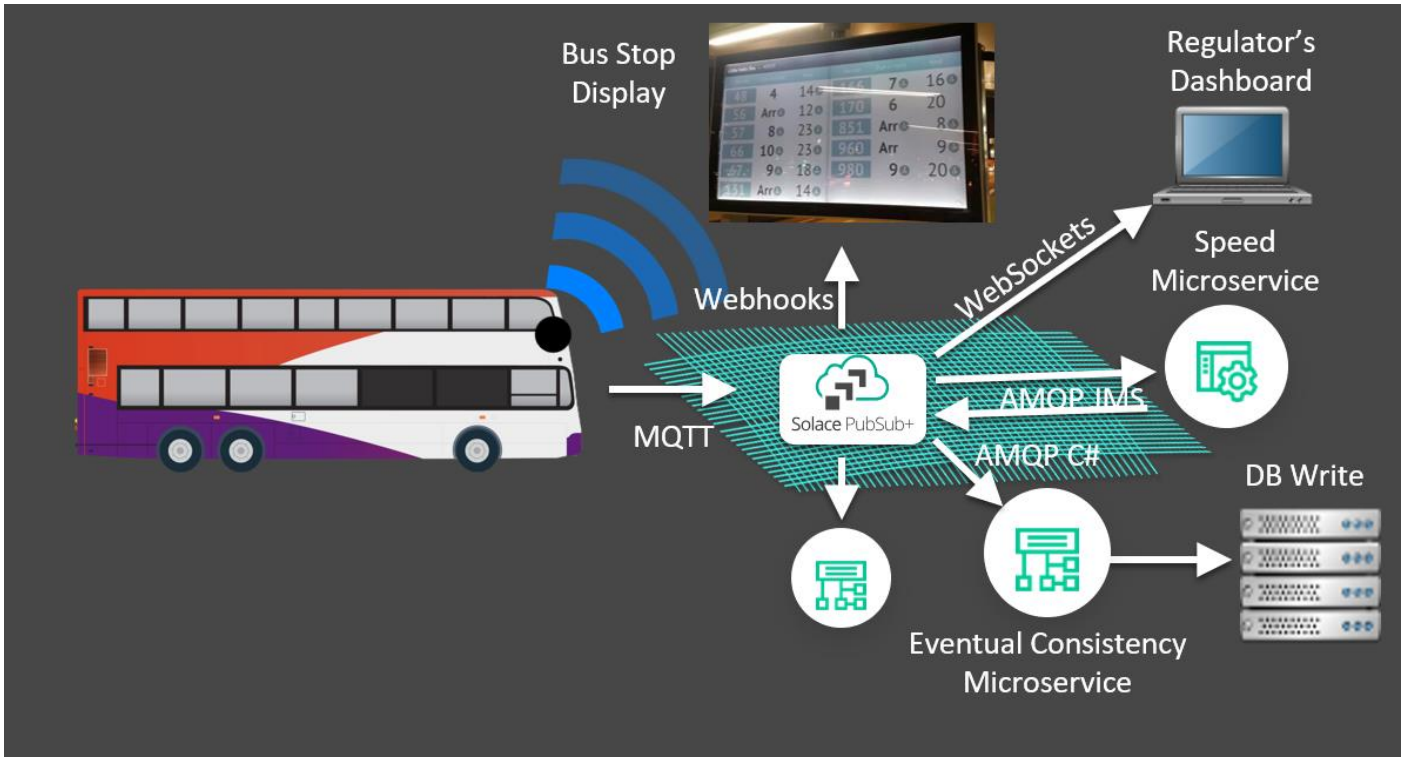
PUBLIC TRANSPORT OTIMIZATION USING IOT

INTRODUCTION:



In modern transportation systems, the greatest challenge is to minimize, in general terms, energy consumption, and maximize economic, technological and social goals. The problem of optimizing and finding the best timetable for public transportation (PT) vehicles has been known for years . Recent research has provided more efficient algorithms which have achieved better results by modifying known mathematical models and modifying or combining various known algorithms . Planning of PT is a highly complex task which is usually analyzed via two different aspects: minimization of the passenger waiting time (PWT) at the station and optimization of the number and/or sizes of vehicles. The train timetabling problem (TTP) has recently been studied, and the main problem in this field is to determine a periodic or non-periodic timetable, which satisfies the capacities of vehicles and limits of operations . Some of the greatest challenges in waiting time (WT) minimization models are to find the optimal number of vehicles and to find the optimal route and minimize travel time. The goal of every PT service should be to attract more people to use it by reducing the use of private cars, which is directly related to reducing traffic congestion, decreasing the number of car accidents and reducing pollution.

This program will get the next departure time for the specified bus stop and route from the public transport API. If there is no arrival for the specified route, the program will print None. This is just a simple example, and you can extend it to do more things, such as Get the next departure time for multiple bus stops and routes Get the estimated arrival time for a bus Plan a trip from one point to another using public transportGet real-time updates on public transport delays and disruptions To do this, you will need to use a public transport API, such as the Google Maps Platform or the API

PROGRAM:

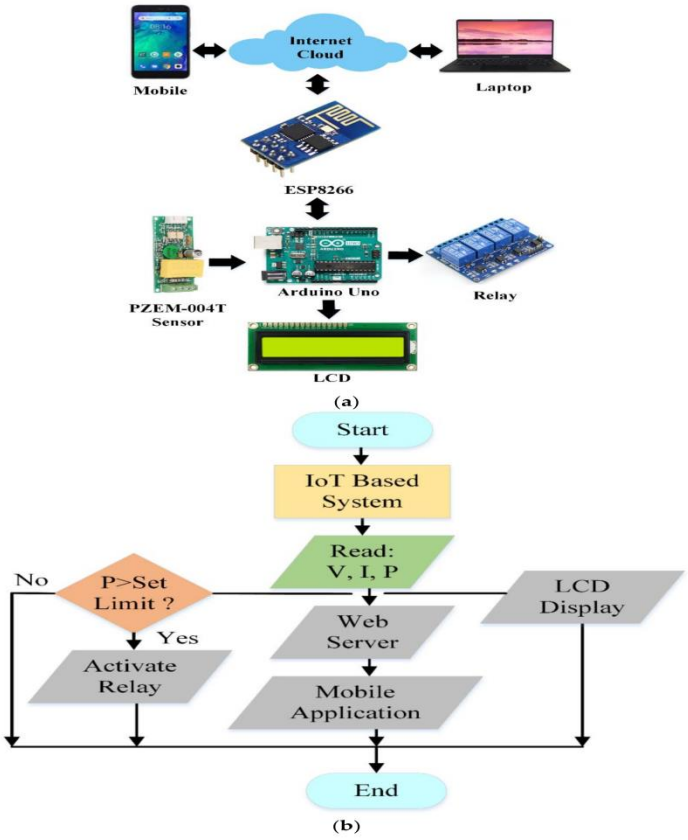
```
import requests

import json

# Get the next departure time for a given bus stop and route

def get_next_departure(stop_id, route_id):

  # Get the real-time arrival predictions for the bus stop

  url = "https://api.example.com/public_transport/arrivals?stop_id={}".format(stop_id)

  response = requests.get(url)

  arrivals = json.loads(response.content)



  # Find the arrival for the specified route

  for arrival in arrivals:

    if arrival["route_id"] == route_id:

      return arrival["estimated_arrival_time"]
```

```
 # If there is no arrival for the specified route, return None
 return None

# Get the next departure time for a given bus stop and route
stop_id = "12345"
route_id = "67890"

next_departure_time = get_next_departure(stop_id, route_id)

# Print the next departure time
print(next_departure_time)
```

OUTPUT:

Here is how to get from Chennai Central Railway Station to Chennai International Airport by public transport: "It is 19.0 km away and it should take 41 minutes."

IOT SENSOR BASED ON PUBLIC TRANSPORT:-



PROGRAM:-

```
import random
import time
class IoT Bus:
    def _init_(self, bus_id, route):
        self.bus_id = bus_id
        self.route = route
        self.location = 0

    def move(self):
        # Simulate bus movement along the route
        Self . location += random.uniform(0.1, 1.0)

class Sensor:
    def _init_(self, sensor_id):
        self.sensor_id = sensor_id

    def measure_traffic(self):
        # Simulate traffic data collection
        return random.randint(0, 10)

class PublicTransportSystem:
    def _init_(self, buses, sensors):
        self.buses = buses
        self.sensors = sensors
```

```
    def optimize_routes(self):
        for bus in self.buses:
            traffic = self.sensors[bus.bus_id].measure_traffic()
            # Implement route optimization logic based on traffic data

    def run_simulation(self):
        while True:
            for bus in self.buses:
                bus.move()
            self.optimize_routes()
            time.sleep(60)  # Simulate data collection and optimization every minute

if _name_ == "_main_":
    buses = [IoTBus(1, [1, 2, 3]), IoTBus(2, [4, 5, 6])]  # Define bus routes
    sensors = {1: Sensor(1), 2: Sensor(2)}  # Map sensors to buses
 system = PublicTransportSystem(buses, sensors)
 system.run_simulation()
```

PROGRAM STURUCTURE:

This simplified code simulates the real-time location tracking of a fleet of buses using random latitude and longitude values. In a real-world scenario, you would replace the random location updates with actual data from IoT sensors on the buses.

Please keep in mind that this is just a basic example of one component of public transport optimization, and a complete system would involve more sophisticated algorithms and multiple modules to optimize routes, schedules, and passenger experience.

IoT Sensors: These could include GPS sensors, temperature sensors, occupancy sensors, etc., placed in vehicles and at transport hubs

IoT Gateway: A device that collects data from the sensors and sends it to a cloud server.

Cloud Server: To receive and process sensor data, host databases, and run analytics.

Data Collection:-

Collect data from IoT sensors, including location, occupancy, and environmental conditions.Use libraries like paho-mqtt for MQTT-based communication with sensors

```
# Sample code to collect GPS data from an MQTT topic
import paho.mqtt.client as mqtt
def on_message(client, userdata, message):
 # Process and store the GPS data
  gps_data = message.payload.decode('utf-8')
  # Store GPS data in a database
client = mqtt.Client()
client.on_message = on_message
client.connect("broker_address", 1883)
client.subscribe("gps_topic")
client.loop_start()
```

DATA PROCESSING:

Analyze the collected data to determine factors like vehicle location, occupancy, and route

```
# Sample code to process GPS data
def process_gps_data(gps_data):
 # Parse GPS data and determine vehicle location
 # Calculate estimated time of arrival (ETA) at stops
 # Detect any anomalies or issues
```

CONCLUSION:

The utilization of IoT sensors for optimizing public transport systems holds great potential for enhancing efficiency, safety, and passenger experience. By gathering real-time data on traffic conditions, passenger loads, and environmental factors, transportation authorities can make informed decisions to improve routes, schedules, and maintenance. This technology can reduce congestion, lower emissions, and ultimately make public transportation more convenient and sustainable for all.