Project Report

On

GPS Toll based System simulation using Python

For

Intel® Unnati Training Program 2024

Submitted by

Students	**College Mentor**
MOHAMMED HARRIS J A	REVATHI DHANARAJ
SUKITHA D	revathi15yadhav@gmail.com
SHRUTI RANI EDLA	
NISHALINI K R	

KIT-KALAIGNARKARUNANIDHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

Coimbatore, Tamil Nadu, India

Table Of Contents

Objective

- Simulate toll calculation using GPS data.
- Account for dynamic factors: traffic congestion, vehicle type, distance travelled.
- Visualize toll zones, vehicle routes, and traffic conditions on an interactive map.

Abstract

The project is a dynamic toll pricing and management system implemented using Python. It calculates toll charges based on several factors such as toll zone, traffic congestion, vehicle type, and distance travelled. It also calculates penalties for speeding and toll waivers based on vehicle status at toll zones. Furthermore, it visualizes toll zones, vehicle routes, and traffic congestion on an interactive map.

Introduction

The simulation of a GPS-based toll system involves various components and steps to accurately model vehicle movements, toll zone crossings, and the corresponding toll calculations. This report provides a detailed overview of the components, implementation, challenges, and benefits of such a system.

Components of the Simulation

Vehicle Movement Simulation

Simulate vehicles moving along predefined routes with GPS coordinates using SimPy for event-driven simulation.

Toll Zone Definition

Define toll zones or points with GPS coordinates using geopandas and shapely for geospatial analysis.

Distance Calculation

Calculate the distance traveled by each vehicle within toll zones using geopy.

Toll Calculation

Compute toll charges based on distance travelled or zones passed, using predefined rates.

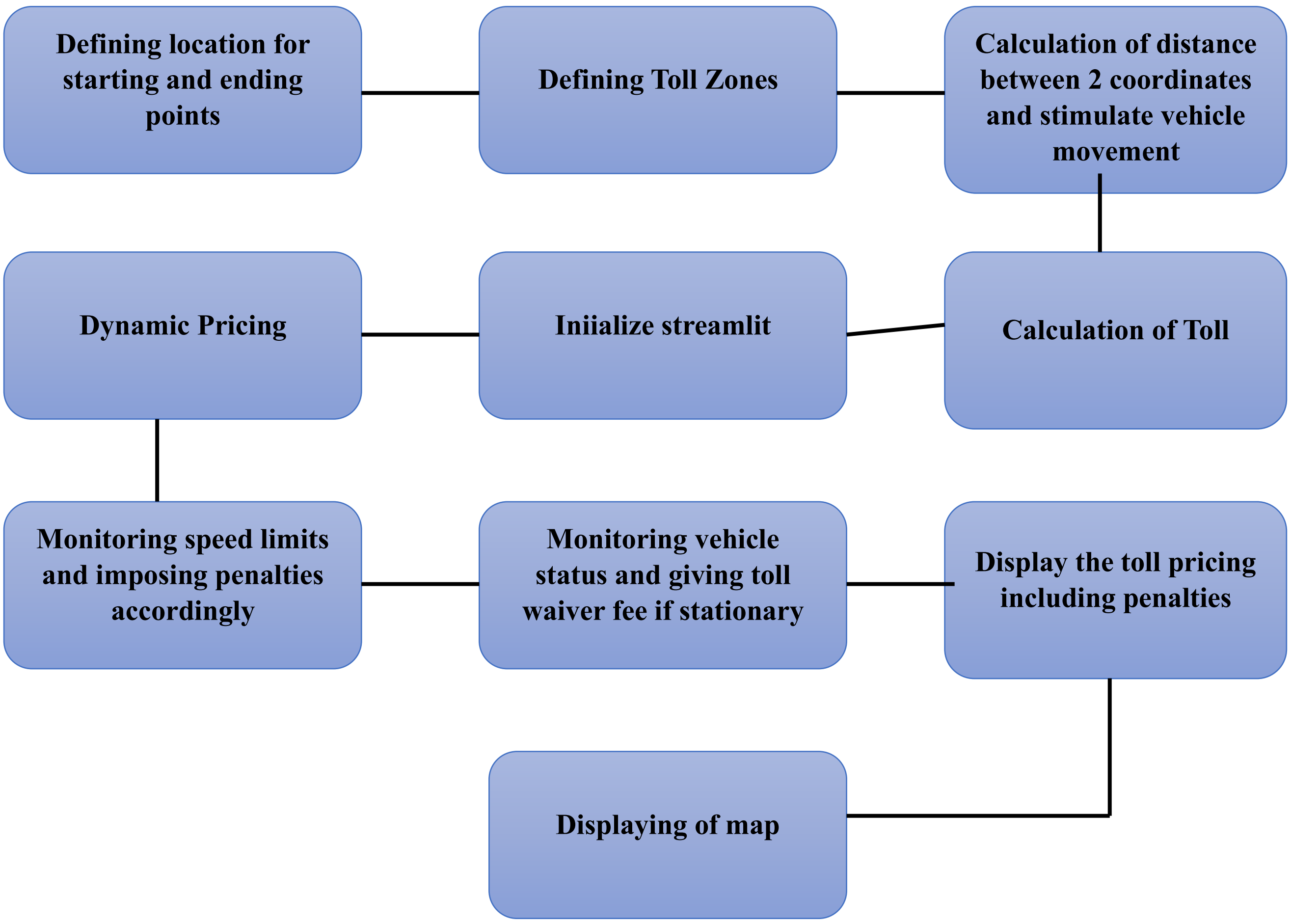
Payment Simulation

Simulate the process of deducting toll charges from user accounts.

Python Libraries and Frameworks

- **Simulation Frameworks**: SimPy
- **Geospatial Analysis**: geopandas, shapely
- **Distance Calculation**: geopy
- **Data Handling**: pandas
- **Visualization**: folium

Flow of project


Defining location for starting and ending points
Defining Toll Zones
Calculation of distance between 2 coordinates and stimulate vehicle movement
Calculation of Toll
Iniialize streamlit
Dynamic Pricing
Monitoring speed limits and imposing penalties accordingly
Monitoring vehicle status and giving toll waiver fee if stationary
Display the toll pricing including penalties
Displaying of map

Detailed Explanation

Defining locations for starting and ending points

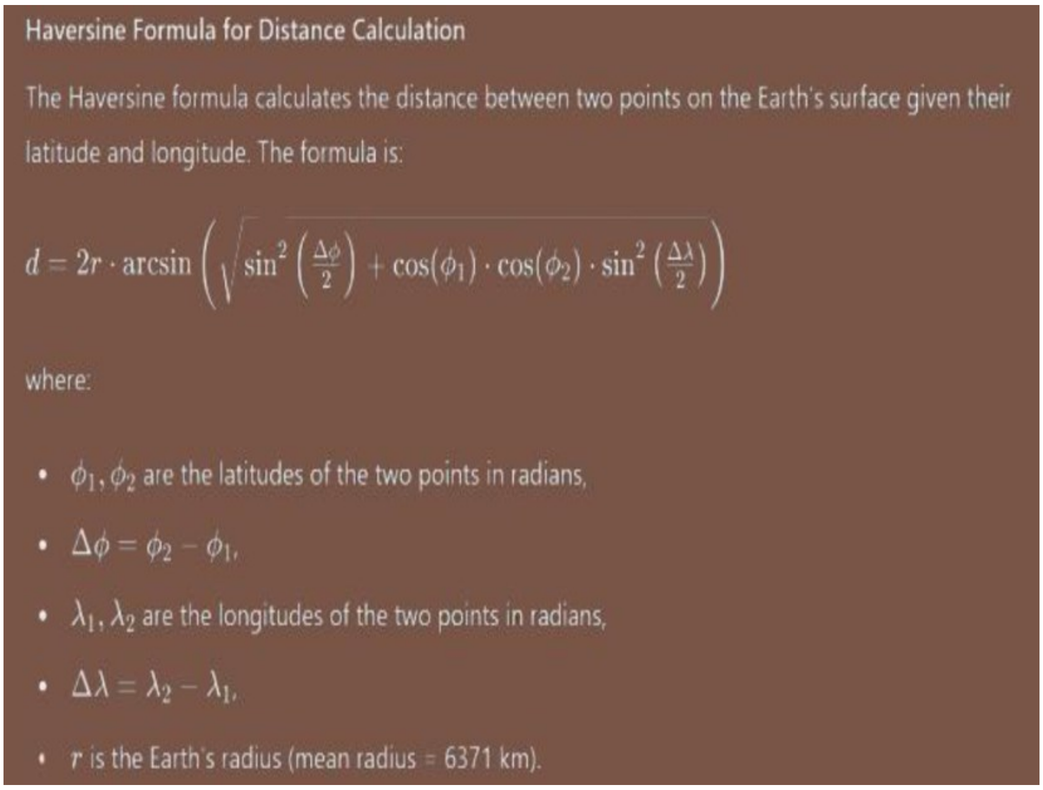
The variable “locations_coords” is a dictionary that maps city names (“Chennai”, “Coimbatore”, “Madurai”, “Salem”) to their respective coordinates as tuples of latitude and longitude.

Defining Toll Zones

The variable “toll_zones” is a dictionary that defines four toll zones (“Toll Zone 1”, “Toll Zone 2”, “Toll Zone 3”, “Toll Zone 4”) using “shapely.geometry.Polygon” objects with coordinates specifying the vertices of each polygon.

Calculation of distance between two coordinates and stimulate vehicle movement

The“calculate_distance” function calculates the distance between two geographical coordinates using the Haversine formula.



Haversine Formula for Distance Calculation

The Haversine formula calculates the distance between two points on the Earth's surface given their latitude and longitude. The formula is:

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cdot\cos(\phi_2)\cdot\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

where:

- ϕ_1, ϕ_2 are the latitudes of the two points in radians,
- $\Delta\phi = \phi_2 - \phi_1$,
- λ_1, λ_2 are the longitudes of the two points in radians,
- $\Delta\lambda = \lambda_2 - \lambda_1$,
- r is the Earth's radius (mean radius = 6371 km).

The coordinates are given as tuples of latitude and longitude, and the function returns the distance in kilometres. The "simulate_vehicle_movement" function calculates the total distance between two locations, identifies the toll zones the route intersects, and computes the distances travelled within each intersected toll zone.

Calculation of Toll

The "calculate_toll" function computes the total toll cost for a vehicle based on the distances travelled within different toll zones and a dynamic price per kilometre, returning the total toll and detailed breakdowns for each zone.

Initialize streamlit

Streamlit is an open-source framework for creating interactive, data-driven web applications. It contains dropdown menus for selecting the start and end locations from the predefined coordinates, and a dropdown menu for selecting the vehicle type from a given list. The title given is "GPS Toll Based System Simulation".

Dynamic Pricing

Dynamic pricing in the GPS-based toll system adjusts the toll rates based on the congestion levels at the start location. Congestion levels are randomly generated for each location, representing the percentage of road congestion. The pricing per kilometre is calculated based on the congestion level and the vehicle type.

Formula and Calculation

The dynamic pricing factor for each vehicle type is calculated using the following formula:

$$\textbf{Price per km} = \textbf{Congestion Level} \times \textbf{Base Rate} \qquad \text{(eqn. 1)}$$

where the base rates are defined as follows:

Car: 0.25 INR/km

Truck: 0.50 INR/km

SUV: 0.35 INR/km

Ambulance: 0.00 INR/km

For example, if the congestion level at the start location is 0.4 (40% congestion), the dynamic pricing factors for each vehicle type would be:

Car: $0.4 \times 0.25 = 0.10$ INR/km

Truck: $0.4 \times 0.50 = 0.20$ INR/km

SUV: $0.4 \times 0.35 = 0.14$ INR/km

Ambulance: $0.4 \times 0.00 = 0.00$ INR/km

Toll Calculation

The toll for each toll zone passed is calculated by multiplying the distance travelled within the zone by the dynamic pricing factor, adjusted by a zone specific multiplier:

Dynamic Amount = Price per km × Distance × Zone Multiplier **(eqn. 2)**

Zone multipliers are predefined as follows:

Toll Zone 1: 1.55

Toll Zone 2: 1.25

Toll Zone 3: 1.35

Toll Zone 4: 1.45

For example, if a Car travels 5 km within Toll Zone 1, the dynamic amount would be:

Dynamic Amount = $0.10 \times 5 \times 1.55 = 0.775$ INR

Monitoring Speed Limits and Imposing Penalties

The system monitors vehicle speeds in different sections of the route and imposes penalties if the vehicle exceeds the speed limit. Speed limits are predefined for different vehicle types and sections.

Penalty Calculation

If the vehicle's speed exceeds the speed limit in any section, a penalty of 500 INR is imposed for each section where the limit is breached.

For example, suppose the speed limits for a Car in three sections are as follows:

Section 1: 80 km/h

Section 2: 100 km/h

Section 3: 120 km/h

If the actual speeds in these sections are:

Section 1: 90 km/h

Section 2: 110 km/h

Section 3: 115 km/h

The penalties would be calculated as follows:
Section 1: Speeding, Penalty = 500 INR

Section 2: Speeding, Penalty = 500 INR
Section 3: Within limit, No Penalty

Total penalty = 500 + 500 = 1000 INR

Toll waiver fee

If toll zones are passed along the route, it calculates the toll for each zone based on dynamic pricing and displays payment vendors for each toll zone, allowing the user to select a payment vendor. It also displays the distance travelled and the amount deducted for each toll zone. Additionally, it randomly selects the vehicle status as "Moving" or "Stationary" and offers a toll waiver of 150 INR if the vehicle is stationary.

Final Amount Deducted

The final toll amount includes the base toll, penalties for speeding, and any waivers (e.g., for vehicles being stationary). The formula for the final toll amount is:

Final Toll Amount = Total Toll + Penalty Amount - Toll Waiver (eqn. 3)

For example, if the total toll is 1000 INR, the penalty amount is 1000 INR, and the toll waiver is 150 INR:

Final Toll Amount = 1000 + 1000 - 150 = 1850 INR

If the calculated final toll amount is negative, it indicates an amount to be credited to the user.

Display the Toll Pricing including penalties

This section calculates the final toll amount, taking into account the total distance travelled, the total toll cost, any penalties incurred for speeding, and toll waiver fees. It displays the total distance travelled, total toll amount, penalty amount for speeding violations, toll waiver fee, and the final toll amount including penalties. If the final toll amount is negative, it indicates that the user is eligible for a refund, and it displays the amount to be credited instead.

Displaying of Map

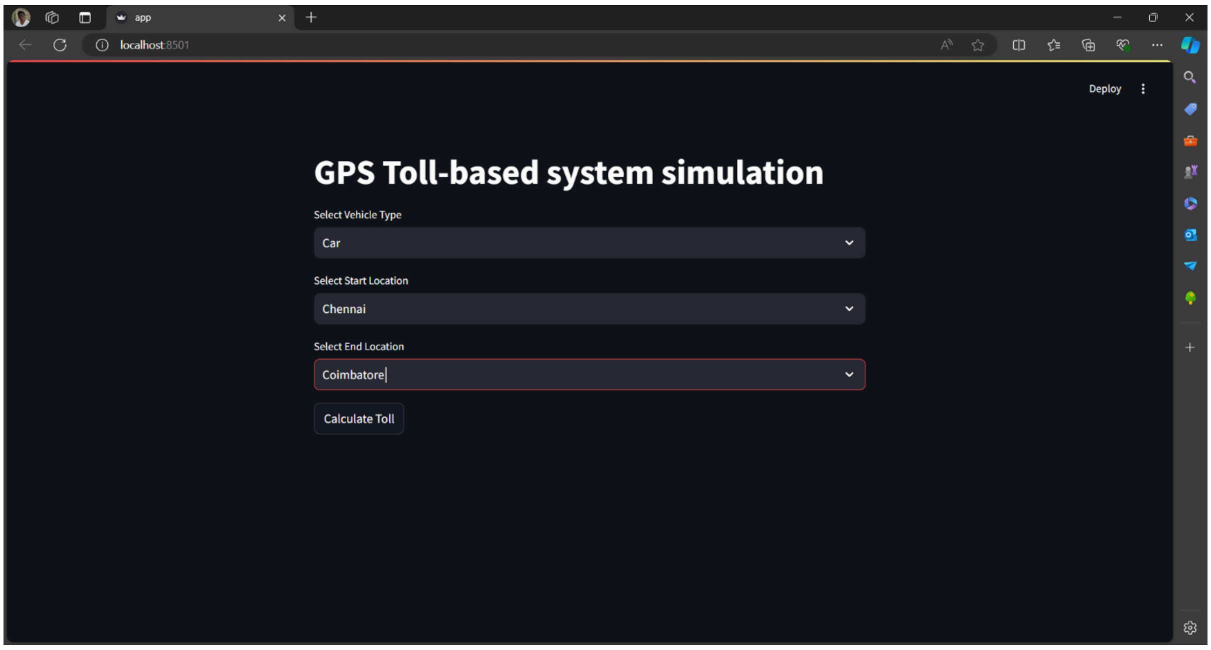
This part of the code displays a map using the Folium library, showing markers for the start and end locations, a blue polyline representing the route between them, and polygons representing toll zones. Additionally, it generates a heatmap overlay indicating congestion levels at various locations based on random values assigned to each location.

The "folium.Map" function initializes the map with the centre coordinates calculated as the midpoint between the start and end locations. Markers for the start and end locations are added using "folium.Marker", with green and red icons, respectively. The route between the start and end locations is drawn as a polyline using "folium.PolyLine".

For each toll zone, a GeoJSON object is created and added to the map using "folium.GeoJson", with a marker at the centre of each zone. The heatmap is generated using "folium.plugins.HeatMap", with data comprising latitude, longitude, and congestion level for each location.

Finally, the map is displayed in the Streamlit app using "st_html", with the height set to 500 pixels.

Glimpse of the Project



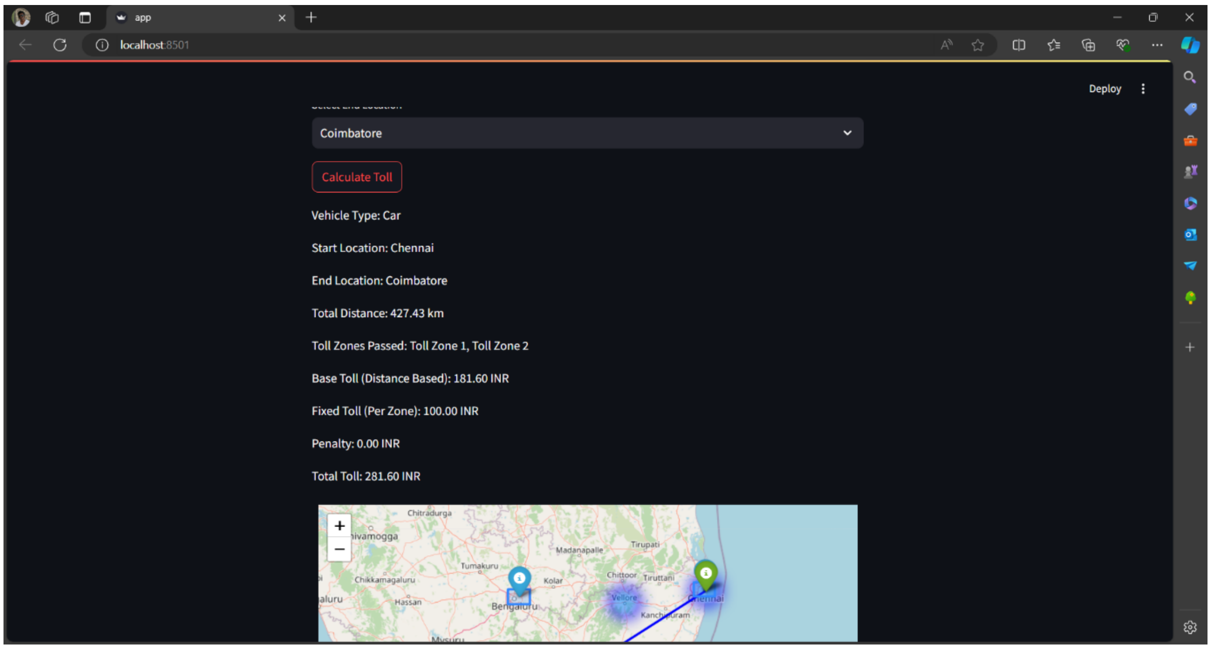
GPS Toll-based system simulation

Select Vehicle Type: Car

Select Start Location: Chennai

Select End Location: Coimbatore

Fig 1



Coimbatore

Vehicle Type: Car

Start Location: Chennai

End Location: Coimbatore

Total Distance: 427.43 km

Toll Zones Passed: Toll Zone 1, Toll Zone 2

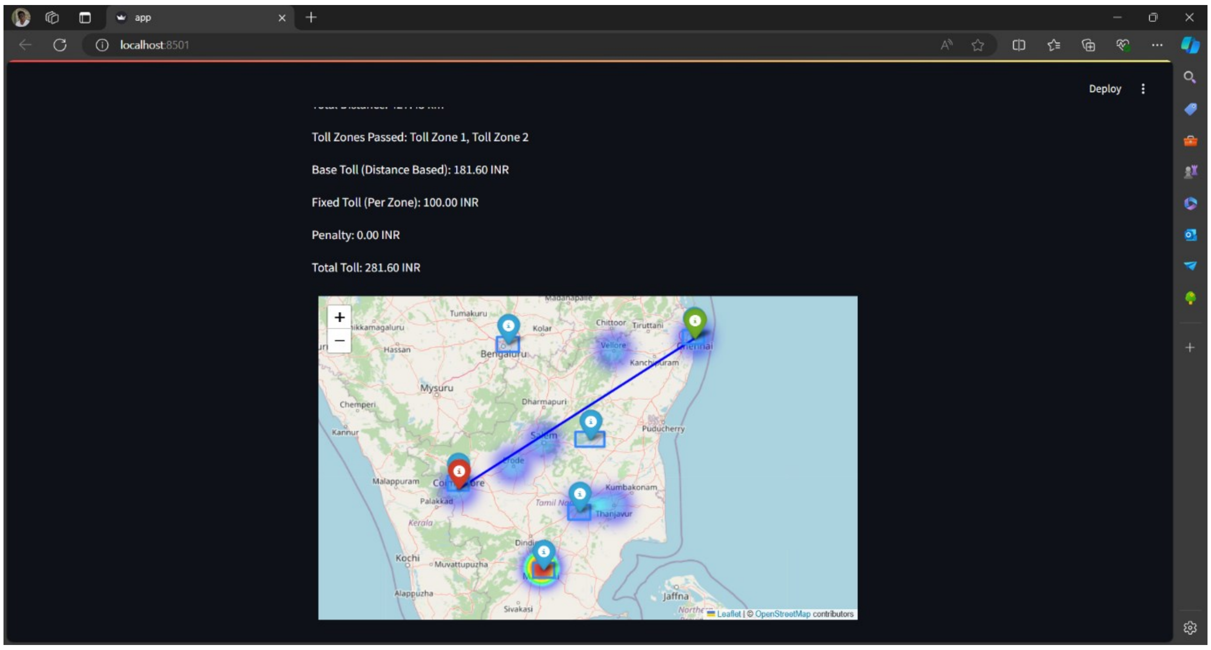
Base Toll (Distance Based): 181.60 INR

Fixed Toll (Per Zone): 100.00 INR

Penalty: 0.00 INR

Total Toll: 281.60 INR

Fig 2



```
Toll Zones Passed: Toll Zone 1, Toll Zone 2

Base Toll (Distance Based): 181.60 INR

Fixed Toll (Per Zone): 100.00 INR

Penalty: 0.00 INR

Total Toll: 281.60 INR
```

Fig 3

Benefits of GPS-based toll calculation

- **Real-time accuracy**: Provides accurate and up-to-date toll calculations, reducing errors and disputes.
- **Convenience**: Eliminates the need for manual toll collection.
- **Reduced congestion**: Helps reduce congestion at toll booths.
- **Increased transparency**: Provides a clear record of toll payments.
- **Improved customer experience**: Provides a seamless and hassle-free experience for drivers.

Future Scope

Here are some potential future enhancements for the app:

1. Use of weather data to adjust toll rates or provide alternate routes during adverse weather conditions. Also, addition of parallel non-toll roads.
2. Use of Google API to simulate vehicle based on the real route travelled.
3. Offering subscription-based models for frequent travellers, providing discounts or unlimited access for a fixed fee.
4. Addition of scenarios like road closures, accidents, and diversions.
5. Detailed analytics on toll road usage, peak travel times, and revenue generation.

6. Use of cloud-based architecture to handle a larger number of users and real-time data processing.
7. Expansion of app to cover more regions and countries, adapting to local tolling systems and regulations.
8. Addition of multiple languages to serve a global audience.

Implementing these enhancements will improve user experience and make the app more versatile, reliable, and widely used.

Conclusion

The dynamic toll pricing and zone management system presented in the project provides a comprehensive solution for calculating toll charges based on real-time factors and optimizing traffic flow. It uses Streamlit for user interaction and Folium for map visualization, offering an intuitive interface for users to plan their journeys and understand the associated costs effectively. Future enhancements could include real-time data integration for congestion updates and more advanced route optimization algorithms.