IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 5, November 2025
Impact Factor: 7.67

Design and Implementation of a Secure Web-Based Parking Management System

Prof. Snehal Rewatkar[1], Mr. Anurag Nand[2], Mr. Anup Tatte[3], Miss. Liza Nagarale[4]
Guide, Computer Science and Engineering Department[1]
Student, Computer Science and Engineering Department[2-4]
Tulsiramji Gaikwad-Patil College of Engineering and Technology, Nagpur, India

Abstract: *The transition from manual to digital administrative processes is essential for modern logistical infrastructure. This paper presents the design and implementation of a Web-Based Parking Management System (PMS) aimed at rectifying systemic inefficiencies, data redundancy, and lack of financial transparency in vehicle parking facilities. The system utilizes a Three-Tier Web Architecture leveraging the LAMP stack (Linux, Apache, MySQL, PHP) to establish a centralized solution for real-time occupancy tracking and financial transaction management. A significant focus of this work is the security hardening of the application, specifically the mitigation of SQL Injection vulnerabilities through Prepared Statements and the implementation of modern cryptographic password hashing. The resulting system automates vehicle identification, enforces data integrity, and provides dynamic decision support through a real-time administrative dashboard.*

Keywords: Parking Management System, Web Application, PHP, MySQL, SQL Injection, Three-Tier Architecture

I. INTRODUCTION

Manual administrative processes within parking facilities often suffer from inefficiencies, leading to traffic congestion and revenue leakage. The primary motivation for this development was to address these issues by transitioning from paper-based records to a robust digital platform.

The project focuses on developing an administrative backend that facilitates staff authentication, vehicle lifecycle management, and detailed operational reporting. The comprehensive objectives guiding this development were:

- Operational Optimization: To automate the assignment of unique verification numbers (Parking IDs) to reduce manual processing time.
- Data Centralization: To secure critical vehicle and transactional data within a relational database, preventing transcription errors.
- Decision Support: To provide a dashboard with up-to-the-minute Key Performance Indicators (KPIs) such as occupancy and daily income.
- Security Hardening: To identify and mitigate severe vulnerabilities, specifically SQL Injection and weak cryptography, which are common in legacy web applications.

II. SYSTEM ARCHITECTURE

The system employs a standard Three-Tier Web Architecture, which is the industry standard for scalable enterprise solutions.

A. Presentation Tier

The user interface is built using HTML, CSS, and the Bootstrap framework. This ensures a responsive design that adapts to various devices, allowing administrators to manage the system via desktop or mobile tablets.

B. Application Tier

The core business logic is engineered using PHP (Hypertext Preprocessor). This tier handles input validation, session management, and the execution of algorithms for fee calculation and ID generation.

DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 5, November 2025 Impact Factor: 7.67

C. Data Tier

MySQL is utilized as the backend relational data store. The database schema allows for the storage of complex relationships between administrators, vehicle categories, and transactional records. The system relies on the ACID properties (Atomicity, Consistency, Isolation, Durability) of MySQL to ensure that financial data, such as parking charges, are recorded reliably without corruption. .

III. METHODOLOGY AND DESIGN

The project followed a structured System Development Life Cycle (SDLC). The core functional modules are designed as follows:

A. Vehicle Check-In Module

• Upon vehicle arrival, the administrator submits a vehicle entry form. The system utilizes the PHP mt_rand() function to programmatically generate a unique, non-sequential Parking ID (e.g., CA-96069). This ID serves as the primary key for retrieval. The system captures the "In-Time" using a server-side timestamp to prevent manipulation.

B. Vehicle Check-Out and Billing

• The Check-Out module is responsible for revenue generation. The administrator retrieves the vehicle record using the Parking ID. The system then requires the input of a final "Parking Charge" and a "Remark". The core logic executes a single, atomic database operation to update the status to 'Out', stamp the definitive "Out-Time", and save the final charge.

C. Reporting and Analytics

• To support management decisions, the system includes a generate-reports.php module. This allows administrators to run dynamic queries filtered by specific "From" and "To" dates, providing a clear audit trail of all transactions within a given period.

IV. IMPLEMENTATION AND SECURITY

The system is implemented using the secure mysqli extension for database connectivity. A critical aspect of this implementation was the security assessment and subsequent hardening of the code.

A. Mitigation of SQL Injection

Initial analysis revealed that legacy coding practices, such as directly concatenating user input into SQL strings, posed a severe SQL Injection (SQLi) risk. To mitigate this, the entire application was refactored to use Prepared Statements. This technique ensures that the SQL command structure is immutable and user data is bound as non-executable parameters.

B. Cryptographic Enhancements

The system abandoned the use of MD5 for password storage, which is no longer considered secure. Instead, the implementation utilizes PHP's password_hash() and password_verify() functions to implement Bcrypt, a slow, salted hashing algorithm that protects administrative credentials against brute-force attacks. whereas the Microsoft Word templates are self-contained.

V. RESULTS

The resulting application provides a user-friendly dashboard that displays real-time analytics. As shown in Fig. 1, the dashboard visualizes total vehicle counters and highlights active transactions.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 5, November 2025
Impact Factor: 7.67

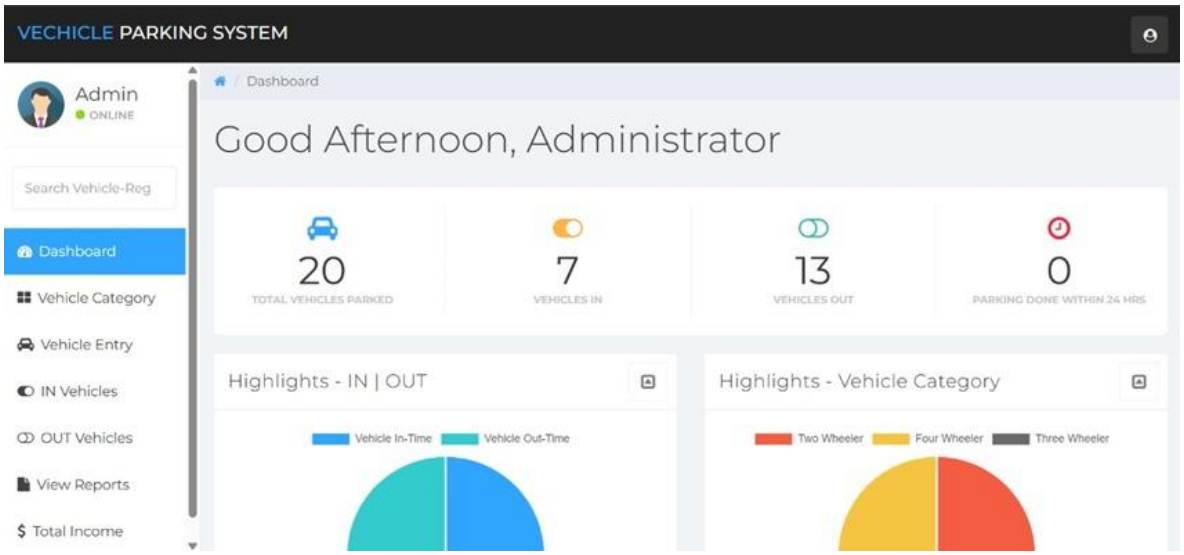

Fig. 1. Administrative Dashboard displaying real-time vehicle counters.

Figure 2 illustrates the "Vehicle In" management interface, where administrators can view current inventory and initiate the exit process.

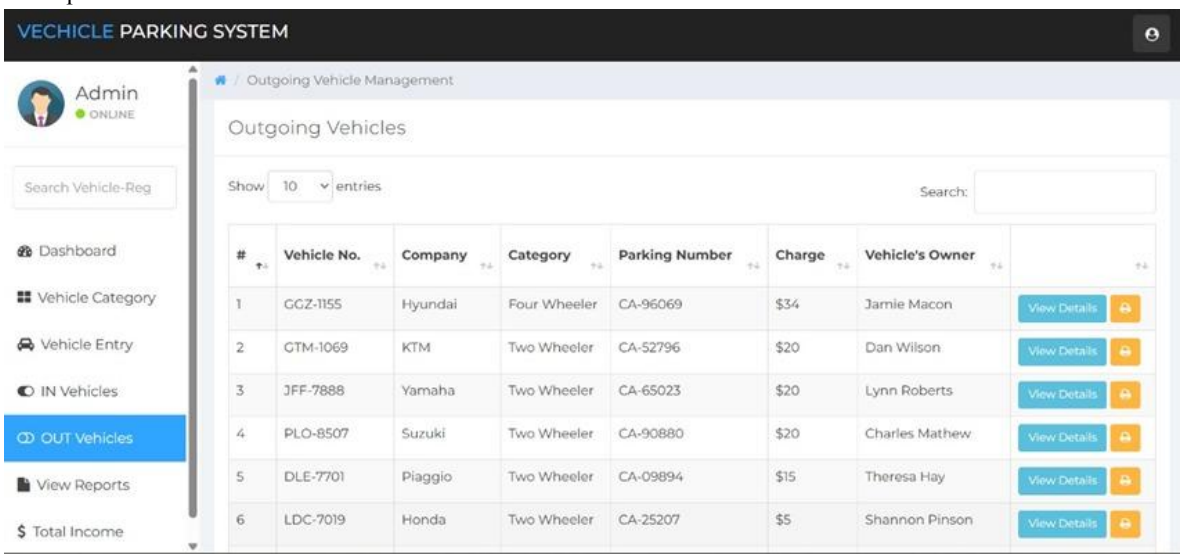

Fig. 2. Incoming Vehicle Management Interface.

VI. CONCLUSION

The developed Parking Management System successfully modernizes logistical infrastructure management. By centralizing data and automating the check-in/check-out process, the system reduces manual errors and enhances operational speed. Furthermore, the rigorous application of security standards—specifically the elimination of SQL Injection and the adoption of robust password hashing—validates the system as a secure, professional-grade solution suitable for deployment.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 5, November 2025
Impact Factor: 7.67

VII. FUTURE SCOPE

Future enhancements for the system include the integration of third-party payment gateways (e.g., Razorpay) for cashless transactions and the incorporation of IoT hardware, such as License Plate Recognition (LPR) cameras, to fully automate the entry and exit validation process.

ACKNOWLEDGMENT

We express our deep sense of gratitude to our project guide, Prof. Snehal Rewatkar, Department of Computer Science and Engineering, for his valuable guidance, continuous encouragement, and constructive suggestions throughout the development of this system.

We also wish to extend our sincere thanks to our Industry Mentor, Miss. Ashwini Digrase from Thunder Cube Pvt. Ltd., for providing technical insights and support during the internship period.

We utilize this opportunity to thank Dr. Swapnili Karemore, Head of the Computer Science and Engineering Department, and Dr. Snehal Abhyankar (SDC) for providing the necessary facilities and infrastructure to complete this work. Finally, we express our gratitude to Dr. P. L. Naktode (Principal) and Dr. Pragati Patil (Vice-Principal) of Tulsiramji Gaikwad-Patil College of Engineering & Technology, Nagpur, for their constant motivation..

REFERENCES

[1] J. Duckett, HTML and CSS: Design and Build Websites, 1st ed. Wiley, 2011.
[2] D. Stuttard and M. Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, 2nd ed. Wiley, 2011.
[3] S. Choudhari, A. Gupta, and N. Kamble, "Security and Privacy of AWS S3," International Journal of Innovative Research in Science, Engineering and Technology, vol. 10, no. 12, 2021.
[4] J. Zhang, et al., "InfiniStore: Elastic Serverless Cloud Storage," arXiv preprint, 2022.
[5] E. Erturk and S. Obrutsky, "Multimedia Storage in the Cloud using Amazon Web Services," arXiv, 2016.
[6] B. Piper and D. Clinton, AWS Certified Solutions Architect Study Guide. Wiley, 2020.
[7] J. Zhang, et al., "InfiniStore: Elastic Serverless Cloud Storage," arXiv preprint arXiv:2209.01496, 2022.
[8] L. Welling and L. Thomson, PHP and MySQL Web Development, 5th ed. Addison-Wesley Professional, 2016.
[9] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7th ed. Pearson, 2015.
[10] I. Sommerville, Software Engineering, 10th ed. Pearson, 2015.