powered chatbot to provide instant customer support. This feature will ensure quick responses to customer inquiries and improve overall satisfaction by offering seamless assistance 24/7.

3. **Maintenance Reminder System:**

   Introducing an automated maintenance reminder system that notifies customers of upcoming service appointments. The system will leverage customer records to schedule reminders based on vehicle maintenance history and usage.

4. **Customer Loyalty Program**

   Establishing a customer loyalty program to reward clients for their engagements, such as purchases and maintenance services. The program will feature a points-based system, encouraging repeat business and fostering long-term customer relationships.

5.2 REFERENCES

1. **C# Programming Guide**. Retrieved from https://learn.microsoft.com/en-us/dotnet/csharp
2. **ASP.NET Core Framework Documentation**. Retrieved from https://learn.microsoft.com/en-us/aspnet/core.
3. **JSON Web Token for Authentication.** Retrieved from https://jwt.io
4. **HTML, CSS and JavaScript**. Retrieved from https://www.w3schools.com.
5. **Modern JavaScript Tutorial**. Retrieved from https://javascript.info
6. **Frontend Framework Documentation**. Retrieved from https://getbootstrap.com.
7. **Deployment and Hosting on Microsoft Azure**. Retrieved from https://azure.microsoft.com.
8. **SQL Server Management Studio Documentation.** Retrieved from https://learn.microsoft.com/en-us/sql/ssms.