PUBLIC TRANSPORT OPTIMIZATION

Team Members:

Varsha V (113321104110)

Velavaluru Rajitha Sai (113321104112)

Veluru Priyadarshini (113321104113)

Yaddala Harshitha (113321104117)

Yuva Shree (113321104120

Phase 3: Development part 1

PROJECT OBJECTIVES:-

The objectives of the public transport is optimization should make the possibilities of efficiency, cost effectiveness, safety, and other important measures are as follows:

- **Enhance Efficiency and Reliability**: To ensure that the passengers should not waste there time by making the passengers to get information of the transport details.
- **Enhance Safety and Security**: To implement safety measures for the passengers and staffs.
- **Financial Sustainability**: To optimize the cost-effectiveness in the public transport services.
- **Data-Driven Decision-Making**: To collect and analyze data on passengers flows, route performance, and other measures to inform decision-making.

CODE IMPLEMENTATION:

To run the public transport optimization with IoT integration Python code, you will need to:

1. Install the required Python libraries:

   - paho.mqtt.client
   - scipy
   - json

2. Replace the distance() function with a function that calculates the distance between two locations in your area. You can use a variety of methods to do this, such as the Google Maps Distance Matrix API or the OpenStreetMap Overpass API.
3. Start the MQTT broker.
4. Start the Python code.
5. Publish sensor data to the MQTT topic public_transportation.
6. Once all sensor data has been received, the code will solve the optimization problem and implement the optimized routes.

```
import paho.mqtt.client as mqtt
import time
import json
from scipy.optimize import linprog
import math

# Define the MQTT broker address and topic
MQTT_BROKER_ADDRESS = "localhost"
MQTT_TOPIC = "public_transportation"

# Create an MQTT client
client = mqtt.Client()

# Connect to the MQTT broker
client.connect(MQTT_BROKER_ADDRESS)

# Subscribe to the MQTT topic
client.subscribe(MQTT_TOPIC)

# Define a callback function to handle incoming MQTT messages
def on_message(client, userdata, msg):
    # Decode the JSON message
    data = json.loads(msg.payload.decode())

    # Get the sensor data
    ridership = data["ridership"]
    location = data["location"]
    predicted_arrival_time = data["predicted_arrival_time"]

    # Update the public transport optimization model
    # ...

# Start the MQTT client loop
client.loop_forever()

# Define the public transport optimization model
def public_transport_optimization(ridership, locations, predicted_arrival_times):
    # Define the variables
    num_vehicles = len(locations)
    routes = [[] for i in range(num_vehicles)]

    # Define the objective function
    def objective(routes):
        return sum([sum([ridership[i] * distance(locations[i], locations[j]) for j in routes[i]]) for i in
range(num_vehicles)])
# Define the constraints
    constraints = []
    for i in range(num_vehicles):
        constraints.append(sum([ridership[j] for j in routes[i]]) <= 50)

    # Solve the optimization problem
    result = linprog(objective, constraints=constraints)

    # Return the optimal routes
    return routes
```

```
# Define the distance function
def distance(location1, location2):

  delta_lat = location2[0] - location1[0]
  delta_lon = location2[1] - location1[1]
  # Calculate the distance in kilometers.
  distance = math.sqrt(delta_lat**2 + delta_lon**2) * 111.325

  return distance
  # Calculate the distance between two locations
    # ...

# Implement the main function
def main():
    # Create a public transport optimization model
    model = public_transport_optimization()

    # Start listening for sensor data
    client.loop_forever()

    # Once all sensor data has been received, solve the optimization problem
    routes = model.solve()

    # Implement the optimized routes
    # ...

if __name__== "_main_":
    main()
```

CONCLUSION:

In conclusion, public transport optimization is a multifaceted and dynamic process aimed at improving the efficiency, safety, and quality of public transportation systems. It involves a combination of planning, data collection, technological integration, and continuous improvement efforts. Key takeaways from public transport optimization include:

- **Efficiency:** Optimization efforts aim to make public transportation systems more efficient by improving route planning, scheduling, resource allocation, and maintenance practices. This leads to cost savings, reduced congestion, and improved passenger experiences.
- **Data-Driven Decision-Making:** Data collection and analysis play a central role in public transport optimization.Real-time and historical data help transportation authorities and operators make informed decisions and respond to changing conditions.
- **Passenger Satisfaction**: Enhancing the passenger experience is a core objective of public transport optimization. This includes improving on-time performance, reducing wait times, ensuring safety, and providing accessible services.
- **Environmental Impact**: Public transport optimization can contribute to reducing the environmental impact of transportation by promoting the use of clean energy sources and optimizing routes to minimize fuel consumption and emissions.
- **Safety and Security:** Safety and security measures are integrated into optimization efforts, with technologies like surveillance cameras, sensors, and real-time monitoring systems helping to ensure the well-being of passengers and staff.

In summary, public transport optimization is a complex and multifaceted endeavor that requires a combination of data-driven decision-making, technological innovation, and a commitment to meeting the needs of passengers and communities. As cities continue to grow and environmental concerns become more pressing, the importance of optimizing public transport systems is expected to increase, making transportation more efficient, accessible, and sustainable for all.