



International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

“SMART CITY PARKING SYSTEM USING PHP”

Dr.Omkar Pattnaik¹, Vivek Chadrakant Chavan², Sohan Prakash Kotwal³, Nitin Manohar Bhamare⁴

Sandip University, Nashik, Scholar Department of Computer Science & Engineering Sandip University, Nashik, India

¹ Assistant Professor, School of Computer Science and Engineering, Sandip University, Nashik

ABSTRACT:

The need for automated solutions has grown as a result of urban parking inefficiencies. The Smart Car Parking System presented in this paper was created only with PHP and MySQL and is housed on a XAMPP server environment. Through a web interface, the system offers a digital solution for real-time parking space management. It has an admin dashboard for slot administration and reporting, slot reservation and cancelation, and secure user authentication. The system is appropriate for offices, residential areas, and college campuses because of its scalability, affordability, and ease of deployment. This study shows how traditional server-side technology may effectively update traditional parking techniques.

Keywords: Real-time monitoring, parking management, slot reservations, web-based applications, PHP, MySQL, smart parking systems, admin dashboards, responsive interfaces, traffic reduction,

INTRODUCTION

Due to the growing number of automobiles and the scarcity of parking spots, parking is a major urban difficulty that cities all over the world must deal with. Effective parking management is now more important than ever as urbanization keeps growing. Drivers in many cities frequently spend a great deal of time looking for parking spaces, which not only wastes fuel and increases pollution but also adds to traffic congestion and commuter stress. Conventional parking systems often use physical entry passes, manual tracking techniques, or simple token-based systems. These antiquated techniques usually lead to ineffective use of available space, annoyance for users, longer wait times, and heavy administrative workloads for parking managers.

Given these difficulties, there is an increasing need to put in place more intelligent, tech-driven solutions that make parking easier for managers and users alike. By utilizing PHP and XAMPP to propose an automated and digital parking spot management system, this study seeks to address these problems. While XAMPP offers a comprehensive, open-source web server package that is simple to install and operate, making it perfect for development and testing purposes, PHP, a popular server-side scripting language, gives the flexibility and functionality needed to create dynamic web applications.

The main goal of the suggested system is to develop a web application that is easy to use, scalable, and lightweight in order to streamline parking spot administration. The time and effort spent looking for parking spots will be decreased because users will be able to register, log in, and book spots in advance based on real-time availability. The user experience can be further improved by providing a visual depiction of the slots that are available and occupied, which enables users to make well-informed decisions fast and easily. Parking lot managers will also be able to effectively manage reservations, adjust availability, track slot usage, and provide reports thanks to the system's administrator interface.

This technology attempts to close the gap between the limits of traditional parking infrastructures and the increasing demands of urban mobility by combining real-time data management with intuitive user interfaces. It is made to be easily adapted to a variety of settings, including public parking lots, office buildings, retail centers, and educational institutions. Additionally, using open-source technologies guarantees that the solution is affordable and available to a variety of organizations with different financial capacities.

Overall, this study's main objective is to show how automation and digital transformation may greatly improve parking management systems' scalability, ease, and efficiency. The project's goal is to use PHP and XAMPP to provide a workable and feasible answer to one of the most enduring issues in transportation management and urban development.

RELATED WORKS

Numerous hardware- and software-based intelligent parking systems have been developed as a result of the growing urban population and car utilization. The basis for this study is the progress of smart parking systems, which has been aided by a number of noteworthy works.

1. Solutions Driven by Mobile Applications

Mobile-based slot booking was the subject of certain initiatives, such as Patel and Mehta's (2019) Android-based parking app. These applications combine live availability and location monitoring, however they are platform-specific and need further work to be cross-platform compatible.

2. Systems Based on the Web Making use of PHP and MySQL

Using PHP and MySQL, Kumar and Singh (2019) presented a web-based parking management system that provides administrators and users with an easy-to-use interface. Although their approach lacked an advanced admin dashboard and real-time slot conflict prevention, it showed that open-source technology might provide scalable and affordable solutions.

3. Data-Driven and Cloud Methods

In order to facilitate remote data access and analytics, contemporary research also investigates the integration of cloud computing with parking systems (Joshi et al., 2021). Although these systems are strong, they are frequently difficult to set up and maintain in environments with limited resources.

EXISTING SYSTEM

Parking systems from the past and early generations have mostly been manual or semi-automated, mostly depending on human presence, human supervision, and crude infrastructure. These systems require users to physically visit the location in order to verify availability, which frequently results in wasted time, traffic, gasoline, and irritated users. Hardware-based smart parking systems that use sensors, RFID tags, or smartphone apps have been implemented in some cities. These systems could consist of:

- **Sensor-Based Detection:** Real-time availability updates are provided by ultrasonic or infrared sensors that identify the presence of cars in each slot.
- **RFID-Based Entry Systems:** Cars with RFID tags can enter gated parking lots automatically.
- **Mobile App Interfaces:** These let consumers use their phones to look for and book slots.
- **Cloud-Based Platforms:** Provide sophisticated analytics and internet data processing and storage.

While these systems offer automation and real-time data, they also come with notable limitations:

- **High Initial Setup Cost:** The cost of deployment is greatly increased by sensors, cameras, and other devices.
- **Dependency on Infrastructure:** Needs internet, steady electricity, and sensor calibration.
- **Limited Accessibility:** Mobile applications frequently only work with particular operating systems (like iOS or Android).
- **Difficult Maintenance:** Hardware parts may break down and need constant assistance.

On the other hand, simple web-based systems constructed with PHP and MySQL provide a more affordable and accessible option; nonetheless, the majority of current implementations lack sophisticated capabilities like:

- Real-time booking conflict avoidance.
- An interface that is mobile-friendly and responsive.
- Data visualization combined with administrative controls.
- Features for booking and canceling are seamless.

MATHEMATICAL MODEL

We define the Smart Car Parking System as a function-based model that mathematically depicts the interaction between the database (MySQL), server-side logic (PHP), and the user. Deterministic processes with well-defined inputs, processing functions, and outputs form the basis of the model.

1. System Definition

Let the system be represented as:

$$S = \{I, P, O, F\}$$

Where:

- I = Input Set
- P = Process Set
- O = Output Set
- F = Function Set

2. Input Set (I)

$$I = \{U, L, S, T, A\}$$

Where:

- U = User credentials (username, password)
- L = Login/Logout action
- S = Slot ID requested
- T = Booking time and date
- A = Action (Book, Cancel, View)

3. Process Set (P)

$P = \{ \text{Authenticate}(U), \text{CheckAvailability}(S, T), \text{BookSlot}(U, S, T), \text{CancelSlot}(U, S), \text{ViewStatus}() \}$

- $\text{Authenticate}(U) \rightarrow$ verifies login credentials
- $\text{CheckAvailability}(S, T) \rightarrow$ checks if slot S is available at time T
- $\text{BookSlot}(U, S, T) \rightarrow$ books slot S for user U at time T
- $\text{CancelSlot}(U, S) \rightarrow$ cancels booking
- $\text{ViewStatus}() \rightarrow$ returns current slot status

4. Output Set (O)

$O = \{ \text{Confirmation}, \text{Slot_Status}, \text{Error_Message}, \text{Dashboard_View} \}$ Where:

- Confirmation $\rightarrow$ Booking or cancellation success message
- Slot_Status $\rightarrow$ List of available/booked slots
- Error_Message $\rightarrow$ Authentication or booking error
- Dashboard_View $\rightarrow$ Admin view of system state

5. Function Set (F)

Mathematically, the system's behavior can be represented by: $O = F(I)$

Where F is a set of deterministic functions:

- F1: $\text{Authenticate}(U) \rightarrow \{ \text{true}, \text{false} \}$
- F2: $\text{CheckAvailability}(S, T) \rightarrow \{ \text{available}, \text{booked} \}$
- F3: $\text{BookSlot}(U, S, T) \rightarrow \text{Confirmation} \mid \text{Error_Message}$
- F4: $\text{CancelSlot}(U, S) \rightarrow \text{Confirmation} \mid \text{Error_Message}$
- F5: $\text{ViewStatus}() \rightarrow \text{Slot_Status}$
- F6: $\text{AdminDashboard}() \rightarrow \text{Dashboard_View}$

Constraints & Conditions

- A user cannot book the same slot already booked for the same time (Slot S must be available).
- Only authenticated users can perform booking and cancellation operations.
- Each slot can be assigned to one user per time block ($\forall S, T \rightarrow \max 1 \text{ booking}$).

PROPOSED SYSTEM

1. The goal of the suggested system is to create a PHP-based smart car parking system that offers a practical, effective, and user-friendly way to manage and use parking spots in cities. By providing a web-based platform that enables users to check availability, reserve spaces, and manage their parking remotely, it overcomes the drawbacks of conventional and hardware-based parking systems.

Essential Elements of the Suggested System:

1. Interface Based on the Web

Any device with an internet connection can access the system's user-friendly and responsive online interface. It is constructed with MySQL as the backend database, HTML, CSS, and JavaScript for the front end, and PHP for server-side scripting.

2. Role management and user authentication

For both administrators and users, the system offers safe login and registration features. Various roles are established:

- o User: Able to look for open times, make, and cancel reservations.
- o Admin: Able to view reservations, control parking spaces, and produce reports.

3. Availability of Slots in Real Time

The real-time status of parking spaces, including whether they are filled, reserved, or open, is visible to users. This data is kept in a MySQL database and is updated dynamically using PHP backend logic.

4. Making a reservation and booking

Parking spaces can be reserved in advance by users. By locking the chosen time window after confirmation, the system stops double booking. Users can also postpone or cancel reservations.

3. The Admin Panel

The administrator has complete control over parking operations, including adding or removing slots, manually updating availability as necessary, and keeping an eye on user activities, thanks to a dedicated dashboard.

4. Confirmation and Notices

Users receive a confirmation message through the interface after making a successful reservation (if extended, they can also receive it by email or SMS). Admins are also informed of cancellations and new reservations.

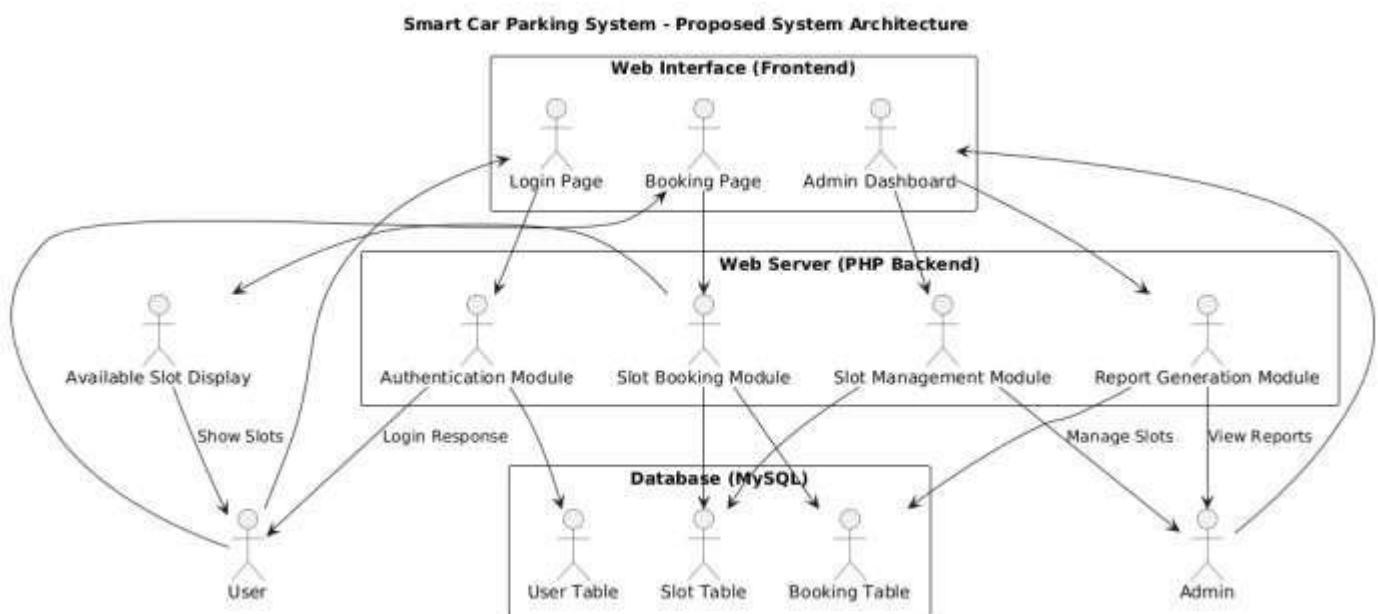
5. Analytics and Reports

To aid in decision-making and system enhancements, the system can produce data on parking slot utilization, total reservations, peak usage hours, and user statistics on a daily, weekly, or monthly basis.

Advantages of the Proposed System:

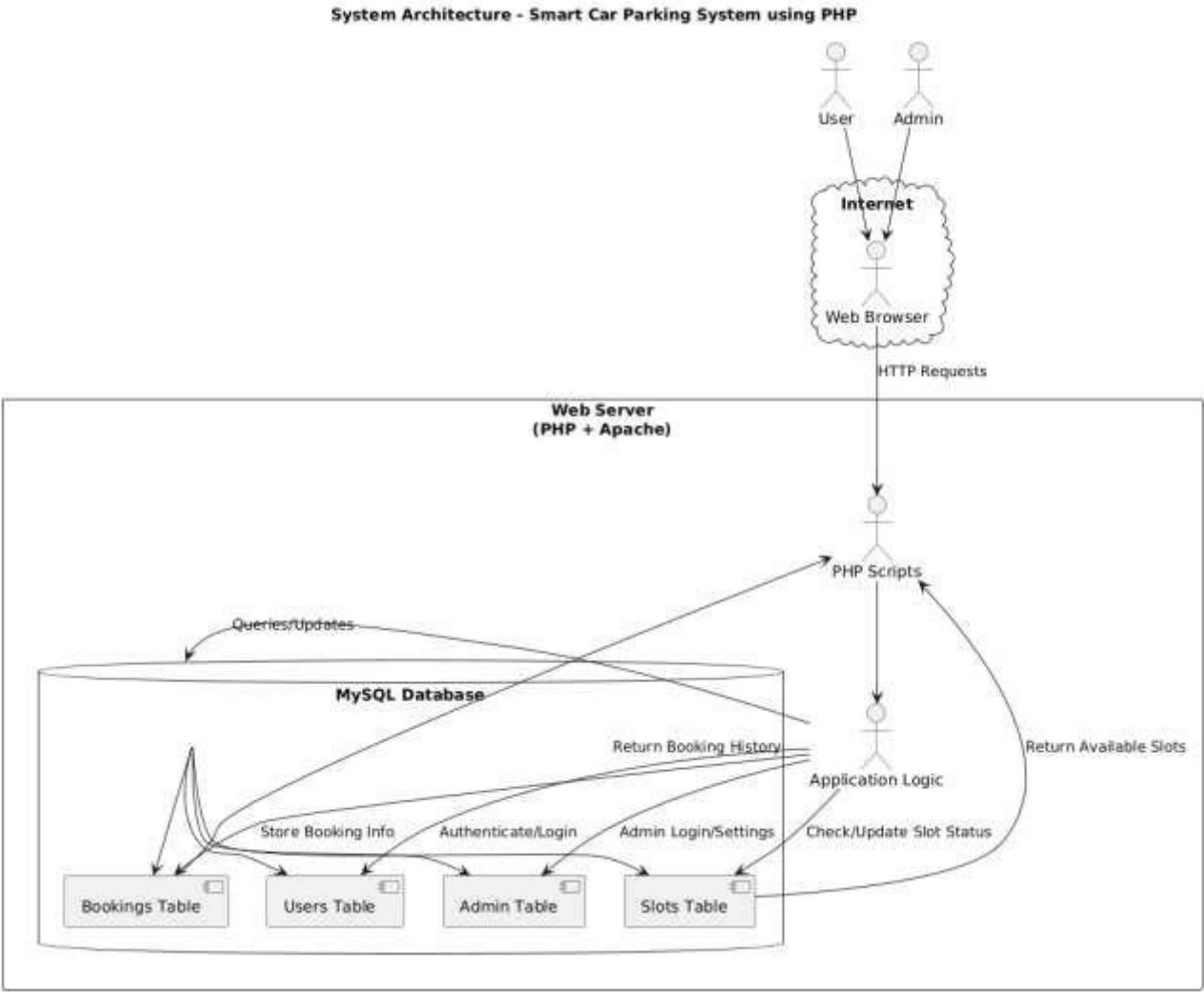
- It does away with the requirement for on-site parking lot monitoring.
- Lessens traffic and saves user time.
- An affordable option that makes use of open-source technologies.
- Adaptable and scalable for public parking lots, office buildings, campuses, and shopping centers.

Fig. 1. Proposed System



A. System Architecture

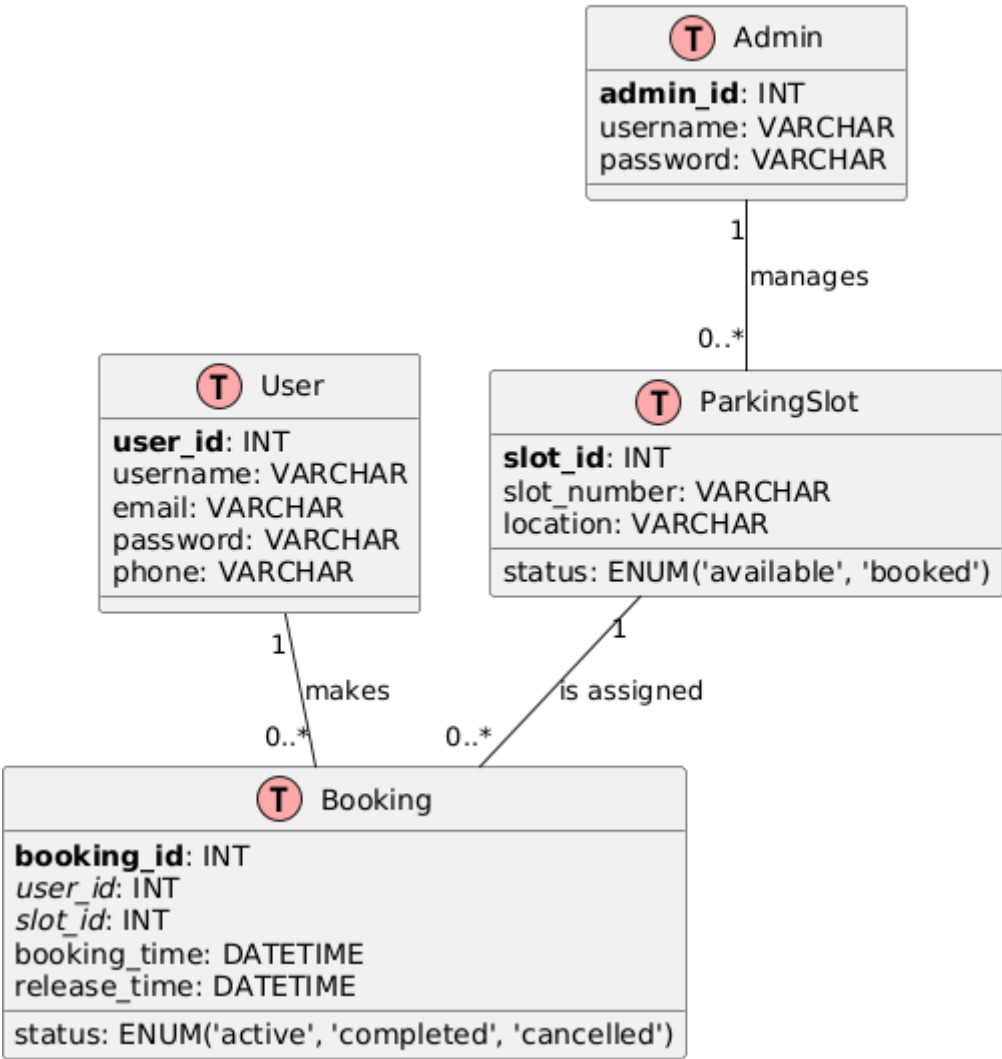
Fig. 2. System Architecture



B. ER Diagram

Fig. 3 : ER Diagram

ER Diagram - Smart Car Parking System



METHODOLOGY

PHP was used to create the Smart Car Parking System using an organized and methodical software development process. successful requirement analysis, modular design, reliable implementation, and successful testing are all guaranteed by the process. The steps listed below describe the methodology:

1. Requirement Analysis

To determine the needs of end users (vehicle owners and administrators) and the necessary technical capabilities, a thorough requirement study was carried out. Key features found consisted of:

- the administrator dashboard for tracking and reporting;
- Real-time slot availability check;
- Slot booking and cancelation;
- User registration and login;
- Responsive web interface

2. System Design

The system was architected using the **MVC (Model-View-Controller)** design pattern to ensure separation of logic and ease of maintenance:

- **Model:** Handles data logic and database interaction using MySQL.
- **View:** Provides user interface with HTML, CSS.

- **Controller:** Manages user requests, session control, and routing logic using PHP.

Database schema was designed with normalized tables for:

- User information
- Parking slot details
- Booking history
- Admin operations

A flowchart and Data Flow Diagrams (DFDs) were used to visualize system operations.

3. Implementation

The system was implemented using the following technologies:

- **Frontend:** HTML5, CSS3, JavaScript (with AJAX for real-time updates)
- **Backend:** Core PHP (without frameworks for simplicity and performance)
- **Database:** MySQL
- **Server:** Apache (XAMPP for local development and testing)

Key functionalities coded:

- Secure user authentication using hashed passwords
- Conflict detection logic to prevent double booking
- Slot status updates after booking or cancellation
- Admin panel for slot and user management

4. Testing and Validation

A series of testing phases were conducted:

- **Unit Testing:** Each module (login, booking, cancellation) was tested in isolation.
- **Integration Testing:** Modules were tested in combination to ensure smooth interaction.
- **System Testing:** End-to-end testing was done for the entire system.
- **User Acceptance Testing:** Feedback was collected from sample users to evaluate usability and performance. Test cases included normal flow, boundary conditions, and invalid input handling.

5. Deployment and Future Scope

The system is hosted on a local server using XAMPP and can be deployed on a live server with minimal configuration changes. Future enhancements may include:

- Integration of IoT-based sensors for automatic slot detection
- SMS/Email booking confirmations
- QR code generation for contactless entry
- Mobile app development for improved accessibility

RESULT AND DISCUSSION

Results:

PHP was used to successfully create and implement the suggested Smart Car Parking System in a test setting. Following tests of the system's different admin and user features, the following results were noted:

- Users were able to create accounts and log in securely.
- PHP session handling was used to preserve login sessions correctly.
- **Slot Booking and Real-Time Availability**
 - In real time, users could see every parking space that was available.
 - After reservations, changes in slot availability were appropriately recorded in the booking system.

- The system prevented two reservations for the same time slot.
- **Admin Control Panel**
 - Parking spaces could be added, changed, or removed by administrators.
 - The admin dashboard offered a thorough summary of every reservation and user action.
 - If necessary, the slot status could be manually modified.
- **The interface's responsiveness and**
 - The user experience was seamless, with little loading time
 - an easy-to-use navigation system; the online interface was fully responsive across PCs, tablets, and smartphones.
- **Confirmation of Booking and Administration**
 - Customers could check their booking history and cancel reservations; they received confirmation messages right away after making a reservation.
- **Data Security and Management**
 - All user and reservation data were successfully stored by MySQL.
 - Encryption was used to safely store user passwords.
 - To stop SQL injection and other security risks, data validation and sanitization were put into place.

Discussion:

The system's main goal of streamlining the parking process via a web-based platform was accomplished, greatly cutting down on the time and effort needed to locate and reserve parking spaces. The selection of PHP and MySQL worked well for creating a system that is scalable, safe, and dynamic. A number of advantages were noted, such as improved ease for users to check the availability of slots and make reservations from a distance, effective administrative control over parking operations, real-time updates for precise slot status, and an intuitive interface that even non-technical users could use. The development process was not without difficulties, though. AJAX integration was required to ensure data consistency and real-time changes without requiring a page refresh, and managing concurrency in bookings required careful session management to prevent conflicts. Furthermore, in a true deployment scenario, manual updates for slot status may be necessary if real-world sensor integration is not available. A dedicated mobile application for even more seamless access, the use of QR codes or RFID technologies for automated gate access, the integration of IoT sensors for automatic slot detection, and the implementation of email or SMS notifications for booking reminders are just a few of the opportunities for system improvement that lie ahead.

Result and Discussion - Smart Car Parking System using PHP

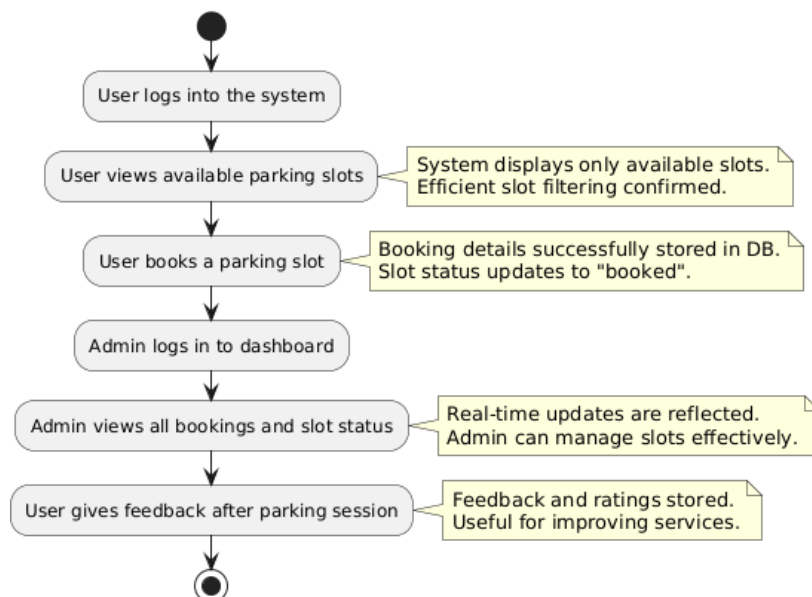
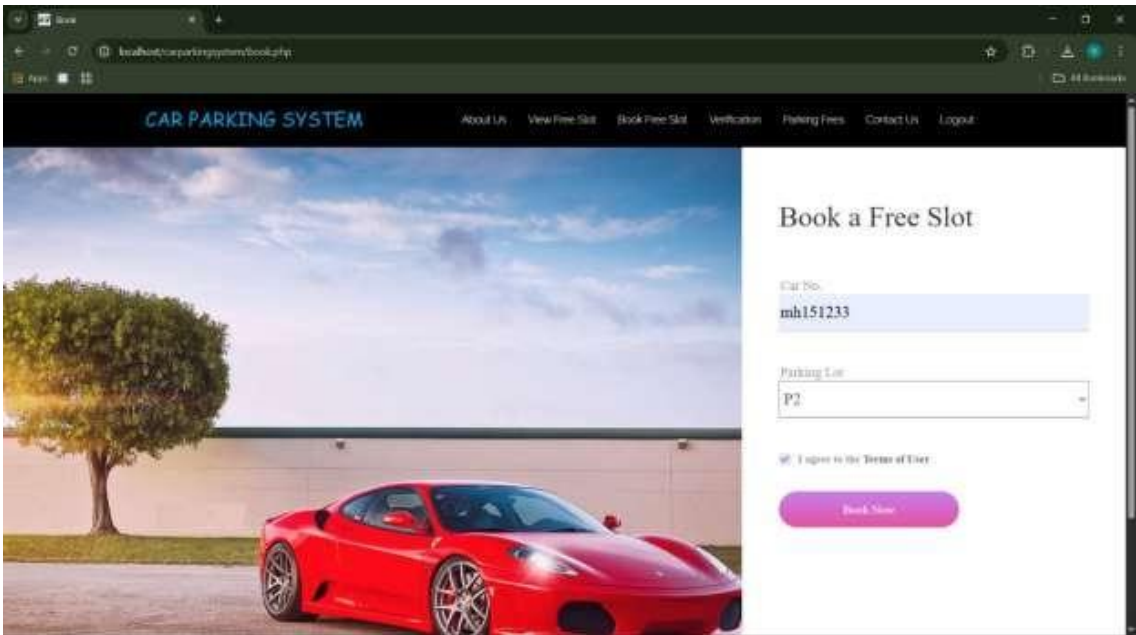
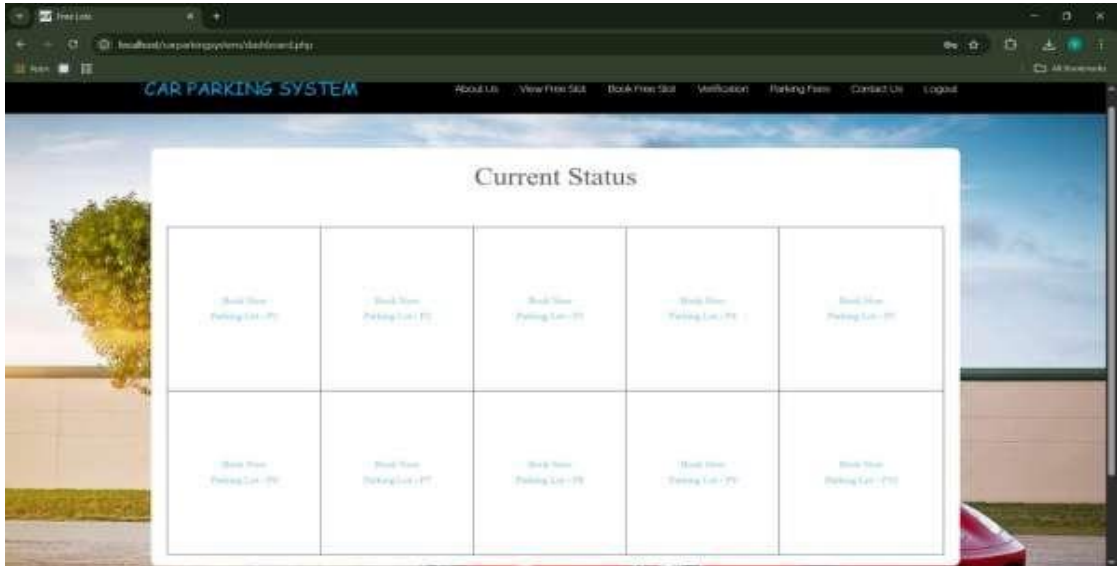
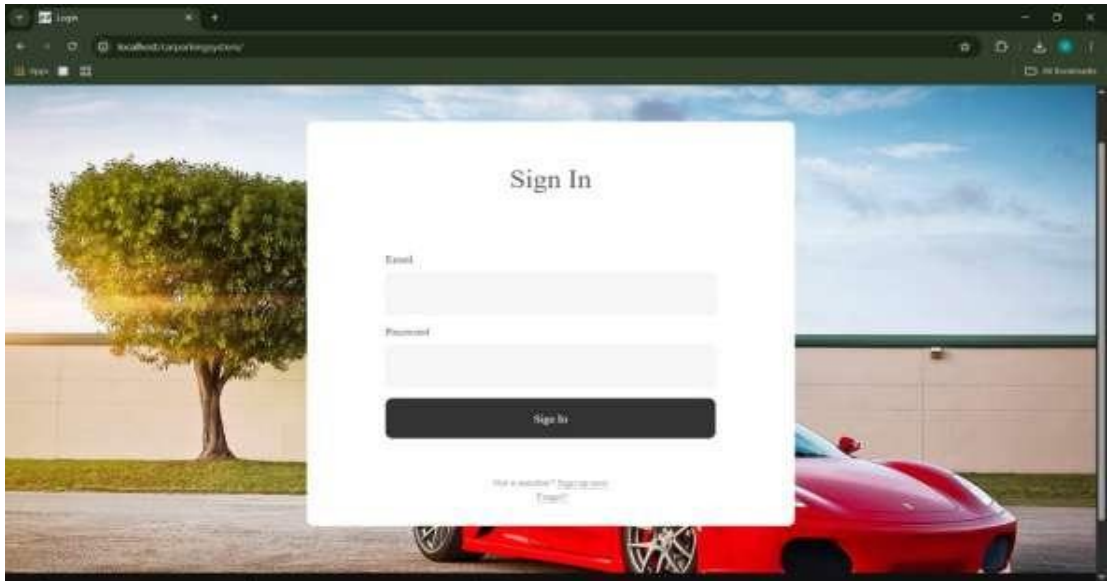
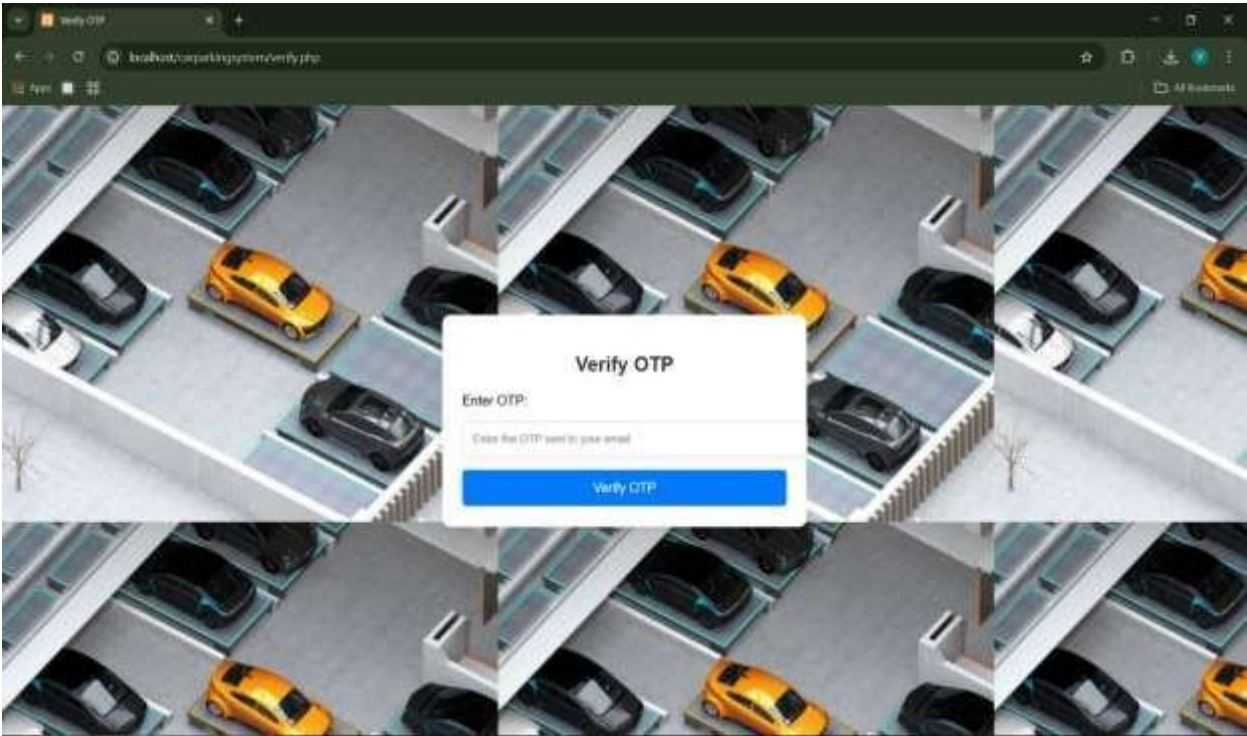
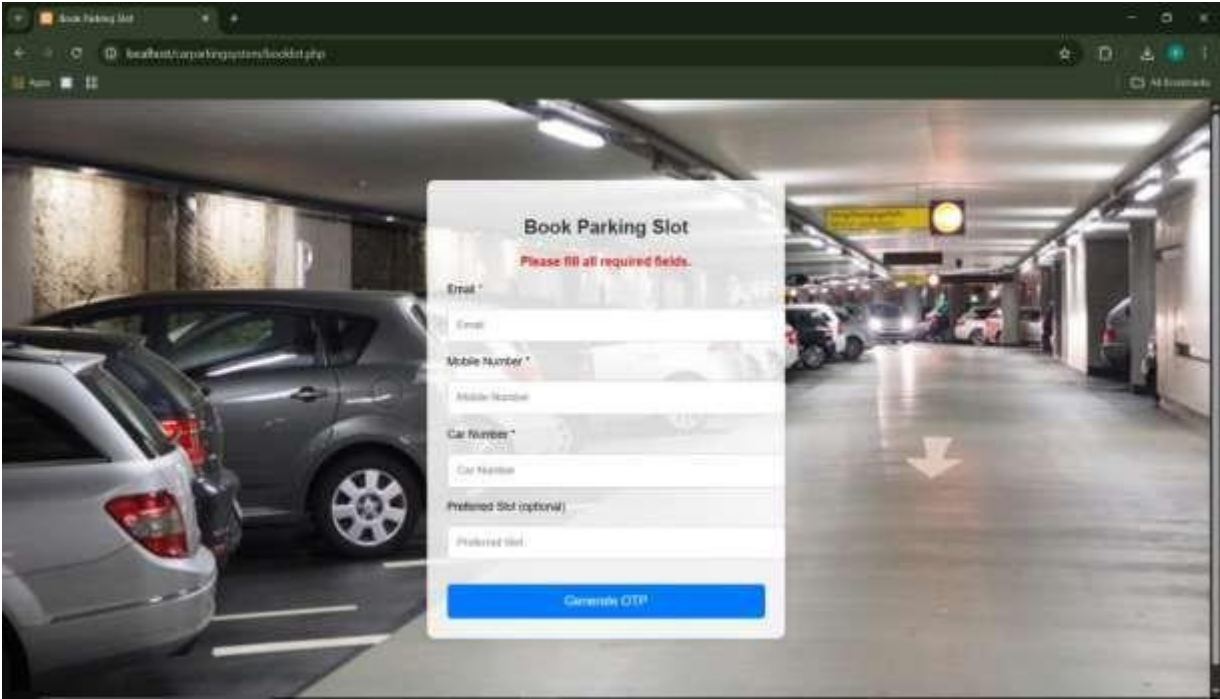
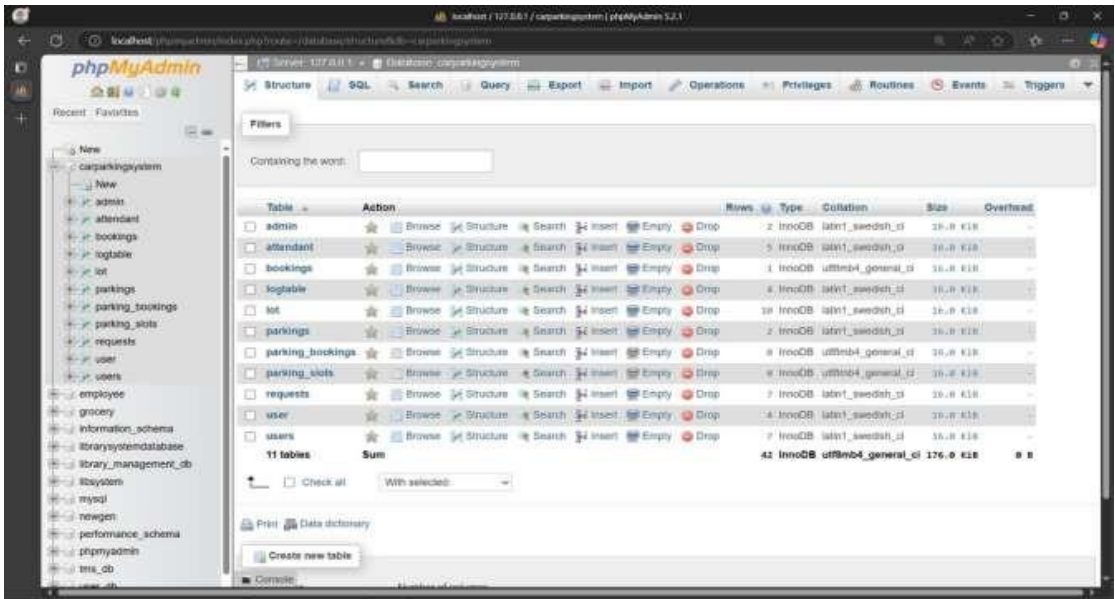
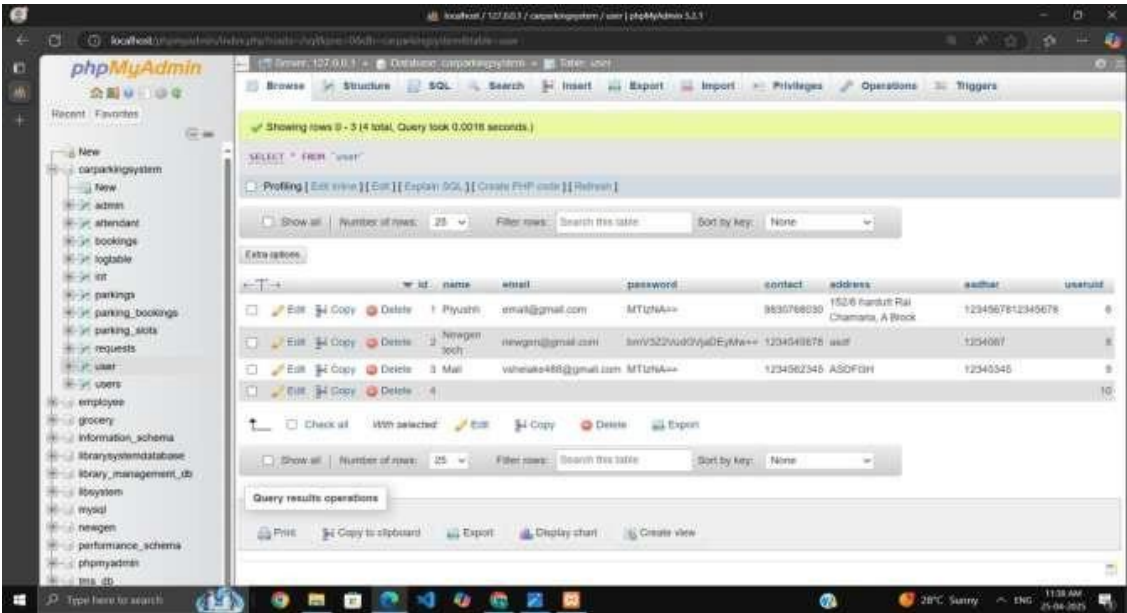
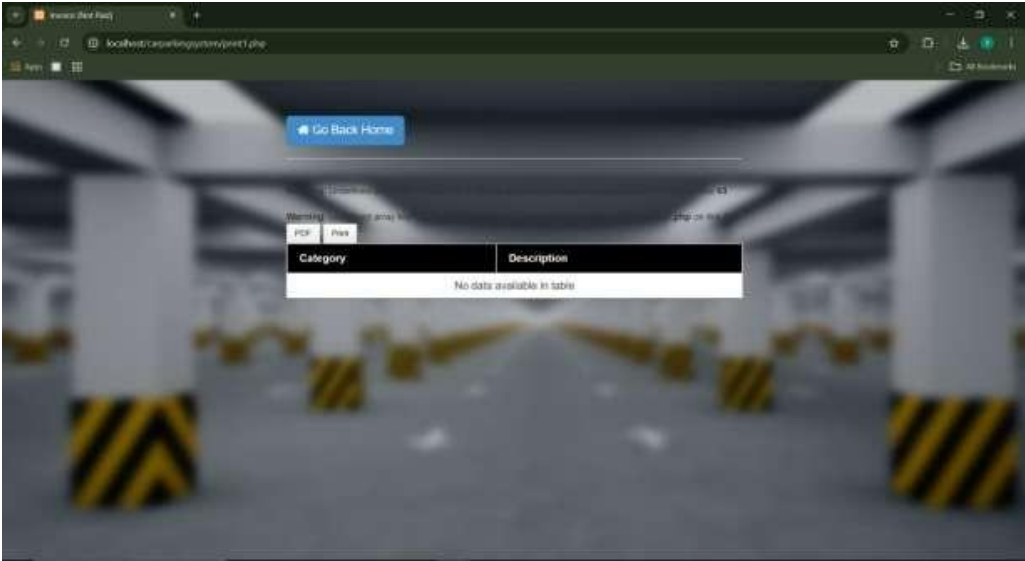


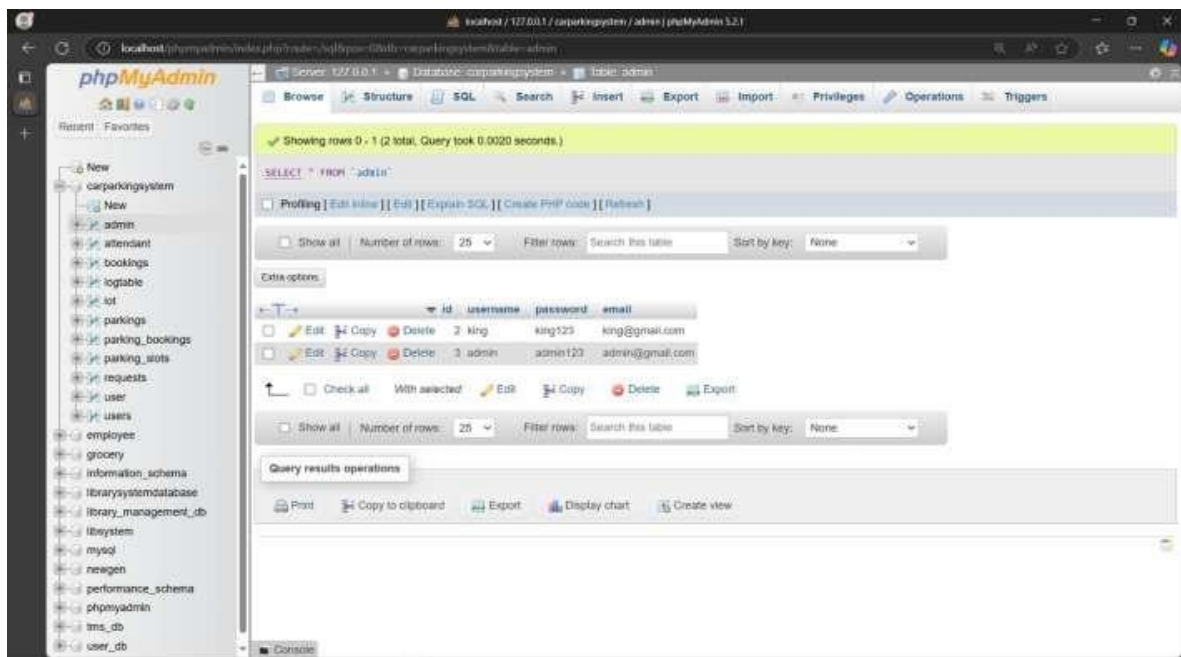
Fig. 4 Results and Discussion

OUTPUT SCREENSHOTS









CONCLUSION

To sum up, the PHP-powered Smart Car Parking System provides a dependable, effective, and affordable way to get beyond the drawbacks of conventional parking techniques. The system offers smooth user authentication, real-time parking spot management, and an easy-to-use admin dashboard for effective operations through the use of web technologies and a strong backend with PHP and MySQL. It effectively maximizes parking space use, lessens traffic congestion, and improves user convenience. Future improvements like real-time sensor integration, specialized mobile applications, and sophisticated analytics can further improve the system's functionality and user experience, making it an even more potent tool for contemporary parking management, even though it currently satisfies the fundamental requirements of a smart parking platform.

Among the system's primary characteristics are:

- Real-time availability checks that guarantee customers can locate available parking spaces fast.
- Simple and safe user authentication for managing accounts and making parking reservations.
- Admin dashboard to manage reservations and track slot availability, increasing operational effectiveness.

This technology guarantees dependability and a seamless user experience through extensive development and testing. The system lowers traffic congestion, maximizes space use, and cuts down on time spent looking for parking by automating parking management.

Further enhancements could include the addition of real-time sensor integration for automatic slot status updates, mobile applications for increased user accessibility, and analytics tools for parking lot operators, even though this system successfully meets the fundamental requirements of smart parking.

FUTURE WORK

The following improvements are suggested in order to better improve the system:

- IoT Integration: Auto-detect slot occupancy using sensors.
- Mobile App: Create an iOS or Android app to make access simpler.
- GPS Navigation: Locate parking lots by integrating Google Maps.
- Online Payments: Include gateways for billing and reservations.
- Dynamic Pricing: Adjust prices according to demand and time.
- Support for Multiple Levels: Expand the system to accommodate parking on multiple floors.
- Admin Analytics: Offer dashboards with revenue and usage data.
- Security Features: Include license plate verification and two-factor authentication.
- Cloud Hosting: Use the cloud to improve scalability and performance.

REFERENCES

1. Sharma, P., Kumar, R., & Singh, A. (2018). IoT based Smart Parking System using Arduino and IR Sensor. International Journal of Computer Applications, 182(37), 12–15.

2. Kumar, V., & Singh, M. (2019). Web-Based Smart Car Parking System Using PHP and MySQL. *International Journal of Engineering Research & Technology (IJERT)*, Vol. 8, Issue 5.
3. Patel, D., Thakkar, H., & Shah, A. (2020). Cloud-Based Smart Parking System. *International Journal of Advanced Research in Computer Science*, Vol. 11, Issue 3.
4. Joshi, S., Mehta, R., & Deshpande, K. (2021). Smart Parking System using IoT and Web Technologies. *IEEE Conference on Emerging Trends in Communication and Computing Technologies (ETCCT)*, pp. 45–49.
5. W3Schools. (2023). PHP Tutorial. Retrieved from: <https://www.w3schools.com/php/>
6. MySQL Documentation. (2023). MySQL Reference Manual. Retrieved from: <https://dev.mysql.com/doc/>
7. Mozilla Developer Network (MDN). (2023). HTML, CSS, and JavaScript Guides. Retrieved from: <https://developer.mozilla.org/>
8. Stack Overflow Community. (2022–2023). Various discussions on PHP login systems, session handling, and database integration.
9. Geeks for Geeks. (2023). PHP MySQL Login System. Retrieved from: <https://www.geeksforgeeks.org/php-login-form-using-mysql/>
10. World Health Organization. (2022). Global Status Report on Road Safety. Retrieved from: <https://www.who.int/publications/>