# Public transportation  Optimization-Phase3

## Aim

To developing a Python script on the IOT devices as per the public transportation optimization and as per the project requirements.

## Program:

```
from sqlalchemy import create_engine, Column, Integer, String, DateTime

from sqlalchemy.ext.declarative import declarative_base

from sqlalchemy.orm import sessionmaker

from datetime import datetime


# Create a database connection
```

```
engine = create_engine('postgresql://username:password@localhost/dbname')
Base = declarative_base()

# Define a data model
class TransportationData(Base):
    __tablename__ = 'transportation_data'
    id = Column(Integer, primary_key=True)
    vehicle_id = Column(String)
    timestamp = Column(DateTime)
    latitude = Column(Float)
    longitude = Column(Float)

# Create tables
Base.metadata.create_all(engine)
```

```
# Insert data into the database
Session = sessionmaker(bind=engine)
session = Session()

data = TransportationData(vehicle_id='123', timestamp=datetime.now(), latitude=12.345, longitude=67.890)
session.add(data)
session.commit()
from ortools.constraint_solver import routing_enums_pb2
from ortools.constraint_solver import pywrapcp

def solve_routing_problem(data):
    manager = pywrapcp.RoutingIndexManager(len(data['dist_matrix']), 1)
    routing = pywrapcp.RoutingModel(manager)
    def distance_callback(from_index, to_index):
        return
data['dist_matrix'][manager.IndexToNode(from_index)][
manager.IndexToNode(to_index)]


    transit_callback_index =
routing.RegisterTransitCallback(distance_callback)


routing.SetArcCostEvaluatorOfAllVehicles(transit_callba
ck_index)


    search_parameters =
pywrapcp.DefaultRoutingSearchParameters()
    search_parameters.first_solution_strategy = (


routing_enums_pb2.FirstSolutionStrategy.PATH_CHEA
PEST_ARC)


    solution =
routing.SolveWithParameters(search_parameters)
```

```
    if solution:
        # Process the solution
        ...

# Define your distance matrix and data

data = {
    'dist_matrix': [
        [0, 10, 20],
        [10, 0, 15],
        [20, 15, 0]
    ],
}

solve_routing_problem(data)
```

## THEORY

This project could involve various components, including data collection from IOT devices, data processing, and optimization algorithms. Below, I'll outline the main components and provide code examples for each step.

1. **Data Collection (IOT Devices):**

   - You can use IOT devices, such as GPS trackers, sensors, and cameras, to collect data about public transportation vehicles, passenger load, and traffic conditions.

2. **Data Processing and Storage:**

   - Use a database to store the collected data for analysis. You can use databases like MySQL, PostgreSQL, or NoSQL databases like MongoDB.

Example code to store data in a PostgreSQL database using Python and SQLAlchemy:

pyt