**SMART PARKING**

**Problem definition:**

The objective of this project is to design, develop, and deploy an innovative parking management system that leverages Internet of Things (IoT) technology. The system will optimize parking space utilization, improve user experience, and reduce traffic congestion in urban areas. An IoT-based parking system is a centralized management that enables drivers to search for and reserve a parking spot remotely through their smartphones. It offers a convenient arrangement for drivers to park their cars when they are looking to avoid potential traffic congestion.

**Detailed Explanation:**

* **Parking Monitoring:** ImplementIoT sensors in each parking space or at entrance/exit points of parking slots. These sensors detect the presence or absence of vehicles and send this data to a central server or cloud platform.
* **Occupancy Monitoring:** Real-time data on parking space occupancy is collected and this information can be used to optimize parking space allocation and pricing strategies.
* **Parking Guidance:** Develop a system to guide a drivers to the nearest available parking spaces using navigation and mapping services. This reduces the time and fuel wasted in searching for parking. This can be accessible through mobile apps, digital displays at stops, or other communication channels.
* **User Interface:** A user-friendly mobile app provides real-time parking availability information to drivers. Users can view the nearest available parking spaces.
* **Reservation and Payment:** Users can reserve parking spaces through the app and make payments electronically. This reduces the need for physical payment.
* **Notifications and Alerts:** The system can send notifications and alerts to users, such as reminders for a parking reservations or notifications about parking availability**.**

**Design thinking:**

**Project objectives:**

* To reduce delays, improves route efficiency, and provides real-time information, collectively reducing the overall travel time for passengers.

**IoT sensor design:**

* Generate ideas for a system that incorporates IR sensors, level sensors, speed sensors, temperature and humidity sensors, GPS module, Wi-Fi module, and an LCD interface.
* Create a physical prototype, connecting IR sensors for obstacle detection, level sensors for fluid monitoring, speed sensor, temperature, and humidity sensors for environmental conditions, GPS module for location tracking, and Wi-Fi module for data transmission.

**Real time transit Information platform:**

* The web-based real-time transit information platform aims to provide a comprehensive, user-friendly, and secure experience for passengers. Continuous improvement through iterative testing and user feedback ensures that the platform remains responsive to changing needs and technological advancements.

**Integration approach:**

* ThinkSpeak is a popular Internet of Things (IoT) platform that allows users to easily build and control IoT projects through a user-friendly mobile app. It provides a simple way to connect various hardware devices, sensors, and microcontrollers to the internet and control them remotely.

**Work Flow:**

* When system is powered up, it goes through initializing phase during which it sets the baud rate for connected devices communicating a Universal Asynchronous Received Transmitted (UART) serial connected devices such as Wi-Fi module, GPS, and serial monitor as well initializes the LCD and DHT sensor.
* After that microcontroller scans all the sensors and reads both analogue and digital sensors.
* The microcontroller then performs analogue to digital conversion for all analogue read sensors. It then processes the data read and computes passenger count, calculate bus speed, compute GPS coordinates, passenger count, temperature, and humidity.
* All computed parameters are displayed on the LCD locally.
* The same values are also sent to serial monitor for testing and debugging purpose.
* These parameters are sent to the ThingSpeak cloud over the internet using ESP8266 module.
* The data sent to cloud is fetched from the cloud and displayed in a web browser for user visualization. In the next step data is fetch from the cloud using channel ID and API key which is then analyzed using the MATLAB.

**Team Members:**

Aishwarya I (2021504005)

Hayagreevan S (2021504009)

Janani M (2021504010)

Naresh Kumar S (2020504025)

Shohail Shahabudeen (2021504040)