PUBLIC TRANSPORTATION ANALYSIS

# PROJECT OBJECTIVES:

The main objective of the public transportation analysis is to optimize public transport services by analyzing passenger data to enhance efficiency, reliability, and user satisfaction.

# DATASET LINK :

[**https://www.kaggle.com/datasets/rednivrug/unisys?select=20140711.CSV**](https://www.kaggle.com/datasets/rednivrug/unisys?select=20140711.CSV)

# ANALYTICAL APPROACH:

Utilizing advanced analytics, our approach involves extracting valuable insights from extensive public transport data. By employing statistical methods, machine learning algorithms, and data visualization techniques, we aim to understand passenger behavior, optimize routes, and improve service reliability. This data-driven approach enables us to make informed decisions, enhance operational efficiency, and elevate the overall commuter experience, leading to a more streamlined and responsive public transport system.

1. **DATA COLLECTION:** Efficient data collection is the backbone of our public transport analysis. We gather real-time and historical data on passenger counts, travel patterns, and service usage. Utilizing cutting-edge technologies such as sensors, GPS, and ticketing systems, we ensure a robust data foundation. This meticulous data collection enables us to derive actionable insights, leading to informed decisions and enhanced public transport services.
2. **DATA CLEANING:** Raw data is cleaned to remove inconsistencies and missing values to ensure data quality and accuracy.
3. **EXPLORATORY DATA ANALYSIS (EDA):** EDA is done to identify trends, correlations and anomalies in the data. This step helps to understand the characteristics of the data.

# DISPLAY METHOD:

The project uses several visualization techniques, including:

1. **TIME SERIES PLOT:**

Time series plots display data points over a specific time interval. In public transport analysis, they can show trends in passenger numbers, boarding and alighting patterns, or service reliability metrics (such as on-time performance) over time.

1. **HEAT MAP:**

Heatmaps are effective for visualizing passenger density and demand across different routes and stops.

1. **NETWORK GRAPH:**

Network graphs depict the connections between stops and routes. Nodes represent stops, and edges indicate routes connecting them.

# IMPLEMENTATION CODE:

Python is used for data analysis and visualization. Libraries such as Pandas, Matplotlib, Seaborn and Folium are used for data manipulation and visualization. Jupiter notebook is used for code documents and analysis

# CONCEPT AND CONCLUSION:

Public transport analysis involves leveraging data to optimize routes, enhance scheduling, and improve commuter experiences. By employing statistical methods and data visualization techniques, transport authorities gain valuable insights into passenger behavior, enabling them to make informed decisions for efficient and user-friendly transit systems.

In conclusion, data-driven public transport analysis is instrumental in creating responsive and streamlined transit networks. By understanding commuter patterns, identifying operational inefficiencies, and implementing targeted improvements, cities can build more accessible, reliable, and passenger-centric public transport systems. These enhancements lead to increased rider satisfaction, reduced congestion, and a sustainable urban transportation ecosystem.

# CODING:

# # Importing necessary libraries

# import pandas as pd

# import matplotlib.pyplot as plt

# import seaborn as sns

# # Load the dataset (assuming it's in CSV format)

# data\_url = '/content/20140711.CSV'

# df = pd.read\_csv(data\_url)

# # Display the first few rows of the dataset

# print("Sample Data:")

# print(df.head())

# # Summary statistics of numerical columns

# print("\nSummary Statistics:")

# print(df[['NumberOfBoardings']].describe())

# # Visualization: Bar chart for NumberOfBoardings vs StopName

# plt.figure(figsize=(12, 6))

# sns.barplot(x='StopName', y='NumberOfBoardings', data=df)

# plt.title('Number of Boardings at Each Stop')

# plt.xticks(rotation=90)

# plt.xlabel('Stop Names')

# plt.ylabel('Number of Boardings')

# plt.show()

# # Visualization: Line chart for NumberOfBoardings trend over Weeks

# plt.figure(figsize=(12, 6))

# sns.lineplot(x='WeekBeginning', y='NumberOfBoardings', data=df)

# plt.title('Number of Boardings Trend Over Weeks')

# plt.xticks(rotation=45)

# plt.xlabel('Week Beginning')

# plt.ylabel('Number of Boardings')

# plt.show()

# End of code

# OUTPUT:



