|  |
| --- |
| **SMART PARKING** |
| Using IoT |
| Summitted by, |
| p.santhosh |
| **au812921106029**  **[santhosh20040117@gmail.com](mailto:santhosh20040117@gmail.com)** |
|  |
|  |
|  |

Abstract:

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.

Design Thinking :

**1. Define :**

* Clearly articulate the problem and establish a design challenge, such as "How might we make parking more efficient and convenient?"
* Set specific goals, like reducing the average time it takes to find parking.

**2. Ideate:**

* Brainstorm potential solutions with a cross-functional team.
* Consider IoT technologies like sensors, cameras, and connectivity to collect data and optimize parking.

**3. Prototype:**

* Create a low-fidelity prototype of the smart parking system.
* Include key components like IoT sensors to detect empty parking spots, a central control system, and a user interface (e.g., mobile app).

**4. Test:**

* Gather user feedback on the prototype.
* Make improvements based on feedback and iterate the prototype if necessary.

**5. Develop:**

* Build the actual IoT-based smart parking system.
* Implement sensors in parking spaces to detect occupancy and transmit data to a central server.

**6. Connect:**

* Ensure all components are interconnected using IoT protocols and networks like Wi-Fi, Bluetooth, or LPWAN (Low-Power Wide-Area Network).

**7. Data Analysis:**

* Collect and analyze data from sensors to monitor parking space occupancy and patterns.
* Use this data to optimize parking allocation and provide real-time information to users.

**8. User Interface:**

* Develop a user-friendly mobile app or web interface that displays parking availability, location, and navigation options.

**9. Testing and Optimizatio :**

* Continuously test the system to ensure accuracy and reliability of sensor data.
* Optimize algorithms for parking allocation and update the user interface based on user feedback.

**10. Deployment :**

* Install IoT sensors in parking areas and set up the central control system.
* Launch the mobile app or web platform for users to access.

**11. Monitoring and Maintenance :**

* Regularly monitor the system for performance, connectivity issues, and sensor maintenance.
* Provide customer support for users encountering problems.

**12. Scale and Expand :**

* Consider expanding the system to cover more parking areas or integrate with other smart city initiatives.