Volume 11, Issue 4, April – 2026 International Journal of Innovative Science and Research Technology
ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr199

Real Time Parking Availability and Booking System

A. Alagar[1]; R. Mohana Brintha[2]; R. Pavithra[3]; S. R. Navya[4]

[1]Assistant Professor, CSE Dept, K.L.N. College of Engineering, Sivagangai, Tamil Nadu, India
[2,3,4]UG Scholar, CSE Dept, K.L.N. College of Engineering, Sivagangai, Tamil Nadu, India

Publication Date: 2026/04/09

Abstract: The intelligent real-time parking availability and booking system presented in this project is intended to lessen traffic jams and the time lost looking for parking spots. To effectively manage users, parking sites, slots, and bookings, the system combines a web-based frontend with a backend powered by Python, Flask, and MongoDB. Users can reserve spots up to one hour in advance, check parking availability in real time, and get email booking confirmations with QR codes for easy access and departure. Locations, slots, users, and system analytics can all be managed via an admin panel. A machine learning model is also included to forecast parking occupancy and suggest the best times to arrive based on past data and temporal trends. In order to ensure effective slot use, the system also has automated procedures for releasing expired reservations. All things considered, the solution improves user convenience, maximizes parking management, and promotes more intelligent urban transportation.

Keywords: *Real-Time Parking Availability, Parking Slot Booking, Parking Management System, Flask Backend, MongoDB Database, JWT Authentication, QR Code Booking, Slot Allocation, Admin Dashboard, Booking System, Parking Occupancy Prediction, Machine Learning Model, Time-Based Prediction, Slot Availability Tracking, Automated Booking Expiry.*

How to Cite: A. Alagar; R. Mohana Brintha; R. Pavithra; S. R. Navya (2026) Real Time Parking Availability and Booking System. *International Journal of Innovative Science and Research Technology,* 11(4), 126-132. https://doi.org/10.38124/ijisrt/26apr199

I. INTRODUCTION

The difficulties of effective parking management in contemporary cities have increased due to urbanization and the sharp rise in car ownership. Fuel usage, traffic congestion, and environmental pollution all increase when drivers spend a lot of time looking for parking spots. Poor parking resource utilization and user annoyance are the results of traditional parking systems' lack of effective allocation mechanisms and real-time monitoring. By combining automation, intelligent decision-making systems, and real-time data processing, recent developments in smart parking technologies seek to solve these problems [1], [9].

The suggested solution makes use of contemporary online technologies to provide a real-time parking availability and booking platform for effective parking management. Through an easy-to-use interface, users can search available parking spots, make reservations in advance, and easily manage their bookings. Scalability, flexibility, and safe data handling are guaranteed by the backend architecture's use of Flask and MongoDB. Usability and operational efficiency are improved by extra features including email reminders, automated booking expiration, and QR code-based entry and exit. Centralized monitoring and administration of parking infrastructure is further made possible by an administrative dashboard, guaranteeing the best possible use of available resources [4], [10].

The system incorporates machine learning techniques to forecast parking occupancy and suggest the best times to arrive in addition to its fundamental features. Predictive methods like Random Forest and Gradient Boosting reduce ambiguity and enhance user decision-making by examining past parking data and temporal patterns. In addition to improving user experience, this clever strategy helps to improve traffic flow and lessen congestion. Automated background procedures maximize slot availability by ensuring that expired reservations are released on schedule. All things considered, the system offers a scalable and effective smart parking solution by combining real-time monitoring, predictive analytics, and user-centric design [6], [13].

II. METHODOLOGY

Data gathering, storage, and user interaction are the first steps in the system's structured, modular design. A MongoDB database is used to store parking locations, spaces, user information, and booking details, guaranteeing scalability and flexibility. Flask is used to construct the backend, and RESTful APIs manage front-end and database interactions. To provide safe access, JWT is used to implement user

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr199

authentication. Real-time visibility is provided by dynamically retrieving available parking spots and locations when a user enters the system. The system ensures accurate information for users and effective parking management by continuously updating slot statuses, such as available, booked, and occupied.

The booking procedure is made to be both effective and simple to use. Users pick a parking spot, pick a time period that is open, and indicate when they plan to arrive within an hour. The system creates a unique booking record and modifies the slot status after the reservation is confirmed. Every reservation has a unique QR code that is used for entry and exit verification. Users receive email notifications with booking information, improving communication and dependability. In order to avoid resource blockage and guarantee the best possible use of parking spots, automatic background operations also keep an eye on booking expiration times and release unused slots.

A machine learning model is incorporated to forecast parking occupancy and recommend the best times to arrive in order to improve system intelligence. In order to train models like Random Forest or Gradient Boosting, historical data is analyzed to extract features like time, day, and exceptional events. By forecasting occupancy rates for upcoming time periods, the trained model assists users in effectively scheduling their parking. A heuristic method offers backup forecasts in the event that the model is unavailable. The system may provide precise recommendations, lessen traffic, and enhance user experience thanks to this predictive mechanism and real-time data updates.

III. PROCESS FLOW

User engagement and system authentication are the first steps in the process. By entering basic information like name, email address, phone number, and car data, a user can access the web application and register. Following a successful registration, the user uses JWT-based authentication to safely log in and control their session. The technology provides access to essential functions including viewing parking locations and scheduling slots after authentication. Real-time data saved in MongoDB is retrieved by the frontend via RESTful APIs that connect to the Flask backend. Users may engage with the parking system effectively while maintaining data integrity and privacy thanks to this first stage, which creates a safe and customized environment.

Following login, the system shows a list of parking spots that are available as well as the current availability of slots. The backend dynamically calculates the total and available slots by retrieving data from the parking locations and parking slot collections. Usability is improved by allowing users to filter slots according to criteria like floor or slot type. In order to provide correct information, the system updates slot statuses on a regular basis. Before making a reservation, users can make well-informed judgments thanks to this real-time visibility. Users can also access comprehensive details about each site, such as occupancy levels and amenities. By ensuring that consumers are fully informed about parking availability, this step lowers ambiguity and enhances the decision-making process as a whole.

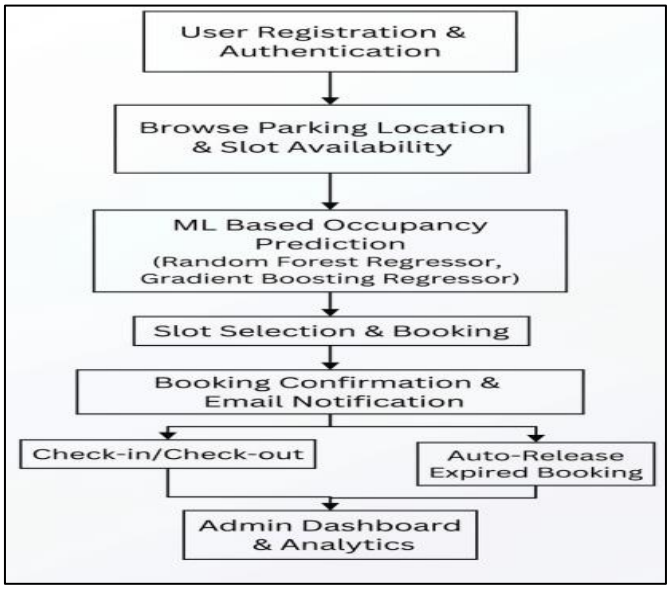

Fig 1 Process Flow

When a user chooses a chosen time slot and indicates an arrival time within the permitted one-hour window, the booking procedure is started. The system verifies the input, verifies the availability of slots, and stops duplicate reservations. After confirmation, the slot status is changed to reflect the reservation and a booking record is produced in the database. The reservation generates a special QR code that serves as a digital pass for entry and departure. Additionally, the method improves communication and dependability by sending a confirmation email with booking details. In addition to preserving data consistency and avoiding slot allocation conflicts, this step guarantees a smooth reservation experience.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr199

The user uses the booking ID or QR code to finish the check-in procedure when they arrive at the parking spot. After confirming the reservation, the system changes its status from pending to active. Concurrently, real-time consumption is shown by the slot state changing to occupied. The check-out procedure is finished when the user exits the parking lot, changing the booking status to completed and releasing the space for further usage. Furthermore, bookings are regularly monitored by an automated background scheduler, which releases slots if individuals don't show up within the allotted time. This guarantees effective use of resources and avoids needless parking slot obstruction.

In order to improve decision-making and maximize parking usage, the system also includes a machine learning module. In order to train models that forecast occupancy rates based on time, day, and unique circumstances, historical parking data is gathered and processed. The technology offers real-time data and predictive insights when users ask for parking availability, indicating the ideal time to arrive. A rule-based heuristic technique guarantees continuity in the event that the trained model is not available. This predictive layer improves planning, lessens traffic, and raises user pleasure. All things considered, the combination of intelligent prediction and real-time processing results in a reliable and effective parking management system.

IV. ALGORITHMS USED

The suggested system effectively manages parking availability and forecasts occupancy by combining machine learning algorithms with rule-based reasoning. Supervised learning methods are used at the center of the predictive module to evaluate past parking data and project future occupancy rates. Features including the timestamp, hour of the day, day of the week, month, holiday indicators, and special event flags are all included in the collection. These characteristics are designed to record behavioral trends and temporal patterns in parking utilization. To improve model performance and consistency, the preparation step involves cleaning the data, converting timestamps into useful properties, and normalizing the data as needed.

Because of its resilience and capacity to manage non-linear relationships in the data, the Random Forest Regressor is one of the most important algorithms used. To increase prediction accuracy and decrease overfitting, it builds several decision trees during training and averages the results. Gradient Boosting Regressor is another model that can be used to improve performance through sequential learning, in which each new model fixes the mistakes of the prior one. The model with the best performance is chosen and saved for deployment after both models are assessed during the training phase using metrics like Mean Absolute Error and R-squared score.

The trained model generates the expected occupancy rate for real-time prediction using input features taken from the current or user-specified time. After that, this forecast is transformed into easily navigable insights, including suggested parking times and expected availability. A heuristic-based approach serves as a backup in situations where the trained model is unavailable. This rule-based method guarantees that the system continues to work even in the absence of the machine learning model by estimating occupancy using predetermined conditions based on weekends, peak hours, and special events.

The system uses effective booking and slot allocation logic in addition to prediction techniques. The system uses database queries to confirm availability when a user tries to reserve a space, and it updates the slot status atomically to avoid conflicts. In addition to enforcing time restrictions like advance booking limits and expiration periods, the booking algorithm guarantees that customers can only reserve one active slot at a time. In order to maintain optimal resource use, a background scheduling algorithm automatically releases the related slots after periodically checking for expired bookings. The system functions dependably, effectively, and intelligently in real-time situations thanks to these coupled algorithms.

V. SYSTEM ARCHITECTURE

The frontend, backend, and database layers make up the system's multi-tier architecture, which guarantees modularity and scalability. The frontend, which offers an interactive user experience for registration, login, parking selection, and booking management, is created using HTML, CSS, and JavaScript. It uses RESTful APIs to interact with the backend. Flask, which serves as the central processing unit for business logic, authentication, and API routing, is used to build the backend. Secure communication and user session management are guaranteed via JWT-based authentication. Flask Blueprints are used to structure a number of modules, including booking, parking, administration, and machine learning, allowing for clear division of responsibilities and simpler maintenance. Effective communication and seamless interaction between system components are guaranteed by this layered architecture.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr199

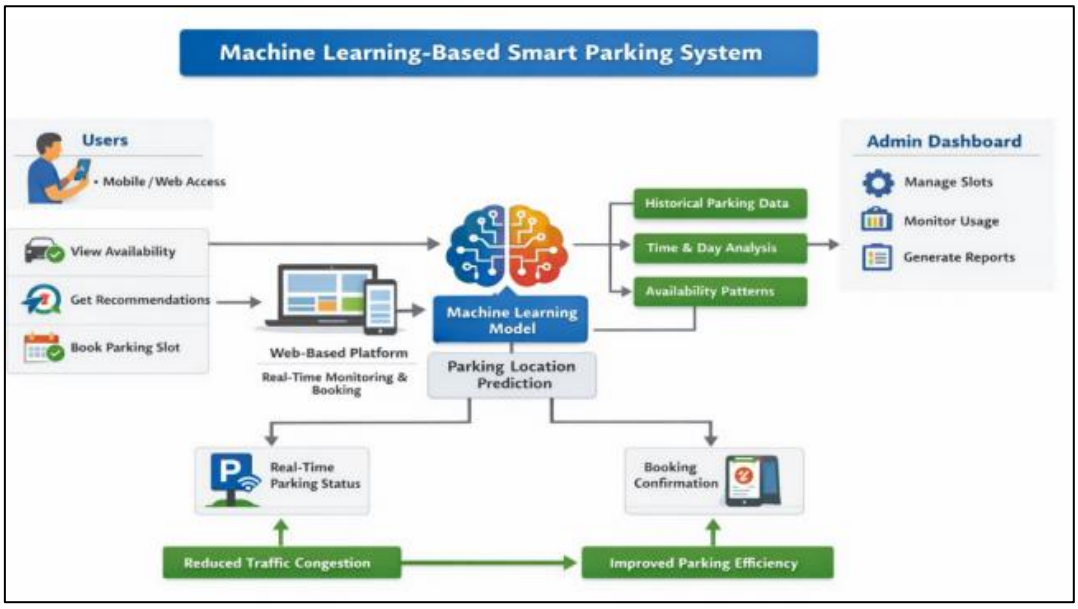

Fig 2 Architecture Diagram

The system's multi-tier architecture, which ensures modularity and scalability, is composed of the frontend, backend, and database layers. HTML, CSS, and JavaScript are used to develop the frontend, which provides an interactive user experience for registration, login, parking selection, and booking administration. It communicates with the backend via RESTful APIs. The backend is constructed using Flask, which acts as the core processing unit for business logic, authentication, and API routing. JWT-based authentication ensures secure communication and user session management. A number of modules, such as booking, parking, administration, and machine learning, are structured using Flask Blueprints to facilitate easier maintenance and a clear division of duties. This layered architecture ensures smooth interaction and efficient communication between system components.

VI. PERFORMANCE ENHANCEMENT

Scalable backend architecture, optimized query handling, and effective database design all improve system speed. To shorten query execution times and boost response times, MongoDB indexing is applied to frequently requested fields including user email, booking status, slot availability, and geographical identifiers. Flask ensures that there is little communication cost between the frontend and backend by using lightweight RESTful APIs. Furthermore, data is processed in an organized way, preventing unnecessary calculations by dynamically determining slot availability only when necessary. In order to keep user-facing procedures quick and responsive, the system further reduces latency by leveraging asynchronous background jobs for non-critical tasks like releasing expiring reservations and sending email notifications.

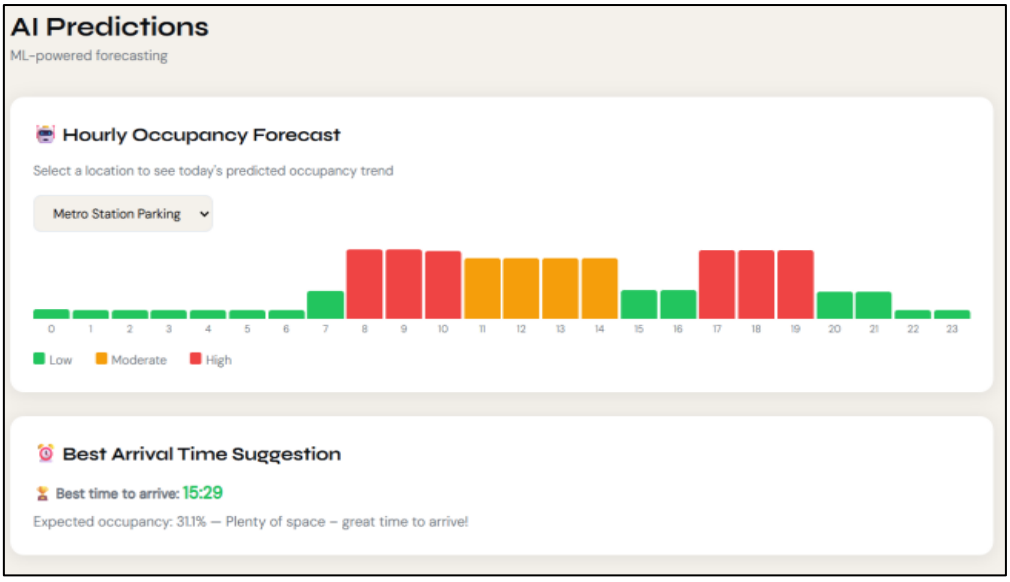

Fig 3 ML Prediction

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr199

Intelligent algorithms and effective resource management enable further advancements. Because the machine learning model is pre-trained and locally stored, it can make fast predictions without having to retrain at runtime. In order to minimize computing complexity without sacrificing forecast accuracy, feature engineering is optimized. By automating recurring operations, a scheduler helps to ensure smooth operation and avoid system overload. Furthermore, appropriate validation and atomic database updates guarantee data consistency in heavy traffic situations and avoid race conditions. Easy scaling is made possible by the modular architecture, which enables the integration of extra servers or services when demand rises. All things considered, these improvements guarantee excellent system dependability, quicker reaction times, and a better user experience even during periods of high activity.

VII. RESULT AND DISCUSSION

The deployment of the real-time parking availability and booking system shows notable advancements in the effective management of parking resources. Through QR-based authentication, the system effectively allows customers to monitor parking availability in real time, reserve spots ahead of time, and finish the booking process with ease. While the admin module efficiently handles locations, slots, and reservations, the integration of secure authentication guarantees the protection of user data. In order to release expired reservations and avoid needless parking space blockage, the automated background scheduler is essential. Because of improved database queries and effective backend processing, performance testing shows that the system can manage several user requests at once with little lag. Overall, the system's goals of lowering manual labor and improving parking service accessibility are met.

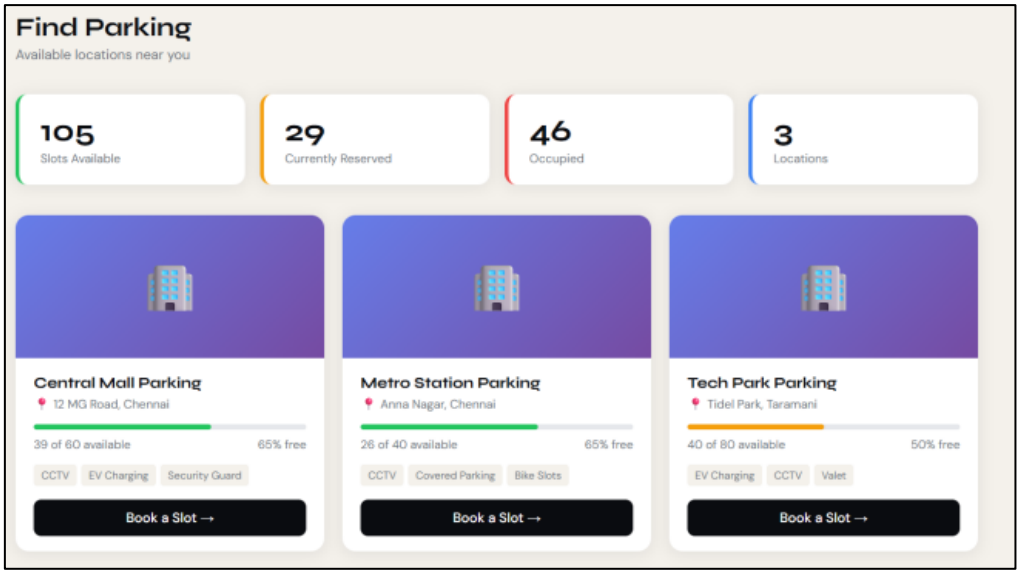

Fig 4 Parking Availability

Based on time-based variables and previous data, the machine learning model offers accurate occupancy projections from a predictive standpoint. For this application, ensemble approaches provide greater accuracy and stability when compared to models like Random Forest and Gradient Boosting. Users can better arrange their arrival times according to the prediction findings, which lessen traffic during peak hours. The heuristic fallback maintains system reliability by guaranteeing continuous operation in situations where the trained model is unavailable. However, the quantity and quality of past data determine how accurate projections are, suggesting room for improvement.Weather and real-time traffic data are examples of extra aspects that can be added to the system to improve it. Overall, the findings demonstrate how well real-time processing and predictive analytics work together to produce a clever and successful parking management system.

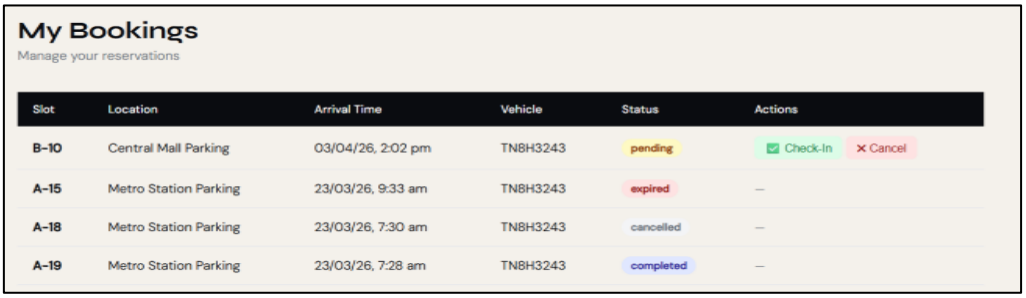

Fig 5 Booking Logs

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr199

VIII. CONCLUSION

The increasing difficulties of parking management in urban areas are effectively addressed by the real-time parking availability and booking system. The system makes it simple for users to find, reserve, and manage parking spaces ahead of time by combining an intuitive web interface with a powerful Flask backend and MongoDB database. Reliability and usability are improved by features like email alerts, automated booking expiration, secure authentication, and QR code-based verification. Additionally, the admin module guarantees effective control of user activities, parking spaces, and slots. All things considered, the system effectively shortens the time spent looking for parking, lessens traffic, and enhances the use of the resources that are available.

Additionally, by forecasting parking occupancy and suggesting the best times to arrive, the incorporation of a machine learning model provides substantial value. Users can avoid moments of high traffic and make well-informed selections thanks to this predictive capabilities. The robustness of the system is maintained by incorporating fallback methods that guarantee uninterrupted functioning even in the event that the model is unavailable. Even while the current implementation performs well, accuracy and accessibility might be significantly improved by adding real-time traffic data, weather information, and support for mobile applications. In summary, the system exhibits a scalable, clever, and useful approach to smart parking, supporting better municipal infrastructure and more effective urban mobility.

IX. FUTURE ENHANCEMENT

By incorporating cutting-edge technologies, the system can be made even more functional and user-friendly. The integration of real-time IoT sensors in parking spaces, which eliminate reliance on manual or system-based status updates and deliver extremely precise and rapid availability updates, is one significant advance. Furthermore, incorporating a mobile application with the web platform would improve user accessibility and convenience. Additionally, GPS and map services can be added to the system to help customers find the closest parking spot through navigation. Additionally, adding an online payment channel would make it easier for customers to finish reservations, increasing the system's viability for practical implementation.

Enhancing the system's intelligence and scalability is another area for improvement. To increase prediction accuracy, the machine learning model can be improved by adding extra variables like weather, traffic patterns, and special events. For dynamic decision-making, real-time data streaming and analytics can be implemented. To effectively manage high demand, the system can also use microservices architecture and cloud deployment. The system can be further improved with features like demand-driven dynamic pricing, loyalty programs for regular users, and sophisticated analytics dashboards for administrators. The application would become a fully intelligent, scalable, and industry-ready parking solution with these enhancements.

REFERENCES

[1]. Elfaki, Abdelrahman Osman, Wassim Messoudi, Anas Bushnag, Shakour Abuzneid, and Tareq Alhmiedat. "A smart real-time parking control and monitoring system." Sensors 23, no. 24 (2023): 9741.

[2]. Govindagoudar, Srushti N., J. J. Jijesh, PK Vineeth Kumar, D. Manju, P. Ramya, and K. M. Manohara. "Smart and Effective Real-Time Management of Car Parking System." In 2025 IEEE North Karnataka Subsection Flagship International Conference (NKCon), pp. 1-6. IEEE, 2025.

[3]. e Jahan, Bazam, Khadija Nadeem, Mehreen Masood, and Dure Sameen Munawar. "AN EFFICIENT AND USER-FRIENDLY ANDROID APPLICATION FOR PARKING SLOT RESERVATION."

[4]. Dharshan, J. Y., Arjun R. Amarnath, Gadamsetty Siva Adithya, and Meena Belwal. "QuickPark: Efficient Parking with Real-Time Pre-Booking." In 2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), pp. 1784-1790. IEEE, 2025.

[5]. Aghakishiyev, Ruslan, and Nazila Rahimova. "ONLINE PARKING RESERVATION." Endless light in science 7, no. март 2 (2024): 134-137.

[6]. Pandimeena, R., V. Saumiya, R. Sarala, and R. Deepalakshmi. "“IntelPark” Website Based Advance Parking Booking System with Number Plate Detection." In 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), pp. 1-7. IEEE, 2024.

[7]. Dhakshinamoorthy, JaiAmudhan, Vishwa Varshini Viswanathan, Yamuneswar Annadurai, and V. P. Dhivya. "Parking Reservation and Smart Allocation using IoT." In 2025 International Conference on Visual Analytics and Data Visualization (ICVADV), pp. 427-434. IEEE, 2025.

[8]. Muthupriya, V., S. Revathi, N. Sabiyath Fatima, I. Karthiga, and Syed Saad Ahmed. "Smart parking system with dynamic pricing using iot." In 2024 8th international conference on i-smac (iot in social, mobile, analytics and cloud)(i-smac), pp. 169-175. IEEE, 2024.

[9]. Aswath, S., N. Kishore, Thinagaran SK, M. Vishwaa, A. Hariharan, and C. S. Vasavi. "Smart car parking system with online reservation." Procedia Computer Science 258 (2025): 2777-2786.

[10]. Kumar, Lingamallu Raghu, Sureddy Sathvika Reddy, Panchalingala Govardhan Reddy, Rayapudi Jagadish, Vittanala Jaya Charan Tej, Neelima Gogineni, and Saurav Dixit. "Advanced parking system." In MATEC Web of Conferences, vol. 392, p. 01085. EDP Sciences, 2024.

[11]. Burns, Aaron, Connor R. Forsythe, Jeremy J. Michalek, and Kate Whitefoot. "Estimating the potential for dynamic parking reservation systems to increase delivery vehicle accommodation." Transportation Research Part A: Policy and Practice 193 (2025): 104380.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr199

[12]. Barbadekar, Ashwinee, Archit Joshi, Arnav Dhobale, and Arya Joshi. "Real-Time Parking System with Navigation." In 2024 International Conference on Electronics, Computing, Communication and Control Technology (ICECCC), pp. 1-7. IEEE, 2024.
[13]. Minh, Quang Tran, Phat Nguyen Huu, Trong Nhan Phan, and Vijayan Sugumaran. "Intelligent Parking Service Integration for Dynamic Urban Areas." The Journal of Engineering 2026, no. 1 (2026): e70168.
[14]. Kulkarni, Jayant, Shravani Tambe, Shrinidhi Deshmane, and Parimal Vasmatkar. "Smart Automated and Real-Time Parking Management System." In 2025 7th International Conference on Inventive Material Science and Applications (ICIMA), pp. 928-932. IEEE, 2025.
[15]. Raji, C. G., Adil Bin Aboobacker, Anfas Muhammad, K. Jamshidha, and Jishana Shemeem. "Android based Integrated Parking System for Real-Time Parking." In 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS), pp. 304-309. IEEE, 2022.