PUBLIC TRANSPORT OPTIMIZATION

PHASE 3

TABLE OF CONTENTS

Public Transport Optimization

Introduction

The Public Transport Optimization Project aims to enhance the efficiency and effectiveness of public transportation systems by leveraging Internet of Things (IoT) devices and developing a Python script to optimize operations. This document provides a comprehensive overview of the project's key components and working principles from start to finish.

Project Overview

Objectives

The primary objectives of the project are as follows:

- *Improve public transport system efficiency.*
- *Enhance passenger experience.*
- *Reduce operational costs.*

- *Minimize environmental impact.*

Key Components

The project consists of the following key components:

IoT Device Deployment

To achieve real-time data collection and monitoring, IoT devices are deployed within the public transportation system. These devices are strategically placed on vehicles, at stations, and along routes to gather essential data.

Data Collection and Transmission

IoT devices collect data related to:

- *Vehicle location and speed.*
- *Passenger occupancy.*
- *Traffic conditions.*

The collected data is transmitted securely to a central database.

Central Database

All data is stored in a central database for analysis and real-time access. The database serves as the backbone of the project, ensuring that decisionmaking processes are data-driven.

IoT Device Setup:

Acquire IoT devices with GPS and other relevant sensors to be installed on public transport vehicles (buses).

Configure IoT devices to collect data such as GPS coordinates, vehicle speed, and environmental conditions.

GPS Sensor:

Application: Used for real-time vehicle tracking and route optimization.

Benefits: Provides accurate location data, helping to monitor vehicle movement, calculate ETA, and optimize routes based on traffic conditions.

Proximity Sensors (Ultrasonic or Infrared):

Application: Detecting the proximity of vehicles to obstacles, objects, or pedestrians.

Benefits: Enhances safety by providing alerts to drivers and helping avoid collisions.

Camera Sensors (CCTV):

Application: Surveillance and monitoring of passengers, driver behavior, and security.

Benefits: Improves safety and security by recording video footage for analysis and incident resolution.

Public Transport Optimization

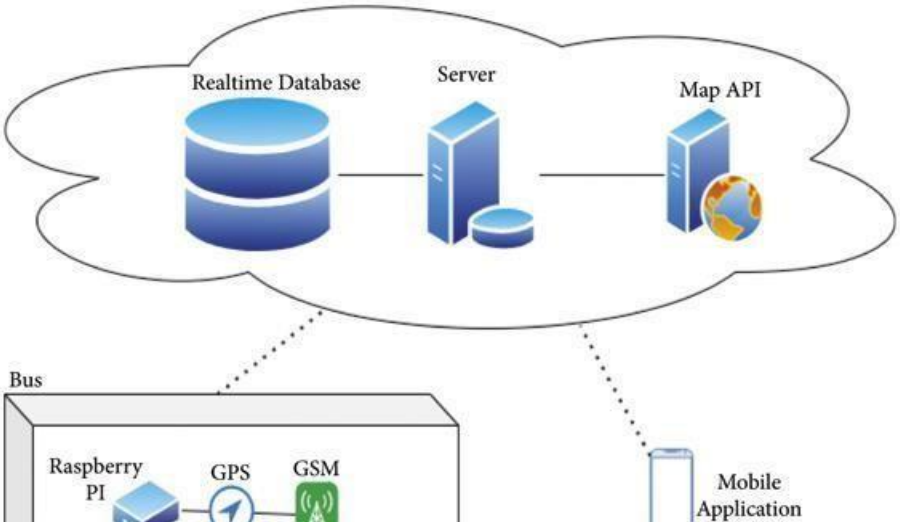
Working Principles

Data Collection and Analysis

- Develop firmware for the IoT devices using a programming language suitable for the IoT platform (e.g., Python for Raspberry Pi or Arduino).
- Program the IoT devices to collect and transmit real-time data to a central server.
- Ensure data is transmitted securely.

Python Script Development

A Python script is developed to process the data received from the IoT devices. The script's



1.Route Optimization

The Python script uses historical and real-time data to optimize routes by:

- Identifying traffic congestion.
- Analyzing passenger demand.
- Recommending alternative routes in case of disruptions.

- Predictive Maintenance :

The script monitors the condition of vehicles and predicts maintenance needs based on sensor data. This proactive approach reduces breakdowns and improves vehicle reliability.

- Passenger Load Balancing
  - Passenger load data is used to optimize vehicle schedules and capacity allocation, ensuring a comfortable and efficient experience for passengers.
  -
- Real-Time Service Updates
  - Passengers receive real-time updates through various channels, including mobile apps and digital displays at stations. These updates include: ○ Estimated arrival times.
  ○ Service disruptions. ○
  Crowding information.

A Python script using the scikit-learn library to create a machine learning model for predicting arrival times based on speed:

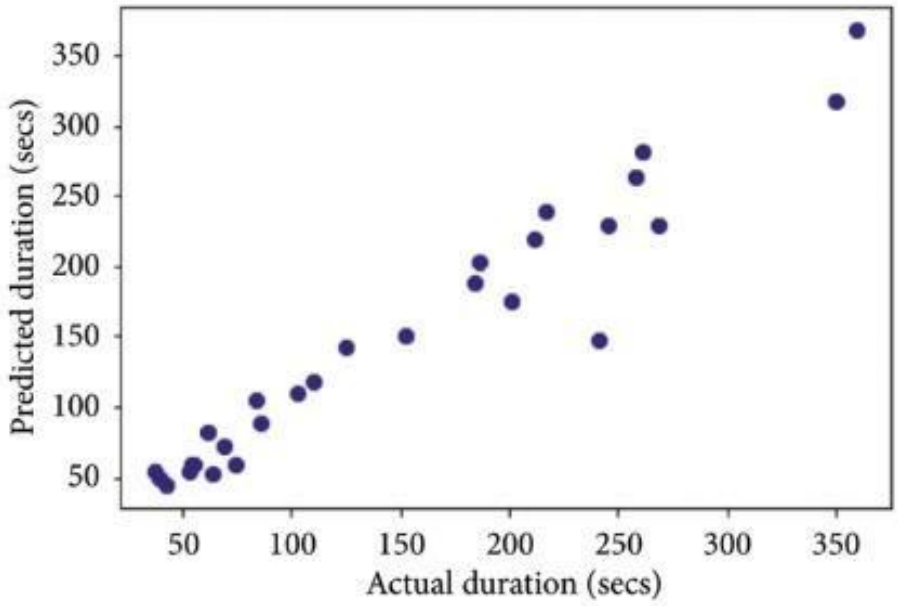
PYTHONSCRIPT CODE

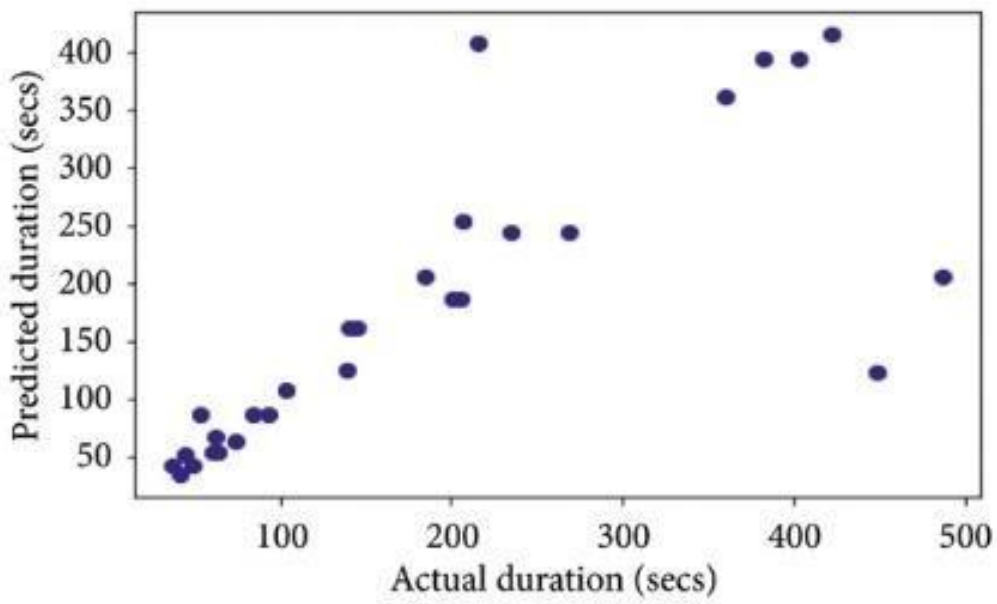
```
import time import random
from sklearn.ensemble import
RandomForestRegressor import
numpy as np
```

```
# Simulated IoT device data
class IoTDevice:     def
__init__(self, device_id):
self.device_id = device_id
self.latitude = 0.0
self.longitude = 0.0
        self.speed = 0.0

    def generate_data(self):
# Simulate GPS data          self.latitude +=
random.uniform(-0.001, 0.001)        self.longitude
+= random.uniform(-0.001,
0.001)          self.speed = random.uniform(5,
30)  #  Simulate speed in km/h

    def get_data(self):         return
{
            "device_id": self.device_id,
"latitude": self.latitude,
            "longitude": self.longitude,
            "speed": self.speed
        }

# Machine learning model for arrival time
prediction class ArrivalTimePredictor:
def __init__(self):
self.model =
RandomForestRegressor(n_estimators=100,
random_state=42)

    def fit(self, X, y):
```

```
        self.model.fit(X, y)

    def predict_arrival_time(self, data)
features =  np.array([data['speed']]).reshape(1, -
1)
estimated_time  =
self.model.predict(features)
        return estimated_time[0]

if __name__ == "__main__":     #
Create an IoT device
    device = IoTDevice(device_id="Bus123")

    predictor = ArrivalTimePredictor()

    # Simulated historical data      historical_speeds
= [random.uniform(5, 30) for _ in range(100)]
historical_times = [10 / speed for speed in
historical_speeds]


predictor.fit(np.array(historical_speeds).reshape(
1, 1), historical_times)

    while True:
        device.generate_data()
        data = device.get_data()

        # Predict arrival time         estimated_time
=
predictor.predict_arrival_time(data)
```

```
        print(f"Device: {data['device_id']}, Latitude:
{data['latitude']}, Longitude: {data['longitude']},
Speed: {data['speed']:.2f} km/h, Estimated Arrival
Time: {estimated_time:.2f} minutes")
time.sleep(60)  # Simulate data update every
minute
```




Predicted duration (secs)
400
350
300
250
200
150
100
50
100
200
300
400
500
Actual duration (secs)

Conclusion

The Public Transport Optimization Project utilizes IoT technology and a Python script to enhance the efficiency of public transportation systems, improve passenger experience, and reduce operational costs. By collecting and analyzing realtime data, the project empowers decision-makers to make informed choices that benefit both the transportation system and its passenger. we use a simple machine learning model, RandomForestRegressor, to predict arrival times based on speed.