|  |
| --- |
| **GARAGE MANAGEMENT SYSTEM** |

A Mini Project Report

submitted by

ADITHYA DAS M P (MES24MCA-2002)

to the APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the Degree

of

Master of Computer Applications



**Department of Computer Applications**

MES College of Engineering

Kuttippuram, Malappuram – 679582

November, 2025

**Declaration**

I undersigned hereby declare that the project report GARAGE MANAGEMENT SYSTEM submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a Bonafide work done by me under supervision of Mr. Hyderali K., HoD , Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

ADITHYA DAS M P (MES24MCA-2002)

06/08/2025

DEPARTMENT OF COMPUTER APPLICATIONS

MES COLLEGE OF ENGINEERING, KUTTIPPURAM



**CERTIFICATE**

This is to certify that the report entitled **GARAGE MANAGEMENT SYSTEM** is a bonafide record of the Mini Project work during the year 2025-26 carried out by **ADITHYA DAS M P(MES24MCA-2002)** submitted to the APJ Abdul Kalam Technological University, in partial fulfilment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor Head of The Department

**Acknowledgment**

I would like to begin by expressing my deep sense of gratitude to the Almighty, whose blessings and guidance have been a constant source of strength and inspiration throughout the completion of my project work titled “**GARAGE MANAGEMENT SYSTEM**”, undertaken as part of the requirements for the degree of Master of Computer Applications at MES College of Engineering, Kuttippuram, under APJ Abdul Kalam Technological University ,Kerala.

I owe my sincere thanks to my project guide, Mr. Hyderali K., HoD, Department of Computer Applications, MES College of Engineering, Kuttippuram, for her valuable guidance, constructive suggestions, and constant encouragement at every stage of the project. Her insights and expertise have been instrumental in shaping this work into its final form.

I also express my heartfelt gratitude to the Head of the Department of Computer Applications for providing me with the necessary facilities, encouragement, and academic support during the course of my project. My sincere thanks also go to all the faculty members and technical staff of the Department of Computer Applications for their timely advice, cooperation, and moral support.

I am especially thankful to my friends and classmates for their valuable feedback, collaboration, and encouragement, which motivated me to enhance the quality of this work.

Finally, I express my deepest gratitude to my family members for their unconditional love, patience, and constant support, which gave me the confidence to successfully complete this project.

ADITHYA DAS M P (MES24MCA-2002)

**Abstract**

The project Garage Management System (GMS) is a modern web-based application developed to efficiently manage and streamline all operational activities within a garage. It addresses the limitations of traditional garage setups, such as poor data storage, inefficient workforce distribution, and lack of real-time tracking. The system includes multiple user roles—Admin, Manager, Supervisor, Mechanic, Security, and Customer—each granted specific access privileges based on their responsibilities.

The Supervisor can monitor spare parts inventory, track vehicles under service, and identify those requiring maintenance notifications. The system also records mechanic working hours, supports payment processing for completed services, and maintains a detailed record of spare parts sold. These features ensure improved transparency, accountability, and efficiency in day-to-day operations.

Developed using HTML, CSS, JavaScript, PHP, and MySQL Server, the Garage Management System offers a user-friendly interface and ensures secure, organized data handling. It minimizes manual errors, enhances coordination among employees, and provides real-time insights into garage performance.

By automating and centralizing garage operations, this project enhances service quality, reduces administrative workload, and improves overall customer satisfaction. The Garage Management System thus provides a reliable, scalable, and efficient digital solution for modernizing garage management.

**Table of Contents**

[List of Figures viii](#_Toc180355201)

[List of Tables ix](#_Toc180355202)

[Chapter 1. Introduction 3](#_Toc180355203)

[1.1 Motivation](#_Toc180355204) 4

[1.2 Objectives 4](#_Toc180355205)

[1.3 Contributions 6](#_Toc180355206)

[1.4 Report Organization 6](#_Toc180355207)

[Chapter 2. System Study 8](#_Toc180355208)

[2.1 Existing System 8](#_Toc180355209)

[2.2 Proposed System 9](#_Toc180355210)

[2.3 Functionalities of Proposed System 9](#_Toc180355211)

[Chapter 3. Methodology 10](#_Toc180355212)

[3.1 Introduction 10](#_Toc180355213)

[3.2 Software Tools 11](#_Toc180355214)

[3.2.1 Html,Css,Javascript 12](#_Toc180355215)

[3.2.2 Php 13](#_Toc180355216)

[3.2.2 MySQL 13](#_Toc180355216)

[3.3 Module Description 14](#_Toc180355214)

[3.3.2 User Module 14](#_Toc180355219)

[3.3.2 Supervisor Module 15](#_Toc180355219)

[3.3.3 Mechanic Module 15](#_Toc180355218)

[3.3.4 Manager Module 15](#_Toc180355219)

[3.4 User Story 16](#_Toc180355220)

[3.5 Product Backlog 20](#_Toc180355221)

[3.6 Project Plan 22](#_Toc180355222)

[3.7 Sprint Backlog 24](#_Toc180355223)

[3.8 Database Design 27](#_Toc180355224)

[Chapter 4. Results and Discussions 31](#_Toc180355225)

[4.1 Results 31](#_Toc180355226)

[Chapter 5. Conclusion 34](#_Toc180355227)

[References 35](#_Toc180355228)

[Appendix 36](#_Toc180355229)

[Appendix A Data Flow Diagram 36](#_Toc180355230)

[Appendix B ER Diagram 39](#_Toc180355231)

[Appendix C Source Code 40](#_Toc180355232)

Appemdix D Screenshot…………………………………………………………………….81

List of Figures

[**Figure**](#_Toc180355233) **4.1**: Index Page 30

**Figure 4.2:** Manager dashboard 31

**Figure 4.3**: Mechanic dashboard 31

List of Tables

**Table 3.1:** List the software tools or languages used for the project development 11

**Table 3**.**2:** User Story…………………………………………………………………………...16

**Table 3.3:**Product Backlog……….………………………………………………………….....20

**Table 3.4:**Project Plan……………………………………………………………………….....22

**Table 3.5:**Sprint Backlog……………………………………………………………………....25

**Table 3.6:**Database Design………………………………………………………………….....27

# Introduction

The rapid advancement of digital technologies and web-based applications has transformed service industries, including the automobile sector. Traditional garage operations often rely on manual processes, paper records, and verbal communication, which result in inefficiency, data loss, and poor coordination. To overcome these challenges, the Garage Management System (GMS) is developed as a centralized, web-based solution for managing all garage activities efficiently.

The system automates major functions such as vehicle service tracking, spare parts inventory management, mechanic work-hour monitoring, service notifications, and payment processing. It provides role-based access for Admin, Manager, Supervisor, Mechanic, Security, and Customer, ensuring each user can perform tasks relevant to their responsibilities. This structured approach improves workflow, transparency, and accountability across all levels of operation.

By integrating all essential modules into one platform, the GMS enhances operational accuracy and enables real-time access to information. Supervisors can monitor ongoing jobs and inventory, Managers can analyze performance, and Customers receive timely updates about their vehicles.

The project is developed using HTML, CSS, JavaScript, PHP, and MySQL, offering a secure, interactive, and user-friendly environment. While HTML and CSS ensure an intuitive interface, JavaScript improves interactivity, PHP manages backend logic, and MySQL handles data storage and retrieval.

In conclusion, the Garage Management System addresses the drawbacks of traditional methods by introducing automation, structured data management, and improved service coordination. It minimizes manual effort, reduces errors, optimizes resources, and enhances customer satisfaction — creating a modern, efficient, and transparent garage management experience.

**1.1 Motivation**

In many traditional garages, operations such as service tracking, inventory management, and mechanic scheduling are handled manually. This often leads to problems like misplaced records, miscommunication among staff, and delays in vehicle servicing. The absence of a centralized system results in inefficiency, customer dissatisfaction, and difficulty in monitoring overall performance. Managing multiple tasks manually also increases the chances of human error, making it challenging to maintain accuracy and consistency in daily activities.

The motivation behind developing the Garage Management System (GMS) is to overcome these issues by introducing a structured, digital platform that streamlines all garage operations. Through this system, users can efficiently manage vehicle services, monitor mechanic work hours, track spare parts inventory, and handle payments in a transparent manner. It enables supervisors and managers to have better control over operations, helping them make informed decisions while improving productivity and accountability within the garage.

By implementing GMS, garages can reduce dependency on paper-based records and move toward an automated environment that saves time and effort. The system aims to enhance customer satisfaction by ensuring faster, more reliable service and creating a well-organized workflow. Overall, the project is motivated by the need to bring modernization, accuracy, and efficiency to the garage management process through the use of web-based technology.

## 1.2 Objectives

The main problem this mini project focuses on is the lack of automation and organization in traditional garage operations, which results in poor data management, inefficient workforce allocation, and lack of real-time tracking. Manual methods make it difficult for garages to monitor vehicles under service, track mechanic working hours, maintain spare parts inventory, and ensure timely customer updates. These limitations reduce productivity, increase errors, and lead to customer dissatisfaction.

The hypothesis of this project is that by developing an independent Garage Management System (GMS) using web technologies such as HTML, CSS, JavaScript, PHP, and MySQL Server, garages can efficiently manage their operations through a centralized digital platform. This system will allow managers, supervisors, and mechanics to coordinate tasks effectively, maintain transparency in operations, and provide better service to customers through automation and real-time updates.

The objectives of the project can be summarized as follows:

1. To design and develop a web-based platform that automates and manages all activities in a garage, including vehicle service tracking, spare parts management, and employee monitoring.
2. To provide role-based access for Admin, Manager, Supervisor, Mechanic, Security, and Customer to ensure secure and efficient workflow management.
3. To maintain real-time records of vehicles under service, spare parts inventory, and service history for easy tracking and reporting.
4. To enable the Supervisor to monitor mechanic working hours and manage notifications for vehicles that require maintenance.
5. To implement an efficient payment module for processing repair and service transactions securely.
6. To reduce manual errors and improve coordination among different users within the garage.
7. To enhance customer satisfaction by providing timely updates and ensuring transparency in service operations.

## 1.3 Contributions

**. System Automation** – Automates all major garage operations such as vehicle service tracking, mechanic work-hour recording, spare parts inventory management, and payment processing, thereby reducing manual effort and human errors.

**Role-Based Access Control** – Provides specific access privileges for Admin, Manager, Supervisor, Mechanic, Security, and Customer to ensure secure and efficient workflow management.

**Spare Parts Inventory Management** – Allows supervisors to monitor stock levels, track parts sold, and manage inventory efficiently, preventing shortages and ensuring timely availability.

**Vehicle Service Tracking** – Enables real-time tracking of vehicles under maintenance, along with details of services performed, estimated completion time, and mechanic assignments.

**Mechanic Performance Monitoring** – Records working hours and assigned tasks for each mechanic to improve accountability and workforce utilization.

**Payment Processing System** – Supports secure and reliable payment management for completed repair and maintenance services, ensuring transparent transactions.

**Customer Satisfaction and Transparency** – Enhances customer experience by providing real-time updates, service reminders, and clear billing details, leading to better service quality and trust.

**1.4 Report Organization**

The project report is organized into five chapters:

**Chapter 1**: Introduction – Provides the background, motivation, objectives, contributions, and organization of the report.

**Chapter 2**: System Study – Describes the existing garage system, its limitations, and the requirements of the proposed Garage Management System along with its functionalities.

**Chapter 3**: Methodology – Explains the methodology used for implementation, including system design, architecture, software tools, and detailed descriptions of each module.

**Chapter 4**: Results and Discussions – Presents the implementation results, screenshots of the developed system, and a discussion of the outcomes with respect to the project objectives.

**Chapter 5**: Conclusion and Future Work – Summarizes the overall contributions, highlights the benefits of the developed system, and suggests possible directions for future enhancements.

# System Study

The project under consideration, Garage Management System (GMS), focuses on creating a digital solution to manage and streamline the daily operations of a garage. The system is designed to simplify service management, spare parts tracking, employee coordination, and customer communication within a single platform. It provides an efficient way to record, monitor, and control all garage-related activities, thereby reducing manual work and improving overall productivity.

Developed using HTML, CSS, JavaScript, PHP, and MySQL Server, the system offers a secure and user-friendly environment for different roles such as Admin, Manager, Supervisor, Mechanic, Security, and Customer. Each role has specific access and responsibilities, ensuring organized workflow and data transparency. The project aims to enhance service efficiency, eliminate human errors, and ensure accurate record management, making garage operations faster, more reliable, and technology-driven.

**2.1 Existing System**

In the existing system, most garage operations are carried out manually, which leads to inefficiency and frequent errors. The details of vehicle services, spare parts used, mechanic assignments, and payments are usually recorded in registers or spreadsheets. This manual process often causes data duplication, misplacement of records, and difficulty in tracking the service history of vehicles. Communication between customers and garage staff is limited, resulting in delays in service updates and lack of transparency. Moreover, managing spare parts inventory becomes challenging, leading to either stock shortages or overstocking. The absence of a centralized system also makes it difficult for managers to monitor employee performance and maintain accurate financial records. Overall, the traditional approach consumes more time, requires more manpower, and lacks reliability and real-time tracking.

**2.2 Proposed System**

The proposed Garage Management System (GMS) aims to automate and streamline all the major operations of a garage in a centralized digital platform. It provides a secure login system for different roles such as Admin, Manager, Supervisor, Mechanic, Security, and Customer, ensuring that each user has access only to relevant features. The system maintains detailed records of vehicle services, spare parts, customer details, and payments, thereby improving accuracy and reducing manual effort. It allows real-time service tracking, automatic billing, and digital report generation for better decision-making. The GMS also includes an efficient inventory management module that monitors spare parts stock levels and provides timely alerts for restocking. By integrating all operations under a single system, the proposed solution enhances transparency, saves time, and increases overall productivity and customer satisfaction.

**2.3 Functionalities of Proposed System**

1. **User Authentication** : Allows secure login and access based on user roles such as Admin, Manager, Supervisor, Mechanic, Security, and Customer.
2. **Admin Panel** :Enables the Admin to manage users, assign roles, and control access to different modules.
3. **Manager Dashboard** : Provides an overview of garage operations, employee activities, and financial summaries.
4. **Supervisor Module** : Allows supervisors to monitor ongoing vehicle services, manage spare parts inventory, and track mechanic performance.
5. **Mechanic Module** : Displays assigned service tasks, records working hours, and updates job completion status.
6. **Customer Module** : Enables customers to view vehicle service details, receive notifications, and make secure online payments.
7. **Inventory Management** – Tracks availability, purchase, and sale of spare parts with automatic stock updates.

# Methodology

Developing a web-based application like the Garage Management System (GMS) requires the use of an effective software development methodology to ensure timely completion, high quality, and fulfillment of project requirements. A well-defined methodology provides a structured approach to planning, designing, developing, and testing the system. Since the GMS involves multiple user roles, real-time service updates, and continuous management of garage operations, the Agile methodology was adopted for its development

**3.1 Introduction**

The Agile methodology is an iterative and incremental approach to software development. Unlike traditional models, Agile focuses on flexibility, adaptability, and continuous collaboration between developers and end-users. The development process is divided into short cycles called sprints, where each sprint results in a working and testable part of the system.

This approach is well-suited for the Garage Management System, as it involves multiple modules like user management, vehicle service tracking, inventory management, and payment processing, which may require frequent updates based on user feedback. Agile allows developers to make quick modifications, incorporate new requirements efficiently, and ensure that the system aligns with user expectations. Regular testing and review at each stage help improve software quality, minimize errors, and ensure faster delivery of usable features.

**3.2 Software Tools**

The following software tools and technologies were used to develop the Garage Management System (see Table 3.1), For example,

**Table 3.1:** List the software tools or languages used for the project development

|  |  |
| --- | --- |
| Operating System | Windows /Linux |
| Front End | HTML, CSS,JAVASCRIPT |
| Back End | PHP |
| Database | MySQL Server |
| IDE | Visual Studio Code |
| Version Control | Git |

## Front End(HTML,CSS,JAVASCRIPT)

## HTML, CSS, and JavaScript were chosen for the front end because they provide a foundational, standards-based framework for building web applications. HTML structures the content, CSS styles the interface, and JavaScript adds interactivity, enabling dynamic and responsive user experiences. Their combined use ensures flexibility, ease of maintenance, and compatibility across browsers.

## Back End(PHP)

PHP is a server-side scripting language used to develop dynamic web applications. It processes data on the server, interacts with databases like MySQL, and generates HTML content sent to the browser. PHP is widely used for tasks such as user authentication, form handling, and content management, making web applications interactive and data-driven.

* **Database(MySQL**)

MySQL is used as the database management system for storing and organizing all project data. It securely stores information such as user details, appointments, spare parts, and transaction records.MySQL was chosen for its reliability, scalability, and compatibility with PHP, ensuring smooth data retrieval and high system performance

* **IDE(Visual Studio Code)**

Visual Studio Code was chosen as the development environment for its wide range of extensions, debugging features, and seamless integration with Git and JavaScript frameworks.

* **Version Control (Git & GitHub)**

Git and GitHub were used for version control, enabling collaborative development,

version tracking, and backup of the project source code.

**3.2.1 HTML,CSS,JAVASCRIPT**

* **HTML (HyperText Markup Language) :** was used to design the structural foundation of the Garage Management System. It defines the layout and content of each web page, including text, images, forms, and links. Since the system involves multiple modules such as login, dashboard, and service records, HTML provides a clear and organized structure that ensures smooth navigation and easy accessibility across all user roles.
* **CSS (Cascading Style Sheets) :** was used to enhance the appearance and presentation of the web pages created using HTML. It controls the layout, colors, fonts, and spacing to ensure a consistent and visually appealing user interface. In the Garage Management System, CSS helps maintain a professional design and responsive layout, making the platform accessible across different devices and screen sizes.
* **JavaScript** : In the Garage Management System, JavaScript is used to make the web application interactive, dynamic, and user-friendly. It enables real-time updates, allowing customers to track vehicle service status, supervisors to monitor inventory, and mechanics to update job progress without reloading the page. JavaScript handles form validation, ensuring that user inputs such as registration details, service bookings, and payments are accurate before submission. It also supports interactive UI elements like dropdowns, modals, date pickers, and notifications, enhancing the overall user experience. Additionally, JavaScript works alongside PHP and MySQL to fetch, insert, and update data asynchronously, reducing server load and improving performance.

**3.2.2 PHP**

PHP was chosen as the backend programming language for developing the Garage Management System due to its strong compatibility with web technologies and databases. It efficiently handles server-side operations such as user authentication, session management, and dynamic content generation. PHP interacts seamlessly with the MySQL database to store and retrieve information related to users, services, and payments.Its ability to embed directly within HTML makes it easier to design interactive and data-driven web pages. The language’s open-source nature, wide community support, and simple syntax make development faster and maintenance easier. Overall, PHP ensures secure, reliable, and efficient backend functionality for managing all garage operations smoothly**.**

**3.2.3 MySQL**

MySQL was selected as the database for the Garage Management System because it is a relational database management system (RDBMS) that stores data in structured tables with predefined schemas. This structure is ideal for managing organized datasets such as users, vehicles, appointments, spare parts, and payments. MySQL ensures data integrity through relationships, primary and foreign keys, and supports complex queries for tracking services, managing inventory, and generating reports. It also provides high reliability, secure transactions, and scalability, making it suitable for web-based applications where consistent and structured data storage is crucial.

**3.3 Module Description**

A software system is often divided into smaller manageable components called modules. Each module is designed to handle a specific part of the overall system functionality. This modular approach improves maintainability, scalability, and clarity of the system. By separating responsibilities, modules allow developers to work on different parts of the application independently. This also makes debugging, testing, and future enhancements more efficient and systematic.

The Garage Management System is designed for various user groups, each having specific roles and access levels. The users and their characteristics are as follows:

**Manager**: Monitors overall garage operations and generates reports. Requires familiarity with service workflows and basic digital skills.

**Superviso**r: Manages inventory, service scheduling, and staff assignments. Requires knowledge of garage operations and basic computer use.

**Mechanic**: Updates service status and logs work hours. Minimal training required in how to login and update service records.

**Customer:** Can book services, view service status, and make payments. Interface designed to be user-friendly; no training required for basic usage.

**3.3.1 User Module**

The User Module manages all customer-side interactions, providing a seamless garage service experience. Customers can securely register and log in with their credentials, ensuring safe access to the system. Once authenticated, customers can book vehicle services, view the status of their vehicles, and track service progress in real-time. The module also supports payment processing for completed services, providing a secure and reliable method for bill settlement. Additionally, customers can view service history, receive notifications for upcoming maintenance, and communicate with garage staff through a simple messaging interface. The module is designed to be user-friendly, allowing customers to easily navigate the system, schedule appointments, and make payments without requiring any prior training. This module enhances transparency, improves customer satisfaction, and ensures timely service delivery

**3.3.2 Supervisor Module**

The Supervisor Module is designed to streamline garage operations and staff management. Supervisors can assign jobs to mechanics, monitor the progress of ongoing services, and view assigned jobs for each mechanic. The module allows supervisors to track inventory of spare parts, ensuring availability for repairs and notifying management when stock is low. Supervisors can also generate service schedules, manage staff assignments, and monitor mechanic work hours, ensuring efficient workflow and proper task distribution. With its organized dashboard, supervisors can make informed decisions, reduce delays, and maintain smooth operations across the garage. The interface is intuitive, requiring only basic knowledge of garage operations and computer usage.

**3.3.3 Mechanic Module**

The Mechanic Module enables mechanics to efficiently manage their assigned tasks and record work details. Mechanics can view assigned jobs, request spare parts when needed, and update the status of each job as it progresses. The module also allows mechanics to log their working hours, providing supervisors and managers with accurate performance data. Designed for ease of use, the interface requires minimal training, focusing on quick updates and task completion. By centralizing work tracking and job updates, this module ensures accountability, improves communication with supervisors, and helps maintain timely service delivery.

**3.3.4 Manager Module**

The Manager Module focuses on overall garage oversight and administrative operations. Managers can view and update customer information, track vehicle service status, and manage the inventory of spare parts, including adding new stock or updating existing quantities. The module also allows managers to generate reports on garage performance, such as completed services, revenue generated, and spare parts usage. Additionally, managers can create and manage bills for services rendered, ensuring accurate financial tracking. The dashboard provides real-time updates and analytical insights, enabling managers to make data-driven decisions, optimize workflows, and enhance overall efficiency. Basic knowledge of service workflows and digital skills is required to operate this module effectively.

**3.4 User Story**

A user story represents a short, simple description of a feature told from the perspective of the end user. For example, in the Garage Management System, a typical user story could be: “As a customer, I want to book vehicle service appointments so that I can get my vehicle serviced conveniently,” or “As a manager, I want to generate reports and manage staff so that I can oversee garage operations efficiently.” These user stories help define the needs of both customers and administrators, guiding the development process in an Agile methodology by focusing on real-world user requirements and expected outcomes.

**Table 3.2**: User Story

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story ID** | **As a type of User** | **I want to** | **So that i can** |
| 1 | USER | Register | Create an account and access garage services |
| Login | Securely access my account with correct credentials |
| Book Vehicle Services | Schedule my vehicle service online without visiting manually. |
| View Service Status | Track the progress of my vehicle being serviced. |
| Pay Bill | Pay the bill for vehicle service conveniently online or at the garage once the service is completed. |
| 2 | Supervisor | Assign jobs to mechanics | Ensure each job is assigned to the right mechanic efficiently |
| View all assigned jobs | Track the progress of jobs and monitor work allocation |
| Update job status | Keep records accurate and inform management of ongoing work |
| Search/filter jobs by mechanic or status | Quickly find specific jobs and manage them effectively |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | Manager | View customer information | Keep track of customer details and service history |
| Update customer information | |  | | --- | |  |  |  | | --- | |  |   Ensure customer records are accurate and up-to-date |
| Add new spare parts to inventory | |  | | --- | | Maintain proper stock levels and prepare for upcoming jobs |  |  | | --- | |  | |
| |  | | --- | |  |  |  | | --- | | Update spare parts stock | | Track available parts and avoid shortages during repairs |
| |  | | --- | |  |  |  | | --- | | View reports on jobs, revenue, and stock | | Make informed decisions about garage operations and performance |
| Generate bills for completed jobs | Ensure customers are billed correctly and payments are recorded |
| Monitor overall garage workflow | Ensure smooth operations and timely service delivery |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | Mechanic | |  | | --- | |  |  |  | | --- | | View assigned jobs | | Know which tasks I need to complete |
| |  | | --- | |  |  |  | | --- | | Request spare parts | | |  | | --- | |  |  |  | | --- | |  |   Ensure I have the necessary parts to complete repairs |
| |  | | --- | |  |  |  | | --- | | Update job status | | |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | Inform supervisors and managers about the progress of each job | |  |  | | --- | |  | |
| |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | Add notes to jobs | | | |  | | --- | | Provide details about work done or issues encountered |  |  | | --- | |  | |
| |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | Mark job as completed | | | Indicate that a repair is finished and ready for billing |
| |  | | --- | |  |  |  | | --- | | View job history | | Check previous work done for reference or follow-ups |

**3.5 Product Backlog**

The product backlog is a comprehensive and prioritized list of all the features, functionalities, enhancements, and technical requirements that need to be developed in the GarageManagement System. It serves as the central reference point for the development team throughout the project lifecycle. Each item in the backlog represents a specific feature or task that contributes to the overall functionality of the system.

The product backlog helps in planning, managing, and tracking the progress of the project by breaking down the system into smaller, manageable components. It is continuously updated and refined based on feedback, changing requirements, and business priorities. The backlog ensures that the most valuable and high-priority features are developed first to deliver maximum benefit to users.

**Table 3.3:** Product Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **NAME** | **PRIORITY** | **ESTIMATE (Hours)** | **STATUS** |
| 1 | User Registration/Login | High | 5 | COMPLETED |
| 2 | Vehicle Entry and Exit Logging | High | 8 | COMPLETED |
| 3 | Spare Parts Inventory System | High | 10 | COMPLETED |
| 4 | Service Request Management | High | 12 | COMPLETED |
| 5 | Mechanic Job Assignment | Medium | 8 | COMPLETED |
| 6 | Customer Billing & Invoices | High | 9 | COMPLETED |
| 7 | Real-Time Service Tracking | Medium | 6 | COMPLETED |
| 8 | Payment Integration | High | 7 | COMPLETED |
| 9 | Admin Dashboard & Reports | High | 10 | COMPLETED |
| 10 | Notifications | Medium | 5 | COMPLETED |

**3.6 Project Plan**

The project plan outlines the roadmap for developing the Garage Management System by breaking down the work into phases and sprints using the Agile methodology. It defines the timeline, tasks, and deliverables such as user registration and login, role-based access control, job assignment, spare parts management, vehicle service tracking, bill generation, and report management features. Each sprint focuses on completing a set of prioritized functionalities from the product backlog, ensuring incremental progress, continuous testing, and efficient development of the system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User**  **StoryID** | **Task Name** | **Start Date** | **End Date** | **Hours** | **Status** |
| 1 | Sprint 1 | 06/08/2025 | 11/08/2025 | 22 | Completed |
| 1 | 12/08/2025 | 18/08/2025 | Completed |
| 2 | 19/08/2025 | 22/08/2025 | Completed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | Sprint 2 | 28/09/2025 | 05/09/2025 | 18 | Completed |
| 3 | 06/09/2025 | 15/09/2025 | Completed |
| 3 | Sprint 3 | 16/09/2025 | 21/09/2025 | 14 | Completed |
| 4 | 22/09/2025 | 28/09/2025 | Completed |
| 4 | Sprint 4 | 29/09/2025 | 05/10/2025 | 14 | Completed |
| 4 | 06/10/2025 | 12/10/2025 | Completed |

**3.7 Sprint Backlog**

The sprint backlog is a subset of the product backlog that lists the specific tasks and features planned to be completed within each sprint cycle. For the Garage Management System, the sprint backlog includes deliverables such as user registration and login, role-based access control, job assignment to mechanics, spare parts management, vehicle service tracking, bill generation, report viewing, and customer service monitoring. It provides a clear, short-term development plan, ensuring that each sprint produces a working increment of the system while keeping progress measurable and focused

**Table 3.5:** Sprint Backlog

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Backlog tem** | **Completion Date** | | **Original Estimation in Hours** | | **Day 1**  **hrs** | | **Day 2**  **hrs** | | **Day**  **3**  **hrs** | **Day**  **4**  **hrs** | **Day**  **5**  **hrs** | | **Day 6**  **hrs** | | **Day 7**  **hrs** | | **Day 8**  **hrs** | | **Day 9**  **hrs** | | **Day 10**  **hrs** |
| SPRINT 1 | | | | | | | | | | | | | | | | | | | | | | |
| Database Design | 06/08/2025 | | 3 | | 1 | | 1 | | 1 | 0 | 0 | | 0 | | 0 | | 0 | | 0 | | 0 |
| UI/UX Wireframing | 12/08/2025 | | 4 | | 1 | | 1 | | 1 | 1 | 0 | | 0 | | 0 | | 0 | | 0 | | 0 |
| User Authentication Setup | 18/08/2025 | | 5 | | 1 | | 1 | | 1 | 1 | 1 | | 0 | | 0 | | 0 | | 0 | | 0 |
| SPRINT 2 | | | | | | | | | | | | | | | | | | | | | | |
| Role Management | 24/08/2025 | | 6 | | 1 | | 1 | | 1 | 1 | 1 | | 1 | | 0 | | 0 | | 0 | | 0 |
| Supervisor Module | 30/08/2025 | | 7 | | 1 | | 1 | | 1 | 1 | 1 | | 1 | | 1 | | 0 | | 0 | | 0 |
| SPRINT 3 | | | | | | | | | | | | | | | | | | | | | | |
| Mechanic work hours tracking | | 05/09/2025 | 6 | 1 | 1 | 1 | | 1 | | 1 | | 1 | | 0 | | 0 | | 0 | | 0 | |
| Payment & Billing integration | | 10/09/2025 | 7 | 1 | 1 | 1 | | 1 | | 1 | | 1 | | 1 | | 0 | | 0 | | 0 | |
| SPRINT 4 | | | | | | | | | | | | | | | | | | | | | | |
| Reports & Analytics | | 15/09/2025 | 1 | 1 | 1 | 1 | | 1 | | 1 | | 1 | | 1 | | 1 | | 0 | | 0 | |
| Final Testing & Deployment | | 20/09/2025 | 8 | 1 | 1 | 1 | | 1 | | 1 | | 1 | | 1 | | 1 | | 0 | | 0 | |
| Total | |  | 50 | 9 | 9 | 8 | | 6 | | 5 | | 4 | | 2 | | 1 | |  | |  | |

**3.8 Database Design**

The database design defines how data is structured, stored, and related within the Garage Management System. Using MySQL, the system is organized into tables such as users, vehicles, appointments, spare parts, payments, and work hours, each designed to manage specific operations. For example, the users table stores details of all user roles, appointments manage service records and job assignments, while spare parts track inventory and sales. The design ensures data consistency, security, and efficient retrieval, supporting core features like service scheduling, inventory management, payment processing, and performance tracking within the garage.

**Table 3.6:** Database Design

| **Collection** | **Attributes** | **Purpose** |
| --- | --- | --- |
| **Roles** | * id (PK) * name | Stores user roles like customer, supervisor, mechanic, manager. |
| **Users** | * id (PK) * name * email * password * role\_id(FK) * is\_active * created\_at * updated\_at | Stores login and user details. |
| **Customers** | * id(PK) * user\_id(FK) * phone * address | Stores customer profile details |

|  |  |  |
| --- | --- | --- |
| **Vehicles** | * id (PK) * customer\_id(FK) * vin * licensePlate * make * model * year * created\_at | Stores vehicle information of customers. |
| **Service\_type** | * id(PK) * name * default\_rate * est\_duration | Stores available garage services. |
| **Appoinments** | * id(PK) * customer\_id(FK) * vehicle\_id(FK) * service\_type\_id(FK) * preferred\_date * status * created\_at | Stores service booking appoinments |
| **Work\_orders** | * id(PK) * appointment\_id(FK) * supervisor\_id(FK) * mechanic\_id(FK) * status * started\_at * completed\_at * notes | Track job assigned to mechanics |
| **Work\_orders\_status\_history** | * id(PK) * work\_order\_id(FK) * status * changed\_by(FK) * changed\_at | Maintain history of status changes for work orders. |
| **Parts** | * id(PK) * sku * name * stock\_qty * reorder\_level * cost\_price * sale\_price | Stores spare parts inventory |
| **Spare\_part\_requests** | * id(PK) * work\_order\_id(FK) * mechanic\_id(FK) * part\_id(FK) * qty * status * manager\_id(FK) * approved\_id * created\_at | Request for spare parts by mechanics. |
| **Stock\_movements** | * id(PK) * part\_id(FK) * delta\_qty * reason * reference\_id * created\_by * created\_at | Track movements for parts inventory. |
| **invoices** | * id(PK) * work\_order\_id(FK) * number * subtotal * tax * discount * total * status * issued\_at | Stores invoice records for billing |
| **Invoice\_items** | * id(PK) * invoice\_id(FK) * type * reference\_id * description * qty * unit\_price | Sores line items in invoices |
| **payments** | * id(PK) * invoice\_id(FK) * method * amount * paid\_at * txn\_ref | Stores customer payments for invoices. |
| **Notifications** | * id(PK) * user\_id(FK) * type * payload * read\_at * created\_at | Stores system notifications for users |
| **Audit\_logs** | * id(PK) * entity\_type * entity\_id * action * actor\_id * before\_json * after\_json * created\_at | Track system actions for auditing |

# Result and Discussions

This chapter presents the outcomes of implementing the Garage Management System (GMS) and discusses how the developed modules fulfill the objectives defined in the earlier chapters. The results include the successful implementation of all key modules—Admin, Manager, Supervisor, Mechanic, and Customer—covering functionalities such as user registration and login, service booking, job assignment, inventory management, mechanic work tracking, payment processing, and report generation. Screenshots of the working system are provided to illustrate the main operations, including service scheduling, spare parts management, and billing functionalities.

The discussion section highlights how these results address the limitations of traditional garage operations, which often suffer from poor data organization, inefficient communication, and lack of transparency. The developed system provides a centralized, digital platform that enables smooth coordination between staff, accurate record-keeping, and real-time updates on services. It also improves workflow efficiency, service quality, and customer satisfaction through automation and data-driven insights. Overall, this chapter demonstrates that the project objectives were successfully achieved and that the Garage Management System serves as a practical and efficient solution for modernizing garage operations.

## Results

This section presents the core working interfaces of the Garage Management System, demonstrating how each implemented module functions in real-world scenarios. The results are illustrated through screenshots of major forms and dashboards such as login and registration pages, service booking forms, job assignment dashboards, inventory management panels, payment pages, and report generation screens. Each figure is properly labeled and accompanied by a brief explanation describing its purpose and role within the system. These results confirm that the proposed system meets its intended goals by providing an organized, user-friendly, and efficient platform for managing all garage activities and improving overall operational performance.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 4.1:** Index Page

displays the AutoCare Garage Management System’s homepage ,showcasing professionals auto repair and maintenance services. if features navigation ,user authentication options, and contact information for an intuitive user experience.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 4.2: Manager Dashboard**

**Figure 4.2** shows the Manager Dashboard featuring quick links for “Spare Parts Management ”,”Reports & Analytics ”, and “ Billing ”, along with sections to monitor “Assigned Jobs” and “Billing details.

A screenshot of a computer

AI-generated content may be incorrect. **Figure 4.3**: Mechanic Dashboard

**Figure 4.3** The mechanic Dashboard displays currently “No jobs assigned yet” and includes a “View All Appointments ” button for mechanics to review their schedule.it offers a focused interface for managing assigned tasks and appointments , alongside a “Logout” option.

# Conclusion

The Garage Management System (GMS) was developed to automate and simplify the daily operations of a garage by providing a centralized, digital platform for managing services, staff, inventory, and customer interactions. It replaces traditional manual methods with an organized, user-friendly, and efficient system that enhances coordination and productivity.

The results demonstrate that the Garage Management System meets its intended objectives by improving operational efficiency, data organization, and communication across all roles. It ensures faster service delivery, accurate record-keeping, and enhanced customer satisfaction. By centralizing data and automating workflows, the system helps garage owners make informed decisions and maintain a transparent service environment.

However, the system also has certain limitations. Currently, it functions primarily as a web-based platform and lacks a mobile application for on-the-go access. Integrating SMS or push notifications for service updates and expanding payment options could further enhance usability. Future enhancements may also include real-time GPS-based vehicle tracking, automated service reminders, and analytics dashboards to monitor performance trends.

Overall, the Garage Management System serves as a practical, scalable, and reliable solution for modernizing garage operations. It bridges the gap between customers and service providers by ensuring organized management, real-time updates, and smooth workflow, ultimately contributing to a more efficient and customer-focused automotive service environment.

# References

[1] S. Sharma and R. Gupta, "Garage Management System - Software Engineering Project Report," Lovely Professional University, [Online]. Available: https://www.lpu.in/

. [Accessed 20 October 2024].

[2] M. Qassem, Thesis, Dissertation and Article Writing, 2014.

[3] S. Vinz., "Figure and Table Lists | Word Instructions, Template & Examples," Scribbr, 13 October 2015. [Online]. Available: https://www.scribbr.com/dissertation/figure-and-table-lists-in-your-dissertation/

. [Accessed 19 October 2024].

[4] TechVidvan, "Garage Management System Project using PHP and MySQL," [Online]. Available: https://techvidvan.com/tutorials/garage-management-system-project/

. [Accessed 21 October 2024].

Appendix

Appendix A Data Flow Diagram

**DFD level 0**

A diagram of a garage management system

AI-generated content may be incorrect.

**DFD level 1**

A diagram of a system

AI-generated content may be incorrect.

**DFD level 2**

A diagram of a form

AI-generated content may be incorrect.

**DFD level 3**

A diagram of a diagram

AI-generated content may be incorrect.

**DFD level 4**

A diagram of a login and manage job

AI-generated content may be incorrect.

**Appendix B ER Diagram**

**A diagram of a software company

AI-generated content may be incorrect.**

**Appendix C Source Code**

**Manager\_dashboard.php**

<?php

session\_start();

include 'db\_connect.php';

// Ensure manager is logged in

if(!isset($\_SESSION['user\_id']) || $\_SESSION['role'] != 'manager'){

    header("Location: ../login.php");

    exit();

}

// Handle Mark as Paid functionality

if(isset($\_GET['mark\_paid']) && is\_numeric($\_GET['mark\_paid'])) {

    $bill\_id = $\_GET['mark\_paid'];

    $update\_stmt = $conn->prepare("UPDATE Bills SET payment\_status = 'paid' WHERE id = ?");

    $update\_stmt->bind\_param("i", $bill\_id);

    if($update\_stmt->execute()) {

        // Set session variable for success message

        $\_SESSION['success\_msg'] = "Bill marked as paid successfully!";

        // Redirect to remove the parameter from URL

        header("Location: manager\_dashboard.php");

        exit();

    } else {

        $error\_msg = "Error updating bill: " . $conn->error;

    }

}

// Fetch assigned jobs

$jobQuery = "

SELECT Jobs.id as job\_id, Users.name as mechanic\_name, Appointments.vehicle\_model, Appointments.vehicle\_number, Jobs.service\_details, Jobs.job\_status

FROM Jobs

JOIN Users ON Jobs.mechanic\_id = Users.id

JOIN Appointments ON Jobs.appointment\_id = Appointments.id

ORDER BY Jobs.id DESC

";

$jobResult = $conn->query($jobQuery);

// Fetch bills

$billQuery = "

SELECT Bills.id as bill\_id, Users.name as customer\_name, Appointments.vehicle\_model, Bills.total\_amount, Bills.payment\_status, Bills.bill\_date

FROM Bills

JOIN Users ON Bills.customer\_id = Users.id

JOIN Appointments ON Bills.appointment\_id = Appointments.id

ORDER BY Bills.bill\_date DESC

";

$billResult = $conn->query($billQuery);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Manager Dashboard</title>

<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

<style>

/\* General \*/

body { font-family: 'Roboto', sans-serif; background: #f0f2f5; margin:0; padding:0; color:#333; }

h1,h2{color:#1a1a1a;}

.container{width:95%; max-width:1200px; margin:20px auto;}

/\* Header \*/

header{

    background:#4da6ff; /\* Light blue \*/

    color:white; padding:20px 25px; display:flex; justify-content:space-between; align-items:center; box-shadow:0 4px 6px rgba(0,0,0,0.1);

}

header h2{margin:0; font-size:26px;}

header nav a{color:white; text-decoration:none; margin-left:15px; font-weight:500; transition:0.2s;}

header nav a:hover{color:#ffe066;}

/\* Module Cards \*/

.module-links{display:flex; gap:15px; flex-wrap:wrap; margin-bottom:25px;}

.module-card{

    background:#80bfff; /\* Light blue card \*/

    color:white; padding:20px 25px; border-radius:10px; flex:1 1 250px; text-align:center; text-decoration:none; font-weight:500; transition:all 0.3s ease; box-shadow:0 4px 8px rgba(0,0,0,0.15);

}

.module-card:hover{background:#3399ff; transform:translateY(-5px);}

.module-card i{display:block; font-size:26px; margin-bottom:10px;}

/\* Table Wrapper \*/

.table-wrapper{overflow-x:auto; margin-top:15px; border-radius:10px; box-shadow:0 4px 6px rgba(0,0,0,0.1);}

table{width:100%; border-collapse:collapse; min-width:600px; background:white;}

table th, table td{padding:12px 15px; text-align:left;}

table th{background:#4da6ff; color:white; font-weight:500;}

table tr:nth-child(even){background:#f8f9fa;}

table tr:hover{background:#d9ebff; transition:0.2s;}

/\* Collapsible Details \*/

.details-row {display: none; background-color: #e6f2ff; transition: all 0.3s ease;}

.clickable-row:hover {cursor:pointer; background-color: #cce6ff;}

/\* Status Badges \*/

.status-pending{color:#856404; background:#fff3cd; padding:4px 8px; border-radius:5px; font-weight:500; display:inline-block;}

.status-in-progress{color:#0c5460; background:#d1ecf1; padding:4px 8px; border-radius:5px; font-weight:500; display:inline-block;}

.status-completed, .status-paid{color:#155724; background:#d4edda; padding:4px 8px; border-radius:5px; font-weight:500; display:inline-block;}

/\* Action Buttons \*/

.action-btn{padding:6px 12px; background:#28a745; color:white; border-radius:5px; text-decoration:none; font-weight:500; display:inline-block; transition:all 0.2s;}

.action-btn:hover{background:#218838; transform:translateY(-2px);}

/\* Responsive headings \*/

h1,h2{font-size:1.8rem;}

@media(max-width:768px){

    h1{font-size:1.5rem;}

    h2{font-size:1.3rem;}

}

/\* Footer \*/

footer{text-align:center; padding:15px; background:#f0f2f5; margin-top:25px; color:#555; font-size:14px;}

/\* Success/Error Messages \*/

.alert {

    padding: 12px 15px;

    border-radius: 5px;

    margin-bottom: 15px;

    font-weight: 500;

    animation: fadeOut 3s ease-in-out forwards;

    animation-delay: 2s;

}

.alert-success {

    background: #d4edda;

    color: #155724;

    border-left: 4px solid #28a745;

}

.alert-error {

    background: #f8d7da;

    color: #721c24;

    border-left: 4px solid #dc3545;

}

@keyframes fadeOut {

    from { opacity: 1; }

    to { opacity: 0; display: none; }

}

</style>

</head>

<body>

<header>

<h2>AutoCare Garage Management System</h2>

<nav>

    <a href="../dashboard.php"><i class="fas fa-tachometer-alt"></i> Dashboard</a>

    <a href="../logout.php"><i class="fas fa-sign-out-alt"></i> Logout</a>

</nav>

</header>

<div class="container">

<h1>Manager Dashboard</h1>

<!-- Module Cards -->

<div class="module-links">

    <a href="view\_inventory.php" class="module-card"><i class="fas fa-cogs"></i> Spare Parts Management</a>

    <a href="reports.php" class="module-card"><i class="fas fa-chart-line"></i> Reports & Analytics</a>

    <a href="create\_bill.php" class="module-card"><i class="fas fa-file-invoice-dollar"></i> Billing</a>

</div>

<!-- Assigned Jobs -->

<h2>Assigned Jobs</h2>

<div class="table-wrapper">

<table>

<tr>

    <th>Job ID</th>

    <th>Mechanic</th>

    <th>Vehicle Model</th>

    <th>Vehicle Number</th>

    <th>Status</th>

</tr>

<?php if($jobResult->num\_rows > 0): ?>

    <?php while($job = $jobResult->fetch\_assoc()): ?>

    <tr class="clickable-row" data-target="details-<?= $job['job\_id']; ?>">

        <td><?= $job['job\_id']; ?></td>

        <td><?= $job['mechanic\_name']; ?></td>

        <td><?= $job['vehicle\_model']; ?></td>

        <td><?= $job['vehicle\_number']; ?></td>

        <td><span class="status-<?= strtolower(str\_replace(' ','-',$job['job\_status'])); ?>"><?= ucfirst($job['job\_status']); ?></span></td>

    </tr>

    <tr id="details-<?= $job['job\_id']; ?>" class="details-row">

        <td colspan="5"><strong>Service Details:</strong> <?= $job['service\_details']; ?></td>

    </tr>

    <?php endwhile; ?>

<?php else: ?>

<tr><td colspan="5" style="text-align:center;">No jobs assigned yet.</td></tr>

<?php endif; ?>

</table>

</div>

<!-- Billing -->

<h2>Billing</h2>

<!-- Success Message (only shown for billing actions) -->

<?php if(isset($\_SESSION['success\_msg'])): ?>

    <div class="alert alert-success" id="successMessage">

        <i class="fas fa-check-circle"></i> <?= $\_SESSION['success\_msg'] ?>

    </div>

    <?php unset($\_SESSION['success\_msg']); // Clear the message after displaying ?>

<?php endif; ?>

<?php if(isset($error\_msg)): ?>

    <div class="alert alert-error">

        <i class="fas fa-exclamation-circle"></i> <?= $error\_msg ?>

    </div>

<?php endif; ?>

<div class="table-wrapper">

<table>

<tr>

    <th>Bill ID</th>

    <th>Customer Name</th>

    <th>Vehicle Model</th>

    <th>Total Amount</th>

    <th>Payment Status</th>

    <th>Bill Date</th>

    <th>Action</th>

</tr>

<?php if($billResult->num\_rows > 0): ?>

    <?php while($bill = $billResult->fetch\_assoc()): ?>

    <tr>

        <td><?= $bill['bill\_id']; ?></td>

        <td><?= $bill['customer\_name']; ?></td>

        <td><?= $bill['vehicle\_model']; ?></td>

        <td>₹<?= number\_format($bill['total\_amount'],2); ?></td>

        <td><span class="status-<?= strtolower($bill['payment\_status']); ?>"><?= ucfirst($bill['payment\_status']); ?></span></td>

        <td><?= $bill['bill\_date']; ?></td>

        <td>

            <?php if($bill['payment\_status']=='pending'): ?>

                <a class="action-btn" href="manager\_dashboard.php?mark\_paid=<?= $bill['bill\_id']; ?>">

                    <i class="fas fa-check"></i> Mark as Paid

                </a>

            <?php else: ?>

                <span class="status-paid"><i class="fas fa-check-circle"></i> Paid</span>

            <?php endif; ?>

        </td>

    </tr>

    <?php endwhile; ?>

<?php else: ?>

<tr><td colspan="7" style="text-align:center;">No bills generated yet.</td></tr>

<?php endif; ?>

</table>

</div>

</div>

<footer>

&copy; <?= date("Y"); ?> AutoCare Garage. All rights reserved.

</footer>

<!-- Collapsible Row Script -->

<script>

document.addEventListener('DOMContentLoaded', function() {

    const rows = document.querySelectorAll('.clickable-row');

    rows.forEach(row => {

        row.addEventListener('click', function() {

            const targetId = row.getAttribute('data-target');

            const detailsRow = document.getElementById(targetId);

            if(detailsRow.style.display === 'table-row'){

                detailsRow.style.display = 'none';

            } else {

                detailsRow.style.display = 'table-row';

            }

        });

    });

    // Auto-hide success message after 3 seconds

    const successMessage = document.getElementById('successMessage');

    if(successMessage) {

        setTimeout(() => {

            successMessage.style.display = 'none';

        }, 3000);

    }

});

</script>

</body>

</html>

**Customer/create\_bill.php**

<?php

session\_start();

include 'db\_connect.php';

if(!isset($\_SESSION['user\_id']) || $\_SESSION['role'] != 'manager'){

    header("Location: ../login.php");

    exit();

}

// Check if coming from "Mark as Paid" with appointment ID

$prefill\_paid = isset($\_GET['pay']) ? $\_GET['pay'] : 0;

$prefill\_appointment\_id = isset($\_GET['appointment\_id']) ? $\_GET['appointment\_id'] : 0;

$appQuery = "

    SELECT Appointments.id AS app\_id, Appointments.vehicle\_model, Appointments.vehicle\_number, Users.name AS customer\_name, Users.id AS customer\_id

    FROM Appointments

    JOIN Users ON Appointments.customer\_id = Users.id

    ORDER BY Appointments.id DESC

";

$appResult = $conn->query($appQuery);

if($\_SERVER['REQUEST\_METHOD'] == 'POST'){

    $appointment\_id = $\_POST['appointment\_id'];

    $customer\_id    = $\_POST['customer\_id'];

    $total\_amount   = $\_POST['total\_amount'];

    $payment\_status = $\_POST['payment\_status'];

    $stmt = $conn->prepare("INSERT INTO Bills (appointment\_id, customer\_id, total\_amount, payment\_status) VALUES (?, ?, ?, ?)");

    $stmt->bind\_param("iids", $appointment\_id, $customer\_id, $total\_amount, $payment\_status);

    if($stmt->execute()){

        header("Location: manager\_dashboard.php?msg=BillCreated");

        exit();

    } else {

        $error = "Error: " . $conn->error;

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Generate Bill</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

<style>

\* { margin: 0; padding: 0; box-sizing: border-box; }

body {

    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

    background: white;

    min-height: 100vh;

    display: flex;

    align-items: center;

    justify-content: center;

    padding: 20px;

}

.container {

    background: #f8fafc;

    border-radius: 15px;

    box-shadow: 0 10px 30px rgba(0,0,0,0.08);

    width: 100%;

    max-width: 500px;

    border: 1px solid #e2e8f0;

}

.header {

    background: #1e293b;

    color: white;

    padding: 25px;

    text-align: center;

    border-radius: 15px 15px 0 0;

}

.header h2 {

    font-size: 24px;

    font-weight: 600;

}

.header i {

    font-size: 32px;

    margin-bottom: 10px;

    color: #3b82f6;

}

.form-container {

    padding: 30px;

}

.form-group {

    margin-bottom: 20px;

}

label {

    display: block;

    margin-bottom: 8px;

    font-weight: 500;

    color: #374151;

    font-size: 14px;

}

input, select {

    width: 100%;

    padding: 12px 15px;

    border: 2px solid #e5e7eb;

    border-radius: 8px;

    font-size: 15px;

    transition: all 0.3s;

    background: white;

}

input:focus, select:focus {

    outline: none;

    border-color: #3b82f6;

    box-shadow: 0 0 0 3px rgba(59, 130, 246, 0.1);

}

input:read-only {

    background: #f9fafb;

    color: #6b7280;

}

.btn {

    width: 100%;

    padding: 14px;

    background: #1e293b;

    color: white;

    border: none;

    border-radius: 8px;

    font-size: 16px;

    font-weight: 600;

    cursor: pointer;

    transition: all 0.3s;

    margin-top: 10px;

}

.btn:hover {

    background: #374151;

    transform: translateY(-1px);

}

.back-btn {

    width: 100%;

    padding: 14px;

    background: #6b7280;

    color: white;

    border: none;

    border-radius: 8px;

    font-size: 16px;

    font-weight: 600;

    cursor: pointer;

    margin-top: 10px;

    transition: all 0.3s;

    text-align: center;

    text-decoration: none;

    display: inline-block;

}

.back-btn:hover {

    background: #4b5563;

}

.error {

    background: #fef2f2;

    color: #dc2626;

    padding: 12px;

    border-radius: 8px;

    margin-bottom: 20px;

    text-align: center;

    border-left: 4px solid #dc2626;

    font-size: 14px;

}

.paid-notice {

    background: #d1fae5;

    color: #065f46;

    padding: 12px;

    border-radius: 8px;

    margin-bottom: 20px;

    text-align: center;

    border-left: 4px solid #10b981;

    font-size: 14px;

}

</style>

</head>

<body>

<div class="container">

    <div class="header">

        <i class="fas fa-file-invoice-dollar"></i>

        <h2>Generate Bill</h2>

        <?php if($prefill\_paid): ?>

            <p style="margin-top:10px; opacity:0.8;">Marking appointment as paid</p>

        <?php endif; ?>

    </div>

    <div class="form-container">

        <?php if(isset($error)) echo "<div class='error'>$error</div>"; ?>

        <?php if($prefill\_paid): ?>

            <div class="paid-notice">

                <i class="fas fa-check-circle"></i> Pre-selecting "Paid" status for this bill

            </div>

        <?php endif; ?>

        <form method="POST">

            <div class="form-group">

                <label>Select Appointment</label>

                <select name="appointment\_id" required onchange="updateCustomer()" id="appSelect">

                    <option value="">-- Choose Appointment --</option>

                    <?php while($row = $appResult->fetch\_assoc()): ?>

                        <option value="<?= $row['app\_id']; ?>"

                                data-customer="<?= $row['customer\_id']; ?>"

                                <?= ($prefill\_appointment\_id == $row['app\_id']) ? 'selected' : '' ?>>

                            <?= $row['vehicle\_model']." (".$row['vehicle\_number'].") - ".$row['customer\_name']; ?>

                        </option>

                    <?php endwhile; ?>

                </select>

            </div>

            <div class="form-group">

                <label>Customer ID</label>

                <input type="text" name="customer\_id" id="customerId" readonly required>

            </div>

            <div class="form-group">

                <label>Total Amount (₹)</label>

                <input type="number" name="total\_amount" step="0.01" required>

            </div>

            <div class="form-group">

                <label>Payment Status</label>

                <select name="payment\_status" required>

                    <option value="pending">Pending</option>

                    <option value="paid" <?= $prefill\_paid ? 'selected' : '' ?>>Paid</option>

                </select>

            </div>

            <button type="submit" class="btn">

                <i class="fas fa-file-invoice"></i> Generate Bill

            </button>

        </form>

        <!-- Back to Dashboard Button -->

        <a href="manager\_dashboard.php" class="back-btn"><i class="fas fa-arrow-left"></i> Back to Dashboard</a>

    </div>

</div>

<script>

function updateCustomer() {

    const select = document.getElementById('appSelect');

    const customerField = document.getElementById('customerId');

    const selectedOption = select.options[select.selectedIndex];

    if(select.selectedIndex > 0) {

        customerField.value = selectedOption.getAttribute('data-customer');

    } else {

        customerField.value = '';

    }

}

document.addEventListener('DOMContentLoaded', function() {

    updateCustomer();

});

</script>

</body>

</html>

**Manager/view\_inventory.php**

<?php

session\_start();

include 'db\_connect.php';

// Ensure manager is logged in

if(!isset($\_SESSION['user\_id']) || $\_SESSION['role'] != 'manager'){

    header("Location: ../login.php");

    exit();

}

$result = $conn->query("SELECT \* FROM SpareParts");

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Inventory Management</title>

<link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

<style>

:root {

    --primary: #4361ee;

    --primary-light: #eef2ff;

    --secondary: #3f37c9;

    --success: #4cc9f0;

    --text: #2d3748;

    --text-light: #718096;

    --border: #e2e8f0;

    --bg: #f7fafc;

    --white: #ffffff;

    --shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1), 0 2px 4px -1px rgba(0, 0, 0, 0.06);

}

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

    font-family: 'Inter', sans-serif;

    background: var(--bg);

    color: var(--text);

    line-height: 1.6;

    padding: 20px;

    min-height: 100vh;

}

.container {

    max-width: 1100px;

    margin: 0 auto;

    background: var(--white);

    border-radius: 12px;

    box-shadow: var(--shadow);

    overflow: hidden;

}

.header {

    background: linear-gradient(135deg, var(--primary), var(--secondary));

    color: white;

    padding: 25px 30px;

    text-align: center;

}

.header h2 {

    font-size: 1.8rem;

    font-weight: 600;

    margin-bottom: 5px;

}

.header p {

    opacity: 0.9;

    font-weight: 300;

}

.table-container {

    padding: 25px;

    overflow-x: auto;

}

table {

    width: 100%;

    border-collapse: collapse;

    margin-bottom: 25px;

}

th {

    background: var(--primary-light);

    color: var(--primary);

    font-weight: 600;

    padding: 15px 12px;

    text-align: left;

    border-bottom: 2px solid var(--border);

}

td {

    padding: 14px 12px;

    border-bottom: 1px solid var(--border);

    color: var(--text);

}

tr:hover {

    background: #f8faff;

    transition: background 0.2s;

}

.actions {

    display: flex;

    gap: 15px;

    justify-content: center;

    margin-top: 20px;

}

.btn {

    display: inline-flex;

    align-items: center;

    gap: 8px;

    padding: 12px 20px;

    background: var(--primary);

    color: white;

    text-decoration: none;

    border-radius: 8px;

    font-weight: 500;

    transition: all 0.3s;

}

.btn:hover {

    background: var(--secondary);

    transform: translateY(-2px);

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.15);

}

.btn-outline {

    background: transparent;

    color: var(--primary);

    border: 1px solid var(--primary);

}

.btn-outline:hover {

    background: var(--primary-light);

}

.price {

    font-weight: 600;

    color: var(--primary);

}

.quantity {

    font-weight: 500;

    padding: 4px 10px;

    border-radius: 20px;

    background: #f0fff4;

    color: #2d7d32;

    display: inline-block;

}

.empty-state {

    text-align: center;

    padding: 40px 20px;

    color: var(--text-light);

}

.empty-state i {

    font-size: 3rem;

    margin-bottom: 15px;

    color: var(--border);

}

@media (max-width: 768px) {

    .container {

        border-radius: 8px;

    }

    .header {

        padding: 20px;

    }

    .table-container {

        padding: 15px;

    }

    table {

        font-size: 0.9rem;

    }

    th, td {

        padding: 10px 8px;

    }

    .actions {

        flex-direction: column;

        align-items: center;

    }

    .btn {

        width: 100%;

        justify-content: center;

    }

}

</style>

</head>

<body>

<div class="container">

    <div class="header">

        <h2>Spare Parts Inventory</h2>

        <p>Manage your inventory efficiently</p>

    </div>

    <div class="table-container">

        <?php if($result->num\_rows > 0): ?>

        <table>

            <thead>

                <tr>

                    <th>ID</th>

                    <th>Name</th>

                    <th>Model</th>

                    <th>Supplier</th>

                    <th>Quantity</th>

                    <th>Price</th>

                </tr>

            </thead>

            <tbody>

                <?php while($row = $result->fetch\_assoc()): ?>

                <tr>

                    <td><?= $row['id']; ?></td>

                    <td><?= $row['part\_name']; ?></td>

                    <td><?= $row['model'] ? $row['model'] : '-'; ?></td>

                    <td><?= $row['supplier'] ? $row['supplier'] : '-'; ?></td>

                    <td><span class="quantity"><?= $row['quantity']; ?></span></td>

                    <td class="price">₹<?= $row['price']; ?></td>

                </tr>

                <?php endwhile; ?>

            </tbody>

        </table>

        <?php else: ?>

        <div class="empty-state">

            <i class="fas fa-box-open"></i>

            <h3>No Spare Parts Found</h3>

            <p>Get started by adding your first spare part to inventory</p>

        </div>

        <?php endif; ?>

        <div class="actions">

            <a href="spare\_parts.php" class="btn">

                <i class="fas fa-plus-circle"></i> Add New Spare Part

            </a>

            <a href="manager\_dashboard.php" class="btn btn-outline">

                <i class="fas fa-arrow-left"></i> Back to Dashboard

            </a>

        </div>

    </div>

</div>

<script>

// Add subtle row animation

document.addEventListener('DOMContentLoaded', function() {

    const rows = document.querySelectorAll('tbody tr');

    rows.forEach((row, index) => {

        row.style.opacity = '0';

        row.style.transform = 'translateY(10px)';

        setTimeout(() => {

            row.style.transition = 'all 0.4s ease';

            row.style.opacity = '1';

            row.style.transform = 'translateY(0)';

        }, index \* 100);

    });

});

</script>

</body>

</html>

**Appendix D Screenshorts**

**Home page**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Manager Dashboard Page**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Spare Parts Management/Add Spare Part Pge**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Spare Parts Management/Spare Parts Inventory Page**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Customer Dashboard Page**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Mechanic Dashboard Page**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Mechanic Dashboard/View All Appointments Page**

**A screenshot of a computer

AI-generated content may be incorrect.**