A COMPREHENSIVE REPORT ON GPS TOLL-BASED SYSTEM USING PYTHON

Submitted By

Team Name: BIT GEEKS

USN	NAME
1BI22CS003	ABHIRAAM S
1BI22CS034	BHUVAN B
1BI22CS054	GAGAN NAGANAD KENI
1BI22CS126	RANJAN P(TEAM LEAD)

In fulfillment of the project report for the

Intel Unnati- An Industrial Training Program - 2024

BANGALORE INSTITUTE OF TECHNOLOGY

Technology for Prosperity

JUNE, 2024

FACULTY MENTOR REVIEW

Our project has been reviewed by the Head of Department(HOD) of the Computer Science and Engineering Department Dr. Girija Mam and we have got very good feedback from her. She even suggested bringing this idea into the real world by building the actual product and implementing it in real-world scenarios. One interesting point is that We came to know about cloud technology integration and data management and also about how QR codes can be effectively deployed on the vehicle number plate and the idea of Regenerating the QR code from a bit of it.

Ma'am also suggested that if you are building this product then it should be economical and affordable so that all the citizens can utilize this technology being adopted in their vehicles.

Our Faculty Mentor Prof. Shruthi B Gowda Ma'am has motivated us throughout this project and we also got her suggestions stage-wise during our improvisation or progress. She even brought to our notice some situations in real-world scenarios where we have to keep the user/driver informed about the best possible route available for his destination with minimum toll charges.
She also advised us to go with the prototyping in the future and to enhance our scalability so that we can get to know how things work in real life. Our mentor suggested we ensure each user can generate a unique number plate and QR code. Prevent duplication of number plates and QR codes. Implement a robust verification system to handle unique assignments.

Reviewed By,

faculty mentor
[Shruthi B Gowda
Assistant Professor]

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have supported and guided me throughout the development of the project "GPS Toll Based Simulation Using Python."

Firstly, we would like to thank my mentor, Dr. Girija Ma'am, and Prof. Shruti Gowda Ma'am, for their invaluable guidance, insightful suggestions, and continuous encouragement. Their expertise and knowledge have been crucial in shaping this project.

We are also grateful to my instructors and professors at Bangalore Institute Of Technology (BIT) whose teachings and support have laid the foundation for my understanding of programming and geographic information systems.

Special thanks to my colleagues and peers, whose collaborative efforts and constructive feedback have significantly contributed to the success of this project. Most importantly we had determined and proactive teammates which made each one of us evolve around each other. Kudos to all.

Finally, we would like to thank my family and friends for their unwavering support and encouragement throughout this endeavor.

This project would not have been possible without the combined efforts of everyone mentioned above. Thank you all for your contributions and support.

ABSTRACT OF THE PROJECT

A toll tax collection system is an infrastructure used to levy taxes from moving vehicles to generate revenue for the upkeep, operation, and expansion of transportation infrastructures like roads, bridges, tunnels, and other transportation facilities. However, the existing toll tax collection system suffers from several challenges, leading to inefficiencies, congestion, revenue spillage, and annoyance for drivers. Tollbooth delays, lengthy lines, and longer processing times are caused by dependency on cash payments. Traffic is backed up as a result, especially during rush hours, which aggravates drivers and wastes time.

To tackle this major drawback of manual toll tax collection system in the Toll Booths we proposed a new dynamic approach called “GPS Toll based System ". Systems for manually collecting tolls necessitate a substantial amount of labor and infrastructure upkeep. This is where our model brings in a major advantage with a contemporary method for automating and streamlining the collection of toll taxes using GPS technology which precisely calculates the distance traveled and applies the relevant toll fees using established rates using GPS devices that are mounted in automobiles.

Our model asks the driver only for the distance they have traveled. Whenever the driver rode out of the highway, that particular geolocation was taken as the endpoint. With the following coordinates our programs generate an appropriate toll tax the user has to pay and it gets automatically detected from the user’s bank account. Once the money is deducted the user gets the notification of the distance traveled, toll tax rate, and the estimated toll tax which is successfully debited. This approach solves the major part of the drawbacks we faced in the manual toll tax collection system.

TABLE OF CONTENTS

LEFT THIS PAGE INTENTIONALLY BLANK

INTRODUCTION

1.1. Evolution of Toll Collection Systems

Toll collection systems have been vital in funding road infrastructure for centuries. Initially, toll collection was manual, with toll collectors stationed at booths where drivers would stop to pay cash. This method, while straightforward, was labor-intensive, error-prone, and caused significant traffic delays, especially during peak hours. The late 20th century saw the advent of Electronic Toll Collection (ETC) systems, which utilized Radio Frequency Identification (RFID) or Dedicated Short-Range Communications (DSRC) to automate the tolling process. Vehicles equipped with RFID tags or transponders could pass through toll plazas without stopping, as toll charges were automatically deducted from pre-paid accounts. Although ETC systems significantly reduced congestion and improved efficiency, they required substantial infrastructure investment and maintenance.

The latest advancement in toll collection technology is GPS-based toll collection systems. These systems track vehicles' locations in real time using Global Positioning System (GPS) technology, enabling seamless and continuous tolling based on actual road usage. GPS-based toll collection eliminates the need for physical toll booths, reducing infrastructure costs and offering greater flexibility and accuracy in toll calculation. Fees can be based on distance traveled, road type, and time of day, providing a more efficient and transparent tolling method.

1.2. Motivation for GPS-Based Toll Collection

Several factors drive the transition to GPS-based toll collection systems. Firstly, traditional toll booths, whether manual or electronic, often create bottlenecks and traffic jams, particularly during peak travel times. GPS-based toll collection eliminates the need for vehicles to slow down or stop at toll booths, ensuring a smoother traffic flow and reducing congestion. Secondly, GPS technology provides precise location tracking, allowing for accurate calculation of toll fees based on the exact distance traveled on toll roads. This level of accuracy is difficult to achieve with traditional tolling methods, leading to fairer and more transparent tolling practices.

Thirdly, the installation and maintenance of physical toll booths and electronic toll collection infrastructure are expensive. GPS-based systems significantly reduce these costs by relying on satellite technology and existing mobile networks. This cost reduction is particularly beneficial for expanding toll networks in regions with limited infrastructure. Fourthly, GPS-based toll collection systems offer a seamless and convenient experience for drivers, with real-time notifications, automatic payments, and detailed trip summaries improving user satisfaction and encouraging compliance with toll regulations.

Additionally, GPS-based systems can be easily scaled to cover extensive road networks, including urban areas, highways, and rural roads. They offer flexibility in implementing dynamic pricing models, such as congestion pricing and time-of-day tolling, to manage traffic demand and optimize road usage. Finally, by reducing traffic congestion and promoting efficient road usage, GPS-based toll collection systems can contribute to lower vehicle emissions and reduced fuel consumption, aligning with global efforts to promote sustainable transportation and combat climate change.

1.3. Global Adoption and Success Stories :

1. **Germany's Truck Tolling System (Toll Collect)**

- Uses GPS technology to track and toll heavy goods vehicles (HGVs) on highways.
- Operational since 2005.
- Significantly reduced traffic congestion.
- Improved road maintenance funding.

2. **Slovakia's Electronic Toll Collection System (Myto)**

- Employs GPS technology for trucks.
- Covers over 2,000 kilometers of highways and major roads.
- Enhanced toll revenue collection.
- Streamlined freight transportation.

3. **Hungary's National Toll Payment Services (NTPS)**

- Introduced a GPS-based toll system for all vehicles over 3.5 tons in 2013.
- Improved toll compliance.
- Contributed to infrastructure development.

These success stories highlight:

- Effectiveness of GPS-based toll collection systems.
- Benefits for other regions and countries.

Inspired by the success of these GPS-based toll collection systems in countries like Germany, Slovakia, and Hungary, our team was motivated to develop a similar system in our country. Observing the significant improvements in traffic management, toll revenue collection, and infrastructure funding in these regions, we recognized the potential benefits of implementing GPS-based tolling in our transportation network. By leveraging GPS technology, this project aims to create an innovative and efficient toll collection system that addresses the limitations of existing systems and meets the growing demands of modern transportation networks.

1.4. Project Aims

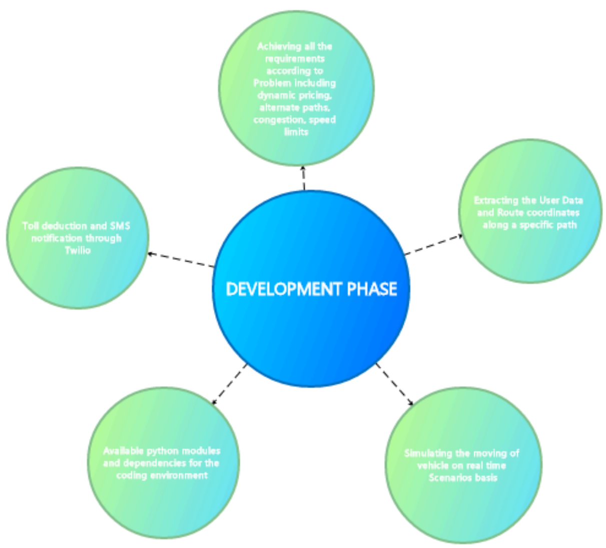
By leveraging GPS technology, our project aims to:

- Create an innovative and efficient toll collection system.
- Address limitations of existing systems.
- Meet the growing demands of modern transportation networks.

Fig.1 Concept of GPS Technology

EXISTING SYSTEM	PROPOSED SYSTEM
Typically relies on RFID tags, manual toll booths, and electronic toll collection systems that use fixed infrastructure.	Utilizes advanced GPS technology to track vehicle location and calculate toll fees based on actual distance traveled.
Often charges a flat fee regardless of the distance traveled	Implements variable toll fees based on vehicle type (e.g., heavy trucks, passenger cars) and distance traveled, ensuring fair pricing.
Charges a fixed amount at toll plazas, not accounting for the actual distance traveled by the vehicle.	Calculates toll fees based on the precise distance traveled by the vehicle, providing a more accurate and fair toll system.
Requires multiple toll plazas along highways, leading to traffic congestion and increased operational costs.	Reduces the need for numerous toll plazas, as toll collection is managed via GPS tracking, decreasing traffic congestion and operational costs.
Requires significant manual monitoring and intervention at toll gates.	Eliminates the need for manual monitoring at toll gates, as the system is automated through GPS technology.
Often involves manual processing and can be prone to errors and delays	Automatically generates bills and processes transactions, ensuring a seamless and efficient toll-collection process
Prone to toll evasion due to manual systems and lack of real-time monitoring	Prevents toll evasion by using GPS technology to accurately track vehicle movements and ensure toll fees are collected for every usage

AN OUTLINE OF THE PROJECT



INNOVATIVE ASPECTS/UNIQUENESS AND SECURITY

4.1. Implementation of QR Code in Vehicle Number Plates for Enhanced Toll Collection and Security

To modernize and streamline toll collection, identity verification, and security protocols, we are introducing an innovative method of integrating QR codes into the number plates of Indian-registered vehicles.

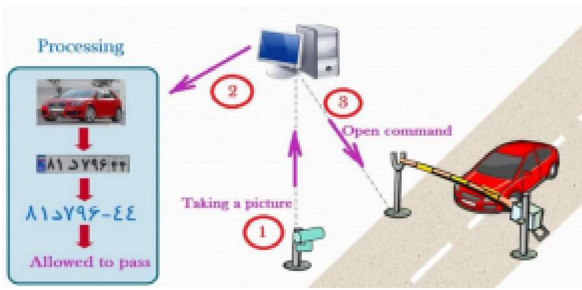


Fig. 2 Existing Toll System

4.1.1. Implementation Methodology

❖ <u>**QR Code Integration :**</u>

1. Design and Placement: QR codes will be integrated into the design of vehicle number plates. The QR code will be prominently placed to ensure it is easily scannable by automated systems at toll booths and other checkpoints.

2. Data Encoding: Each QR code will encode a unique identifier corresponding to the vehicle's registration details, owner's identity, and a secure link to the vehicle's account for financial transactions.

3. Security Measures: Advanced encryption techniques will be employed to protect the encoded information within the QR codes, ensuring data integrity and security against potential fraud.

Fig.3 QR Integration

Functionalities:

❖ **<u>Toll Money Deduction</u>**

- **Automated Scanning:** When a vehicle approaches a toll booth, automated scanners will read the QR code on the number plate.
- **Transaction Processing:** The scanned data will be transmitted to a central system that processes the toll fee by deducting the appropriate amount from the vehicle owner's linked account.
- **Confirmation and Receipt:** A confirmation of the transaction will be sent to the vehicle owner, along with an electronic receipt.

This advanced system aims to enhance efficiency and security through the following key features:

❖ **Automatic Toll Payment:**

- Each vehicle will have a unique QR code embedded in its number plate.
- As a vehicle passes through a toll booth, the QR code will be scanned, and the toll amount will be automatically deducted from the owner's account.
- This eliminates the need for manual transactions, reducing wait times and traffic congestion at toll plazas.

❖ **Identity Verification:**

- The QR code will store essential vehicle and owner information, including registration details, insurance status, and the owner's identity.

- Real-Time Access: Law enforcement and regulatory authorities can access the vehicle's registration and owner's identity details by scanning the QR code.
- Verification Protocols: The system will cross-reference the QR code data with the national vehicle database to ensure authenticity and accuracy

❖ **Criminal Detection:**

- The system will be integrated with national databases to identify vehicles flagged for criminal activities.
- Upon scanning the QR code, authorities will receive immediate alerts if the vehicle is reported stolen or involved in any criminal incidents.
- This real-time information will aid in swift action and enhance public safety.

❖ **Enhanced Security Measures:**

- The QR code will be designed to be tamper-proof, making it difficult to alter or forge.
- Regular updates and maintenance of the QR code database will ensure the accuracy and reliability of the information stored.

❖ **Convenience and Efficiency:**

- Vehicle owners will benefit from a seamless and hassle-free toll payment experience.
- The system will reduce administrative burdens and operational costs for toll operators and law enforcement agencies.

4.2. Enhanced Feature: Distance-Based Toll Exemption

In addition to the core functionalities of QR code integration for toll money deduction and identity verification, an enhanced feature will be introduced to offer toll exemptions based on the proximity of the vehicle owner's residence to the toll booth.

❖ <u>**Proximity-Based Toll Exemption**</u>

- **Residence Verification:** The system will verify the vehicle owner's registered address and calculate the distance from the toll booth.

- **Geofencing:** A geofencing mechanism will be implemented, creating a 20km radius around each toll booth.
- **Automatic Exemption:** If the vehicle owner’s registered address falls within this 20km radius, the system will automatically exempt the vehicle from toll charges.
- **Dynamic Adjustment:** The exemption will be dynamically applied based on the vehicle's entry and exit points within the toll zone.

<u>Implementation Steps :</u>

- o **Address Database Integration:** The system will integrate with the national vehicle registration database to access and verify the owner's residential address.
- o **Distance Calculation Algorithm:** An algorithm will calculate the distance between the toll booth and the vehicle owner’s registered address in real-time.
- o **Exemption Criteria:** Vehicles meeting the 20km proximity criterion will be flagged, and the toll charge will be bypassed for these vehicles.

Benefits of Proximity-Based Toll Exemption

1. **Local Resident Relief:** Provides financial relief to residents who frequently travel short distances and cross toll booths.
2. **Increased Fairness:** Ensures that toll charges are applied more equitably, reflecting the actual distance traveled by the vehicle.
3. **Community Support:** Demonstrates support for local communities, fostering goodwill and cooperation between residents and toll authorities.
4. **Reduced Administrative Burden:** Automates the toll exemption process, reducing the need for manual checks and verifications.

By integrating this proximity-based toll exemption feature, the system becomes more adaptable and responsive to the needs of residents, enhancing the overall efficiency and fairness of toll collection. This approach not only improves user experience but also aligns with equitable tolling practices.

4.3. GPS Integration for Vehicle Movement Monitoring and Toll Collection

❖ **<u>GPS Units in Vehicles</u>**

1. **Real-Time Tracking:** Each vehicle will be equipped with a GPS unit to provide accurate and real-time tracking of its movements. Popular and reliable GPS units suitable for Indian standards include:

-MapmyIndia ICENAV 7464 PRO

-JioTrack GPS Tracker

-Letstrack Prima Vehicle Tracker

- Akari Gt02A GPS Tracker

- Autowiz Obd GPS Car Tracker

-iTrack GPS Tracker

Fig. 4 Different GPS Units Available Across India

2. **Distance Calculation:** The GPS data will be used to calculate the exact distance traveled by the vehicle, ensuring precise toll charges.

3. **Enhanced Security:** GPS tracking enhances vehicle security by enabling quick location tracking in case of theft or unauthorized use.

Costs of Implementation

- **Initial Installation:**
  - o **Hardware Costs:** The cost of purchasing GPS units, such as MapmyIndia ICENAV 7464 PRO, Autowiz Obd GPS Car Tracker, or JioTrack GPS Tracker, ranges from ₹2,000 to ₹10,000 per unit.
  - o **Installation Costs:** Professional installation services may cost between ₹500 and ₹2,000 per vehicle.
- **Data Management:**

  Implementing a robust data management system to handle the large volume of GPS data will involve:
  - o **Server Costs:**

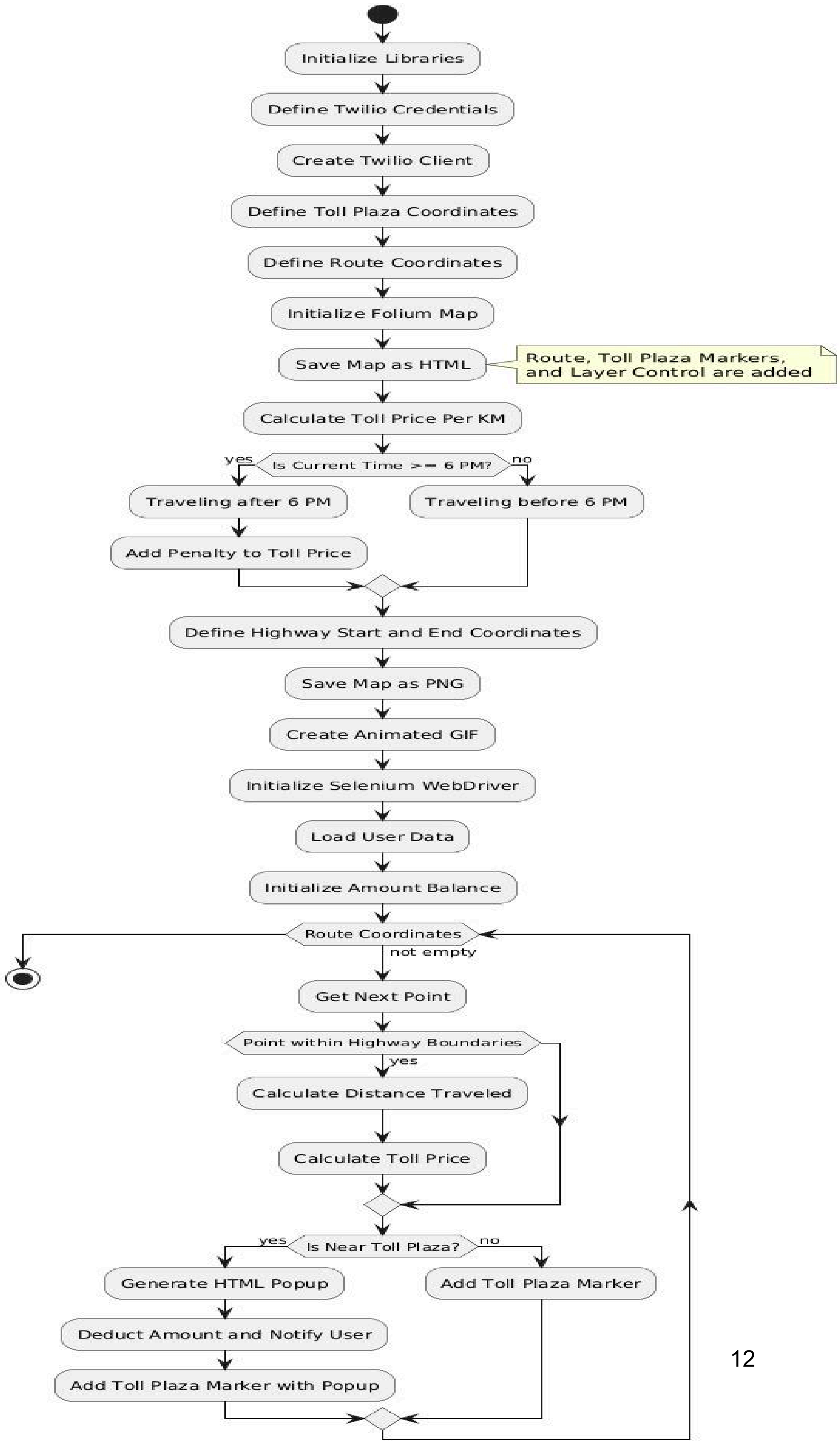
    Investment in servers and cloud storage solutions for data handling.
  - o **Software Development:**

    Custom software solutions for data analysis and integration with toll-collection systems.

Benefits of GPS Integration

1. **Accurate Toll Collection:** GPS data ensures toll charges are based on the actual distance traveled, promoting fairness and accuracy in toll collection.
2. **Improved Traffic Management:** Real-time data on vehicle movements can be used for better traffic management and congestion reduction.
3. **Enhanced User Experience:** Provides vehicle owners with detailed travel logs and precise toll charge calculations, improving transparency and trust in the system.
4. **Emergency Assistance:** Enables quick response in case of emergencies by providing accurate location data.

FLOWCHART OF THE PYTHON PROGRAM



SEQUENCE DIAGRAM

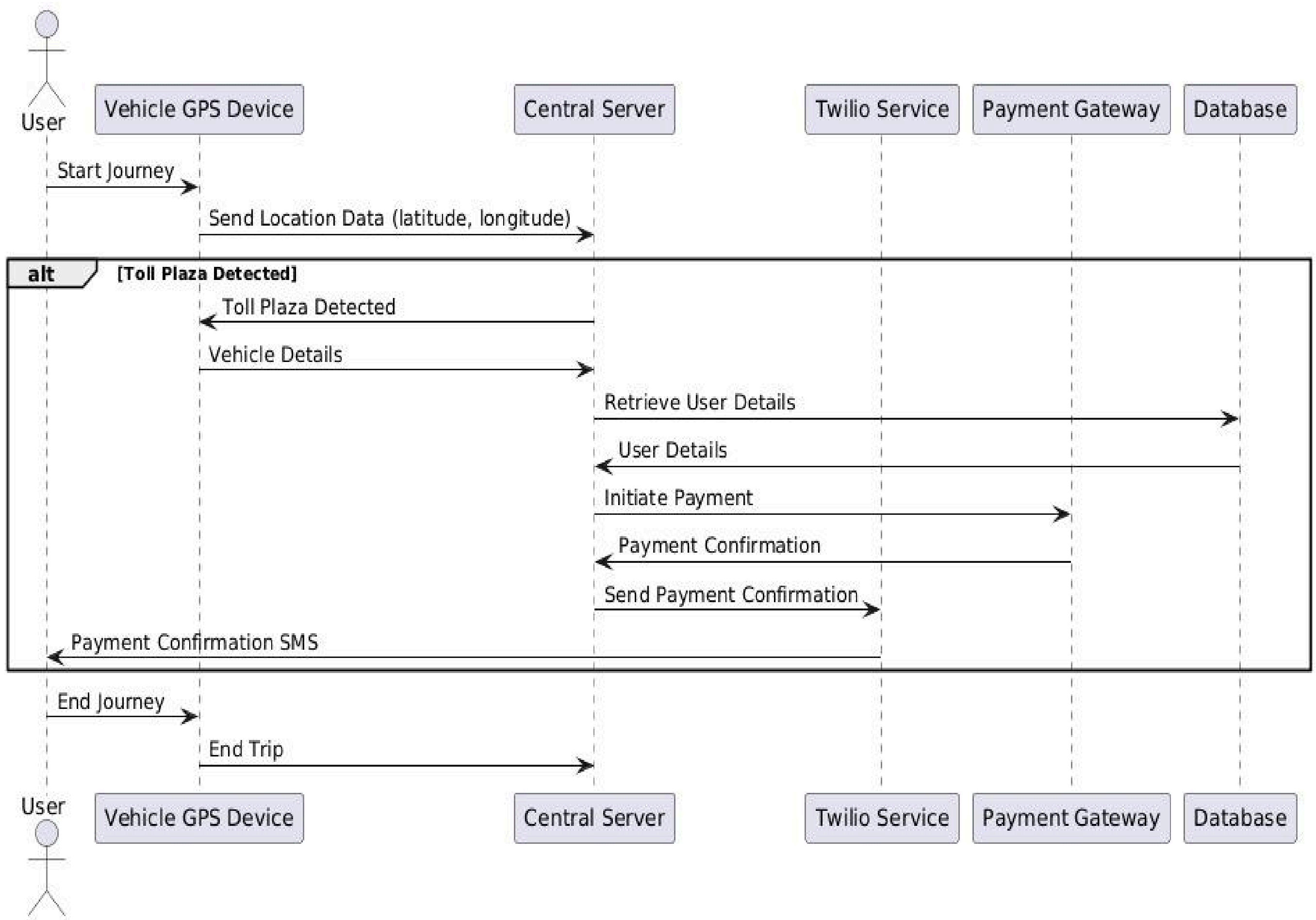


Fig. 5 CI\CD among these components

IMPLEMENTATION OF THE CODE

5.1. How does the code work?

The GPS-based automated toll collection system is designed to efficiently manage toll payments for vehicles traveling along specific routes between Bangalore and Chitradurga. It leverages GPS coordinates, real-time data processing, and automated notifications to streamline the toll payment process.

Key Components

1. **Route Definition and Mapping**:
    - The system utilizes predefined route coordinates between Bangalore and Chitradurga, mapped using latitude and longitude points.
    - These routes are visualized using Folium maps, which display the starting point, endpoints, and toll plaza locations.
2. **Toll Plaza Integration**:
    - Toll plaza locations along the route are predefined with specific coordinates.
    - Each toll plaza is marked on the map, and the system monitors vehicle proximity to these points using geodesic distance calculations.
3. **Automated Toll Calculation**:
    - As vehicles move along the route, the system calculates toll amounts based on the distance traveled.
    - The toll price per kilometer is dynamically set, and the system deducts the appropriate amount from the vehicle owner's account balance.
4. **Real-time Notifications**:
    - Users receive automated SMS notifications about toll deductions, ensuring transparency and providing updates on their account balances.
    - Twilio integration enables seamless communication between the system and users, ensuring timely notifications.
5. **Penalty and Alternate Route Handling**:

  - The system includes penalty logic for vehicles traveling after a specified time (e.g., 6 PM), adding penalties to toll charges.
  - It also supports alternate route detection and toll calculation adjustments, ensuring accurate billing based on the chosen route.
6. **User Account Management**:
  - User data, including vehicle IDs, account balances, and phone numbers, are managed through a CSV file.
  - The system retrieves account information, deducts toll amounts, and updates balances accordingly.

❖ Workflow

1. **Initialization**:
  - The system initializes with predefined route coordinates, toll plaza locations, and user data.
2. **Route Monitoring**:
  - Vehicles are monitored as they travel along the route using GPS coordinates.
  - Geodesic distance calculations determine the distance traveled between points.
3. **Toll Calculation and Deduction**:
  - Toll amounts are dynamically calculated based on the distance traveled by the vehicle.
  - Deductions are made from the user's account balance, with real-time updates sent via SMS notifications.
4. **Exception Handling**:
  - The system handles exceptions such as penalties for late-night travel and provides notifications to users about these additional charges.

RELATED LIBRARIES/MODULES AND METHODS USED

- **Imports**: We've imported various libraries such as folium for creating interactive maps, PIL for image manipulation, selenium for web automation, imageio for creating GIFs, `io` for handling streams, geopy for geolocation calculations, pandas for data manipulation, twilio for SMS notifications, phonenumbers for phone number formatting, numpy for numerical operations, and datetime for handling dates and times.
- **Twilio Integration**: We used Twilio to send SMS notifications about toll deductions.
- **Map Initialization**: We initialized a folium map with different tile layers (OpenStreetMap, Stamen Terrain, Toner, Watercolor, CartoDB Positron, and Dark Matter).
- **Markers and Polyline**: We added markers for start and end points, and toll plazas, and drew a polyline representing the route.
- **Dynamic Toll Calculation**: Toll prices are dynamically calculated based on distance traveled, with different conditions such as penalty charges if traveling after 6 PM.
- **Web Scraping and Automation**: Selenium is used to capture screenshots of the map and automate certain interactions.
- **Data Handling**: You read user data from a CSV file (`user_data.csv`) to simulate account balances and handle transactions.
- **Simulation Logic**: The main logic involves simulating a vehicle moving along predefined route coordinates, checking proximity to toll plazas, calculating toll amounts, and deducting them from the user's account balance.
- **Alternate Paths**: There’s a provision to handle alternate paths and skip toll calculations when the vehicle is on an alternate route.
- **Map Rendering**: Finally, you save the map as an HTML file (`route_map.html`), capture it as an image (`route_map.png`), and create an animated GIF.

Particularly we have used some of the important modules in our code with specific purposes :

- ❖ **Folium for Route Visualization**

Purpose: Folium is utilized to visualize the predefined route between Bangalore and Chitradurga using interactive maps.

Explanation:

- **Folium** is a Python library that integrates with Leaflet.js maps, allowing us to create interactive and customizable maps directly within Python scripts.
- **Why We Use It**: By plotting the route coordinates and toll plaza locations on a map, Folium enhances visual understanding and management of the toll collection system. It provides stakeholders and users with a clear overview of the route, toll plaza positions, and geographical context, aiding in system monitoring and user interaction.

- ❖ **Geopy for Geodesic Distance Calculation**

Purpose: Geopy is employed for calculating geodesic distances between GPS coordinates of vehicles and toll plaza locations.

Explanation:

- **Geopy** is a Python library that provides geocoding, reverse geocoding, and distance calculations using various geodesic models.
- **Why We Use It**: The toll calculation in our system relies on accurate distance measurements between vehicle positions and toll plazas. Geopy's geodesic calculations ensure precise distance computations based on the Earth's curvature, essential for determining toll charges per kilometer traveled. This accuracy is crucial for fair and transparent billing, enhancing user trust and system efficiency.

- ❖ **Twilio for SMS Notifications**

Purpose: Twilio is integrated to send real-time SMS notifications to users regarding toll deductions and system updates.

Explanation:

- **Twilio** is a cloud communications platform that allows programmable SMS, voice, and messaging services.
- **Why We Use It**: SMS notifications play a vital role in user communication within our system. Twilio enables automated, timely notifications to users about toll deductions, account balances, and system updates. This real-time communication fosters transparency, keeps users informed about financial transactions, and enhances user experience by providing immediate feedback on their toll payments.

❖ **CSV Handling for User Account Management**

Purpose: CSV (Comma-Separated Values) files are utilized for managing user data, including vehicle IDs, account balances, and contact information.

Explanation:

- **CSV Handling**: Python's built-in CSV module(pandas' dataframe) is used for reading and writing CSV files efficiently.
- **Why We Use It**: User account management involves storing and retrieving critical information such as vehicle IDs, account balances, and contact details. CSV files provide a straightforward and structured format for storing this data, facilitating easy integration with Python scripts handling toll calculations and user notifications. This method ensures robust data management and seamless interaction between the toll collection system and user accounts.

❖ **Image Processing (PIL, imageio)**

Implementation Purpose: PIL (Pillow) and imageio are used for processing vehicle images captured at toll booths for identification and verification purposes.

❖ **Web Scraping (Selenium)**

Implementation Purpose: Selenium can be used to automate the retrieval of traffic and road condition information from websites for real-time updates.

- **Data Manipulation (Pandas)**

Implementation Purpose: Pandas is used for storing and processing toll transaction data, including vehicle IDs, timestamps, and toll charges.

- **Phone Number Formatting (phonenumbers)**

Implementation Purpose: phonenumbers is used for validating and formatting user phone numbers for toll payment notifications.

- **Numerical Operations (NumPy)**

Implementation Purpose: NumPy is used for mathematical calculations related to toll charges or statistical analysis of toll transaction data.

Other Supporting Features Include :

- **Path Determination (OpenRouteService)**

Implementation Purpose: OpenRouteService is used for determining the path by extracting all the coordinates for a route, which are stored in `pathfile.txt`

- **Performance Optimization (Visvalingam-Whyatt Algorithm)**

Implementation Purpose: The Visvalingam-Whyatt algorithm is used to simplify paths by reducing the number of points, thus increasing performance.The algorithm views each vertex of a polyline as a point in a Cartesian coordinate system. The area is calculated using the determinant formula, which is efficient for 2D computations:

Given a chain of 2d points $\{p_i\} = \left\{ \begin{bmatrix} x_i \\ y_i \end{bmatrix} \right\}$, the importance of each interior point is computed by finding the area of the triangle formed by it and its immediate neighbors. This can be done quickly using a matrix determinant.[1] Alternatively, the equivalent formula below can be used[2]

$$A_i = \frac{1}{2} \left| x_{i-1}y_i + x_i y_{i+1} + x_{i+1}y_{i-1} - x_{i-1}y_{i+1} - x_i y_{i-1} - x_{i+1}y_i \right|$$

The output of the Visvalingam algorithm is a simplified polyline with fewer vertices, where the remaining vertices are considered the most significant in terms of preserving the overall shape and visual appearance of the original polyline.

TEST CASES & SNIPPETS

Essential Prerequisites for Input:

- User's Data
- Route Coordinates set
- Twilio Account is a must
- A parsed GPX file which is used to monitor vehicle speeds via different timestamps
- Make sure that all the dependencies have been installed in your local system

SCENARIO 1:

WHEN THE VEHICLE(ASSUME A CAR) IS MOVING FROM BANGALORE TO CHITRADURGA WITHOUT TRESPASSING IN ANY ALTERNATIVE PATH

OUTPUT:

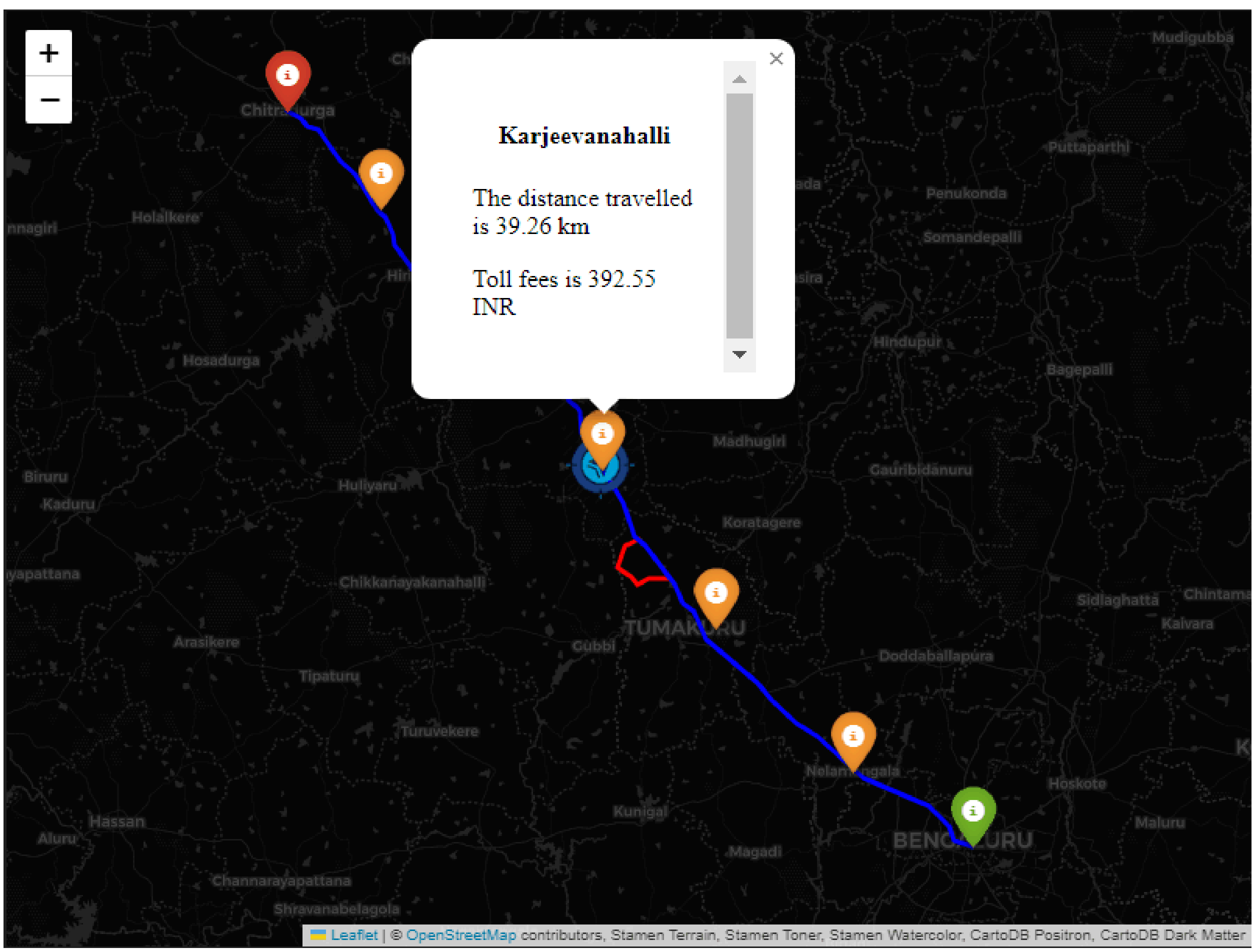


Fig. 6 Vehicle is moving from Bangalore to Chitradurga via Highway Road Only

SCENARIO 2:

WHEN THE VEHICLE(ASSUME A CAR) IS MOVING FROM BANGALORE TO CHITRADURGA BY TRESPASSING IN AN ALTERNATIVE PATH NEAR TUMKUR

OUTPUT:

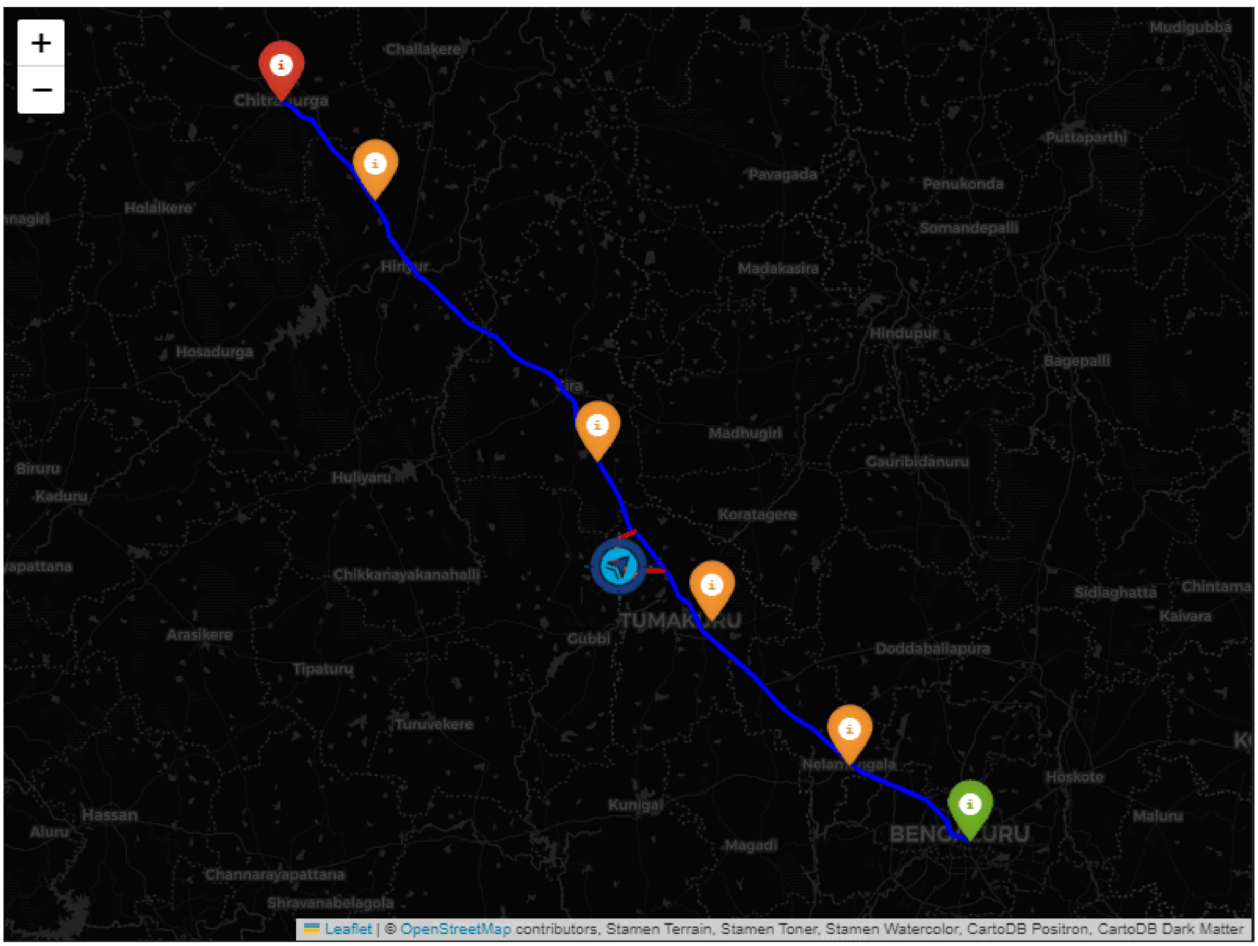


Fig. 7 We can notice that the vehicle has taken a non-highway path.

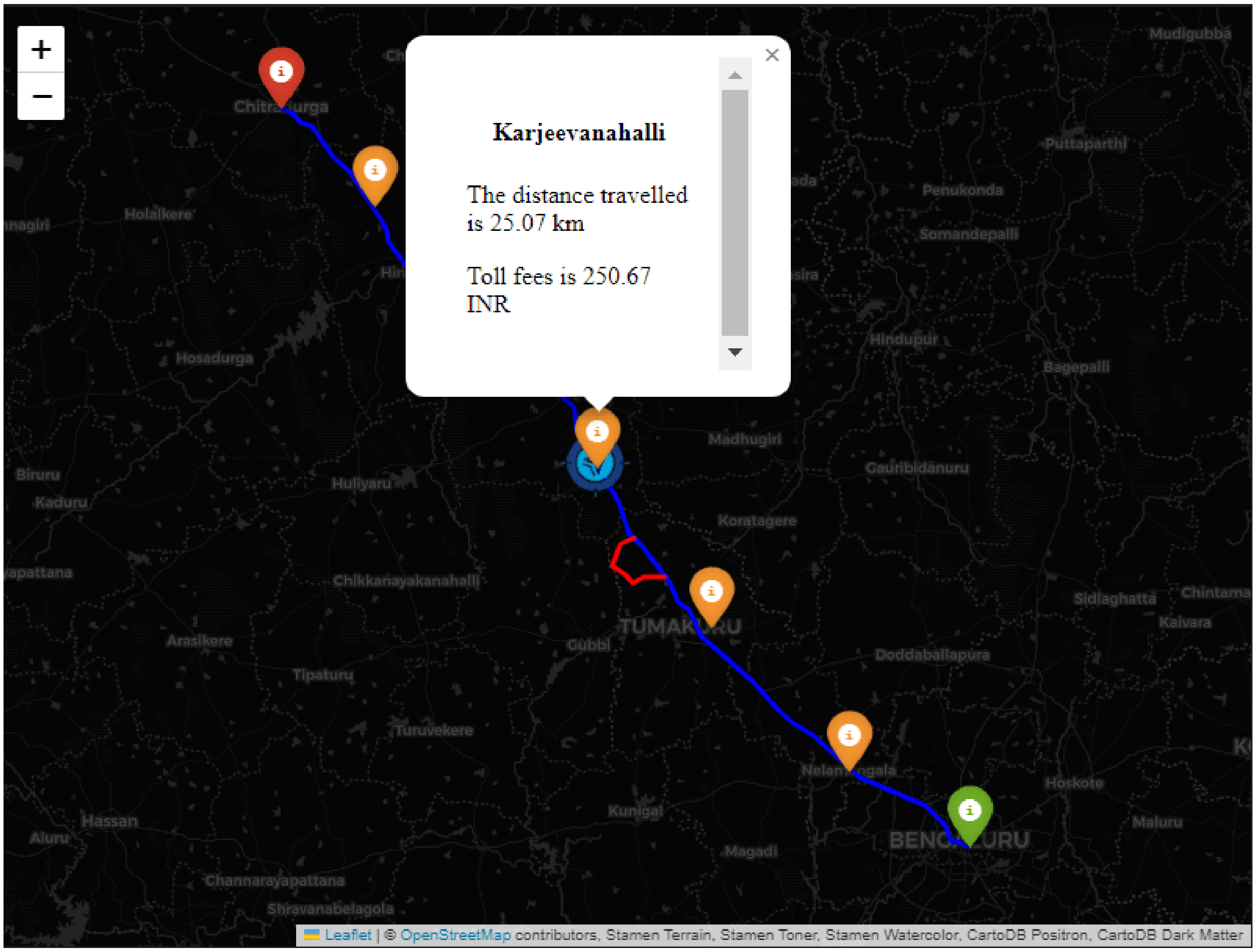


Fig. 8 Here, we can also see that Toll Calculation and Distance Traveled Calculations were suspended during the period when the car was traveling on a non-highway road.

PLEASE NOTE:

For Source Input and Animated Output(GIF) please click on this GitHub Repository Link:
Github Repo

To see our speed limit feature and our Map Route, please click on this link
Github Repo

- All the exception cases and printing statements will be visible on the terminal once we run the above-mentioned source code

TWILIO AND ITS SMS NOTIFICATION SYSTEM

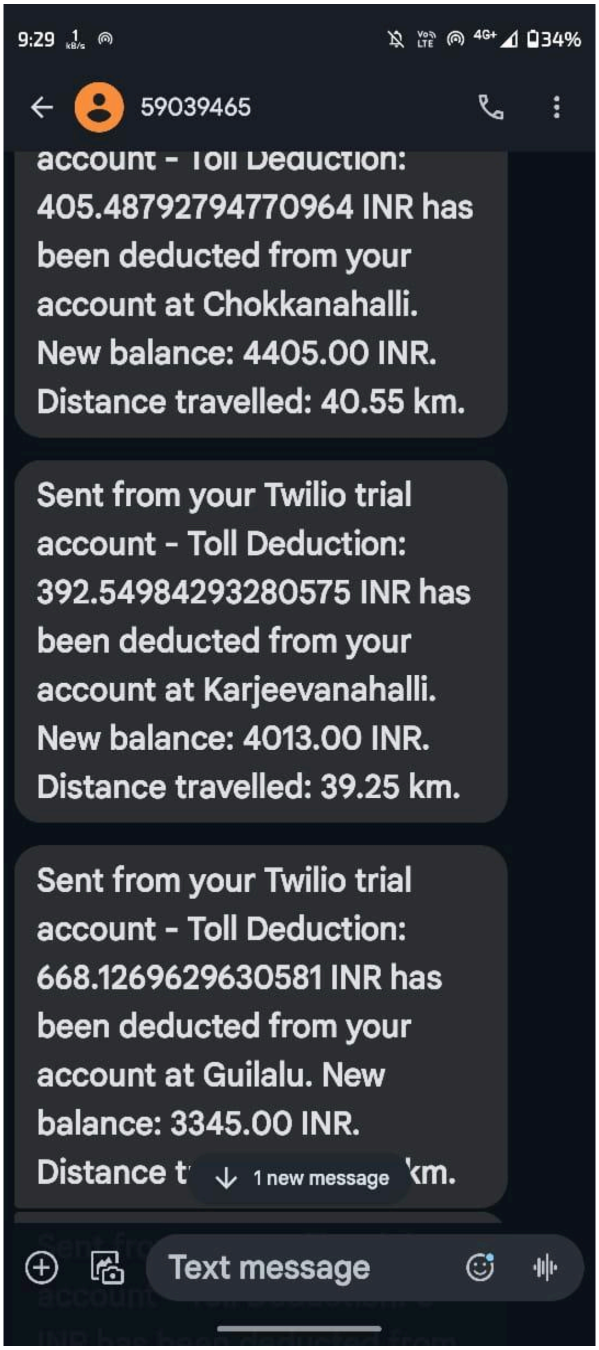


Fig. 9. SMS Notification while travelling on Highway

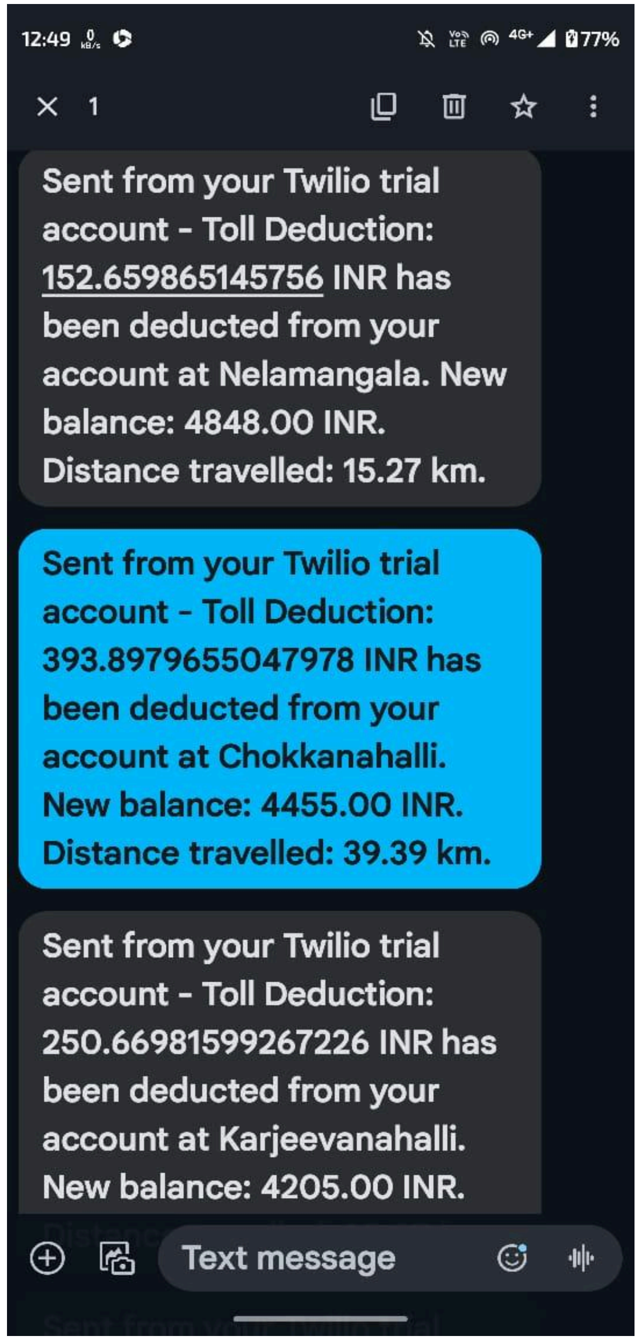


Fig. 10 SMS Notification while traveling on non-Highway for some period

FUTURE SCOPE AND ENHANCEMENT

Here are some potential future enhancements for the project:

- ❖ Use of weather data to adjust toll rates or provide alternate routes during adverse weather conditions.
- ❖ Use of Google API to simulate a vehicle based on the real route traveled.
- ❖ Offering subscription-based models for frequent travelers, providing discounts or unlimited access for a fixed fee.
- ❖ Addition of scenarios like road closures, accidents, and diversions.
- ❖ Detailed analytics on toll road usage, peak travel times, and revenue generation.
- ❖ Use of cloud-based architecture to handle a larger number of users and real-time data processing.
- ❖ Expansion of the app to cover more regions and countries, adapting to local tolling systems and regulations.
- ❖ 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.
- ❖ Use profiling tools to identify bottlenecks in the code and optimize them for better performance. Techniques such as loop unrolling, minimizing redundant calculations, and parallel processing can be explored.
- ❖ Implement caching mechanisms for frequently accessed data, such as toll booth locations and user data, to reduce redundant database queries and improve response times.
- ❖ Develop QR codes that serve multiple purposes, including user identification, payment mode selection, and emergency alerts.
- ❖ Implement a system for generating dynamic QR codes that update in real-time with relevant user information and transaction details.
- ❖ Include emergency contact information and alert mechanisms in the QR codes to provide quick assistance in case of accidents or medical emergencies.
- ❖ Dynamic reporting of the real-time data and storing them with all policies and safety measures, so that we'll be ensured of no security or data breaches.

CONCLUSION

The GPS-Based Automated Toll Collection System represents a significant advancement in toll management, leveraging modern technologies to provide a seamless and efficient experience for users. By integrating GPS technology, we can accurately calculate toll charges based on the distance traveled, ensuring a fair and transparent toll collection process. The use of Python and various libraries has enabled the development of a robust and versatile system that addresses numerous challenges faced by traditional toll collection methods.

Key components of the system include mapping and pathfinding using Folium and OpenRouteService for interactive mapping and path determination, which efficiently calculates routes and displays them to users. Geopy is employed for precise distance calculations, ensuring accurate toll charges based on the user's journey. Data handling and analysis are facilitated by Pandas and NumPy, providing powerful tools for data manipulation and numerical operations. Communication integration is achieved through Twilio and phone numbers, allowing for seamless SMS integration and phone number formatting, and enhancing user communication. Additionally, image processing and web scraping capabilities are provided by PIL, imageIO, and Selenium, contributing to the system's overall functionality.

Real-time monitoring is enabled through Streamlit, which allows for the deployment of an app for real-time speed limit monitoring, enhancing safety and compliance on the road. The system also includes innovative features such as the Visvalingam-Whyatt algorithm for performance optimization, extensive route point extraction stored as pathfile.txt, and comprehensive logging mechanisms for future analysis and cloud storage.

Looking ahead, several future enhancements have been identified to further improve the system. These include continuously refining algorithms and code for better efficiency, integrating real-time traffic data and predictive models to manage congestion near toll plazas, and implementing geo-fencing for toll exemptions within a 10km radius of the user's residence. Enhancements to the payment gateway process, such as improved security and a better user experience, are also planned. Additionally, the development of regenerative QR codes for user identification, payment modes, and emergency alerts is a key focus.

REFERENCES

Websites and Blogs:

- A Basic Idea of how the animation works is retrieved from:
https://stackoverflow.com/questions/68899833/how-to-create-animation-of-vehicle-moving-form-a-to-b-along-a-route

- Timestamps Retrieved from:
https://gotoes.org/strava/Add_Timestamps_To_GPX.php

- The code for retrieving the route points along the path retrieved from:
https://medium.com/analytics-vidhya/measure-driving-distance-time-and-plot-routes-between-two-geographical-locations-using-python-39995dfea7e

Libraries and Frameworks

- *folium*: Used for creating interactive maps and adding markers, polylines, and layers.
    - Documentation: folium
- *Pillow (PIL)*: Used for image processing and saving map screenshots.
    - Documentation: Pillow
- *Selenium*: Used for automating web browser interactions to capture map screenshots.
    - Documentation: Selenium
- *webdriver-manager*: Simplifies the management of browser drivers.
    - Documentation: webdriver-manager
- *geopy*: Used for calculating geographical distances between points.
    - Documentation: geopy
- *pandas*: Used for data manipulation and handling user data.
    - Documentation: Pandas

APIs

- *Fast2SMS API*: Used for sending SMS notifications. (Added extra features)
    - Documentation: Fast2SMS API
- *Open Route Service*: This can be used for route planning and directions. This is only used for retrieving the real coordinates along the path.
    - Documentation: openrouteservice

Data Sources

- *Geographical Coordinates*: The coordinates for Bangalore, Chitradurga, and toll plazas are predefined.
- *User Data*: User data is assumed to be loaded from a CSV file.

Software Tools

- *Google Chrome* and *Chromedriver*: Used with Selenium for capturing map screenshots.
  - Documentation: ChromeDriver and Selenium ChromeDriver
- Python. (n.d.). Retrieved from https://www.python.org

Online Resources

- Stack Overflow: For various coding issues and solutions related to the above libraries and tools.
  - Example: Selenium WebDriver headless mode