Smart Public Restrooms

Documentation

Project Objectives:

The Smart Public Restrooms project aims to create a modern and efficient restroom experience for the public by integrating IoT technology, mobile app development, Raspberry Pi, and a real-time restroom information system. The key objectives include:

1. **Improving User Experience**: Enhancing the overall restroom experience for the public by providing real-time information on restroom availability and cleanliness.

2. **Optimizing Restroom Management**: Assisting restroom facility managers in monitoring and maintaining restrooms efficiently.

3. **Resource Conservation**: Reducing water and electricity consumption by optimizing restroom facilities' usage.

IoT Sensor Setup:

IoT sensors play a pivotal role in capturing data for real-time restroom information. The setup includes the following sensors:

1**.Occupancy Sensors**: Installed at restroom entrances to detect if the restroom is occupied or vacant.

2. **Toilet Flush Sensors**: Placed inside each toilet to monitor flush frequency and identify any potential issues.

3. **Water Flow Sensors:** Installed in water supply lines to track water usage and detect leaks.

4. **Quality Sensors:** These sensors monitor air quality, humidity, and temperature inside the restroom, ensuring a comfortable environment.

Mobile App Development

A mobile app has been developed to enable users to access real-time restroom information. Key features include:

1. **Restroom Locator**: Helps users find the nearest smart public restroom.

2. **Real-time Availability**: Displays which restrooms are currently occupied and which are available.

3. **Cleanliness Ratings**: Allows users to rate and provide feedback on restroom cleanliness.

4. **QR Code Entry**: Users can enter restrooms using QR codes generated by the app, eliminating the need for physical keys.

5. **Wait Time Estimation**: Provides estimated wait times for occupied restrooms.

Raspberry Pi Integration:

Raspberry Pi serves as the central hub for data collection, processing, and communication. It collects data from IoT sensors, processes it, and then sends the information to the mobile app and the restroom information platform. The Raspberry Pi also controls automated features like smart locks and alerts for maintenance.

Code Implementation:

The code implementation involves multiple components:

1. **IoT Sensor Code**: Each sensor has specific code to capture data and send it to the Raspberry Pi.

2. **Raspberry Pi Code**: This code collects, processes, and manages data from sensors, communicates with the mobile app, and controls smart locks.

3. **Mobile App Code**: Developed for iOS and Android platforms, the app code interacts with the Raspberry Pi to display real-time information and enable user interaction.

Diagrams and Screenshots:

Enhancing User Experience and Restroom Management:

The real-time restroom information system enhances user experience and restroom management in several ways:

1. **Enhanced Convenience**: Users can quickly locate available restrooms, reducing the time spent searching for facilities.

2. **Improved Cleanliness**: Real-time feedback and ratings encourage restroom operators to maintain clean and hygienic facilities.

3. **Resource Optimization:** The system helps conserve resources by detecting leaks and monitoring water usage.

4. **Data-Driven Management**: Facility managers can use data collected by IoT sensors for proactive maintenance, reducing downtime.

5. **Cost Reduction**: By optimizing resource usage and reducing maintenance costs, the system can lead to cost savings.

In summary, the Smart Public Restrooms project leverages IoT, mobile app development, Raspberry Pi, and real-time data to create a more user-friendly and efficient public restroom experience, benefiting both the public and facility managers.

a smart public restroom project involving IoT sensors, a transit information platform, and Python integration requires careful planning and execution. Here are step-by-step instructions to replicate this project:

1. IoT Sensors for Smart Public Restroom:

IoT sensors can monitor various aspects of the public restroom to ensure cleanliness, safety, and efficiency.

1. **Choose Sensors:**
    - Select appropriate sensors for your use case, such as occupancy sensors, motion detectors, air quality sensors, or smart soap dispensers.
    - Ensure the sensors can communicate with a central system (e.g., Raspberry Pi) using standard communication protocols like GPIO, I2C, or UART.
2. **Set Up Raspberry Pi**:
    - If you're using a Raspberry Pi as the central hub, set up the Raspberry Pi with the necessary software and network connectivity.
    - Install an operating system like Raspberry Pi OS.

3. **Develop Data Collection Code:**
    - Write Python scripts to interface with the sensors and collect data. You may use libraries specific to your sensors.
    - Store or transmit sensor data to a central server or platform.
4. **Store and Transmit Data:**
    - Store the collected data locally or send it to a central database or platform.
    - Consider using secure communication protocols (e.g., HTTPS) if transmitting data over the internet.
5. **Create a GitHub Repository:**
    - Create a GitHub repository for your IoT sensor code and data collection scripts.
    - Upload your code and documentation to the repository.

2. Transit Information Platform:

For a smart public restroom, the transit information platform will provide real-time information about restroom status.

1. **Choose Technologies:**
    - Decide on the technologies for your platform. You can use a web-based platform with HTML, CSS, JavaScript, and a backend framework (e.g., Flask, Django, or Node.js).
2. **Develop the Platform:**
    - Create a web application for users to access real-time data about restroom occupancy, cleanliness, and safety.
    - Develop a backend service to handle data storage and retrieval.
3. **Database Setup:**
    - Set up a database (e.g., PostgreSQL, MySQL, or NoSQL) to store sensor data and user information.
4. **Implement API:**
    - Create an API to provide real-time restroom status data to your mobile app or other clients.
5. **Deployment:**
    - Deploy your web platform on a server or a cloud platform like AWS, Azure, or Heroku.

6. **Create a GitHub Repository:**
    - Set up a GitHub repository for your web platform code.
    - Upload your code and documentation to the repository.

3. Integration Using Python:

Python is used to integrate the IoT sensors with the transit information platform.

1. **Data Transmission:**
    - Modify your IoT sensor code to send data to the Transit Information Platform's API using Python's requests library or other communication methods.
2. **Data Processing:**
    - Write Python code on the web platform side to process and store the incoming sensor data in the database.
3. **Data Retrieval:**
    - Use Python to retrieve sensor data from the database and format it as needed to be served via the API.

4. Mobile App for Restroom Status:

Develop a mobile app to display real-time restroom status to users.

1. **Choose Mobile App Development Framework:**
    - Select a framework for mobile app development, such as React Native, Flutter, or native development tools (e.g., Android Studio for Android apps or Xcode for iOS apps).
2. **Develop the Mobile App:**
    - Create a mobile app that allows users to view real-time data about the smart public restroom.
    - Implement features for user interaction and data visualization.
3. **Integrate with API:**
    - Use the API from the Transit Information Platform to fetch and display real-time sensor data in the mobile app.
4. **Testing and Deployment:**
    - Test the mobile app on real devices to ensure it works as expected.
    - Deploy the app to app stores (Google Play Store, Apple App Store).