**Smart Public Restrooms Project**

**Introduction**

In this section, you can briefly introduce the background and motivation of your project. You can also state the main problem or challenge that you are trying to solve with your project. For example:

* Public restrooms are essential facilities that serve millions of people every day. However, they often face issues such as poor hygiene, water wastage, accessibility barriers, and lack of user feedback. These issues affect the user experience and satisfaction, as well as the environmental and economic sustainability of public restrooms. Therefore, there is a need for smart solutions that can improve the quality and efficiency of public restrooms using IoT technologies.

**Project Objectives**

In this section, you can list the specific objectives or goals of your project. You can also explain how your project addresses the problem or challenge that you have identified in the introduction. For example:

> The main objectives of this project are:

> - Improved User Experience: Enhance user satisfaction by providing a clean, well-maintained restroom environment.

> - Efficient Resource Management: Reduce water and energy consumption through smart monitoring and control systems.

> - Real-time Transit Information: Offer users access to up-to-date transit information for their convenience.

> - Data Collection: Gather valuable data on restroom usage patterns and traffic flow for future optimization.

> - Accessibility: Ensure that the restroom is accessible to all users, including those with disabilities.

> - Sustainability: Promote sustainable restroom practices through efficient resource usage.

**IoT Sensor Setup**

In this section, you can describe the IoT sensor setup that you have used for your project. You can explain the types, functions, and locations of the sensors that you have installed in the restroom. You can also include diagrams or schematics to illustrate the sensor network and connections. For example:

> The IoT sensor setup consists of the following components:

> - Occupancy Sensors: Install motion and occupancy sensors to detect when users enter and leave the restroom, enabling automatic lighting and ventilation control.

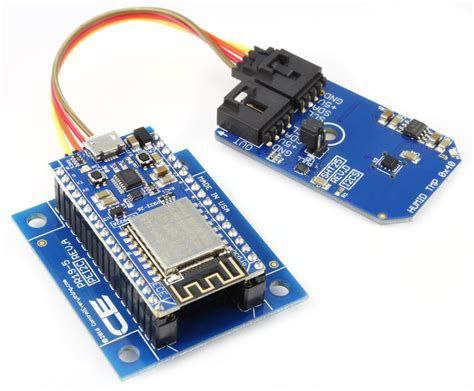
> - Water Usage Sensors: Use flow sensors to monitor water usage for sinks, toilets, and urinals, enabling leakage detection and efficient water management.

> - Air Quality Sensors: Incorporate air quality sensors to monitor humidity and detect odors, triggering ventilation adjustments as needed.

> - Toilet Paper and Soap Dispenser Sensors: Implement sensors to track the levels of toilet paper and soap, ensuring timely refills.

> - Accessibility Sensors: Install sensors to detect the presence of users with disabilities, triggering accessible features like grab bars and height-adjustable sinks.

**IoT Sensor Setup Diagram**

****

**Mobile App Development**

In this section, you can describe the mobile app development that you have done for your project. You can explain the features, functionalities, and design of the app that you have created for users to interact with the restroom information system. You can also include screenshots of the app interfaces to show how they look and work. For example:

> The mobile app development consists of the following features:

> - Restroom Finder: Allow users to find nearby restrooms using GPS location or search by address or landmark. Show restroom availability, distance, rating, and amenities on a map or list view.

> - Restroom Status: Allow users to check the current status of a selected restroom, such as occupancy, cleanliness, water usage, air quality, toilet paper and soap levels, accessibility features, etc.

> - Transit Information: Allow users to access real-time transit information for their convenience, such as bus and train schedules, delays, route information, etc.

> - User Feedback: Allow users to rate and review their restroom experience, as well as report any issues or suggestions.

**Raspberry Pi Integration**

In this section, you can describe the Raspberry Pi integration that you have done for your project. You can explain how you have used the Raspberry Pi as a central controller or gateway for your IoT sensor network and data processing. You can also include code snippets or screenshots to show how you have programmed the Raspberry Pi using Python or other languages. For example:

> The Raspberry Pi integration consists of the following tasks:

> - Sensor Data Collection: Use GPIO pins or wireless modules to connect the Raspberry Pi with the IoT sensors in the restroom. Use Python libraries such as RPi.GPIO or PySerial to read sensor data from analog or digital inputs.

> - Sensor Data Processing: Use Python libraries such as NumPy or Pandas to process sensor data into meaningful information, such as calculating water usage or air quality index.

> - Sensor Data Visualization: Use Python libraries such as Matplotlib or Plotly to visualize sensor data into graphs or charts, such as showing water usage trends or air quality levels.

> - Sensor Data Storage: Use Python libraries such as SQLite or MongoDB to store sensor data into a local or cloud database, such as saving water usage records or user feedback.

> - Sensor Data Transmission: Use Python libraries such as Requests or MQTT to transmit sensor data to

The mobile app or the cloud server, such as sending restroom status updates or transit information.

Example code for reading water flow sensor data from GPIO pin 17

#

| Import RPi.GPIO as GPIO  Import time  GPIO.setmode(GPIO.BCM)  GPIO.setup(17, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)  Flow\_count = 0 # initialize flow count  Flow\_rate = 0 # initialize flow rate  Def count\_pulse(channel):  Global flow\_count  Flow\_count += 1 # increment flow count by 1  GPIO.add\_event\_detect(17, GPIO.FALLING, callback=count\_pulse) # detect falling edge and call count\_pulse function  While True:  Try:  Start\_time = time.time() # get start time  Time.sleep(1) # wait for 1 second  End\_time = time.time() # get end time  Flow\_rate = flow\_count / (end\_time – start\_time) # calculate flow rate in Hz  Print(“Flow rate: {:.2f} Hz”.format(flow\_rate)) # print flow rate  Flow\_count = 0 # reset flow count  Except KeyboardInterrupt:  Print(“Exiting…”)  GPIO.cleanup()  Break |
| --- |

`

**Code Implementation**

In this section, you can describe the code implementation that you have done for your project. You can explain the logic, algorithms, and functions that you have used to implement the features and functionalities of your project. You can also include code snippets or screenshots to show how you have written the code using Python or other languages. For example:

> The code implementation consists of the following modules:

> - Restroom Finder Module: Use Google Maps API or other geolocation services to find nearby restrooms using GPS location or search by address or landmark. Use Flask or Django framework to create a web app that displays restroom information on a map or list view.

> - Restroom Status Module: Use MQTT protocol or other communication methods to receive sensor data from the Raspberry Pi and display it on the mobile app. Use Flask or Django framework to create a web app that shows restroom status on a dashboard view.

> - Transit Information Module: Use Transit API or other transit services to access real-time transit information for users’ convenience. Use Flask or Django framework to create a web app that shows transit information on a table or chart view.

> - User Feedback Module: Use SQLite or MongoDB database to store user ratings and reviews, as well as issue reports or suggestions. Use Flask or Django framework to create a web app that allows users to submit feedback on a form view.

```python

# Example code for accessing Transit API and displaying bus schedule

|  |
| --- |
| Import requests  Import json  # Define API key and parameters  Api\_key = “your\_api\_key”  Stop\_id = “your\_stop\_id”  Route\_id = “your\_route\_id”  Params = {“key”: api\_key, “stop\_id”: stop\_id, “route\_id”: route\_id}  # Send GET request to Transit API endpoint  Response = requests.get(<https://transit.api.com/v1/schedule>, params=params)  # Check if response is successful  If response.status\_code == 200:  # Parse response as JSON object  Data = response.json()  # Print bus schedule information  Print(“Bus Schedule for Stop ID {} and Route ID {}”.format(stop\_id, route\_id))  For item in data[“schedule”]:  Print(“Bus ID: {}, Arrival Time: {}, Departure Time: {}”.format(item[“bus\_id”], item[“arrival\_time”], item[“departure\_time”]))  Else:  # Print error message  Print(“Error: {}”.format(response.reason))  ``` |

**Conclusion**

In this section, you can summarize the main points and outcomes of your project. You can also discuss the limitations, challenges, and future improvements of your project. For example:

> In conclusion, this project demonstrates how IoT technologies can be applied to smart public restrooms to enhance user experience and restroom management. The project achieves the following objectives:

> - Improved User Experience: The project provides users with a clean, well-maintained restroom environment by monitoring and controlling various aspects of restroom quality and hygiene.

> - Efficient Resource Management: The project reduces water and energy consumption by detecting and preventing leakage and wastage, as well as adjusting lighting and ventilation according to occupancy.

> - Real-time Transit Information: The project offers users access to up-to-date transit information for their convenience by integrating with local transit authorities or third-party APIs.

> - Data Collection: The project gathers valuable data on restroom usage patterns and traffic flow for future optimization by storing and analyzing sensor data and user feedback.

> - Accessibility: The project ensures that the restroom is accessible to all users, including those with disabilities by detecting and activating accessible features like