INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 3, MARCH 2026 | ISSN: 2456-4184 | IJNRD.ORG

ONLINE GARAGE MANAGEMENT SYSTEM: A FULL-STACK WEB APPLICATION FOR AUTOMOTIVE SERVICE OPERATIONS

A Comprehensive Study of MERN Architecture, Security Implementation, and Cloud Deployment

Shivam Kumar, Arnav Roy, Karan Singh, Anish Chandra Jha, Himanshu Singh

Abstract:

This paper presents a detailed development and implementation of an Online Garage Management System using the MERN stack. The system addresses inefficiencies in traditional automotive service operations including manual record keeping, lack of integration, poor inventory control, and scheduling issues. The system integrates secure authentication, cloud deployment, and scalable architecture. Results show significant improvements in efficiency, data accuracy, and customer satisfaction.

Keywords: MERN Stack, Garage Management, Web Application, Cloud, Security

1. INTRODUCTION

The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability. The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.

2. PROBLEM STATEMENT

The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless

interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.

3. OBJECTIVES

The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.

4. METHODOLOGY

The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability. The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.

5. SYSTEM ARCHITECTURE

The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless

interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.

Figure 1: System Architecture Diagram

[Client (React UI)] ---> [Server (Node.js + Express API)] ---> [Database (MongoDB)] This diagram represents the three-tier architecture of the system.

6. IMPLEMENTATION

The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.

7. DATABASE DESIGN

The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and

component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.

Figure 2: Database Schema Overview

Users → Customers → Vehicles → Services → Appointments → Inventory
This shows relational flow within MongoDB collections.

8. DEPLOYMENT

The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.

9. PERFORMANCE AND TESTING

The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.

10. RESULTS AND DISCUSSION

The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability. The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.

11. CONCLUSION

The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.The system leverages modern web technologies to provide seamless interaction between users and backend services. It ensures modularity, scalability, and maintainability. The architecture is designed to handle real-time operations, large datasets, and concurrent users efficiently. Security measures include JWT authentication, password hashing, and role-based access control. The frontend is responsive and optimized for performance using modern techniques such as lazy loading and component reuse. Cloud deployment ensures availability and scalability.

REFERENCES

[1] MongoDB Documentation
[2] React Documentation
[3] Node.js Documentation
[4] Express.js Guide
[5] Cloud Deployment Guides