ParkEase - Smart Parking Management System

Problem Statement:

The parking situation at **Koneru Lakshmaiah Educational Foundation (KLEF)** is becoming increasingly difficult due to limited parking spaces across multiple floors. Currently, there is no automated way to track which parking spaces are vacant or occupied, leading to frustration and wasted time spent searching for available parking. The existing process relies on manual inspection, which is inefficient and does not provide real-time data.

Proposed Solution:

ParkEase is a **smart parking management system** designed to address these challenges. It utilizes **ESP32 sensors** to detect whether parking spots are occupied or vacant. The data from these sensors will be sent to a centralized server, where it will be stored and processed. A **web-based dashboard** will display real-time information about parking availability across multiple floors, making it easier for students, faculty, and staff to find available spots.

The system will:

- Provide real-time updates on parking spot availability.
- Offer a visual representation of parking spaces (vacant or occupied).
- Optimize parking space usage and reduce congestion.

Objectives:

1. **Real-time Parking Monitoring**: Track the status of parking spaces (vacant/occupied) in real-time using ESP32 sensors.
2. **Web Dashboard**: Provide a user-friendly dashboard to view parking availability.
3. **Efficient Space Utilization**: Ensure optimal use of parking spaces.
4. **Reduce Time Spent Searching for Parking**: Save time and reduce frustration by providing real-time data on parking availability.
5. **Scalability**: Ensure the system can be expanded to accommodate more parking spots or floors.

Scope:

- **Phase 1**: Develop the web application to monitor parking availability using a small number of test systems and ESP32 sensors.
- **Phase 2**: Scale up the system to monitor more parking floors and integrate additional features such as notifications.
- **Phase 3**: Develop mobile app integration for easy access on smartphones.

Tech Stack:

- **Frontend**: **React** (for building dynamic user interfaces)
- **Backend**: **Spring Boot** (for handling API requests and processing sensor data)
- **Database**: **MySQL** (for storing parking space data)
- **IoT Devices**: **ESP32** sensors (to detect parking spot status)
- **Web Framework**: **React** for frontend development, **Spring Boot** for backend APIs
- **Real-time Updates**: **WebSockets** or **HTTP polling** for real-time communication between frontend and backend

How the System Works:

1. **ESP32 Sensors in Parking Spaces**: Each parking spot is equipped with an **ESP32** sensor. The sensor detects the presence or absence of a vehicle in the parking space and sends this data to the backend system (either via Wi-Fi or Bluetooth).
2. **Backend System (Spring Boot)**: The backend processes data received from the ESP32 sensors. When a sensor detects a parked car, the backend stores this information in a database. If the car leaves the spot, the backend updates the database to reflect the spot as vacant.
3. **Frontend Dashboard (React)**: The user-facing web dashboard shows the parking availability across multiple floors. It will use real-time data from the backend to display the status of parking spaces. The dashboard will provide a clear visual representation of each parking floor and the status of individual spots (vacant/occupied).
4. **Real-time Updates**: The system will update the parking spot status in real-time. As soon as a car parks or leaves, the sensor will send updated data to the backend, which will push the update to the frontend dashboard.

Target Audience:

- **Students**: To find available parking spots quickly and easily, especially during peak hours.
- **Faculty and Staff**: To ensure efficient use of parking spaces near their departments or offices.
- **Visitors**: To help visitors find available parking spots without having to search around the parking area.

Project Benefits:

- **Increased Efficiency**: Users can find available parking spots in real-time, reducing time spent searching for parking.
- **Better Parking Utilization**: By tracking parking space occupancy, the system ensures that spaces are used efficiently.

- **Environmental Impact**: Less time spent circling the parking lot reduces fuel consumption and helps reduce the carbon footprint.
- **Scalable**: The system can easily be expanded to accommodate additional parking floors or spots.

Potential Challenges:

- **Sensor Reliability**: Ensuring that the ESP32 sensors provide accurate and reliable data on the status of parking spots.
- **Real-time Data Handling**: Ensuring smooth, real-time communication between sensors, backend, and frontend.
- **System Adoption**: Ensuring that users are aware of and actively using the dashboard to check for parking availability.

Tech Stack Overview:

- **Frontend (React)**: React provides a dynamic, component-based architecture ideal for handling the real-time data updates from the backend.
- **Backend (Spring Boot)**: Spring Boot simplifies backend development by providing a set of tools and configurations for handling data, APIs, and real-time communication with the frontend.
- **Database (MySQL)**: A reliable relational database will store parking spot data, including the status (vacant or occupied) and the location of each sensor.
- **IoT (ESP32)**: The ESP32 sensor is the core hardware for detecting parking space occupancy. It's low-cost and efficient for such use cases, with easy integration with a backend via Wi-Fi or Bluetooth.

Project Timeline:

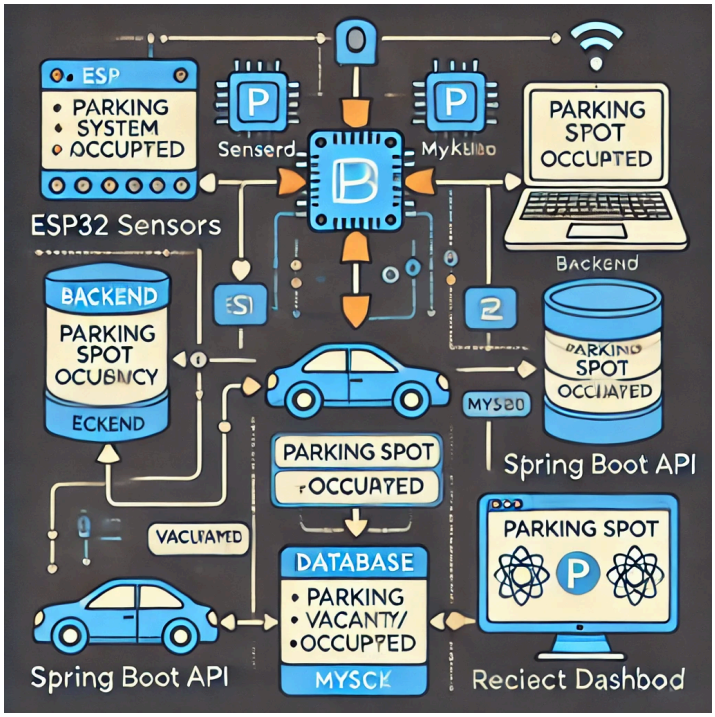
1. **Phase 1 (Testing and Setup)**:
    - Setup ESP32 sensors in 1-2 parking spots for testing.
    - Build the initial backend (Spring Boot) to receive and store data from sensors.
    - Develop the frontend dashboard using React to display the status of test parking spots.
2. **Phase 2 (Scaling and Features)**:
    - Add more parking spaces and sensors.
    - Implement real-time updates and refine the frontend dashboard for scalability.
    - Begin collecting parking data to analyze usage patterns and improve the system.
3. **Phase 3 (Deployment and Mobile Integration)**:
    - Deploy the system to cover multiple floors of the parking area.
    - Implement mobile app integration for better accessibility.

- Continuously monitor and improve system reliability and user experience.

ParkEase System Flow

1. **ESP32 Sensors detect whether a parking spot is occupied or vacant.**
2. **The data is sent from the ESP32 Sensors to the Backend (Spring Boot API).**
3. **The Backend stores the data in the Database (e.g., MySQL).**
4. **The Frontend (React Dashboard) fetches the data from the Backend to display real-time parking status.**

Diagram



Explanation of the Diagram

1. **ESP32 Sensors:**
   - **Each sensor detects if a parking spot is vacant or occupied and sends this status to the Backend.**
2. **Backend (Spring Boot API):**
   - **The backend receives data from the ESP32 sensors and processes it.**
   - **It updates the status of each parking spot in the Database (e.g., MySQL).**
3. **Database:**
   - **The Database stores the vacant or occupied status for each parking spot.**

4. **React Dashboard:**
    - **The React Dashboard fetches real-time parking spot availability from the Backend.**
    - **It displays the parking availability status in an easy-to-understand, visual format (e.g., color-coded spots for vacant and occupied).**

How Data Flows

1. **Sensors detect parking status and send the information.**
2. **The Backend processes the information and updates the Database.**
3. **The Frontend (React) requests real-time data from the Backend and displays the parking spot status to the user.**

Conclusion:

ParkEase is a **smart parking management system** designed to improve the parking experience at **KLEF** by providing real-time information about available parking spots. Using **ESP32 sensors** combined with a web dashboard, the system offers an efficient and scalable solution to monitor and utilize parking spaces. By reducing time spent searching for parking, it not only saves users time but also contributes to a more sustainable and optimized parking environment.