PHASE 2: INNOVATION

SMART PROJECT USING IOT

In this project ,making a smart parking system which makes the people to make and feel better to park their vehicle without any tension. By developing this project consumers can book the parking place online without any interruption by their convenience .

THE REQUIREMENTS:

Parking Sensors: These are the primary IoT devices used to detect the presence of vehicles in parking spots. There are different types of sensors available, including ultrasonic, infrared, magnetic, and camera-based sensors.

Microcontrollers/Embedded Systems: You'll need microcontrollers or embedded systems like Arduino, Raspberry Pi, or specialised IoT development boards to connect and process data from the parking sensors.

Communication Modules: IoT devices need a way to transmit data to a central server or cloud. You can use Wi-Fi, cellular, LoRaWAN, or other communication modules depending on the range and connectivity requirements.

Central Server or Cloud Platform: A central server or cloud platform is where data from parking sensors is collected, processed, and analysed. Popular choices include AWS, Azure, Google Cloud, or specialised IoT platforms.

User Interface: Create a user-friendly web or mobile application that allows users to check parking spot availability and reserve spots if necessary.

Power Supply: Ensure a stable power supply for sensors and microcontrollers. Battery-powered sensors may require periodic battery replacements or recharging.

Database: Store data about parking spot availability, reservations, and user information in a database for retrieval and analysis.

- Ultrasonic sensor
- led(display the car state)
- Xbee series2
- Arduino
- Database sql

These are the major requirements

FLOW OF PROCESS:

Step 1: Define Objectives and Scope

Determine the primary goals of your smart parking system, such as improving user convenience, optimising space utilisation, reducing congestion, or generating revenue.Identify the specific location or parking facility where you want to implement the system.

Step 2: Select Parking Sensors

Choose the type of parking sensors based on your budget and requirements.Consider factors like accuracy, installation complexity, and maintenance requirements.

Step 3: Choose IoT Devices

Select microcontrollers for IoT devices to collect data from parking sensors and manage communication.Ensure compatibility with the chosen sensors and communication protocols.

Step 4: Decide on Communication Protocols

Determine the communication method for transmitting sensor data to a central server or cloud platform.Options include Wi-Fi, cellular (3G/4G/5G), LoRaWAN, or Ethernet.

Step 5: Set Up Central Server or Cloud Platform

Choose a cloud platform (e.g., AWS, Azure, Google Cloud) or set up your own central server to receive, process, and store data from the sensors.
Configure the necessary cloud services and databases.

Step 6: Sensor Deployment

Install sensors in each parking space. Ensure they are securely mounted to prevent tampering.Consider environmental factors, such as weather resistance for outdoor installations.

Step 7: Connect Sensors to IoT Devices

Wire the sensors to your selected microcontrollers or IoT devices.
Develop firmware or software to read data from the sensors.

Step 8: Data Transmission

Use the communication modules on IoT devices to send sensor data to the central server or cloud platform.Implement secure data transmission protocols to protect sensitive information.

Step 9: Data Processing

- Develop algorithms on the central server or cloud platform to:
- Process incoming data from sensors.
- Determine parking spot availability in real-time.
- Update the status of parking spots in a database.

Step 10: User Interface Development

- Create a user-friendly web or mobile application for users to interact with the system.
- Include features such as:
- Real-time parking spot availability display.
- Reservation functionality (if needed).
- User registration and login.
- Notifications and alerts.

Step 11: Database Management

- Set up and configure a database to store data related to:
- Parking spot availability.
- Reservations.
- User accounts and payment information (if applicable).

Step 14: Testing and Optimization

- Thoroughly test the entire system to ensure all components work together seamlessly.
- Optimise for performance, scalability, and reliability.
- Address any bugs or issues that arise during testing.