IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Smart Peer-To-Peer Parking Management System Using IOT And Flutter

Prof Nitisha. K. Rajgure [1] , Mr. Varun Ganesh Deshmukh [2] , Mr. Shreyas Rakesh Jantre[3] , Mr. Ajay Dinkar Karagir[4] , Mr. Vighnesh Vinayak Kode[5],[1]Assistant Professor, [2]BE-Student, [3]BE-Student, [4]BE-Student, [5]BE-Student

Computer Department,

Zeal college of engineering & research

Abstract: The fast growth of city populations has made parking problems worse, causing traffic jams, wasted time, and poor use of space. The Smart Peer-to-Peer Parking System solves these problems by combining IOT hardware, a cloud-based backend, and a mobile app that is easy to use to make parking easy. The system uses IR and ultrasonic sensors linked by ESP32 modules to keep an eye on the availability of parking spots in real time. A firebase backend handles reservations, payments, alerts, and navigation data. Users can check the status of their slots in real time, reserve spaces, make secure payments, and get notifications through the mobile app. Also, automated data collection helps parking owners keep track of their spaces, see how many people are using them, and see how much money they are making. The system cuts down on traffic jams in cities, lowers emissions, and makes operations more efficient by encouraging the smart use of underused private parking spaces and supporting dynamic pricing models. This data-driven method not only helps drivers find clear, dependable, and affordable parking, but it also helps landowners and businesses make more money and run their businesses more smoothly. In the end, the suggested solution promotes long-term urban mobility and creates a working relationship between drivers and parking providers.

***Index Terms* -** ESP32, Firebase, Cloud-based, Parking system, Mobile application, IOT, Real-Time Data.

Introduction

The supply and demand for organized parking in cities are significantly out of balance as a result of the sharp rise in car ownership. This situation makes drivers waste fuel and time looking for open spots, which greatly worsens traffic congestion and pollutes the environment. This problem is made worse by the fact that commercial pay-and-park operators struggle to automate occupancy tracking, handle pre-bookings, and analyze revenue trends, and private owners with unused parking spaces frequently lack an appropriate channel to rent them out. We suggest an intelligent parking ecosystem that integrates all stakeholders using a single, real-time system. The foundation of our solution is 1) Firebase, which provides a scalable, real-time backend and authentication services; 2) Flutter, which facilitates the development of cross-platform mobile applications and a consistent user experience; and 3) Internet of Things devices, particularly ESP32 microcontrollers and IR/Ultrasonic sensors, which enable accurate, real-time slot occupancy detection. We suggest an intelligent parking ecosystem that integrates all stakeholders using a single, real-time system. i) Flutter, a cross-platform mobile platform, is the foundation of our solution. ii) Firebase, for a scalable, real-time backend and authentication services; iii) IoT devices, particularly ESP32 microcontrollers and IR/Ultrasonic sensors, for accurate, real-time slot occupancy detection; and iv) application development and a consistent user experience.

1.1 Motivation

Improve the urban parking experience for business owners, private space providers, and car owners. Deal with the issues of disorganized parking, ineffective space use, and lack of real-time availability. Simplify parking access in the area by using a centralized mobile platform. Boost openness and communication between pay-and-park service providers, private owners, and users. Use an IoT-enabled dashboard to automate manual parking operations in order to maximize revenue tracking and monitoring. Assist business owners and private owners in making extra money while maintaining end users' convenience and dependability.

1.2 Problem Statement

Imagine being a car owner in a busy city struggling to find reliable parking. You waste valuable time searching for spaces, face uncertainty about availability, and often end up paying high charges without transparency. Private parking owners, on the other hand, have vacant spaces but lack a simple and secure way to rent them out. Businesses offering pay-and-park services also face challenges in managing operations manually tracking occupancy, handling bookings, and monitoring sales becomes inefficient and error-prone. This leads to poor utilization of parking infrastructure, loss of revenue for owners, and frustration for users. There is a clear need for a centralized system that connects users, private owners, and businesses while leveraging technology to provide real-time availability, automated management, and a seamless parking experience.

I. LITERATURE SURVEY

- In paper **"Scalable Smart Parking System Using Sensors and Flutter Mobile Application"** by S. Aditya and Ketaki Suryawanshi [1]Aditya and Suryawanshi present a scalable smart parking solution employing sensor nodes for slot occupancy, a central server, and a Flutter-based mobile application for user interface, booking, and live updates. The system utilizes modular sensor arrays, RESTful API integration, and cross-platform user deployment, focusing on software maintainability, dynamic updates, and real-time slot allocation. [1] The study does not elaborate on payment system integration, advanced analytics for demand prediction, or monetization strategies for administrators and stakeholders. Leveraging Flutter for cross-platform agile development directly aligns with our objectives for operational efficiency, user convenience, and streamlined mobile access.
- In this paper **"A Mobile Application Monitoring System Using Internet of Things (IOT) And Firebase"** by Nor Farahidah Za'bah and A syrafuddin Mohamad [2] Za'bah and Mohamad introduce a monitoring system utilizing IoT and Firebase for real-time status detection—specifically targeting lecturer presence in university offices using proximity sensors, Raspberry Pi as IoT gateway, and Firebase for cloud storage and user interface delivery. The system architecture is modular and leverages REST APIs for real-time updates accessible via a mobile hybrid app, enabling authenticated viewing and automatic updates of status data every five minutes.[2] The work does not investigate scalability to larger environments, security concerns (like unauthorized access/data breaches), or integration with predictive analytics for trend analysis. This approach informs our project's objectives of enabling real-time monitoring and robust database-driven insights, which are critical for slot occupancy and operational transparency in parking systems.
- The paper **"Architecture Design of an IoT-based Smart Parking System"** by Kratika Bhardwaj et al. [3] proposes an IoT-driven smart parking system model using embedded sensors, IoT gateways, and centralized control for dynamic space allocation and real-time slot status through cloud-based processing. The authors detail hardware designs with sensors and controllers, network protocols, data processing flows, and application-level requirements for efficient user experience and data accuracy.[3] The system's cybersecurity considerations, large-scale data integration challenges, and fallback strategies for hardware failures in dense deployments are not deeply addressed. Technically, this blueprint helps our own system design for maximizing space utilization, intelligent allocation, and achieving seamless system interoperability for operational efficiency.
- In paper **"IoT-Based Smart Parking Management System Using ESP32 Microcontroller"** by Joni Welman Simatupang et al. [4] Simatupang and colleagues describe a smart parking solution with ESP32 microcontrollers, ultrasonic sensors, MQTT protocol for wireless communications, and a webbased front end—allowing real-time reservation, vacancy detection, RGB LED slot indicators, and time-based fee calculation. The experimental setup shows a sensor accuracy of 94.72% and provides full-stack system architecture, including device firmware, communication, and user access modules.[4] This work does not deal with malicious reservation abuse, advanced payment systems, or the integration with predictive analytics for

demand management. It is valuable for our objectives by providing tested methodologies for accurate real-time monitoring, dynamic slot visibility, and automated fee calculation, directly supporting space utilization and operational insights.

- The paper **"The Smart Parking Management System"** by Amira A. Elsonbaty and Mahmoud Shams [5] details an IoTbased smart parking architecture using Arduino, IR sensors for real-time slot detection, NodeMCU Wi-Fi modules, cloud databases, and an Android app interface. It implements realtime parking status updates, booking services, automated payment calculations, and supports user authentication and feedback.[5]There is a lack of discussion concerning scalability for large/city-wide deployments, integration with advanced predictive analytics and revenue insight systems, or resilience against communication/network outages. The work's full-stack approach to sensing, database integration, and useroriented mobile interaction strengthens our project's goals in real-time monitoring, user satisfaction, and providing reliable revenue insights.

II. PROPOSED SYSTEM AND METHODOLOGY

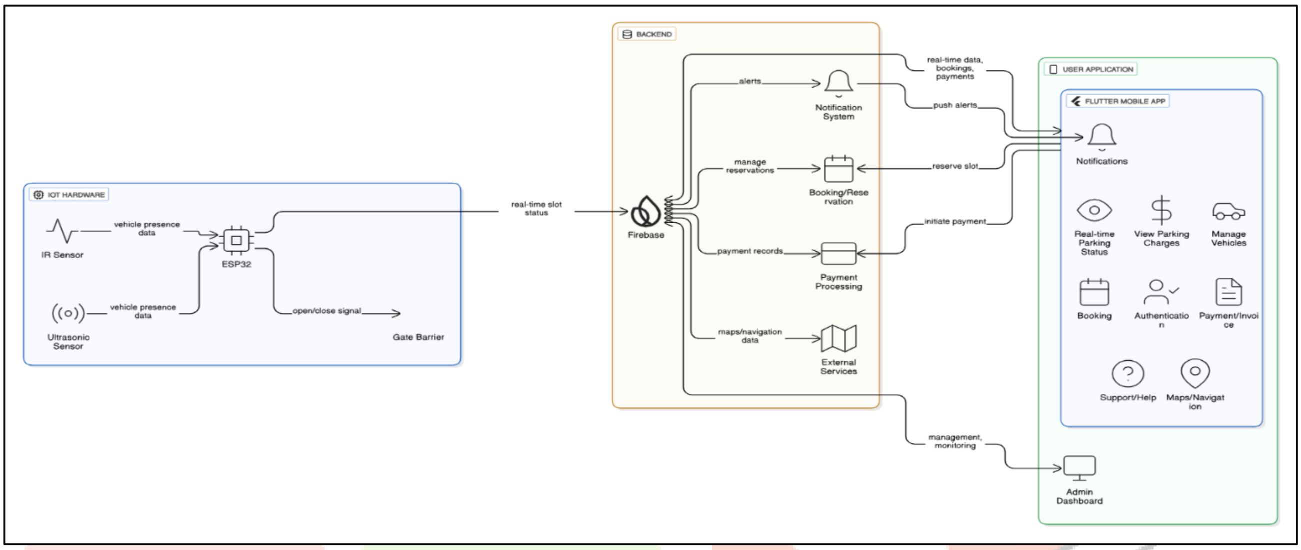

Fig. 1: System Block Diagram

3.1 System Design

The system consists of three major stakeholders:

- **IoT Hardware Block**

This block is responsible for detecting vehicles and controlling the physical barrier:

1. Sensors: An IR Sensor and an Ultrasonic Sensor detect the presence of a vehicle.
2. Controller: An ESP32 microcontroller receives the vehicle presence data from both sensors.
3. Action: The ESP32 sends an open/close signal to the Gate Barrier to control access.
4. Data Out: The ESP32 sends the real-time slot status to the Backend.

- **Backend Block**

The Backend handles the core logic, data storage, and external communications.
Firebase appears to be the primary data store and integration point.

1. Real-time Slot Status: It receives this from the IoT Hardware.
2. Notification System: Sends alerts and push alerts (e.g., parking time expiring) to the User Application based on real-time data.
3. Booking/Reservation: Manages reservations and handles requests to reserve a slot from the User Application.
4. Payment Processing: Handles payment records and initiates payments with the User Application.
5. External Services: Provides maps/navigation data to the User Application and likely relies on external APIs (e.g., Google Maps) for this functionality.

- **User Application Block**

This is what the user interacts with, primarily a Flutter Mobile App, and the system management tool, the Admin Dashboard.

Flutter Mobile App Components:
1. Notifications: Receives alerts and push alerts (e.g., when a reserved spot is available).
2. Real-time/View Parking Status: Displays real-time data on available spots, bookings, and payments.
3. Booking: Allows the user to reserve a slot.
4. Manage Vehicles: Lets users add or remove their vehicle profiles.
5. Authentication/Invoice: Handles user login and displays payment/invoice details.
6. Support/Help & Maps/Navigation: Provides assistance and displays navigation based on data from External Services.

3.2 Hardware Components

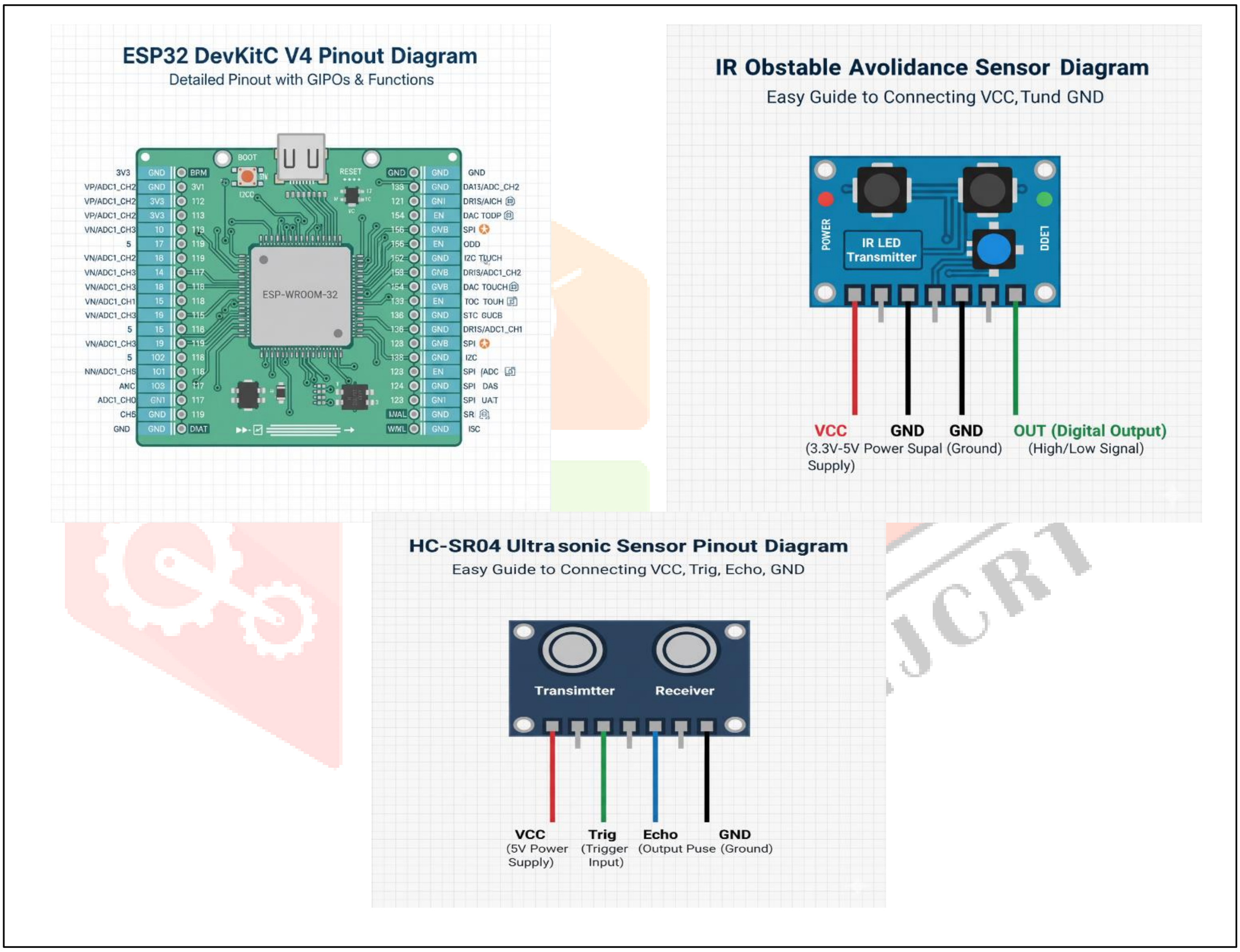

Fig. 2: Main compoents

Here are the descriptions for each component:

- ESP32 DevKit V1

The ESP32 DevKit V1 is a powerful, low-cost microcontroller development board based on the ESP32 chip . It integrates Wi-Fi and Bluetooth connectivity, making it ideal for IoT applications like the smart parking system. It features numerous general-purpose input/output (GPIO) pins, allowing it to interface with various sensors (like the IR and Ultrasonic) and actuators (like the gate barrier). Its dual-core processor and extensive peripheral support enable complex tasks, such as handling sensor data, managing real-time status updates, and communicating with the Firebase backend. It serves as the central control unit in the IoT hardware block.

- Ultrasonic Sensor (HC-SR04)

The Ultrasonic Sensor (HC-SR04) is used for measuring distance based on high-frequency sound waves. It has four pins: VCC, GND, Trigger (Trig), and Echo. The ESP32 sends a signal to the Trig pin to emit a sound pulse, and the sensor measures the time it takes for the echo to return to the Echo pin. In the parking system, it provides vehicle presence data by detecting if an object (a car) is within a certain, very short range, complementing the IR sensor for higher accuracy in determining if a parking spot is occupied.

- IR Sensor (Infrared)

The IR (Infrared) Obstacle Avoidance Sensor is a simple, cost-effective device used to detect the presence of an object within a close, line-of-sight range. It typically has three pins: VCC, GND, and OUT (Digital Output). It works by emitting an IR light beam and checking for a reflection; if the reflection is received, the OUT pin goes low, indicating an obstacle. In the smart parking system, this sensor provides quick, binary vehicle presence data to the ESP32, which is essential for determining the real-time slot status and triggering the gate barrier operation.

3.2 Methodology

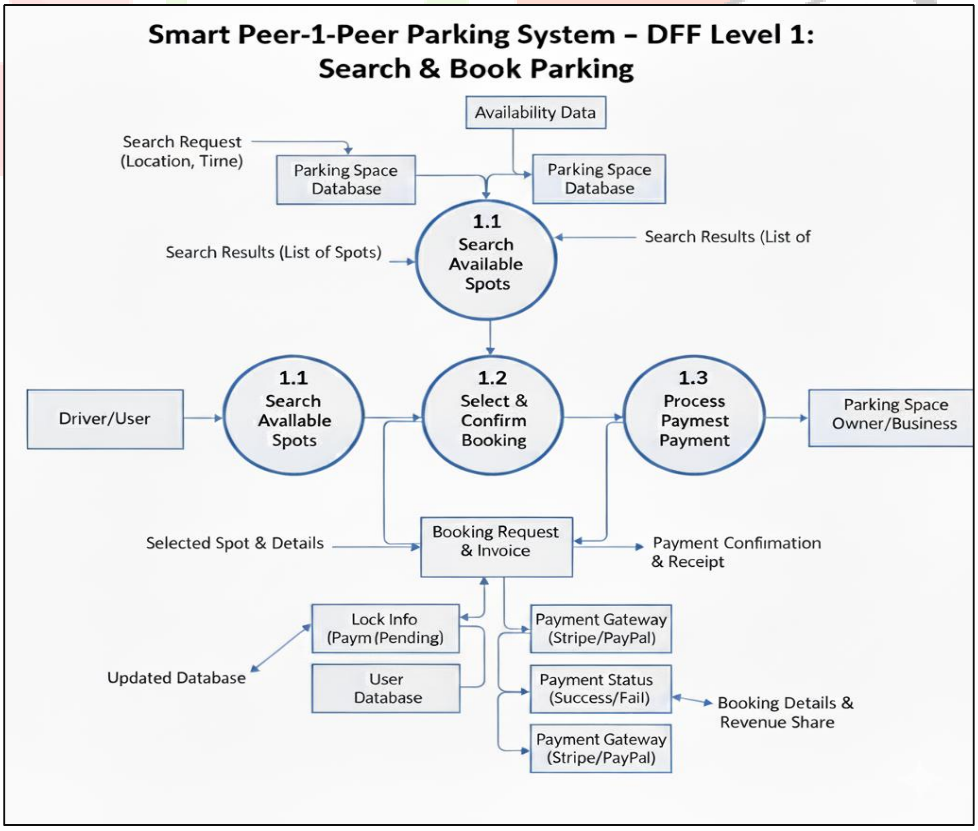

Fig. 3: Data Flow Diagram

Here are the four key points explaining the diagram:

1. Search for Available Spots (Process 1.1): The process begins with a Driver/User making a Search Request (specifying location and time). This request interacts with the Parking Space Database (which holds Availability Data) to generate Search Results (List of Spots) for the user.

2. Selection and Confirmation (Process 1.2): The user receives the search results and submits the Selected Spot & Details to the Select & Confirm Booking process. This generates a Booking Request & Invoice. The process also updates the User Database and holds the spot via a Lock Info (Payment Pending) status, updating the system's Database.

3. Payment Processing (Process 1.3): The system initiates payment by sending the request to a Payment Gateway (Stripe/PayPal). The gateway returns a Payment Status (Success/Fail). Upon success, the gateway sends a Payment Confirmation & Receipt to Process 1.3, which then generates Booking Details & Revenue Share for the Parking Space Owner/Business.

4. Key Data Flows and External Entities: The system relies heavily on the Parking Space Database for spot availability and the User Database for user information. The external entities are the Driver/User, the Parking Space Owner/Business, and the Payment Gateways (Stripe/PayPal), which handle the actual fund transfer.

III. COMPARSION WITH EXISTING SYSTEMS

Table -1: comprehensive analysis

App Name	Search and Reserve	Real time availability	Digital pass/contactless payment	Advance Booking	IoT-based Occupancy Tracking	Other Notable Features
Password Parking	Yes	Yes	Yes	No	No	Rate changes & Notifications
Blinkay	Yes	Yes	Yes	No	No	Navigation options
BestParking	Yes	No	Yes	Yes	No	Saving upto 50%
Hotspot Parking	Yes	Yes	Yes	No	No	Focused on Canada
Proposed System	Yes	Yes	Yes	Maybe	Yes	Unified platform for discovering and pre-booking multiple pay-and-park services

IV. APPLICATIONS

- **Urban and Smart City Parking:** The system facilitates urban and smart city parking by offering drivers real-time discovery and booking of available slots, substantially reducing traffic congestion and the time spent searching for parking.

- **Commercial Complexes and Offices:** For commercial complexes and offices, the platform streamlines parking management, providing digital tools that assign, monitor, and optimize parking spaces for employees and visitors, ensuring organized and efficient usage.

- **Empowering Vehicle Owners:** The mobile application allows users to quickly locate nearby parking spaces, book them in advance, and ensure hassle-free parking in busy urban areas.

- **Optimizing Private Spaces:** Private property owners can list their vacant parking spots on the platform, generating passive income while improving utilization of underused spaces.

- **Streamlining Business Parking Services:** Pay-and-park operators can digitize their services, enabling users to explore availability, pre-book slots, and make secure online payments.

- **Automating Parking Operations:** The IoT-enabled admin dashboard provides real-time monitoring of occupancy, revenue tracking, and space management for business owners.

V. CONCLUSION

The goal of the Smart Parking System is to address the growing parking issues in cities by combining real-time location tracking, IoT-based occupancy detection, and a mobile application into a single platform. The technology makes it simple for users to locate, reserve, and rent parking spots and gives businesses the ability to effectively manage their parking services through the use of Flutter, Firebase, and ESP32 sensors. This solution seeks to enhance urban traffic flow, lessen congestion, and provide new revenue opportunities with features like secure payment processing, analytics dashboards, and a scalable architecture. The project's overall goal is to develop a parking management strategy that is more intelligent, effective, and sustainable.

REFERENCES

[1] Arif md. Sattar, Mritunjay Kumar Ranjan, Shilpi Saxena, "Accelerating Cross-platform Development with flutter framework", stm journals, 2023.

[2] NOR FARAHIDAH ZA'BAH , ASYRAFUDDIN MOHAMAD," A mobile application monitoring system using internet of things (iot) and firebase", Journal of Information Systems and Digital Technologies, Vol.3, No. 1, 2021.

[3] Aqeel Ahlam-Rauf,Subang Jaya,Lillian-Yee-Kiaw Wang, Khee- Siang Bee, Wan-Ru Thang, " Architecture Design of an IoT- based Smart Parkin System", IEEE, 29 January 2024.

[4] Mahalakshmi Bollimuntha, Bhavani Mathi, Navya Tanikonda, Karthika Pokuri, Snigdha Tumu, "Real Time Parking Assistance Using ESP32-MCU", IJNRD, Volume 10, Issue 4 April 2025.

[5] Ms. Ketaki Bhoyar , Mrs. Mohini Avatade , Mrs. S. V. Gavhane , Kedar Bhoyar Asst. Prof, "Smart Parking System Using IOT and Cloud Computing ", IJCRT, Volume 6, Issue 2 April 2018.

[6] Sonam Khedkar1, Swapnil Thube2, "Real Time Databases for Applications", International Research Journal of Engineering and Technology, Volume: 04 Issue: 06,June -2017.

[7] Diya Pandey, Pratham Mistry, Priyanshu Panchal, Sulalah Mirkar,, " Scalable Smart Parking System Using Sensors and Flutter Mobile Application", IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), 09 May 2025.

[8] Amira. A. Elsonbaty and Mahmoud Shams, " The smart parking management system", International Journal of Computer Science & Information Technology (IJCSIT) Vol 12, No 4,August 2020.

[9] Xiaoxia Zheng1 , Wenxi Feng1 , Ning Wang1 , and Huhemandula," IoT-Enhanced Smart Parking Management with IncepDenseMobileNet for Improved Classification", IEEE Access-VOLUME 11, 2023.

[10] Allah Ditta a,, Muhammad Maroof Ahmed a , Tehseen Mazhar bc, , Tariq Shahzad d ,Yazan Alahmed e , Habib Hamam," I Number plate recognition smart parking management system using IoT", sciencedirect Measurement: Sensors 37 (2025) 101409