# TECHNICAL REPORT

## GPS TOLL BASED SYSTEM SIMULATION USING PYTHON

### TEAM NAME : CODE COMMANDOS

### DATE OF SUBMISSION : 15/07/2023


## Abstract

This report presents a GPS toll-based system simulation using Python, aimed at modelling the dynamics of vehicle movement, toll zone interactions, and toll charge calculations. The simulation framework incorporates various Python libraries to manage geospatial data, simulate vehicle movements, calculate distances, and visualize results. The goal is to provide a comprehensive understanding of how such systems can be implemented and analysed for efficiency and accuracy.


## Introduction

The integration of GPS technology into modern transportation systems has revolutionized traffic management and toll collection. Traditional toll collection methods, such as manual toll booths, have significant drawbacks including traffic congestion, higher operational costs, and increased chances of human error. To address these issues, GPS-based toll systems have been developed, allowing for more efficient, accurate, and seamless toll collection processes.

In a GPS toll-based system, vehicles are equipped with GPS devices that track their movement in real-time. As vehicles travel along roads and highways, their GPS coordinates are continuously monitored and recorded. When a vehicle enters a predefined toll zone, the system calculates the toll charges based on the distance travelled within that zone or the number of zones crossed. This automated process eliminates the need for manual toll booths, reducing congestion and operational costs while ensuring accurate toll collection.

The objective of this project is to simulate a GPS toll-based system using Python. This simulation aims to replicate real-world scenarios where vehicles move along predefined routes, cross multiple toll zones, and incur toll charges based on their travel. By utilizing various Python libraries and frameworks, the simulation provides a comprehensive view of how GPS toll-based systems operate.

# Motivation Behind the Project

The primary motivation behind this project is to explore the practical application of GPS technology in toll-based systems, addressing inefficiencies and inaccuracies in traditional toll collection methods. By simulating the entire process from vehicle movement to toll payment, the project aims to demonstrate the feasibility and effectiveness of using Python for such simulations. This comprehensive approach helps in identifying and understanding the challenges, potential bottlenecks, and areas for improvement in real-world toll systems. Additionally, the project seeks to showcase how Python's robust libraries and frameworks can be leveraged to create accurate and efficient simulations, paving the way for future advancements in toll-based transportation management.

# Data Source

The simulation uses predefined routes with GPS coordinates to model vehicle movements along specific paths. Toll zones are established using geospatial coordinates, allowing the system to track when a vehicle enters or exits these zones. The interactions between vehicle paths and toll zones are analysed to ensure accurate toll charge calculations based on the distance travelled within the zones or the number of zones crossed. The primary data sources include:

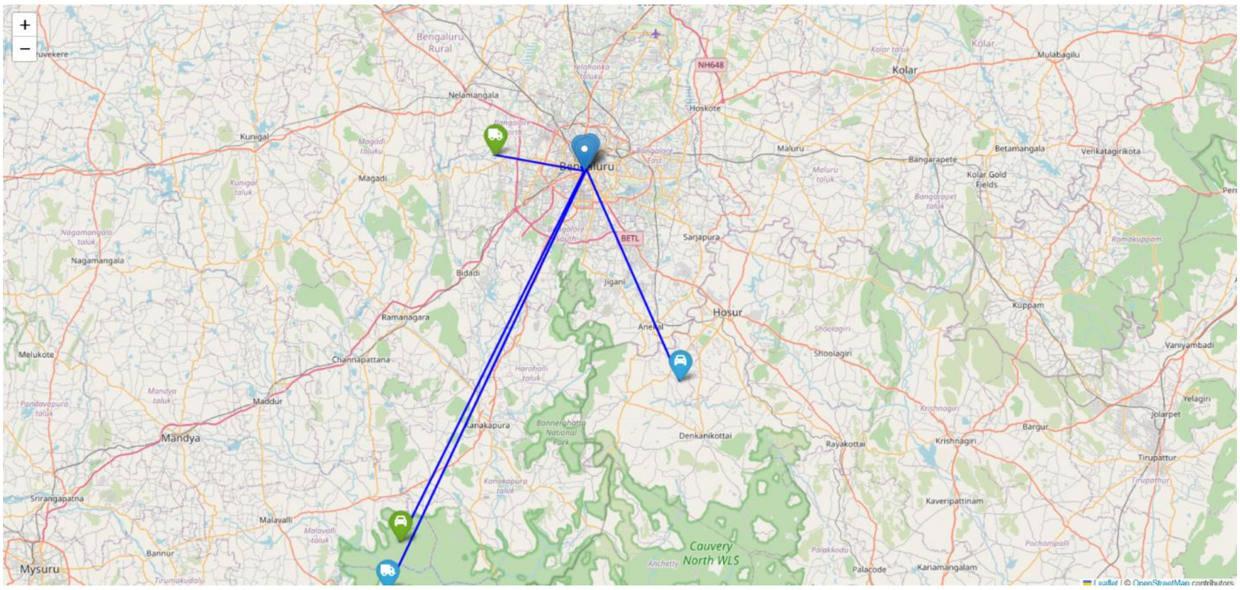
- GPS coordinates for vehicle routes
- Geospatial data for toll zone definitions
- Predefined rates for toll charge calculations

# Work

The simulation involves several key components and workflows:

1. **Vehicle Movement Simulation:**
   - Vehicles are simulated to move along predefined routes with GPS coordinates. This is achieved using the SimPy framework for event-driven simulation.
2. **Toll Zone Definition:**
   - Toll zones are defined using GPS coordinates, leveraging geopandas and shapely for geospatial analysis.
3. **Distance Calculation:**
   - The distance travelled by each vehicle within toll zones is calculated using the geopy library.
4. **Toll Calculation:**
   - Toll charges are computed based on the distance travelled within toll zones or the number of zones crossed. Predefined rates are used for these calculations.
5. **Payment Simulation:**
   - The process of deducting toll charges from user accounts is simulated, demonstrating the financial transactions involved.
6. **Visualization:**
   - The simulation results, including vehicle movements and toll zone locations, are visualized using matplotlib and folium.

# Result




```
Alert: Vehicle 1 has passed through a toll gate.
Alert: Vehicle 2 has entered Toll Gate 1
Alert: Vehicle 2 has entered Toll Gate 2
Alert: Vehicle 2 has entered Toll Gate 3
Alert: Vehicle 2 has entered Toll Gate 4
Vehicle 2: Distance Traveled = 6.00 km, Toll = 100.00 Rs, User Balance = 6400.00 Rs
Alert: Vehicle 2 has passed through a toll gate.
Alert: Vehicle 1 has entered Toll Gate 1
Alert: Vehicle 1 has entered Toll Gate 2
Alert: Vehicle 1 has entered Toll Gate 3
Vehicle 1: Distance Traveled = 7.00 km, Toll = 60.00 Rs, User Balance = 5580.00 Rs
Alert: Vehicle 1 has passed through a toll gate.
Alert: Vehicle 2 has entered Toll Gate 1
Alert: Vehicle 2 has entered Toll Gate 2
Alert: Vehicle 2 has entered Toll Gate 3
Alert: Vehicle 2 has entered Toll Gate 4
Vehicle 2: Distance Traveled = 7.00 km, Toll = 100.00 Rs, User Balance = 6300.00 Rs
Alert: Vehicle 2 has passed through a toll gate.
Alert: Vehicle 1 has entered Toll Gate 1
Alert: Vehicle 1 has entered Toll Gate 2
Alert: Vehicle 1 has entered Toll Gate 3
Vehicle 1: Distance Traveled = 8.00 km, Toll = 60.00 Rs, User Balance = 5520.00 Rs
Alert: Vehicle 1 has passed through a toll gate.
   Vehicle ID  Distance Traveled  Remaining Balance
0           1                8.0               5520
1           2                7.0               6300
```

The above image is a map with markers and a route drawn over a specific region. Here we have used Bangalore map. It has the following representations:

- **Markers:** Represent the locations of vehicles, toll booths, or other relevant points.
- **Route:** The blue line represents the route taken by a vehicle, indicating the path from one point to another.

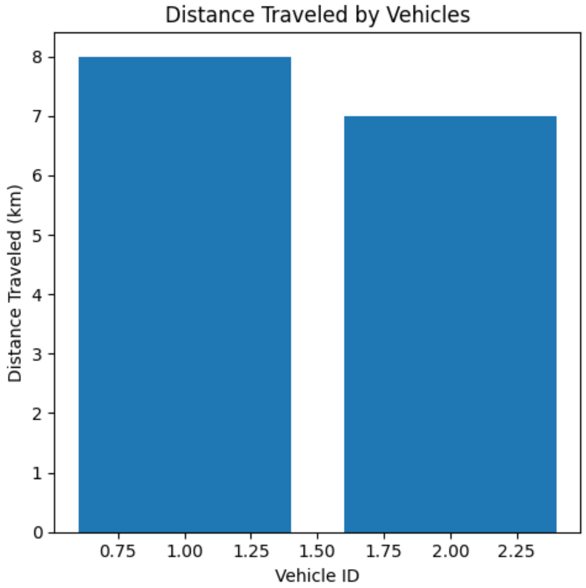
This map visualization is essential for understanding the geographical movement of vehicles in the simulation. It helps in:

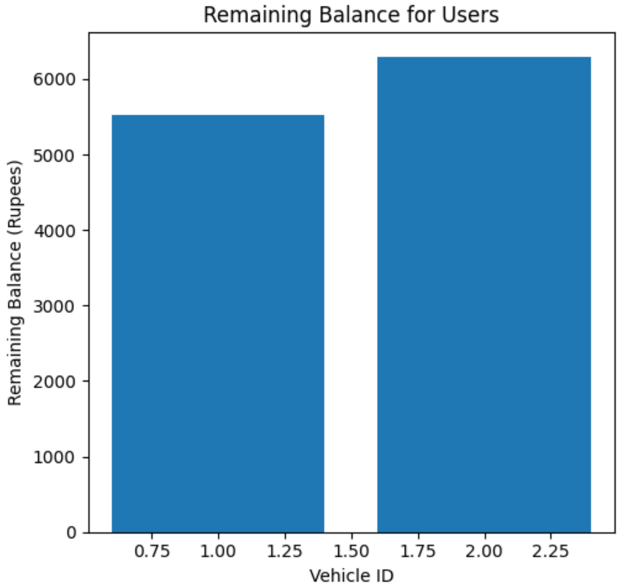
- **Tracking Vehicle Movements:** Monitoring how vehicles traverse through toll zones.
- **Defining Toll Zones:** Visualizing the boundaries and locations of toll zones.
- **Distance Calculation:** Using the visual route to aid in calculating the distance travelled within toll zones, which is a key factor in determining toll charges.

Below image describes the Visualization graphs. The first chart shows two bar charts representing different aspects of a GPS toll- based system simulation.

1. **Distance Travelled by Vehicles (Left Chart):**
    - **X-axis:** Vehicle IDs
    - **Y-axis:** Distance travelled in kilometres (km)
    - **Interpretation:** This chart indicates the total distance travelled by each vehicle. Vehicle ID 1 travelled the most distance, while Vehicle ID 2 travelled the least.
2. **Remaining Balance for Users (Right Chart):**
    - **X-axis:** Vehicle IDs
    - **Y-axis:** Remaining balance in Rupees
    - **Interpretation:** This chart shows the remaining balance in user accounts after deducting toll charges. Vehicle ID 2 has the highest remaining balance, while Vehicle ID 1 has the lowest.

These charts help visualize the relationship between the distance travelled and the remaining balance of the users, which is crucial for analysing the efficiency and financial aspects of the toll system.




# Links of the result:

Result model link :

https://ajayreddy19.github.io/Intel/

GitHub link :

https://github.com/AjayReddy19/Intel