

# PUBLIC TRANSPORT AND OPTIMIZATION

USING IN IOT

## Project Description

---

This project aims to optimize public transportation services through the deployment of IoT devices equipped with various sensors. The collected data will be processed using Python scripts to improve operational efficiency, provide real-time information to passengers, and enhance the overall public transportation experience.



# Sensors for Deployment:

---

- To achieve effective public transport optimization, several types of sensors can be deployed in vehicles and at transport infrastructure locations. Here are the key sensors and their applications:

# 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
- 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

# Passenger Counting Sensors:

**Application:** Used to monitor passenger loads on vehicles.

- **Benefits:** Allows for optimization of vehicle capacity, leading to better resource allocation and service planning.

**Temperature and Climate Sensors:**

**Application:** Monitoring and maintaining comfortable climate conditions inside vehicles.

**Benefits:** Ensures passenger comfort and safety by regulating heating, ventilation, and air conditioning systems.





# 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.



# Details

**Project Steps:** The project steps remain consistent with the previous outline. Here is how the sensors are integrated into the process:

**Project Planning:** Define the scope, objectives, budget, and timeline for the project.

**Select IoT Devices with Sensors:** Research and choose IoT devices that include the necessary sensors.

**Deployment of IoT Devices:** Install IoT devices with sensors in public transport vehicles and at key infrastructure locations.

**Data Collection:** IoT devices with sensors will collect data, including GPS, passenger counts, temperature, proximity, and camera footage.

**Data Processing and Storage:** Develop a data processing pipeline to clean and store the collected sensor data.

**Python Script Development:** Create Python scripts to analyze and process sensor data, implementing algorithms for optimization, real-time passenger information, and more.

**Real-time Passenger Information:** Develop a user interface for passengers to access real-time information based on sensor data.

**Optimization Algorithms:** Utilize sensor data to improve route efficiency, minimize delays, and optimize resource allocation .



# DETAILS

## Visualization and Reporting:

Create data visualization tools and generate reports for operational decision-making, leveraging the sensor data.

## Testing and Validation:

Thoroughly test the system, including sensor accuracy and script functionality.

# Python program

```
import serial

import requests

# Configure the serial port (update the port name accordingly)

ser = serial.Serial('COMX', 9600)

# ThingSpeak settings

thingspeak_api_key = '6EKT0ALDBXGG60Q1'

thingspeak_url = f'https://thingspeak.com/channels/2303456/private_show'

try:

    while True:

        # Read passenger count from Arduino

        passenger_count = ser.readline().strip().decode('utf-8')

        # Send data to ThingSpeak
```

Thank you



**THANK YOU!**



# Python program

```
response = requests.get(f'{thingspeak_url}&field1={passenger_count}'  
if response.status_code == 200:  
    print(f"Passenger count sent to ThingSpeak: {passenger_count}")  
    else:  
        print("Failed to send data to ThingSpeak")  
except KeyboardInterrupt:  
    ser.close()  
    print("Connection closed")
```