Phase 5: Project Demonstration & Documentation

COLLEGE CODE :3105  
COLLEGE NAME : DHANALAKSHMI SRINIVASAN COLLEGE OF ENGINEERING AND TECHNOLOGY  
DEPARTMENT:BE.COMPUTER SCIENCE ENGINEERING  
STUDENT NM-ID:9CEDD197710A97EA18344C2E7F54E22C  
ROLL NO:310523104087  
DATE:14/05/2025  
TECHNOLOGY-PROJECT NAME: Problem Definition & Design Thinking

SUBMITTED BY  
Your Name and Team Members:Y.MOHAMED NAFIS,S.MUTHUCHEZHAYAN,A.RAGUL BALA,K.MANI MARAN,S.KEERTHI VARMAN

# Title: Enhancing Public Transportation through Smart Route Optimization

## Abstract:

This project leverages AI and optimization algorithms to improve public transportation efficiency through smart route planning. The system analyzes real-time traffic data, passenger density, and schedules to dynamically recommend optimal bus/train routes. It aims to minimize travel time, reduce congestion, and improve commuter satisfaction. The platform includes a route recommendation engine, traffic analysis dashboard, and feedback loop for continual improvement.

## Index

(to be filled post-report compilation)

## 1. Project Demonstration

### Overview:

Demonstration of the smart transportation system highlighting:

- Real-time route optimization.  
- Passenger load prediction and schedule adjustment.  
- Traffic-aware routing engine.  
- Performance metrics: travel time savings, route accuracy.  
- User and operator interface.

## 2. Project Documentation

- System Architecture: Data flow diagrams showing data ingestion, model prediction, and UI interaction.  
- Code Documentation: Explanations of optimization algorithms and interface logic.  
- User Guide: How commuters and transit authorities interact with the system.  
- Admin Guide: System monitoring, updates, and maintenance.  
- Testing Reports: Simulation results showing reduced delays and optimized schedules.

## 3. Feedback and Final Adjustments

- Collect feedback from simulated environments or pilot users.  
- Fine-tune optimization thresholds and rerouting triggers.

## 4. Final Project Report Submission

- Summarizes objectives, technical approach, testing results, and deployment feasibility.

## 5. Project Handover and Future Work

- Suggestions: integrate GPS-based live data, mobile app support, and ML-based demand forecasting.  
- Handover includes all documentation and modular codebase.

## Appendix: Python Code Implementation

import networkx as nx  
import heapq  
  
# Graph-based route optimization  
def build\_city\_graph():  
 G = nx.Graph()  
 # Sample city graph with distances (nodes are bus stops or intersections)  
 edges = [  
 ("A", "B", 5), ("A", "C", 10), ("B", "D", 3), ("C", "D", 4),  
 ("D", "E", 8), ("C", "F", 7), ("F", "E", 2)  
 ]  
 for u, v, w in edges:  
 G.add\_edge(u, v, weight=w)  
 return G  
  
def shortest\_path(graph, start, end):  
 try:  
 path = nx.dijkstra\_path(graph, start, end, weight='weight')  
 distance = nx.dijkstra\_path\_length(graph, start, end, weight='weight')  
 return path, distance  
 except nx.NetworkXNoPath:  
 return [], float('inf')  
  
def main():  
 graph = build\_city\_graph()  
 print("Welcome to Smart Public Transport Route Optimizer")  
 start = input("Enter starting point: ").strip().upper()  
 end = input("Enter destination: ").strip().upper()  
   
 print("\nCalculating optimal route...")  
 path, dist = shortest\_path(graph, start, end)  
   
 if path:  
 print(f"Optimal Route: {' -> '.join(path)}")  
 print(f"Estimated Distance: {dist} units")  
 else:  
 print("No available route between the given points.")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()