## PROBLEM STATEMENT 06:

# GPS TOLL BASED SYSTEM SIMULATION USING PYTHON

# ABSTRACT

This project is a GPS-based toll system simulation using Python. It includes vehicle movement in toll zones, distance calculation, toll computation and payment processing. The key Python libraries used are SimPy for event simulation, GeoPandas and Shapely for geographic data, GeoPy for distance calculations, Pandas for data handling, Matplotlib for visualization and Folium for interactive mapping. We get to practice programming and transportation systems and see real world applications of advanced Python libraries.

# Introduction

GPS technology has advanced so fast and is integrated into transportation systems, we manage and use road networks in a whole new way. One of the applications of this technology is in toll based systems which facilitates efficient and automated toll collection. This project aims to simulate a GPS based toll system using Python for first to fourth semester students.

The simulation will mimic real world scenarios where vehicles traverse toll zones, the system will calculate distance traveled, determine toll applicable and process payment. By using this simulation students will get practical experience in various aspects of transportation systems and programming and will reinforce their theoretical knowledge through hands-on application.

To achieve this the project uses:

1. SimPy - For discrete event simulation to model vehicle movement and payment process.
2. GeoPandas and Shapely - To handle and manipulate geographic data to define toll zone boundaries.
3. GeoPy - For geocoding and calculating distance between geographic points.
4. Pandas - For data manipulation and management.
5. Matplotlib - For data visualization.
6. Folium - To create interactive maps to visualize vehicle routes and toll zones.

The simulation has:

Vehicle Movement Simulation: Vehicle movement along routes with speed and stops.

Toll Zone Definition: GeoPandas and Shapely to define toll zone boundaries.

Distance Calculation: GeoPy to calculate distance within toll zones.

Toll Calculation: Toll charge based on distance and other factors.

Payment Simulation: SimPy to simulate payment.

# METHODOLOGY

## Objective

- Develop a simulation of a GPS-based toll system using Python.

## Libraries Used

- **SimPy:** For simulating vehicle movement and payment processes.
- **GeoPandas and Shapely:** To define toll zone boundaries.
- **GeoPy:** For calculating distances between points.
- **Pandas:** For managing data.
- **Matplotlib:** For visualising data.
- **Folium:** For interactive maps.

## Simulation Components

- **Vehicle Movement:** Simulate vehicles moving along routes.
- **Toll Zone Definition:** Define geographic boundaries for toll zones.
- **Distance Calculation:** Calculate distances traveled within toll zones.
- **Toll Calculation:** Determine toll charges based on distance.
- **Payment Simulation:** Simulate payment processes.

## Implementation Strategy

- Develop modular components for each simulation aspect.

- Integrate Python libraries for data processing and visualisation.
- Ensure components work together to simulate a complete toll system.

**Testing and Validation**

- Test to ensure simulation accuracy.
- Validate calculations of distances and toll charges.
- Address any errors to improve simulation reliability.

**Documentation**

- Document development process and methodology.
- Provide usage instructions and interpret simulation results.
- Summarise project objectives and future enhancements.

# CHALLENGES ENCOUNTERED

1. Accurately defining geographic boundaries for toll zones and ensuring precise detection of vehicle entry and exit.

2. Ensuring precise calculation of distances traveled within toll zones, considering varying routes and vehicle paths.

3. Accurately modeling the toll payment process, including handling different payment methods and transaction delays.

4. Managing large datasets for vehicle information, toll zones, and transactions, and ensuring seamless integration between components.

5. Creating clear visual representations of vehicle movements, toll zones, and simulation results.

6. Ensuring the simulation runs efficiently, especially with large datasets and complex calculations.

7. Identifying and resolving errors in a complex simulation environment.

# FUTURE PROSPECTS

The GPS-based toll system simulation project has several avenues for future development and enhancement:

1.Implement more sophisticated pricing strategies, such as dynamic pricing based on traffic conditions, time of day, or vehicle type.

2. Incorporate real-time traffic data to adjust toll calculations dynamically based on current road conditions and traffic congestion.

3.  Extend the simulation to cover larger geographic areas with varying toll zones and road types.

5. Utilize machine learning algorithms to predict traffic patterns, optimize toll pricing, and enhance route planning.

6. Develop a more interactive and user-friendly interface with advanced visualization features and real-time updates.

7. Explore integration with smart city infrastructure, such as vehicle-to-infrastructure (V2I) communication and automated toll collection systems.

# CONCLUSION

The developed simulation project of the GPS based toll system is creating a real-time mode of toll collection and vehicle movement. It helps in understanding the toll system better. Though the accuracy and data handling are the major disadvantageous constraints faced, this simulation project has opened many future research work hands in practicing python for transportation related simulations. With additional implementations like usage of real time traffic data, usage of complex form or advanced methods for pricing will enhance with more features making it a resourceful learning tool as well as facilitating significance in practical approach.

# REFERENCES

- https://realpython.com/simpy-simulating-with-python/
- https://geopandas.org/
- https://shapely.readthedocs.io/
- https://www.w3schools.com/python/pandas/pandas
- https://www.analyticsvidhya.com/blog/2021/10/introduction-to-matplotlib-using-python-for-beginners/

# RESULT

The GPS based toll system simulation could bring out several outcomes from the study. The simulation application simulated the real-time vehicle traffic in a city region with GPS performance support. With this simulation model, some GPS based toll system features and operations could be analyzed before its development as in real system implementation. The results showed that the GPS based toll system can create and configure the toll zones accurately. Vehicle entry and exit on a particular zone were detected correctly so that it generated accuracy in calculating appropriate vehicle fare or charge for passing through several consecutive linked zones. Calculation for total distance of each passing vehicle by activating related function procedure was precise to get accurate value of fares charged to vehicles driver or user for utilizing defined street segments road between two specific points of interest (POIs). Toll charge computation also compares closely with predetermined values taken from one regions on other alternative routes regions road distance covered as a preview region to another destination region’s road distance which is published officially via regional public service.