REVA UNIVERSITY
Bengaluru, India
Established as per the section 2(f) of the UGC Act, 1956,
Approved by AICTE, New Delhi

School of Computing Science and Engineering

Research Based Mini Project (Code: B20EF0603)

Course Code: B20EF0603	**Mini Project Progress Report – Phase 1**	**Year: 2023-2024** **Sem: 6th**

Project Group Members Details

Group No:	**Group Name:**

Sl. No	SRN	Full Name	Sec	Mobile No.	Email_ID	Sign
1	R21EF100	C. Sruthi	B	8074117258	2107300@reva.edu.in	
2	R21EF110	K. Deva Harshini	B	8790631741	2106974@reva.edu.in	
3	R21EF111	K. Tejaswini	B	9108637868	2104519@reva.edu.in	
4	R21EF102	C. Dinesh	B	9390040359	2106473@reva.edu.in	

Project Details

Project Title:	**Smart Parking System**	
Guide Details:	**Name:** Geetha B **Designation:** Assistant Professor	**Mobile No:** 7019231762
Remarks by Guide:		**Guide Signature** **Date:**

Abstract:

The Internet of things (IoT) describes devices with sensors, processing ability, software and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks. In the upcoming years, IoT-based technology will offer advanced levels of services and practically change the way people lead their daily lives. IOT is a system of interrelated things, computing devices, mechanical and digital machines, objects, animals, or people that are provided with unique identifiers. The Internet of things encompasses electronics, communication, and computer science engineering. And the ability to transfer data over a network requiring human-to-human or human-to-computer interaction.

Existing solutions for this project include using different types of sensors. The moto of different ideas is to just make a parking slot available using sensors and storing that data in a database. This idea can be useful to detect an empty parking slot and book it in prior to placing the vehicle. But the drawback for this is that when a person books a slot before entering, this leads to the unavailability of parking place for the vehicles which arrive at that booked time. This makes the wastage of that slot for a very long time. Because a person can book a slot at any time even if he is arriving 10 hours before that. There will be ambiguity in parking slots. Parking slots may not be available if many vehicles arrive one by one continuously. And there is no automatic message generation of where the vehicle is placed. This happens a very number of times where the person forgets his parking place. So, there is a drawback of generating a notification of the parking place lane number. These ideas don't have the automatic deduction of the amount for parking which involves manual work. These are the existing solutions and their drawbacks which cause lags in the usage of the proposed system.

The proposed Intelligent Parking System (IPS) consists of an Iot framework that collects real-time data, sends it to the cloud, and thereby suggests to the user a suitable place for parking the vehicle at a nearby location. Smart parking in IoT (internet of things) is a system integrated with a sensor that sends parking data via web applications about free and occupied parking places. This process involves the following data path: The microcontroller transfers data to the cloud. Parking data is read by Ultrasonic Sensors & Arduino. IR sensor is effective for indoor parking Iot, and sonar sensor can be used for indoor and outdoor application. After reading and processing the data Arduino sends the number of engaged and free slot of the parking to Raspberry Pi using serial port. The system should support multiple types of parking spots such as Compact, Large, Disabled, Motorcycle, etc.-The system should support parking of different vehicles like car, truck, van, motorcycle, etc.- Each parking floor should have a display board showing any free parking spot for each spot type. Another approach is to give the empty space pillar number to the driver at the ticket collection by avoiding the confusion by seeing on the boards. And that place is marked as full after allotting and marking as empty after clearance. The future work will be developing fastags for automatic money deduction to reduce the manual work and time. And then at the parking slot if any person had a cardiac arrest, he/she will be having a smart watch can detect the heartbeat rate and send the immediate notification to the ambulance for quicker response. These are the ideas which enhance the project scope, reduce time and manual work.

Table of Contents

List of Tables

Table 1: Sensor Data Format

Field	**Description**	**Example**
sensor_id	Unique identifier for the sensor	S001
location	Physical location of the sensor	Parking Lot A, Slot 1
Status	Availability status of the parking slot	available or occupied
distance	Distance measured by the sensor	150
timestamp	Date and time of data recording	2024-02-25T12:30:45

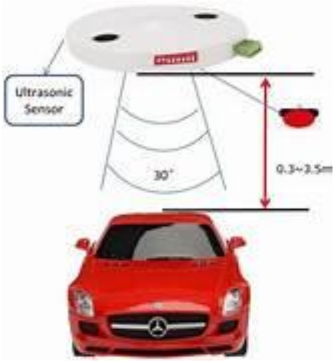
Table 2: Firebase Database Schema

Collection	**Document Id**	**Fields**
Parking_slots	S001	- location: "Parking Lot A, Slot 1"
		- status: "available" or "occupied"
		- sensor_id: "S001"
		- distance: 150
		- timestamp: "2024-02-25T12:30:45"
Users	UID123	- username: "john"
		- email: "john@example.com"
		- parking_history: ["S001", "S002"]

Table 3: Mobile App Features

Feature	**Description**
User Authentication	Secure login and registration for users
Map Integration	Display of parking lot layout with available slots
Real-time Updates	Instant updates on parking slot status
Slot Reservation	Allow users to reserve a parking slot
Parking History	Record and display user's parking history
Notification	Push notifications for slot availability and reservation
Profile Management	Edit user profile, view reservation history
Feedback/Support	Provide feedback or seek support within the app

List of Figures



Data flow diagram:

Ultrasonic Sensor

|
|

Microcontroller
Unit (MCU)

|
|

Firebase
Real-time DB

|
|

Mobile App
User Interface

List of Symbols, Abbreviations and Nomenclature

IoT: Internet of Things
MCU: Microcontroller Unit
API: Application Programming Interface
UID: User Identification
GPS: Global Positioning System

I. Introduction:

In urban environments, the increasing number of vehicles poses a significant challenge in managing parking spaces efficiently. Finding an available parking slot can be time-consuming and frustrating for drivers, leading to traffic congestion and environmental concerns. To address these issues, the implementation of a Smart Parking System has become imperative.

The Smart Parking System leverages cutting-edge technologies to streamline the parking process. One crucial component of this system is the use of ultrasonic sensors to monitor the availability of parking slots in real-time. These sensors detect the presence of vehicles, providing accurate and instantaneous data on parking occupancy.

The integration of Firebase as the backend solution enhances the system's capabilities by offering a robust cloud-based database. Firebase facilitates seamless communication between the ultrasonic sensors, the mobile app, and the server, ensuring the availability of up-to-date information. Real-time data synchronization enables users to access accurate parking slot status instantaneously.

The motivation behind developing this Smart Parking System lies in its potential to revolutionize urban mobility. By leveraging ultrasonic sensors and a cloud-based infrastructure, the system optimizes parking space utilization and enhances the overall parking experience for users. Reduced traffic congestion, decreased environmental impact, and improved user satisfaction are among the anticipated benefits of this innovative solution.

Front-End and Back-End Technologies

Front-End: Mobile App

The mobile app's user interface is developed using modern frameworks such as React Native or Flutter for a cross-platform experience.

User interactions, real-time updates, and intuitive design contribute to a seamless user experience.

Back-End: Firebase

Firebase serves as the backend solution, offering a real-time NoSQL database to store parking slot data.

Authentication and authorization services provided by Firebase ensure secure access to user-specific information.

Firebase Cloud Functions can be employed to handle server-side logic and automate tasks, enhancing the system's efficiency.

The integration of ultrasonic sensors, Firebase, and a user-friendly mobile app aims to create a Smart Parking System that not only addresses the challenges of urban parking but also sets the stage for a more connected and efficient urban transportation ecosystem.

I. Literature Survey:

Brief overview of the importance of smart parking systems in urban environments.
Introduction to the integration of ultrasonic sensors, Firebase, and mobile apps for enhanced functionality. Summary of key research highlighting the role of smart parking systems in optimizing parking space utilization and reducing traffic congestion. Recap of studies showcasing the effectiveness of ultrasonic sensors in real-time data collection and their application in detecting vehicle presence. Synthesis of literature discussing the advantages of cloud platforms, with a focus on Firebase, for storage, communication, and real-time data processing in smart parking applications. Summary of research emphasizing the importance of user-friendly mobile apps, including features like real-time updates, reservation systems, and navigation functionalities. Re-organization of information on the benefits and capabilities of Firebase as a real-time database and its role in enhancing the efficiency of smart parking systems. A new interpretation of literature exploring the broader context of the Internet of Things in urban mobility and how interconnected devices contribute to intelligent transportation ecosystems. Evaluation of studies focusing on user experience factors and how they shape the acceptance of smart parking applications. Synthesis of key findings and recommendations based on the literature review, highlighting the significance of integrating ultrasonic sensors, Firebase, and mobile apps in smart parking systems.

II. Objectives:

The primary objective of this project is to design and implement a Smart Parking System that utilizes ultrasonic sensors for real-time detection of parking slot occupancy. The system will be integrated with Firebase as the backend solution and a user-friendly mobile app to provide accurate and timely information about available and occupied parking slots.

Sensor Integration:
Integrate ultrasonic sensors into the parking infrastructure to accurately detect the presence of vehicles in each parking slot.
Real-time Data Collection:
Establish a seamless communication mechanism between the ultrasonic sensors and the Firebase database to ensure real-time data collection on parking slot occupancy.
Firebase Integration:
Implement Firebase as the backend solution for storing and managing sensor data, ensuring data consistency, and providing scalable and synchronized access.
Mobile App Development:
Develop a user-friendly mobile app that interfaces with the Firebase database to display real-time information about available and occupied parking slots.
User Interaction and Reservation:

Implement user-friendly features in the mobile app for intuitive user interaction, allowing users to view parking slot status and reserve available slots.

Real-time Updates:

Enable real-time updates in the mobile app, ensuring users receive instantaneous information about changes in parking slot status.

Security and Authentication:

Implement secure authentication mechanisms within the mobile app, ensuring that users have authenticated access to the system and their data.

Testing and Validation:

Conduct comprehensive testing, including unit testing, integration testing, and system testing, to validate the functionality and reliability of the entire system.

Performance Evaluation:

Evaluate the performance of the Smart Parking System in terms of response time, data accuracy, and overall user experience.

Documentation and User Manuals:

Provide thorough documentation, including code documentation and user manuals, to facilitate system maintenance and user understanding.

III. Methodology:

The methodology for implementing the Smart Parking System begins with a thorough analysis of system requirements, encompassing parking slot details, sensor placement, and desired mobile app functionalities. Ultrasonic sensors are strategically integrated into parking slots, interfacing with a chosen microcontroller unit (MCU) that processes real-time sensor data and sends control signals. A Firebase project is set up to serve as the backend, providing a real-time database with secure authentication. The integration phase involves establishing a seamless connection between the MCU and Firebase to ensure up-to-date parking slot information. For the mobile app, a cross-platform framework is selected, and user-friendly features such as real-time updates and a reservation system are developed. User authentication is implemented using Firebase Authentication. Rigorous testing is conducted, encompassing unit testing, integration testing, and system testing to validate the functionality and performance of the entire system. Comprehensive documentation is prepared, including system architecture, code documentation, and user manuals. The deployment phase involves real-world implementation, and feedback from users is collected for iterative improvements. This methodology ensures a systematic and comprehensive approach to realizing a reliable and efficient Smart Parking System.

Block Diagram:

```
 Ultrasonic                    Microcontroller                 Mobile App
   Sensors                       Unit (MCU)

       |                            |
       |                            |
       | Distance Data              | Control Signals
       |                            |

  Sensor Data      ----------->  Data Processing     -----------> User Interface
  Collection                       & Decision                      & Interaction
                                  Algorithm

       |                             |
       |                             | Real-time Updates
       |                             |
       |                             |

  Firebase                        Mobile App                     User Database
  Database                     Integration & Sync
```

IV. Modules identified:

The Smart Parking System implementation is divided into several key modules:

Sensor Module: Collects real-time occupancy data using ultrasonic sensors.
MCU Processing Module: Processes sensor data, makes decisions, and controls the system.
Firebase Integration Module: Establishes real-time synchronization between MCU and Firebase.
Mobile App Interface Module: Develops a user-friendly interface for real-time parking updates.
User Authentication Module: Ensures secure access using Firebase Authentication
Reservation System Module: Allows users to reserve available parking slots.
Real-time Updates Module: Provides instant updates on parking availability.
Testing and Validation Module: Ensures reliability through unit, integration, and system testing.
Performance Evaluation Module: Assesses responsiveness and efficiency.
Documentation Module: Creates comprehensive system documentation and user manuals.
Deployment Module: Implements the system in a real-world environment.
Feedback and Iteration Module: Gathers user feedback for iterative improvements.

V. Work progress / plan & Implementation.

Implementation Timeline:

Week 1-2: Sensor setup and integration with Arduino.

Week 3-4: Raspberry Pi integration and communication establishment.

Week 5-6: Cloud platform integration.

Week 7-8: User interface development (mobile app or web portal).

Week 9-10: Testing, bug fixing, and optimization.

The status of the project is at developing mobile app which is going on.

VI. Sample Code

```
public class SignupActivity extends AppCompatActivity {

    EditText signupName, signupEmail, signupUsername, signupPassword;
    TextView loginRedirectText;
    Button signupButton;
    FirebaseDatabase database;
    DatabaseReference reference;
        signupName = findViewById(R.id.signup_name);
        signupEmail = findViewById(R.id.signup_email);
        signupUsername = findViewById(R.id.signup_username);
        signupPassword = findViewById(R.id.signup_password);
        signupButton = findViewById(R.id.signup_button);
        loginRedirectText = findViewById(R.id.loginRedirectText);

                database = FirebaseDatabase.getInstance();
                reference = database.getReference("users");

                String name = signupName.getText().toString();
                String email = signupEmail.getText().toString();
                String username = signupUsername.getText().toString();
                String password = signupPassword.getText().toString();
                HelperClass helperClass = new HelperClass(name, email, username, password);
                reference.child(username).setValue(helperClass);

                Toast.makeText(SignupActivity.this, "You have signup successfully!",
Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(SignupActivity.this, LoginActivity.class);
                startActivity(intent);
            }
        );
```

VII. Conclusions

In conclusion, the development of the Smart Parking System is progressing according to plan, with a dedicated focus on creating a user-friendly mobile app. The initial phases have seen successful completion, including requirement gathering, technology stack selection, and the initiation of front-end development using React Native. The implementation has prioritized real-time updates and basic user authentication, providing a solid foundation for the subsequent integration of the reservation system and additional features. The phased development approach, coupled with iterative testing and user feedback, ensures a systematic and user-centric implementation process. As we move forward, the project team is committed to delivering a robust and efficient Smart Parking System, leveraging the integration of ultrasonic sensors, Firebase, and the mobile app to optimize parking space utilization and enhance the overall urban mobility experience.

VIII. References:

[1] IoT based Smart Parking System using ESP8266 NodeMCU. (n.d.). Circuitdigest.com. Retrieved February 25, 2024, from https://circuitdigest.com/microcontroller-projects/iot-based-smart-parking-system-using-nodemcu

[2] Macharla, M. (2020, June 20). IoT based Smart Parking System. IoTEDU. https://iot4beginners.com/iot-based-smart-parking-system/