Smart Parking Solution

A Project Report

Submitted By:

Jintu Nath (210710007019)
Mrinmoy Shyam (210710007028)
Nilakhya Mandita Bordoloi (210710007029)
Parag Choudhury (210710007032)

Department of Computer Science and Engineering
Jorhat Engineering College, Jorhat, Assam

Semester: 7th
Date: December 12, 2024

Contents

Abstract

This project report delves into the comprehensive development and implementation of a Smart Parking Solution that leverages cutting-edge IoT technologies to address parking challenges in urban areas of India. With the ever-growing urbanization and increasing number of vehicles, finding a parking spot has become a major concern, leading to traffic congestion and fuel wastage. Our solution integrates ultrasonic sensors, ESP32, and a web-based server to provide real-time monitoring and efficient management of parking slots.

This system offers users a streamlined interface for booking, locating, and managing parking spaces. With features like Google Maps integration, pre-booking options, wallet functionality, and detailed dashboards for both users and service providers, it stands out as a user-centric and scalable solution. Future enhancements include automated billing, advanced navigation technologies, and mobile application support. By combining affordability, scalability, and innovative features, this project aims to revolutionize parking management in India.

Chapter 1

Problem Statement

The rapid pace of urbanization has led to an exponential increase in the number of vehicles on the roads, making efficient parking management a critical need. Drivers spend an average of 20-30 minutes searching for parking in crowded areas, leading to frustration, increased fuel consumption, and traffic congestion. Moreover, the lack of real-time parking availability data exacerbates the problem, resulting in inefficient use of available parking spaces.

Our objective is to develop a Smart Parking Solution that tackles these issues by utilizing IoT technologies and web-based systems to provide real-time updates on parking slot availability. The solution will enable users to locate, book, and navigate to parking spaces seamlessly, thereby saving time and reducing traffic congestion. By leveraging ESP32 microcontrollers and ultrasonic sensors, this system aims to offer a low-cost, scalable, and efficient solution to urban parking challenges.

Chapter 2

Approach

2.1 System Overview

Our Smart Parking Solution is designed to provide a seamless and efficient parking experience for users while ensuring scalability and reliability. The system architecture integrates multiple components to achieve real-time parking slot monitoring and user interaction.

1. **ESP32 for processing and server communication:** The ESP32 microcontroller processes data from ultrasonic sensors and communicates with the web server to update parking slot statuses.
2. **Ultrasonic sensors for slot detection:** Sensors placed at each parking slot detect the presence or absence of vehicles by measuring the distance to an object. If a vehicle is within 50 cm, the slot is marked as occupied.
3. **Web server backend:** The server acts as the central hub, storing and updating parking data in real-time and providing APIs for frontend communication.
4. **Dashboard visualization:** The user-friendly dashboard displays parking lot layouts, indicating available and occupied slots with visual markers.

2.2 Working Mechanism

The system operates as follows:

- Users can search for nearby parking spots using the "Search for Parking" button. This sends a request to the server for information on available spots.
- Alternatively, users can browse through a list of parking spaces manually.
- The server receives real-time data from ultrasonic sensors connected to ESP32 microcontrollers at each parking spot. Sensors detect whether a slot is occupied based on the proximity of a vehicle.
- Detected data is processed and transmitted to the server, which updates the parking slot status as available or occupied.
- The frontend dashboard, built with ReactJS, displays this information and allows users to navigate to their chosen parking spot using Google Maps integration.

- Additional features include pre-booking, favorite spots, wallet management, and vehicle dashboards for tracking parking history and expenses.

For service providers, a dedicated dashboard offers insights into earnings, statistics, and maintenance requests. The admin panel ensures smooth onboarding and verification of new parking spaces, along with comprehensive system monitoring.

Chapter 3

Tech Stack

The following technologies and tools have been employed in this project:

- **ESP32:** A versatile microcontroller that processes sensor data and connects to the web server for data transmission.
- **Ultrasonic Sensors:** Detects vehicle presence by measuring the distance between the sensor and an object.
- **MySQL:** A robust relational database management system for storing and managing parking data.
- **NodeJS:** Server-side scripting language used for backend logic and API development.
- **ReactJS:** A modern JavaScript library for building the dynamic and interactive web dashboard.
- **Figma:** Tool for wireframing and designing user interfaces and experiences.
- **Postman:** API testing tool used for ensuring seamless communication between hardware and software components.

Chapter 4

Interfaces

4.1 Home Page

The homepage of Parklink serves as the gateway to the platform. It provides basic information about the service and offers options to navigate to either the user signup or service provider signup pages.

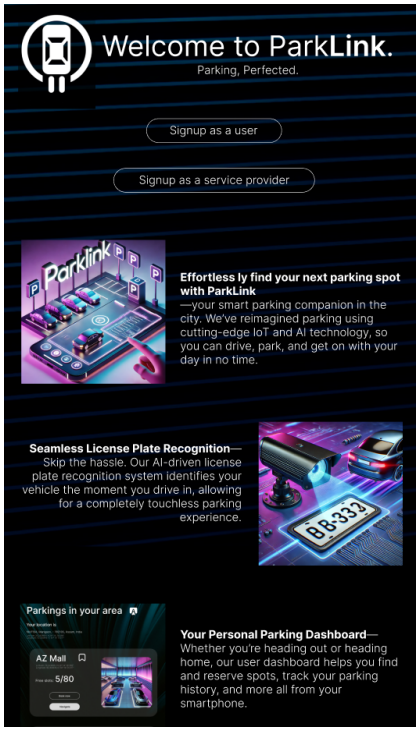

Figure 4.1: Parklink Home Page

4.2 Signup Page for User

This is the signup page where users can register for the service. It collects necessary information to create a user account.

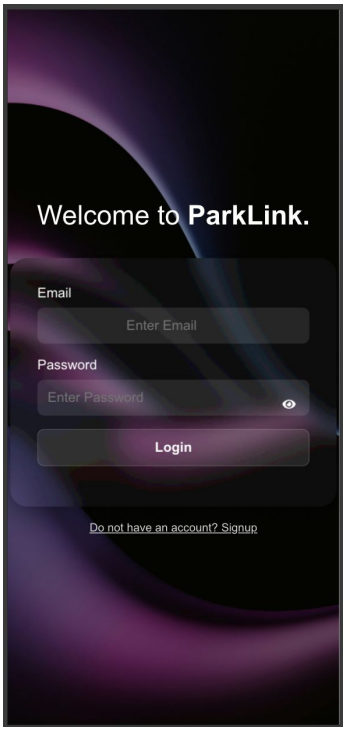

Figure 4.2: User Signup Page

4.3 User Homepage

Once a user logs in, they are directed to their homepage. The buttons on the page are enlarged to ensure easy navigation.

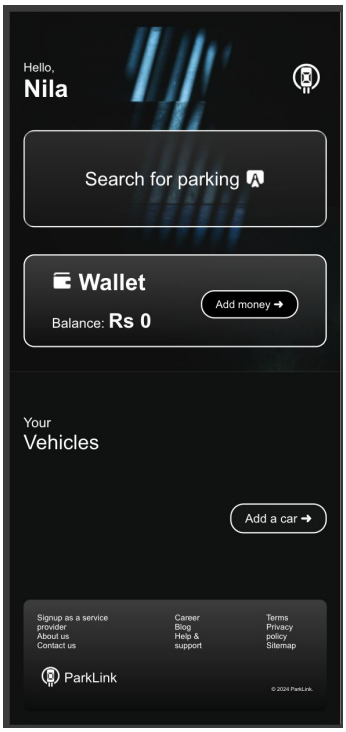

Figure 4.3: User Homepage

4.4 User Homepage After Adding Cars

This shows the updated homepage of the user after they have added their cars to the profile. It reflects the available parking spaces and vehicle details.

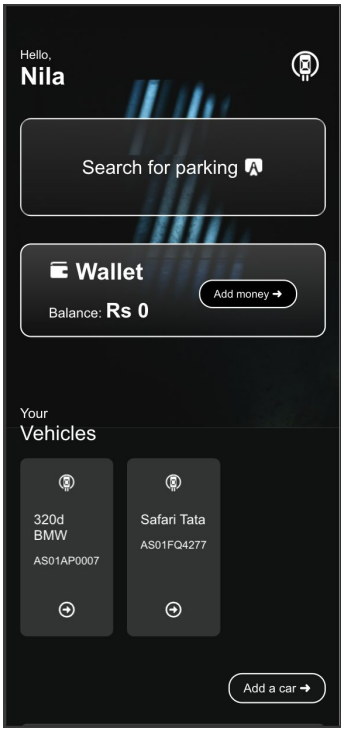

Figure 4.4: User Homepage After Adding Cars

4.5 Available Parking Spots

This page displays all the available parking spots in the vicinity. The user can browse and select from the options available.

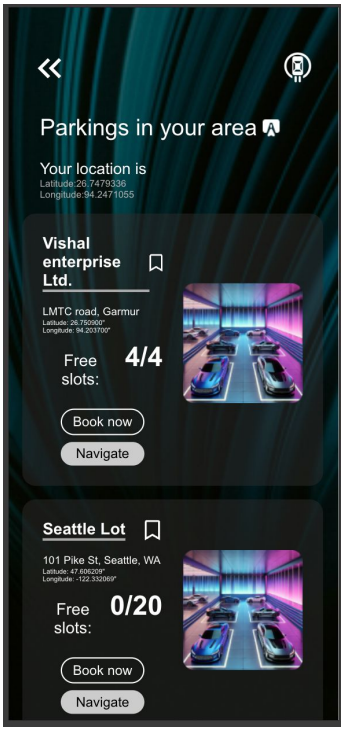

Figure 4.5: Available Parking Spots

4.6 After Booking the Parking Slot

After the user books a parking slot, they are presented with a confirmation screen displaying the details of the booking.

Figure 4.6: Booking Confirmation

4.7 Navigate to the Parking Location

Once the user clicks the navigate button, a new window opens providing the navigation details to the parking location.

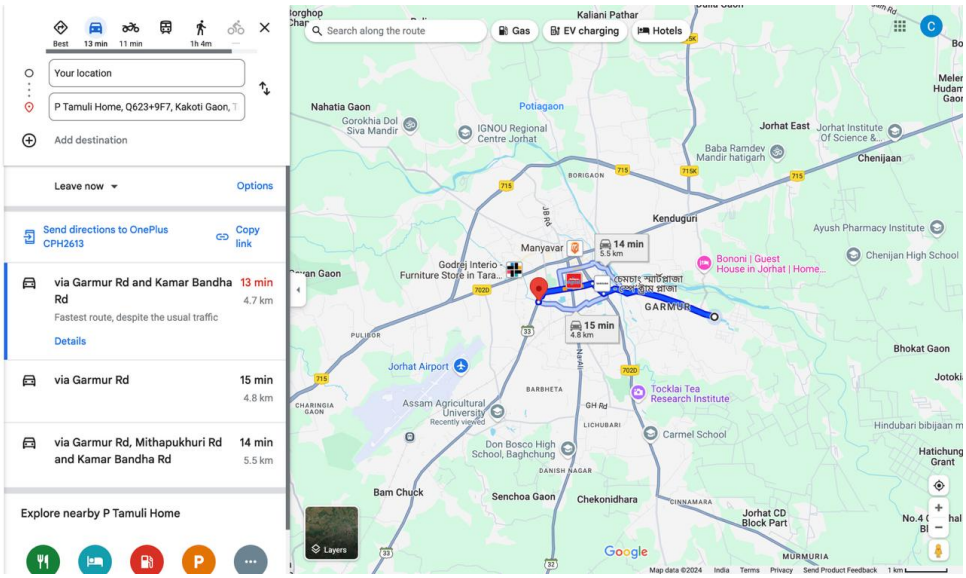

Figure 4.7: Navigation Details

4.8 Add Money Phase

In this phase, users can add money to their account for future bookings. The payment options are shown on this page.

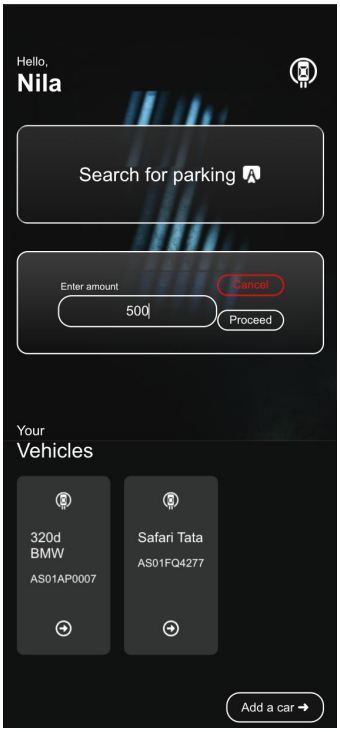

Figure 4.8: Add Money to Account

4.9 Service Provider Homepage

The service provider homepage displays all relevant information needed at a glance. This includes an overview of parking lots, availability, and other important stats.

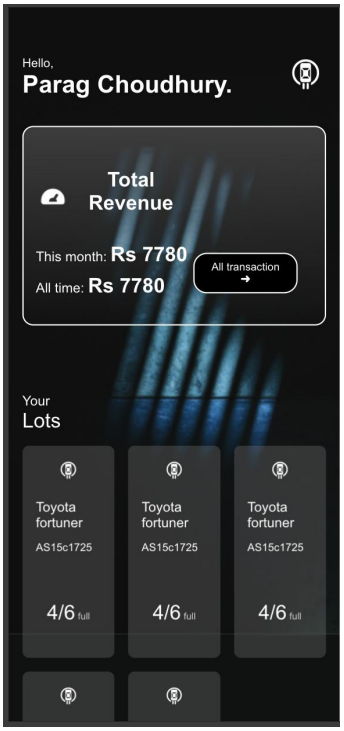

Figure 4.9: Service Provider Homepage

4.10 Add Parking Lots

The service provider can add new parking lots through this page. The details such as location, price, and availability are entered here.

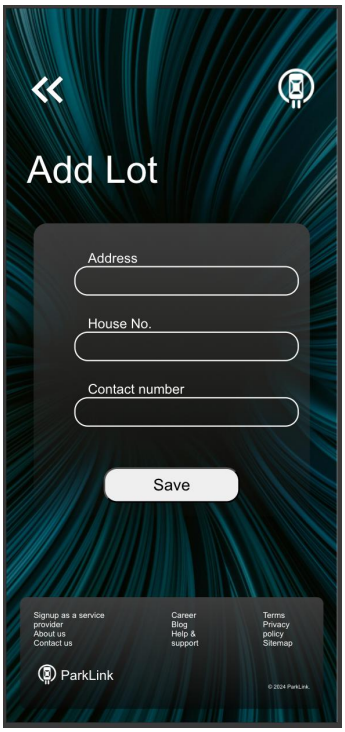

Figure 4.10: Add Parking Lot

4.11 Waiting for Confirmation

After adding the details of the new parking lot, the service provider awaits confirmation from the Parklink admin before the lot can be listed for booking.

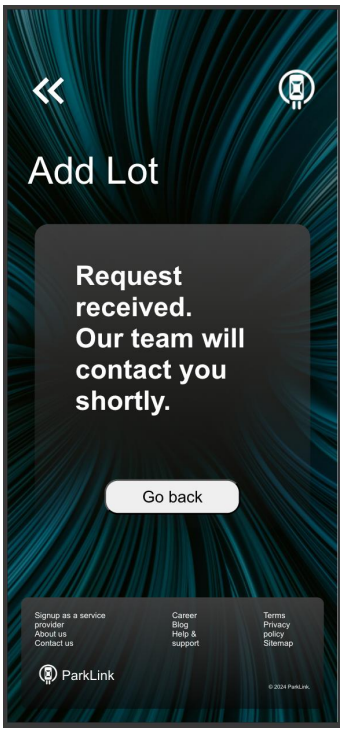

Figure 4.11: Awaiting Confirmation for Added Lot

4.12 System Schema

The system schema provides an overview of how the entire system operates, from user interactions to service provider processes.

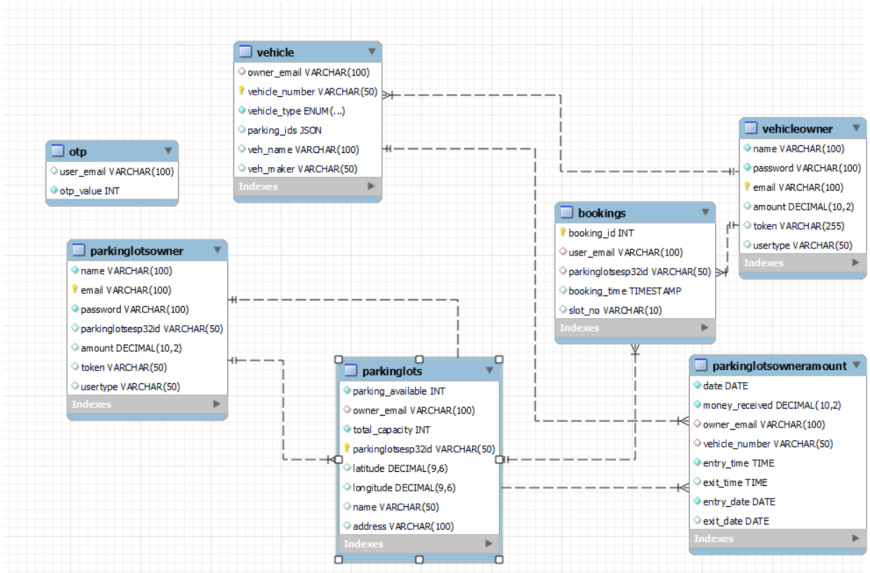

Figure 4.12: System Schema

4.13 Development Phases

The project development was divided into the following phases:

1. **UI/UX Design:** Initial design of user interfaces and branding elements using Figma. The logo and visual theme were also finalized during this phase.
2. **Frontend Development:** Created using ReactJS, the frontend includes user, provider, and admin interfaces to cater to different user roles.
3. **Backend Development:** Implemented using NodeJS for server-side operations and MySQL for data management, ensuring robust and efficient data handling.
4. **Hardware Integration:** ESP32 microcontrollers were configured to interact with ultrasonic sensors and transmit data to the server.

Chapter 5

Features

5.1 Technical Features

- **Scalability:** The system architecture supports the addition of multiple parking lots across diverse locations, with real-time synchronization of data.
- **Flexibility:** Modular design allows easy integration of future features like automatic billing, advanced analytics, and mobile application support.
- **Durability:** ESP32 and ultrasonic sensors are designed to operate reliably in various environmental conditions.
- **Consistency:** Ensures accurate and synchronized data transmission between hardware components and the server.
- **ACID Properties:** MySQL database adheres to atomicity, consistency, isolation, and durability, ensuring reliable data transactions.

5.2 Unique Perspectives

- **Enhanced User Experience:** The intuitive interface, combined with Google Maps navigation and pre-booking options, offers a seamless parking experience.
- **Comprehensive Dashboards:** Tailored dashboards for users, providers, and administrators ensure efficient management and monitoring.
- **Integrated Wallet System:** Simplifies transactions and reduces reliance on external payment systems.
- **Provider Helpline and Maintenance:** Differentiates the solution by offering free helpline and maintenance services to parking providers.
- **Customizability:** Adaptable to both urban and rural settings, making it suitable for diverse market needs.

Chapter 6

Advantages and Disadvantages

6.1 Advantages

- Reduces time spent searching for parking, thereby alleviating traffic congestion.
- Offers real-time monitoring of parking slots for improved efficiency.
- Scalable for implementation in large parking lots or structures.
- Enhances user convenience with features like pre-booking and navigation.
- Enables data-driven decision-making for service providers.

6.2 Disadvantages

- Relies on stable internet connectivity for real-time updates.
- Initial setup costs could be a barrier in cost-sensitive markets.
- Sensor accuracy may be impacted by extreme weather conditions or misalignment.
- Implementation in rural areas may face challenges related to infrastructure and connectivity.

Chapter 7

Future Scope

- **Number Plate Recognition:** Enables automatic vehicle detection using ESP32 CAM and billing.
- **Mobile App Development:** Expands accessibility by enabling slot monitoring and booking via a mobile app.
- **Advanced Analytics:** Offers insights into parking trends, user behavior, and revenue generation.
- **Workforce Management Dashboard:** Develop a dashboard for the Parklink administration to look over operations like approving new parking lots, maintenance of parking lots etc.
- **Fast Tag Integration:** (Further prospect) Streamlines parking fee deduction for vehicles with Fast Tags.
- **GPS-based Fast Tags:** (Further prospect) Introduces advanced tracking and billing capabilities.
- **What3Words Integration:** (Further prospect) Provides precise navigation to specific parking slots.

Chapter 8

Challenges in Indian Context

The following challenges are anticipated:

- **Infrastructure Gaps:** Limited infrastructure in certain cities may hinder deployment.
- **Cost Sensitivity:** High initial setup costs could deter adoption.
- **Connectivity Issues:** Poor internet connectivity in some regions may impact system performance.
- **Privacy Concerns:** Resistance to technologies like number plate recognition due to legal and privacy concerns.

Chapter 9

Conclusion

In conclusion, Parklink offers an innovative IoT-based solution to address the growing urban parking challenges faced by rapidly developing countries, such as India. With the increasing number of vehicles and the scarcity of parking spaces, the system provides an effective way to optimize parking management. By offering real-time parking availability updates and automating processes, Parklink helps reduce the time spent searching for parking, leading to lower fuel consumption, time savings, and reduced traffic congestion.

The use of IoT-enabled sensors that detect parking spot availability and transmit data to a central server ensures seamless communication with users via mobile or web applications. This not only helps drivers find parking spots quickly but also simplifies the payment process through automation, eliminating the need for manual billing.

Parklink is designed with scalability in mind, making it adaptable to different environments—from small parking lots to large, multi-level facilities. Its cloud-based infrastructure allows for real-time data synchronization and provides the flexibility to accommodate future urban growth. Additionally, the system's use of low-maintenance, durable sensors helps minimize operational costs, while maximizing parking space utilization.

The environmental benefits of Parklink are significant, as it reduces emissions by preventing vehicles from idling while searching for parking. The overall improvement in traffic flow and air quality, along with the promotion of sustainable urban mobility, underscores the system's broader positive impact.

Despite some challenges, such as infrastructure limitations, initial setup costs, and the transition from legacy systems, the long-term advantages of Parklink are clear. As IoT technologies become more accessible, the system's ability to streamline urban parking will become increasingly crucial. In the future, features like automated mobile payments, GPS-based navigation to parking spots, and app-based reservations will make the platform even more user-friendly.

Parklink's integration of IoT with parking management marks a transformative step towards creating smarter, more sustainable cities. By alleviating parking shortages, reducing congestion, and improving urban mobility, Parklink is well-positioned to help cities adapt to the demands of an increasingly urbanized world.

Chapter 10

References

- ESP32 Documentation: https://www.espressif.com/en/products/socs/esp32
- Ultrasonic Sensors: https://www.maxbotix.com/
- PHP Documentation: https://www.php.net/manual/en/
- MySQL Documentation: https://dev.mysql.com/doc/
- ReactJS Official Documentation: https://react.dev/
- Google Maps API: https://developers.google.com/maps/documentation/
- Figma Resources: https://www.figma.com/resources/
- Postman API Testing: https://www.postman.com/
- IoT-based Parking Management Research Papers and Articles

Chapter 11

Appendix

11.1 Github Repository Link

Figure 11.1: https://github.com/nilaMan16/Parklink

11.2 Test Cases and Results

Comprehensive testing ensured the system's functionality across various scenarios. Below are examples:

Test Case	Expected Outcome	Result
Ultrasonic Detection	Correctly identifies distance	Passed
API Response	Returns available slots within 2 seconds	Passed
Dashboard Display	Real-time updates of slot status	Passed
Google Maps Navigation	Accurate navigation to slot	Passed