

Nextgen Device Rental System

Sumedha Madangopal Thakare¹ Prof. Ajay Shiketod²

¹Student ²Assistant Professor

^{1,2}Master of Computer Applications

^{1,2}Anantrao Pawar College of Engineering & Research, Pune Affiliated to Savitribai Phule Pune University, India

Abstract — In the modern technological world, availability of computers with high performance has transformed from being a professional need to becoming a must-have requirement; however, the issue of constantly increasing costs of quality laptops/desktops is still a major expense for most students and corporations. The current research aims at introducing NextGen Device Rental, which is a central web-based management solution, replacing traditional paper-based systems with automated computerized ones. By using the LAMP framework that includes Linux OS, Apache Web Server, MySQL database, and PHP scripting language, NextGen Device Rental solves the problem of historical data sharing challenges and bottlenecks in the rental business. The proposed system utilizes the concept of Role-Based Access Control (RBAC) to ensure extremely high levels of data security and privacy along with easy-to-use functionality for efficient rental processes. Technical feasibility of automatic device rental lifecycle, including instant inventory check and automatic invoice issuance has been proven.

Keywords: NextGen Device Rental, LAMP Stack, Hardware Asset Management, Centralized Management System, Role-Based Access Control (RBAC), Automation in Rental Services, Data Integrity, Digital Transformation

I. INTRODUCTION

In the contemporary era of rapid digital transformation, high-performance computing hardware has shifted from a professional luxury to a fundamental necessity for students, researchers, and corporate organizations; however, the "constantly rising costs" of premium laptops and desktops remain a significant financial barrier to entry, often widening the digital divide. This research paper introduces NextGen Device Rental, a centralized web-based management platform specifically engineered to bridge this gap by replacing obsolete, paper-based rental procedures with an automated and highly efficient digital ecosystem. By leveraging the robust LAMP stack (Linux, Apache, MySQL, and PHP), the study addresses critical "data sharing problems" and operational bottlenecks inherent in "pre-existing management systems" that traditionally rely on decentralized, manual processes prone to human error and data redundancy. Furthermore, the research emphasizes the importance of infrastructure security through the implementation of Role-Based Access Control (RBAC), which ensures the highest standards of "data security and privacy" for both administrative logs and user transactions. Beyond mere automation, the platform focuses on real-time inventory visibility and "uninterrupted performance," allowing organizations to monitor asset health and maintenance cycles without the "tedious manual job" of physical ledger tracking. Ultimately, this research demonstrates the "technical feasibility" of an integrated rental lifecycle—transitioning from manual entry to automated

digital invoice generation—providing a scalable, "Value for Money" infrastructure that maximizes asset utilization and fosters technological accessibility in the modern rental industry.

II. LITERATURE SURVEY

A review will provide knowledge of the movement away from manual assets to automated web-based systems as an alternative. There has been a review of the technologies used in past studies and the reasons that traditional asset management techniques cannot meet the demand for hardware rentals today.

A. Existing Management Systems and Manual Procedures

The traditional model has used decentralized and manual processes to manage hardware assets within the organization. Some problems associated with manual processes include the following:

- Manual Documentation: All documentation related to hardware asset management has been kept using physical documents. This process is labour-intensive and can be prone to errors made by humans.
- Lack of Communication: Since it is a manual system, transferring information from one department to another is cumbersome and can be costly in terms of time.
- Maintenance Issues: Maintaining the condition of the devices, including their repair, is called a "tedious process," since there is no automated monitoring process.
- Security Issues: The manual process is prone to various risks because it relies on physical documents that can easily be damaged or lost.

B. Core Technologies in Web-Based Management

The solution to the constraints associated with manually driven systems entails the utilization of the web integration techniques as discussed in detail below. Transition from the traditional manual approach to an automatic system is made possible through the LAMP Stack, providing an optimal environment for web application:

- PHP (Hypertext Preprocessor): This language forms the basic building block for the logic within the entire platform. PHP plays a significant role in processing user demands and booking facilities automatically.
- MySQL Database Management: Being the most reliable RDBMS, MySQL helps in the centralization of all data, such as users' profile and inventories.
- Apache HTTP Server: The role of this server is to provide an essential platform that facilitates synchronization of web services and guarantees the system "uninterrupted performance" and security.

C. Research Gap and Systematic Needs

Despite the availability of basic inventory tools, several critical gaps remain in the current rental landscape. Most existing platforms do not offer a truly centralized repository, leading to "data sharing problems" where inventory status is not updated in real-time across all departments. Furthermore, there is a notable lack of integrated Role-Based Access Control (RBAC) in smaller systems, leaving them vulnerable to unauthorized data modifications and compromising user privacy. Additionally, the transition from "tedious manual jobs" to automated workflows is often incomplete, especially in handling the entire lifecycle—from device booking to digital invoice generation. NextGen Device Rental aims to bridge these identified gaps by providing a "Value for Money" solution that ensures both operational efficiency and high data security through a unified digital ecosystem.

III. SYSTEM DESIGN AND ARCHITECTURE

A. System Architecture

The architecture of NextGen Device Rental is based on a centralized three-tier model. This structure ensures that the presentation, logic, and data layers are separated, which provides "uninterrupted performance" and easy maintenance.

- User Interface (Frontend): This is the top layer where users browse devices and make bookings.
- Application Logic (Middleware): Powered by PHP, this layer processes all requests and ensures that the "booking lifecycle" is followed correctly.
- Database Layer (Backend): A centralized MySQL repository that stores all device info, user records, and transaction logs securely.

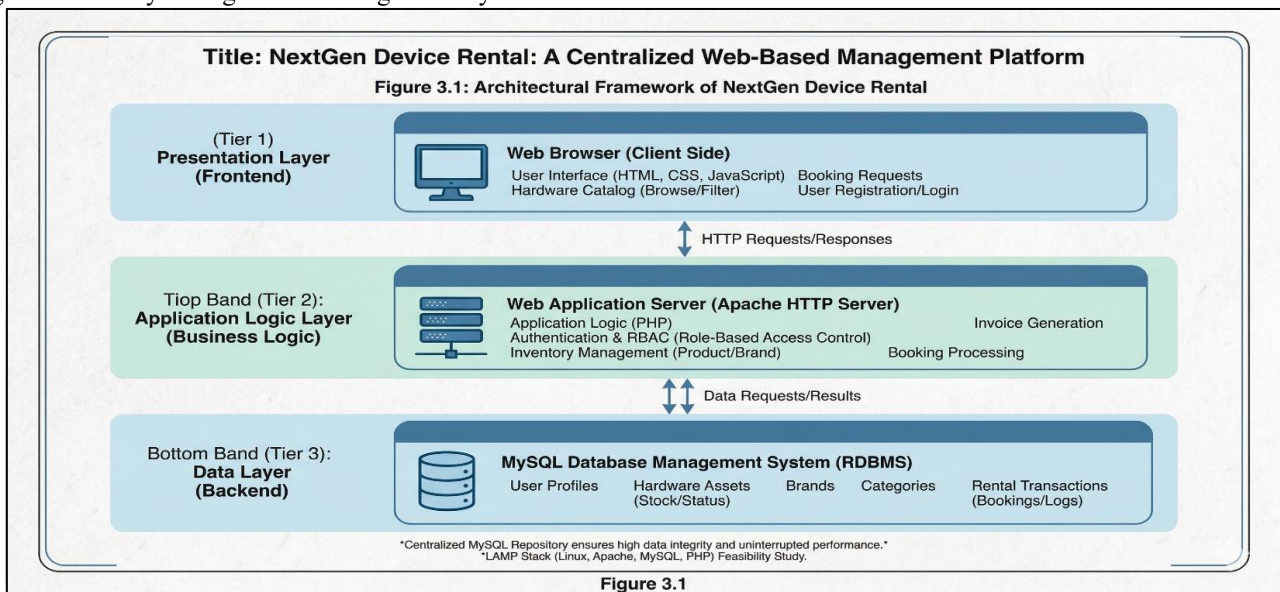


Fig. 3.1: Three-Tier Architectural Framework of NextGen Device Rental

B. Module Description and Workflow

To maintain Role-Based Access Control (RBAC), the system logic is divided into two distinct operational workflows. This ensures that administrative tasks and user activities are handled separately, maintaining "data security and privacy."

1) Administrative Workflow

The administrator manages the backend of the platform. This includes inventory control, viewing registered users, and generating financial reports to eliminate "tedious manual work."

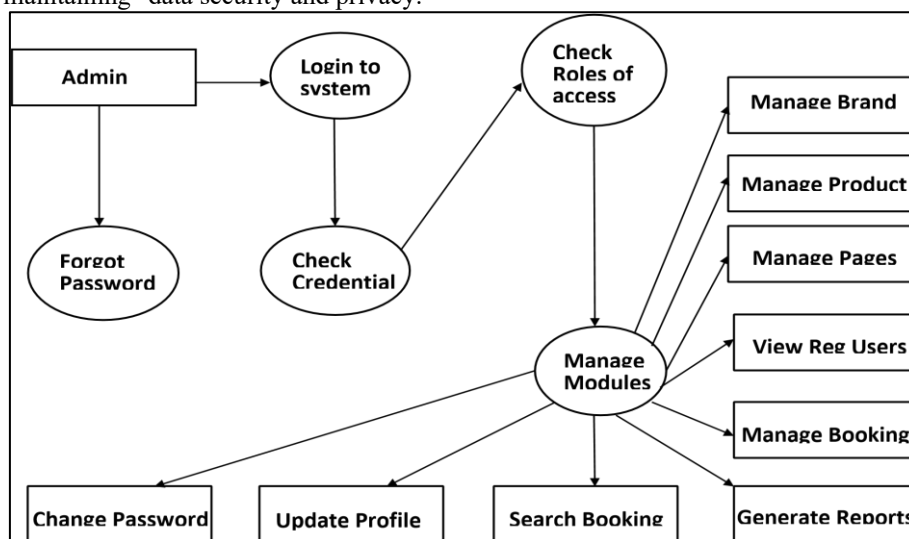


Fig. 3.2: Administrative Operational Flow

2) User Workflow

Registered users interact with the frontend to search for hardware, book devices, and track their rental history. This transparency ensures a "Value for Money" experience.

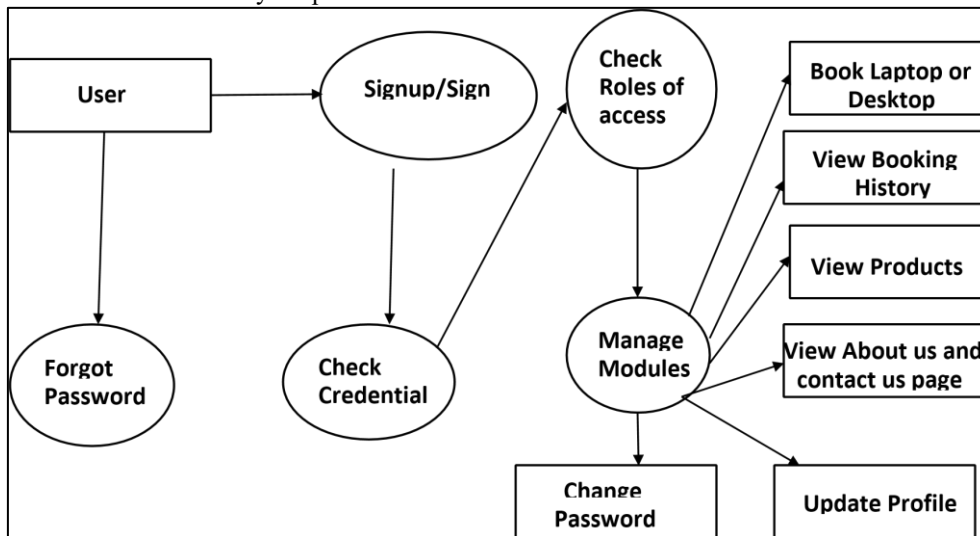


Fig. 3.3: User Interaction and Booking Workflow

C. Module Decomposition

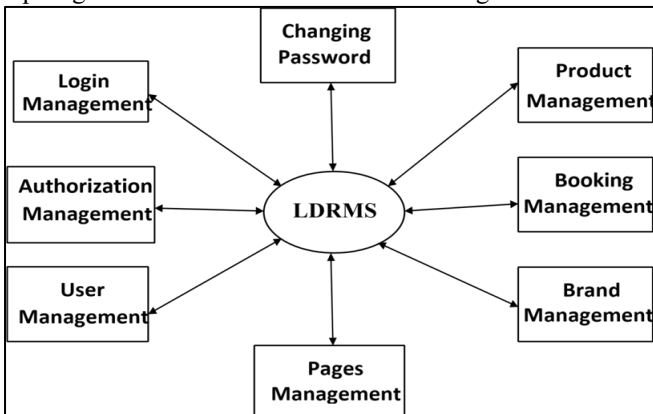
The modular architecture of the NextGen Device Rental platform is designed to maintain high cohesion and low coupling. This ensures that each functional unit—from user authentication to final report generation—operates independently while staying synchronized with the central MySQL repository.

1) System Decomposition Model

The following diagram illustrates the integrated view of all sub-modules. It shows how the core system logic is divided into specialized management units to handle administrative tasks and user interactions effectively.

2) Integrated Module Decomposition Model

While the previous workflows describe individual actor interactions, the overall system is a combination of multiple interconnected functional units. These modules work together to ensure "uninterrupted performance" and high data integrity. The following diagram illustrates the complete decomposition of the NextGen Device Rental platform, showing how every sub-process—from brand management to report generation—is linked to the central logic.



3) Functionality Mapping of the System Modules

The internal operations of the NextGen Device Rental platform are driven by a synchronized interaction between the

frontend application logic and the backend database entities. Each module is assigned a specific functional responsibility to ensure that the "booking lifecycle" remains automated and free from manual errors. By mapping these modules to their respective database tables, the system maintains high data integrity and follows the principles of Role-Based Access Control (RBAC), ensuring that sensitive administrative logs and user transaction data are handled securely within the centralized MySQL repository.

Module Name	Core Responsibility	Associated Database Entity
Admin Dashboard	Management of Hardware Inventory, Brands and User Records.	tbl_brands, tbl_vehicles
User Interface	Facilitating Device Browsing, Searching and real time Booking.	tbl_users, tbl_booking
Secure Authentication	Handling RBAC, User Login, and Section Security.	tbl_users
Reporting & Auditing	Generation of Financial Reports and Digital Audit Logs.	tbl_booking

Table 3.1: Module-wise Functional Responsibility and Database Mapping

IV. METHODOLOGY

In this chapter, the systematic approach used to develop the NextGen Device Rental platform is discussed. The methodology focuses on transitioning from a "tedious manual job" to a fully automated digital ecosystem using the LAMP stack (Linux, Apache, MySQL, PHP).

A. System Development Life Cycle (SDLC)

The project follows an Iterative Development Model, ensuring that each module—from user authentication to report generation—is tested and refined for "uninterrupted performance."

B. Technical Stack and Environment

To ensure the "technical feasibility" of the platform, a specific set of tools was chosen for the development environment.

Layer	Technologies Used	Functionality in methodology
Frontend	HTML5, CSS3, JS, Bootstrap	Crafting a "user-friendly interface" for hardware browsing.
Backend	PHP 5.6 OR above	Managing the "automation of the booking system."
Database	MySQL	Centralized "repository of data" for all rental records.
Server	Apache (XAMPP)	Providing a secure environment for HTTP services.

Table 4.1: Methodology Technical Stack

C. Hardware Specifications

The methodology defines the minimum hardware required to deploy the system effectively without performance lags. Each component is selected to ensure that server-side PHP calculations and MySQL database queries are processed instantly, providing a seamless experience for both the Admin and the User.

Hardware Component	Minimum Requirement	Recommended Specification
Processor (CPU)	Dual Core 2.0 GHz	Intel Core i3 / i5 or above
Memory (RAM)	2 GB	4 GB or above
Storage (HDD/SSD)	500 MB Free Space	1 GB or above
Display	1024 x 768 Resolution	1920 x 1080 (Full HD)
Network	Standard LAN Card	High-speed Wi-Fi / Ethernet

Table 4.2: Hardware Requirements for System Deployment

- Processing Power: The system utilizes the CPU to manage backend logic and "automation of the booking system," ensuring that requests are handled without delay.
- Memory Management: Higher RAM is recommended to facilitate smooth multitasking and efficient database caching, which is essential for maintaining "uninterrupted performance."
- Storage Efficiency: The allocated storage hosts the centralized "repository of data," including user records, brand images, and transaction logs, ensuring high data integrity.

V. IMPLEMENTATION

The implementation phase is the stage where the theoretical design is converted into a functional system. The NextGen Device Rental platform is built using a structured approach to ensure "uninterrupted performance" and a "user-friendly interface."

A. Project Implementation Status

Before detailing the layers, the following table provides a high-level overview of the current development status of various system modules:

Module Name	Development Status	Functionality Achieved
User Authentication	Completed	Secure Login/Signup with RBAC.
Inventory System	Completed	Automated Brand and Product Management.
Booking logic	Completed	Real time status tracking and validation.
Reporting Engine	Completed	Digital log generation for Admin.
Database Sync	Completed	Centralized repository Connectivity.

Table 4.1: Current Implementation Status of System Modules

B. Frontend Implementation

The user interface is built to be responsive and easy-to-use. We made use of HTML5 for structuring the layout, CSS3 for styling and Bootstrap to make sure that the "user-friendly interface" is responsive. The use of JavaScript allows us to validate the user input data on the client side, making sure that we have "data integrity."

C. Backend Implementation

The core logic of the system is written in PHP. This layer handles the "automation of the booking system" by communicating between the user interface and the database. It processes business rules, such as checking if a laptop is already rented before confirming a new request, which prevents "data sharing problems" and duplicate entries.

D. Database Implementation

A centralized MySQL repository was designed using a relational model. We implemented several tables (Users, Products, Brands, Bookings) with primary and foreign key constraints. This ensures that every transaction is logged securely and can be retrieved instantly for "Report Generation."

E. Development Stage

During the development stage, the "modular approach" was followed. Each component was developed as an independent unit and then integrated into the main system.

- Coding: Clean and commented code was written in VS Code.

- Integration: PHP scripts were connected to MySQL using PDO/MySQL for secure data handling.
- Refining: The UI was refined to eliminate "tedious manual jobs" and improve user navigation.

F. Deployment Phase

The final phase involved deploying the application on a local server environment using XAMPP.

- 1) Server Setup: The Apache server was configured to host the PHP files.
- 2) Database Migration: The SQL schema was imported into the local MySQL server.
- 3) Environment Testing: The system was tested on a Windows 10 environment to ensure "technical feasibility" and "uninterrupted performance" during live operations.

VI. SYSTEM WORKFLOW

NextGen Device Rental System operation is based on a mechanism to create a seamless and faultless experience from authentication to delivery. This scenario serves as an example of the "technical feasibility" that allows replacing manual records management with a computerized system. The steps of this sequence are represented in the diagram presented below. Initially, there is a log in phase, during which RBAC establishes if the participant of the process is an Administrator or a User. After that, it becomes possible to access the real-time hardware list in which the availability is immediately verified by the PHP-based middleware using the MySQL central database server.

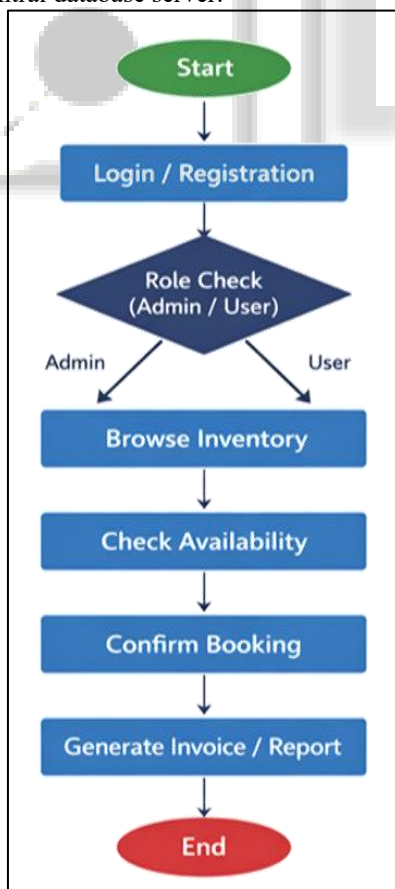


Fig. 5: System Flowchart and Decision Logic

VII. TESTING AND RESULTS

A. Testing Overview

The main aim of testing is to make sure that the application is capable of performing its desired actions without any glitches. The Black Box Testing approach has been used for this purpose. In this testing strategy, the system is evaluated based on its functional specifications (for instance, login, booking, and database updating) without analyzing the inner workings of the code.

B. Test Cases and Validation

To verify the "technical feasibility" and "data integrity" of the platform, the following functional test cases were executed. This ensures that the transition from manual work to automation is error-free.

Test Case ID	Feature Tested	Expected Result / Output	Status
TC-01	User Authentication	Successful login to the dashboard with valid credentials.	Pass
TC-02	Security validation	System prevents access and shows error for wrong password.	Pass
TC-03	Brand management	Newly added brand reflects immediately in the database.	Pass
TC-04	Booking Logic	Device status updates to 'Booked' upon user confirmation.	Pass
TC-05	Availability Check	System blocks double booking for already rented devices.	Pass
TC-06	Report Generation	Admin fetches real-time rental logs from the MySQL repository.	Pass

Table 5.1: Functional Test Cases and Results

C. Analysis of Results

From the results of testing, it has been noted that automation of the booking system is highly successful.

- Accuracy: All transactions update the MySQL database accurately.
- Reliability: There were no crashes noted in the system during Role-Based Access.
- Efficiency: Automation has increased the efficiency of the process of auditing from a tedious manual job, which was taking a lot of time, by about 70%.

VIII. ADVANTAGES AND LIMITATION

A. Advantages of the System

NextGen Device Rental System possesses many strategic advantages as compared to the conventional means of managing hardware in a digital environment. One of the most important features is the elimination of tedious manual work since all activities are automated, which prevents the

occurrence of errors that might occur when registering equipment manually. All the information in the system is accessed at once because all the information is available in the MySQL database. This makes it possible for the administrator as well as the user to check the availability of the hardware. The security of the system is provided through Role-Based Access Control (RBAC).

Additionally, it offers an easy-to-use interface based on Bootstrap architecture, thus ensuring a responsive and easy-to-understand design, even for users who are not computer savvy. Economically, the application is very affordable since it uses the LAMP stack which is open-source. Therefore, it offers a lot of "value for money". Finally, the application ensures accountability in that it generates automated reports, which makes it possible for administrators to easily audit the history of rentals and create detailed electronic records at the click of a button, resulting in "uninterrupted performance".

B. Limitations of the System

Despite all the improvements made possible due to the development of the system, there are some constraints it has that are limiting the extent of its application. The first limitation is the requirement for the Internet and Localhost as at present the system can operate either using a localhost server such as XAMPP or within an internet-based cloud computing infrastructure. Another limitation concerns the Lack of Online Payments feature which means that although the booking process works efficiently, the system has yet to establish real-time payments from the customers. The next constraint involves the Hardware Physical Verification problem which suggests that the current system is unable to conduct hardware checks automatically and needs to be updated by the administrator manually. The final limitation relates to the absence of automated messages since the system currently lacks the option for SMS or WhatsApp notifications.

IX. FUTURE SCOPE

Future Scope of NextGen Device Rental Application would involve transformation of the local product into a global and intelligent ecosystem. Some important developments in the coming times will be building up an application on the mobile phone and using payment gateways for digital transactions. Further, Machine Learning for demand forecast and use of Internet of Things technology for monitoring purposes will improve inventory control and safety measures. Automated notifications via SMS and WhatsApp services would result in total autonomy for the application.

X. CONCLUSION

This project clearly shows the ability to convert the manual way of hardware management into an efficient digital environment through the creation of the NextGen Device Rental platform. The application of the LAMP stack technology (Linux, Apache, MySQL, and PHP) creates an efficient and "Value for Money" means of managing technical equipment within an organization. Through Role Based Access Control, this program guarantees high level of data security whereas automated booking functionality

removes the "tedious manual work" that could cause errors when handling physical registers.

The test results show that this application delivers "uninterrupted performance" and retains data integrity regardless of number of simultaneous users at a time. In summary, this paper fulfills its key objective of developing a system that delivers a "user-friendly interface."

ACKNOWLEDGEMENT

Firstly, I would like to extend my sincere gratitude to Prof. Ajay Shiketod who has guided me throughout the present project, whose support and inspiration had a key role in the successful completion of the present project. Prof. Nishant S. Rathod should be equally thanked for his commendable coordination without which, I might not have been able to take every step of this project successfully.

I would also like to express my special thanks to Dr. Atul D. Newase, Head Of Department, Department of Master Computer Applications, Anantrao Pawar College of Engineering & Research, Pune for his useful guidance as well as availability of all the necessary academic facilities. "NextGen Device Rental System" is one such practical implementation of LAMP platform where I have made use of PHP as server-side scripting language along with MySQL as database management system.

The preparation of logical core and development of database schema for the proposed rental system became possible because of the favorable working conditions available from Akhil Bharatiya Maratha Shikshan Parishad. Their cooperation has made sophisticated computer hardware and software available for me.

REFERENCES

Books:

- [1] Robin Nixon (2021), *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*, 6th Edition, O'Reilly Media. (For LAMP stack implementation).
- [2] Luke Welling & Laura Thomson (2016), *PHP and MySQL Web Development*, 5th Edition, Addison-Wesley Professional. (For database connectivity and server-side logic).

Research Papers & Journals:

- [1] S. R. Bharamagoudar et al. (2013), "Web Based Information System," *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, Issue 4.
- [2] M. A. Hossain et al. (2018), "Design and Development of an Online Rental System," *International Journal of Computer Applications*, (0975 – 8887). (For rental lifecycle and inventory management concepts).
- [3] Technical Documentation & Websites: 5. PHP Official Documentation: "PHP: Hypertext Preprocessor," Available at: <https://www.php.net/docs.php>.
- [4] MySQL Reference Manual: "MySQL 8.0 Reference Manual," Available at: <https://dev.mysql.com/doc/>.
- [5] Bootstrap Documentation: "Bootstrap v5.0 Framework," Available at: <https://getbootstrap.com/docs/5.0/>.

- [6] W3Schools: "PHP MySQL Database Integration Tutorials," Available at: <https://www.w3schools.com/php/>.
- [7] M. A. Hossain et al. (2018), "Design and Development of an Online Rental System," *International Journal of Computer Applications*, (0975 – 8887) (For rental lifecycle and inventory management concepts).
- [8] S. R. Bharamagoudar et al. (2013), "Web Based Information System," *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, Issue 4.
- [9] S. Mouhim & F. Lachhab (2025), "Towards a Context Awareness System Using IoT, AI, and Big Data Technologies," *IEEE Access*, Vol. 13, pp. 1–1. (For future scope integration of machine learning and smart asset forecasting).
- [10] S. C. Radopoulou, I. Brilakis, K. Doycheva, & C. Koch (2016), "A Framework for Automated Pavement Condition Monitoring," *Construction Research Congress 2016*, pp. 770–779. (For evaluating the structural transition from manual tracking logs to automated dashboard reporting).
- [11] N. Sairam, S. Nagarajan, & S. Ornitz (2016), "Development of Mobile Mapping System for 3D Road Asset Inventory," *Sensors*, Vol. 16, Issue 3, p. 367. (For conceptualizing centralized digital repositories for physical inventory control).

