**(M.Harish – au732721106016)**

**PUBLIC TRANSPORTATION OPTIMIZATION**



**1.Planning and Equipment Selection:**

⦁ Determine the types of IoT sensors needed. Common sensors for public transportation optimization include GPS, passenger counters, and possibly environmental sensors.

⦁ Choose the hardware and communication protocols for the IoT devices. Common choices include Raspberry Pi, Arduino, or specific IoT development boards.

**2.IoT Sensor Deployment:**

⦁ Install the selected sensors on public transportation vehicles. This may require collaboration with the transit agency or vehicle owner.

⦁ Ensure proper power supply and secure mounting of sensors.

**3.Data Gathering and Processing:**

⦁ Develop or configure the firmware for the IoT sensors to collect and process data. For example, you can use Python on Raspberry Pi for this task.

⦁ Implement code to read GPS coordinates, count passengers, and any other relevant data.

**4.Real-Time Data Transmission:**

⦁ Establish a connection to the transit information platform. This might involve setting up a server or cloud infrastructure.

⦁ Develop a Python script to send real-time data from the IoT sensors to the transit information platform. You can use IoT communication protocols like MQTT or HTTP for this purpose.

Here's a basic example of a Python script for sending GPS and passenger count data via MQTT:

**python code:-**

import paho.mqtt.client as mqtt

import time

# Define MQTT broker and topics

broker\_address = "mqtt.broker.com"

topic\_gps = "public\_transport/gps"

topic\_passenger\_count = "public\_transport/passenger\_count"

# Simulated data for GPS and passenger count

def get\_gps\_data():

# Implement code to retrieve GPS data

pass

def get\_passenger\_count():

# Implement code to count passengers

pass

# Create an MQTT client

client = mqtt.Client("TransportSensor")

# Connect to the MQTT broker

client.connect(broker\_address)

while True:

gps\_data = get\_gps\_data()

passenger\_count = get\_passenger\_count()

# Publish data to respective MQTT topics

client.publish(topic\_gps, gps\_data)

client.publish(topic\_passenger\_count, passenger\_count)

time.sleep(10) # Send data every 10 seconds

# Disconnect from the MQTT broker

client.disconnect()

**5.Data Storage and Analysis:**

⦁ Set up a database or data storage solution to store the incoming data.

⦁ Implement data analytics and visualization tools to make sense of the collected data.

**6.Optimization Algorithms:**

⦁ Develop or integrate optimization algorithms that make use of the real-time data to improve public transportation efficiency. This may include route optimization, scheduling, or predictive maintenance.

**7.User Interface:**

⦁ Create a user interface for transit operators and passengers to access real-time information, such as vehicle location and estimated arrival times.

**8.Testing and Deployment:**

⦁ Thoroughly test the entire system in a controlled environment.

⦁ Deploy the system on a small scale and gather feedback from transit operators and passengers.

**9.Scalability and Maintenance:**

⦁ Plan for the scalability of the system as more vehicles are equipped with sensors.

⦁ Establish maintenance procedures to ensure sensors and software are up to date and functioning correctly.

**10.Regulatory Compliance:**

⦁ Ensure that the system complies with data privacy and transportation regulations.

Remember that building an IoT-enabled public transportation optimization system is a complex project that may require collaboration with transit agencies, software developers, and IoT specialists. The specific details and requirements of your project will depend on your location and goals.

**Team Members:-**

1. **K.S. HARI PRASATH**
2. **M. HARISH**
3. **K. KISHORE SARATHI**
4. **R. SANTHOSH**