International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Parking Space Detection And Navigation System

S Revathi[1], Ananya Mallepally[2], Deepthi Vaddagani[3], Navya Kolloju[4]
[1]Assistant Professor, Department Of Information Technology, Bhoj Reddy Engineering College For Women, Hyderabad, India.
[2,3,4]B.Tech Student, Department Of Information Technology, Bhoj Reddy Engineering College For Women, Hyderabad, India.
Mail Id;navyakolloju20@gmail.com[4]

Abstract
The rapid increase in the number of vehicles in urban areas has created significant challenges in managing parking spaces efficiently, as drivers often spend a considerable amount of time searching for available parking slots, leading to traffic congestion, fuel wastage, and environmental pollution. Existing parking systems are mostly manual or lack real-time updates, making them inefficient and unreliable. To overcome these issues, the Parking Space Detection and Navigation System has been developed as a smart solution that enables users to locate, book, and manage parking spaces effectively through a simple interface. The system integrates location-based services using Google Maps to provide real-time navigation and nearby parking availability, allowing users to book parking slots instantly or reserve them in advance. The application is built using a Flutter-based frontend, a Node.js and Express backend, and MongoDB for database management, where parking availability is maintained through a centralized database. The system also includes a detector module for booking verification during check-in, ensuring proper validation of parking usage. Additionally, users can add their own parking spaces, and administrators can monitor slot usage and manage operations efficiently. By providing real-time updates, seamless booking features, and an easy-to-use interface, the system improves parking management, reduces traffic congestion, and optimizes the utilization of available parking resources.

Keywords: *Parking System, Navigation, Real-Time Availability, Booking System, Flutter, Node.js, MongoDB, Google Maps API*

Introduction
Urban areas face increasing challenges in managing parking spaces due to the rapid growth of vehicles and limited parking facilities, which makes it difficult for drivers to find available parking slots efficiently. In many cities, drivers are forced to search manually for parking, leading to unnecessary delays, increased fuel consumption, and traffic congestion. Traditional parking systems are often not integrated with real-time technologies, making them unreliable and inefficient for both users and parking operators. To address these issues, the Parking Space Detection and Navigation System has been developed as an effective solution that enables users to locate nearby parking spaces and navigate to them using location-based services. The system provides real-time information about parking availability and allows users to book slots either instantly or in advance, improving convenience and reducing search time. It also supports user authentication, parking slot management, and booking history tracking, ensuring a complete parking solution. By integrating modern technologies such as Flutter for frontend development, Node.js and Express for backend processing, MongoDB for data storage, and Google Maps API for navigation, the system ensures better coordination between users and parking space providers, ultimately enhancing parking efficiency and reducing congestion in urban environments. The system also improves transparency by providing accurate slot status to users and reducing dependency on manual monitoring methods. It enables better utilization of available parking resources by minimizing idle or unused spaces. Overall, the system contributes to a more organized and efficient parking infrastructure in modern urban cities.

Existing System:
Most parking systems currently depend on manual checking or basic digital tools to manage parking spaces, where drivers have to search for available slots on their own. In many cases, parking operators manually monitor which spaces are occupied or free, making the process slow and inefficient. Some systems provide limited digital support, but they do not offer real-time updates or accurate information about slot availability. Static indicators or manual records are commonly used, which are not connected to a centralized system, making it difficult for users to view overall parking availability at once. As a result, these systems lack proper integration, accuracy, and efficiency in handling parking management.

Proposed System:
The Parking Space Detection and Navigation System provides a smart and efficient solution for managing parking spaces using real-time technologies. The system allows users to locate nearby parking slots through GPS-based location tracking and navigate to them using integrated map

services. It provides real-time updates on parking availability, helping users quickly identify free slots without manual searching. It provides real-time updates on parking availability, helping users quickly identify free slots without manual searching. The system also enables users to book parking spaces instantly or reserve them in advance, improving convenience and reducing waiting time. In addition, parking space owners can add and manage their slots with details such as location, timing, and charges, which are stored in a centralized database. By combining location services, booking features, and centralized data management, the system enhances efficiency and simplifies the overall parking process.

Related Work

Existing smart parking systems use technologies such as IoT sensors, RFID, and basic image processing techniques to detect parking space occupancy and assist users in finding available slots. According to [1], these systems help improve parking efficiency by providing information about free and occupied spaces. However, they often face challenges in terms of accuracy due to environmental conditions such as lighting variations, vehicle types, and physical obstructions. In addition, these systems depend on hardware sensors, which increases installation and maintenance costs. In large-scale environments, managing multiple sensors can also become difficult. Some systems may also fail to update data properly in real time, which can lead to incorrect availability information. Because of these limitations, the overall efficiency of such systems may reduce in practical situations, especially in busy urban areas.

Modern parking systems are also developed using web-based and cloud-supported architectures to manage parking data and user interaction. These systems allow users to check parking availability and sometimes book parking spaces in advance through mobile or web applications. As discussed in [2], centralized databases help in storing user and parking data securely and support communication between frontend and backend components. However, many of these systems still depend on limited real-time updates, which may cause delays in reflecting the actual parking status. Some systems also do not provide proper navigation support, making it difficult for users to reach the selected parking location. In addition, lack of proper integration between different modules can reduce system performance. These issues highlight the need for a more efficient and well-integrated parking system.

Recent advancements include the use of machine learning models such as object detection algorithms for identifying vehicles and monitoring parking spaces. As mentioned in [3], techniques like YOLO are used to improve detection accuracy and enable monitoring using real-time video data. These systems can provide better results compared to traditional methods. However, they require high computational power, continuous data processing, and complex system setup, which increases overall cost and implementation difficulty. They may also face issues in different environmental conditions such as poor lighting or camera angle variations. Because of these challenges, such systems are not always suitable for simple and scalable applications. Therefore, there is a need for a system that provides real-time updates, easy implementation, and efficient performance, which is addressed in the proposed system.

Design

System Architecture:

The system architecture of the Parking Space Detection and Navigation System is designed as a modular full-stack application that supports efficient parking management through location-based services and controlled booking workflows. The architecture begins with the user, who interacts with the system through a frontend interface developed using Flutter, where the user can perform operations such as registration, login, searching for nearby parking spaces, adding parking slots, and booking parking. The frontend communicates with the backend service through HTTP APIs, which is developed using Node.js and manages core functionalities such as authentication, parking management, booking operations, and request validation. The backend service interacts with the MongoDB database to store and retrieve user information, parking locations, booking records, and slot availability status. The system also includes a car detection module, represented as a separate component in the architecture, which is used to verify vehicle presence during booking confirmation. In the current implementation, this module is integrated as a backend endpoint that returns simulated detection results. The backend processes all incoming requests and updates booking status based on different stages such as pending, arrival confirmation, reservation, and checkout. This architecture ensures proper communication between frontend, backend, detection module, and database, enabling synchronized updates and effective parking management.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

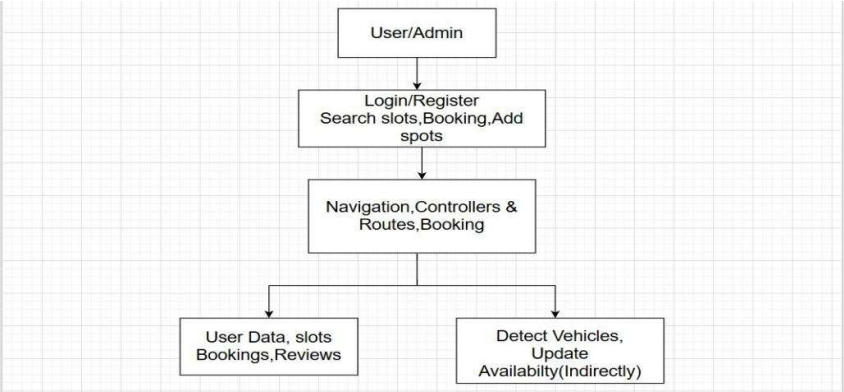


Fig. 1 System Architecture

Technical Architecture:

The technical architecture of the Parking Space Detection and Navigation System defines a modular and scalable infrastructure for implementing a smart parking solution. The frontend is developed using Flutter, which provides a responsive interface for user registration, login, searching nearby parking spaces, adding parking slots, and booking parking. It acts as the primary interaction layer and communicates with the backend through HTTP requests. The backend is developed using Node.js with Express, which handles all core functionalities including authentication, parking management, booking operations, and data processing. The backend interacts with the MongoDB database to store and manage user information, parking locations, booking history, and slot availability data.

The system also includes a car detection module represented as a separate component in the architecture, which is responsible for verifying vehicle presence during booking confirmation. In the current implementation, this module is integrated as a backend endpoint that simulates detection results instead of performing real-time machine learning inference. The backend coordinates communication between the frontend, database, and detection module, ensuring smooth data flow and real-time updates. This architecture enables efficient parking management, reliable data handling, and scalable system performance suitable for real-world applications.

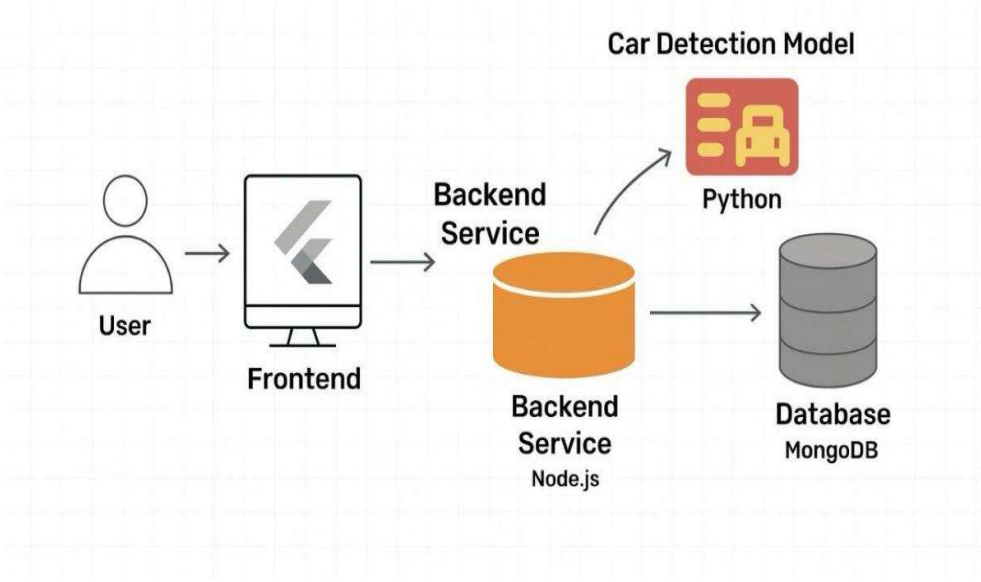


Fig. 2 Technical Architecture

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Methodology:

The Parking Space Detection and Navigation System is centered on developing an efficient and user-friendly solution for managing parking spaces using location-based services and real-time data handling. The process begins with collecting user data through the frontend application developed using Flutter, where users can perform actions such as registration, login, searching for nearby parking spaces, and booking slots. Location data is obtained using GPS services, which is then used to fetch nearby parking locations from the backend system. The backend, developed using Node.js and Express, processes user requests and interacts with the MongoDB database to store and retrieve information related to users, parking slots, and booking history.

The system follows a structured workflow for parking management, including searching, booking, verification, and checkout processes. During booking, the system supports both instant booking and advance reservation, ensuring flexibility for users. A detection module is included in the system to verify vehicle presence during booking confirmation. In the current implementation, this module operates as a simulated detector within the backend, providing predefined responses for validation purposes. The backend updates parking slot availability based on booking actions such as check-in, reservation, and checkout, ensuring accurate and real-time status updates.

The user interface is designed using modular Flutter components, providing smooth navigation for searching parking spaces, viewing details, booking slots, and managing user profiles. Data management is handled through MongoDB, which securely stores user information, parking details, and booking records, ensuring consistency and reliability. This structured and modular approach ensures ease of maintenance, scalability, and efficient performance. By integrating frontend, backend, and database components effectively, the system delivers a reliable and practical solution for modern parking management.

Implementation

The implementation of the Parking Space Detection and Navigation System comprises multiple interconnected software components built using modern web and mobile technologies. Designed with a modular full-stack architecture, the system emphasizes scalability, maintainability, and efficient real-time data handling. The application operates through a Flutter-based frontend, a Node.js backend, and a MongoDB database, enabling seamless interaction between users, parking providers, and system services. Below is a detailed breakdown of the system's core modules and their implementation.

Frontend Implementation:

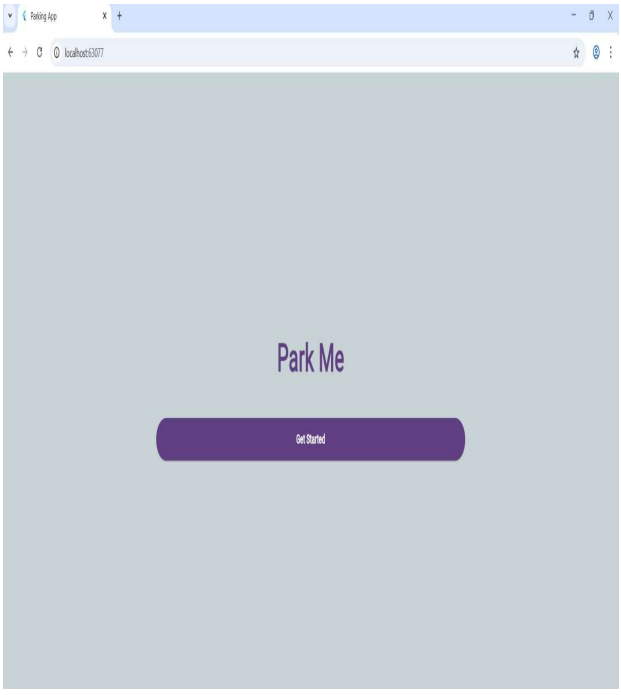
- The frontend of the Parking Space Detection and Navigation System is developed using **Flutter**, providing a responsive and cross-platform user interface with screens for authentication, parking search, booking, and profile management.
- Dynamic UI components are built using Flutter widgets, including login, map view, parking details, booking confirmation, and booking history, ensuring smooth navigation across the application.
- State management is implemented using the **Provider package**, which enables efficient data flow and real-time updates across different screens.
- The system integrates **Google Maps Flutter** for displaying nearby parking locations and uses the **Geolocator package** to fetch real-time user location, while HTTP requests are used for communication with backend APIs.

Backend Architecture and Security Integration

- The backend of the system is developed using **Node.js with Express**, which handles core application logic and API processing.
- The backend provides RESTful APIs for authentication, parking management, booking operations, and user profile handling.
- Authentication is implemented using **JWT (JSON Web Tokens)**, ensuring secure communication between frontend and backend.

- The backend performs validation of user inputs, booking conditions, and availability checks before processing requests.

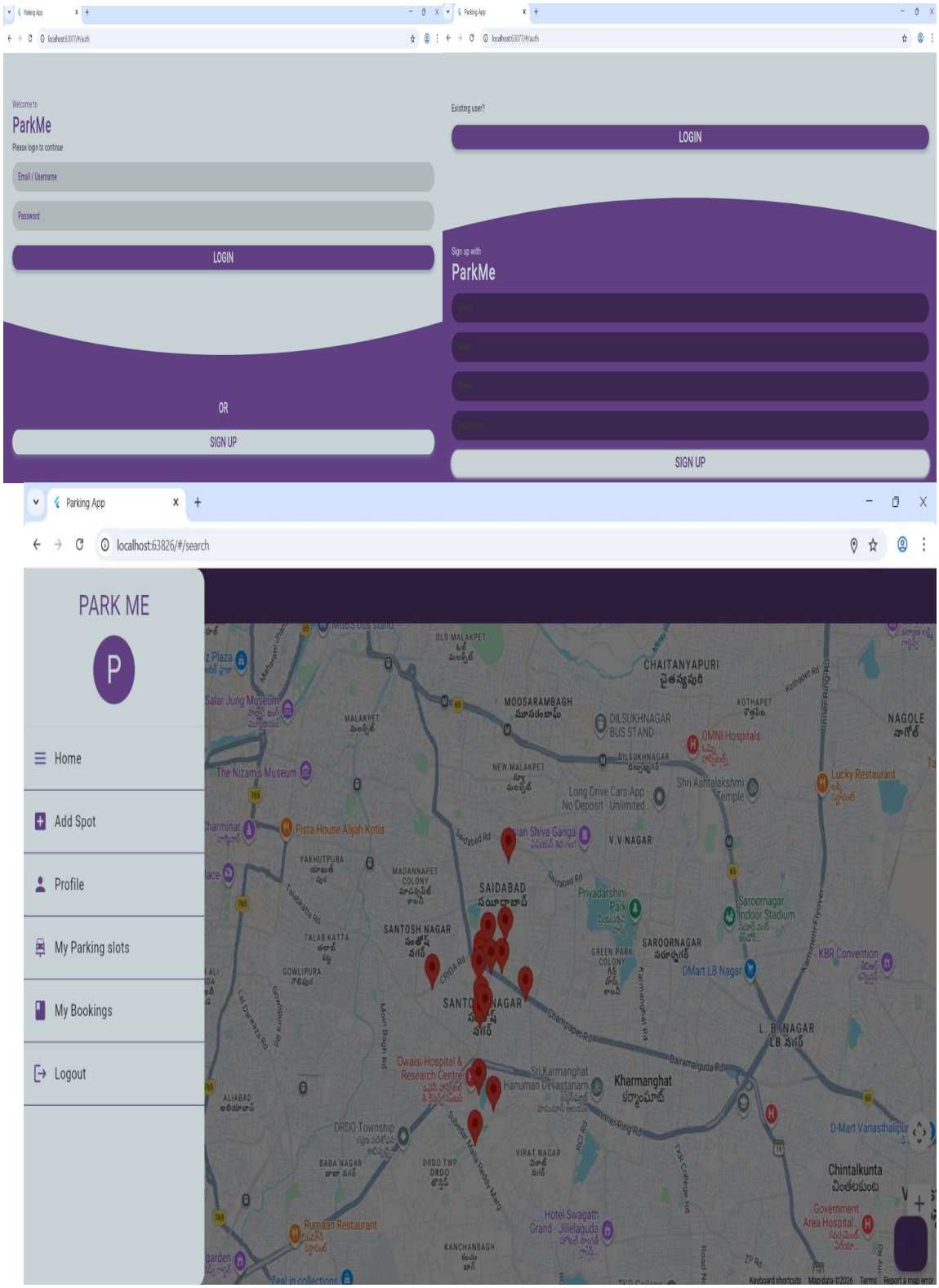
Screenshots



International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

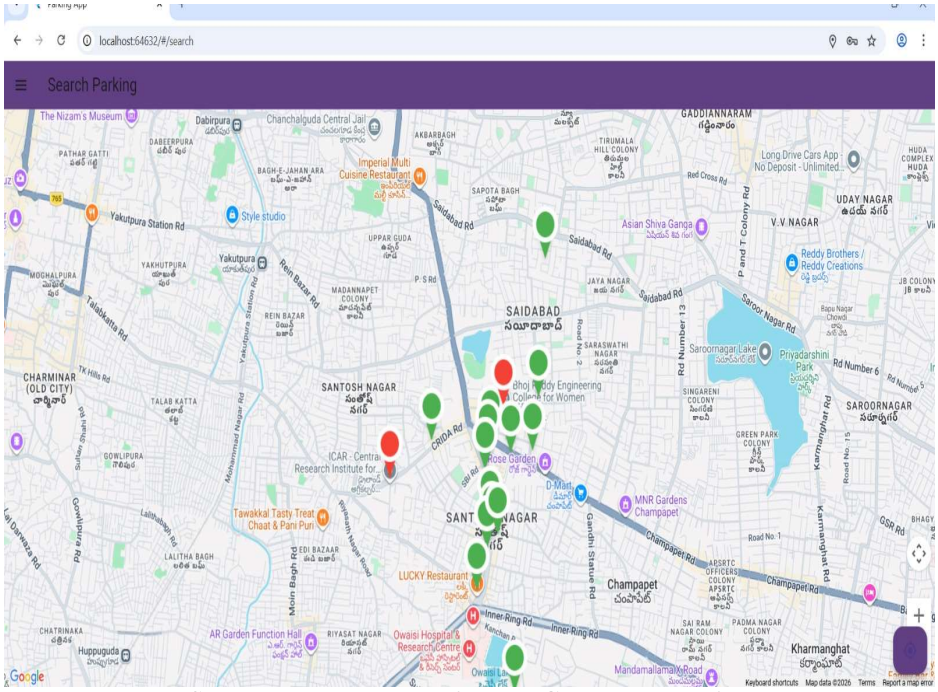


Screenshot From Login to Access – Creating a New Account

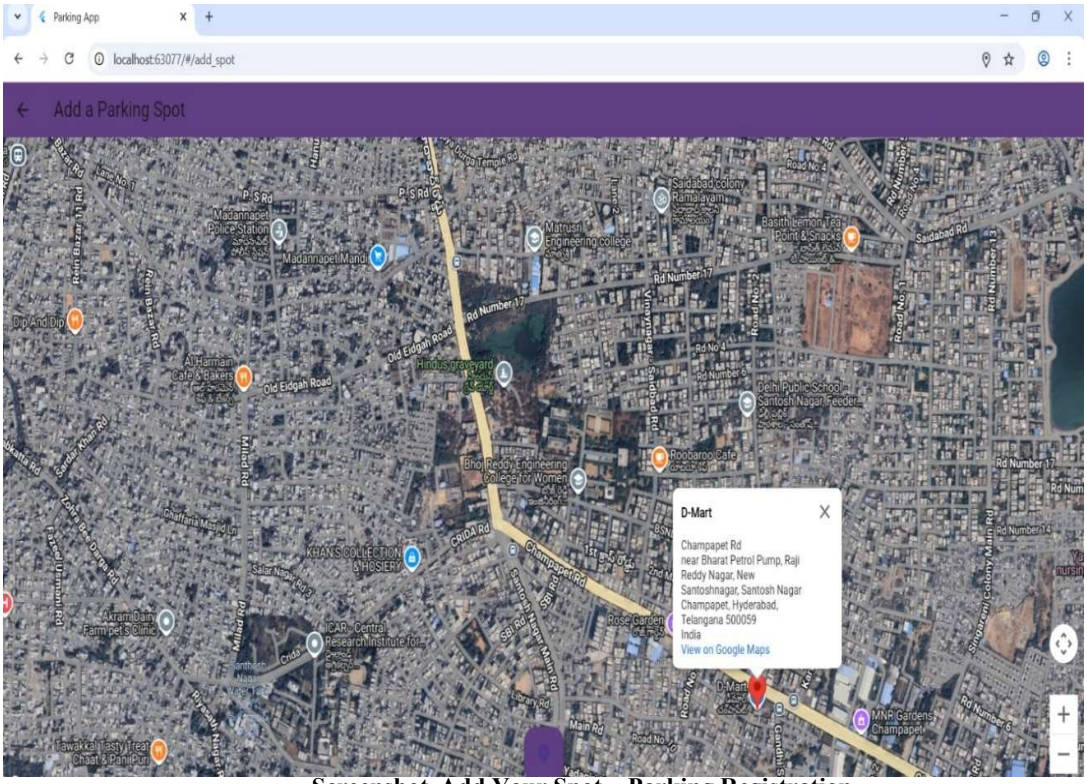
International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026



Screenshot Nearby Parking at a Glance – Map View



Screenshot Add Your Spot – Parking Registration

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

screenshot Car Detector System Demonstration
Test Cases

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Test Cases

Test ID	Test Case Description	Input	Expected Output	Status
TC_01	Valid Login	Correct Email & Password	User logged in successfully, redirected to home	Pass
TC_02	Invalid Login	Incorrect email/password	Error message displayed	Pass
TC_03	Fetch User Location	Allow location permission	Current location retrieved successfully	Pass
TC_04	Search Parking Spots	Valid latitude & longitude	Nearby parking spots displayed on map	Pass
TC_05	Add Parking Spot	Valid parking details	Parking spot added successfully	Pass
TC_06	View parking details	Valid spotID	Parking details displayed	Pass
TC_07	Book Now (Available slot)	isEmpty = true	Booking created with PENDING status	Pass

ISSN 2347–3657

Volume 14, Issue 1, 2026

TC_08	Book Now (Unavailable slot)	isEmpty = false	Booking rejected (slot occupied)	pass
TC_09	Car Detection Success	carPresent = true	Booking confirmed (status = ARRIVED)	pass
TC_10	Car Detection Failure	carPresent = false	Booking cancelled	pass
TC_11	Valid Advance Booking	Future time, no conflict	Booking reserved successfully	pass
TC_12	Invalid Time for Advance Booking	Past time	Error message diaplayed	pass
TC_13	Advance Booking Conflict	Overlapping booking time	Booking rejected	pass
TC_14	Successful Checkout	Valid bookingID	Bill generated, slot becomes available (isEmpty = true)	pass
TC_15	View Booking History	Logged-in user	Booking history displayed	pass

Conclusion

The Parking Space Detection and Navigation System represents an efficient and practical solution for modern parking challenges by integrating location-based services, real-time data management, and user-friendly interfaces. Built using Flutter for the frontend, Node.js for backend processing, and MongoDB for data storage, the system provides a seamless platform for users to search, book, and manage parking spaces. It eliminates the need for manual searching and reduces traffic congestion by providing real-time parking availability and navigation support. The system ensures secure access through authentication mechanisms and maintains accurate booking and parking records for reliability and transparency. Features such as instant booking, advance reservation, and centralized parking management enhance user convenience and system efficiency. The modular and scalable architecture allows easy maintenance and future enhancements, including integration of real-time detection models and advanced analytics. Overall, the system offers a reliable, user-friendly, and scalable solution that addresses urban parking issues and supports the development of smart city infrastructure. In addition, the system improves resource utilization by ensuring optimal use of available parking spaces. It also reduces user effort and time through automated processes and streamlined booking workflows. The integration of modern technologies further strengthens the system's capability to adapt to future advancements in smart transportation systems.

Future Scope

The Parking Space Detection and Navigation System holds significant potential for future enhancements by integrating advanced technologies to improve efficiency, accuracy, and user experience. Future developments may include the integration of artificial intelligence and machine learning models for real-time parking space detection using live camera feeds, enabling more accurate identification of available and occupied slots. The system can be extended with smart payment integration to support secure and seamless digital transactions for parking bookings. Incorporating IoT-based sensors in parking areas can further enhance real-time monitoring and automate slot status updates without manual intervention. The use of cloud-based deployment can ensure better scalability, centralized data management, and accessibility across multiple locations. Advanced analytics and dashboards can be introduced to provide insights into parking usage patterns, peak hours, and revenue generation, helping administrators make better decisions. Integration with smart city infrastructure and navigation systems can improve traffic management and reduce congestion in urban areas. Additionally, support for multi-platform access including mobile, web, and desktop applications can enhance usability and reach. These advancements will make the system more intelligent, scalable, and adaptable to future smart transportation and urban development needs.

References

[1] Sondkar, S., Najare, K., Momin, M., Shevkari, N., & Nadaf, S. (2024). *IoT for Smart Parking System*. 2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA). IEEE. https://doi.org/10.1109/ICAIQSA64000.2024.10882248

[2] Lin, T., Rivano, H., & Le Mouël, F. (2017). *A Survey of Smart Parking Solutions*. IEEE Transactions on Intelligent Transportation Systems, 18(12), 3229–3253.

[3] Yasser, A., et al. (2020). *Smart Parking Systems: A Comprehensive Survey*. Journal of Traffic and Transportation Engineering, 7(6), 789–812.

[4] Flutter Documentation. (2024). *Flutter Official Documentation*. Retrieved from https://docs.flutter.dev

[5] Node.js Documentation. (2024). *Node.js Official Documentation*. Retrieved from https://nodejs.org

[6] MongoDB Documentation. (2024). *MongoDB Manual*. Retrieved from https://www.mongodb.com/docs

[7] Google Maps Platform. (2024). *Maps SDK Documentation*. Retrieved from https://developers.google.com/maps