IOT_Phase1

SMART PARKING

Project Definition:

The project involves integrating IoT sensors into public transportation vehicles to monitor ridership, track locations, and predict arrival times. The goal is to provide real-time transit information to the public through a public platform, enhancing the efficiency and quality of public transportation services. This project includes defining objectives, designing the IoT sensor system, developing the real-time transit information platform, and integrating them using IoT technology and Python.

Workflow:

Project Objectives:

- **Real-time Parking Space Monitoring:** Provide real-time data on available parking spaces.
- **Mobile App Integration:** Offer users the ability to check parking space availability through their smartphones.
- **Efficient Parking Guidance:** Direct drivers to the closest available parking space, minimizing time and fuel wastage.

IoT Sensor Design:

- **Sensor Selection:** Ultrasonic sensors or infrared sensors to detect the presence of a vehicle in a parking space.
- **Data Transmission:** Low power Wi-Fi or dedicated IoT networks for data transmission.
- **Sensor Placement:** Strategically place sensors in each parking spot to maximize accuracy.

Real-Time Transit Information Platform:

Mobile App Design:

- **Interactive Map:** Display a map of the parking area with real-time indications of available and occupied spots.
- **Notifications:** Alert users when a parking spot becomes available.
- **Navigation:** Offer turn-by-turn directions to the nearest available parking spot.
- **Feedback Mechanism:** Allow users to report sensor errors or provide suggestions.

Integration Approach:

Raspberry Pi Integration:

- Utilize Raspberry Pi as the local server/gateway to collect data from sensors.
- Use Python scripts or Node.js for reading the data from sensors and transmitting it to the cloud or directly to the app.

Cloud Integration:

- Integrate Raspberry Pi with cloud platforms like AWS, Google Cloud, or Azure to process and store data.

App Integration:

- APIs to fetch real-time data from the cloud and display it on the app.

Testing:

- Initial testing with a limited number of parking spots to check the system's efficiency and accuracy.

Test:

- Deploy the system in a controlled environment or a single parking lot.
- Collect user feedback on the mobile app interface, accuracy of the data, and the overall experience.
- Monitor sensor accuracy and address any technical glitches.

Implement:

- Based on the feedback and test results, refine the system.
- Gradually roll out the system to multiple parking areas or lots.
- Keep an iterative approach, continuously updating the system based on user feedback and technological advancements.