GLA UNIVERSITY

Department of computer Science and enineering

Project Report on
GPS Toll-based System Simulation using Python

Submitted By

MAYANK SHARMA & JATIN KHETAN

Mentor:

ANKUR RAI , ASSOCIATE PROFESSOR

INDEX

INTRODUCTION

1.1 PROJECT OVERVIEW

The rapid growth of vehicular traffic has led to the need for efficient toll-collection systems. Traditional toll booths often cause traffic congestion and delays, leading to significant time loss and increased fuel consumption. We propose a GPS toll-based system that automates toll collection using real-time GPS data to address these issues. This system aims to streamline the tolling process, reduce human intervention, and enhance the overall efficiency of toll management.

Our project leverages modern technologies such as GPS tracking, geospatial data processing, and web-based interfaces to create a seamless and user-friendly toll collection system. By integrating hardware components like GPS devices and Arduino with robust software frameworks such as Django and Python libraries like NumPy and GeoPy, we ensure accurate toll calculation based on the distance travelled on highways.

1.2 OBJECTIVES

The primary objectives of this project are:

- **Automate Toll Collection:** Develop a system that automatically calculates and collects tolls based on vehicle movement and highway usage.
- **Improve Efficiency:** Minimize traffic congestion and reduce delays at toll booths by eliminating the need for manual toll collection.
- **Enhance Accuracy:** Utilize GPS and geospatial data to accurately determine the distance traveled on toll highways and calculate corresponding toll charges.

1.3 NEED OF THE PROJECT

The need for an automated GPS toll-based system is driven by several critical factors affecting current toll-collection methods, traffic management, and economic efficiency. This section outlines the key reasons highlighting the necessity for this project, supported by relevant facts and figures.

1.3.1 Traffic Congestion and Delays

- **Increased Vehicle Ownership:** As of 2022, the number of registered vehicles in India reached over 295 million, with an annual growth rate of around 8%. This surge has resulted in significant traffic congestion, particularly at toll plazas.
- **Time Loss:** On average, a vehicle spends 5-10 minutes at a toll booth during peak hours, cumulatively resulting in a loss of over 15-30 hours per year per commuter.
- **Economic Impact:** Traffic congestion costs the Indian economy approximately $22 billion annually, including fuel wastage, productivity loss, and environmental damage.

Growth in the number of registered vehicles

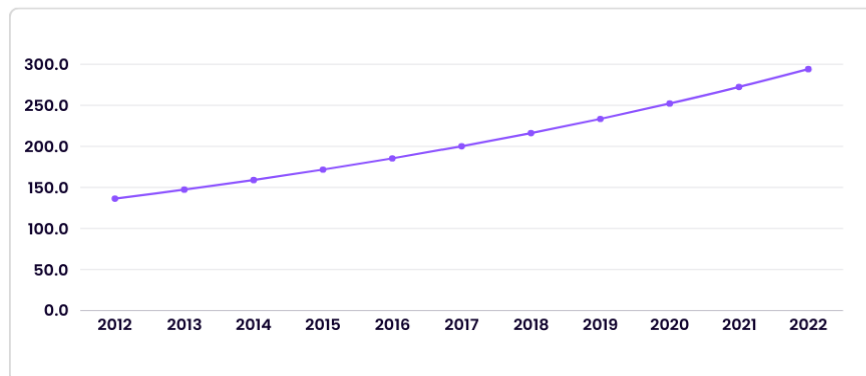


Figure: Growth in Vehicle Registration (2012-2022)

Key Insights:

- Trend Observation: The line graph will show a clear upward trend, indicating the consistent growth in the number of registered vehicles over the decade.
- Growth Rate: The graph will visually demonstrate the annual growth rate of around 8%, with the number of registered vehicles steadily increasing each year.
- Projections: This graph can also be used to make future projections by extending the line if the growth rate remains constant.

1.3.2 Inefficiencies in Toll Collection

- **Manual Toll Collection:** Traditional toll collection methods involve manual processes prone to errors, delays, and revenue leakages. Manual toll booths often suffer from insufficient change, human error, and mismanagement.
- **Revenue Loss:** A study by the Transport Corporation of India and IIM Calcutta estimated that manual toll collection results in revenue losses of up to 25% due to leakages and inefficiencies.
- **Administrative Costs:** The cost of operating manual toll booths, including salaries for toll operators and maintenance, is significantly high and impacts the overall efficiency of toll operations.

1.3.3 Technological Advancements

- **Adoption of GPS Technology:** The widespread availability and affordability of GPS devices provide an opportunity to leverage this technology for automated toll collection. GPS-based systems can accurately track vehicle movement and calculate tolls based on the distance traveled.
- **Integration with Digital Payment Systems:** The rise of digital payment platforms in India, with over 200 million active users on UPI alone, provides a robust framework for seamless toll payments.

1.3.4 User Convenience

- **Cashless Transactions:** Automated toll systems enable cashless transactions, reducing the need for cash handling and minimizing contact points, which is particularly important in the post-COVID era.
- **Real-Time Updates:** Users can receive real-time updates on toll charges and account balances, enhancing transparency and user satisfaction.

LITERATURE OVERVIEW

2.1 OVERVIEW OF GPS TECHNOLOGY

History and Development: The Global Positioning System (GPS) was initially developed by the U.S. Department of Defence in the 1970s and became fully operational in 1995. Originally intended for military applications, GPS technology has since become an integral part of various civilian applications.

Applications in Various Fields:

- **Navigation:** Widely used in automotive, maritime, and aviation industries for real-time navigation.
- **Surveying and Mapping:** Essential for accurate geospatial data collection and map-making.
- **Agriculture:** Enhances precision farming techniques by guiding machinery and optimizing field operations.
- **Emergency Services:** Assists in location tracking for quick response in emergencies.

2.2 TOLL COLLECTION SYSTEMS

Toll systems are essential for funding the construction, maintenance, and operation of highways and other critical infrastructure. With the rapid increase in vehicle ownership and the consequent rise in traffic, efficient toll collection methods are crucial for sustaining road quality and safety. Toll revenues provide a consistent financial stream that supports ongoing infrastructure projects, ensuring that highways remain safe and functional for all users.

Traditional Toll Collection Methods: Traditional toll booths rely on manual collection, where drivers stop to pay tolls using cash or cards. This method often causes traffic congestion and delays.

Modern Automated Toll Systems: Automated systems such as Electronic Toll Collection (ETC) use RFID tags, ANPR cameras, and GPS technology to streamline toll collection. These systems reduce wait times, minimize human error, and improve overall efficiency.

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

The system architecture for the GPS toll-based system integrates predefined routes and software components seamlessly. Instead of installing GPS devices in vehicles, users select routes using our application. This data is transmitted to a central server built with Django, which processes the coordinates and matches them against predefined highway routes.

Using python and JavaScript libraries like Geopandas, Leaflet.js, the system calculates the distance traveled on these routes. The processed data is stored in a MySQL database, enabling real-time toll calculation based on distance traveled. This architecture ensures accurate, efficient, and automated toll collection.

3.2 HARDWARE AND SOFTWARE COMPONENTS

Our GPS toll-based system integrates advanced hardware and robust software for efficient toll calculation. Instead of installing GPS devices in vehicles, we use predefined routes. The software, built on the Django framework, leverages Python libraries like GeoPandas for geospatial data manipulation and NumPy for numerical computations. This setup ensures precise toll computation and enhances system reliability.

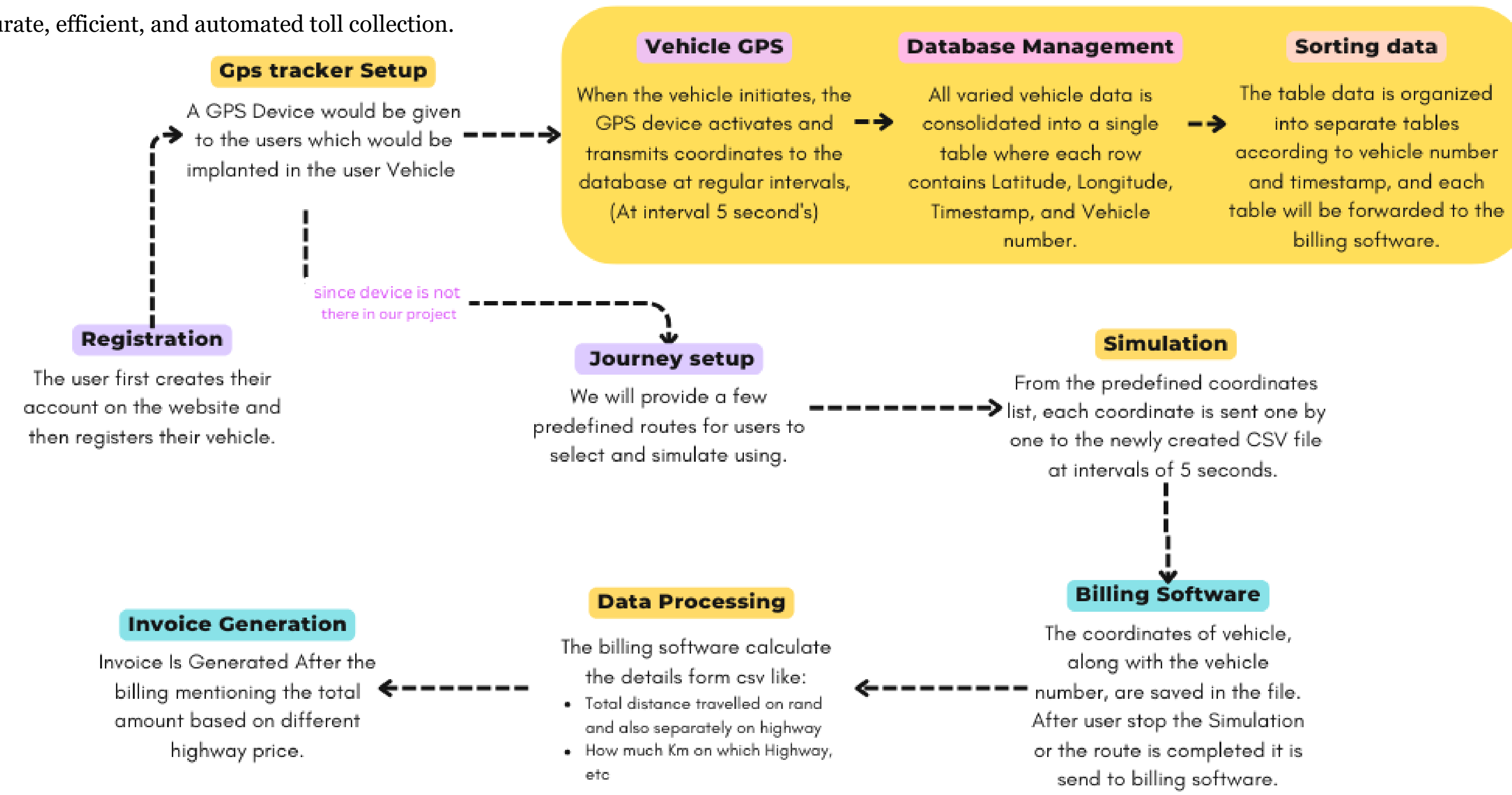


Figure: Flowchart showing the system design for GPS toll based system project

IMPLEMENTATION

4.1 DATA COLLECTION

4.1.1 Collecting Vehicle Coordinates

The process of collecting vehicle coordinates is central to our GPS toll-based system. Initially, the plan was to install dedicated GPS devices in vehicles. However, we adapted our approach to utilize predefined routes. We have collected many CSV routes file each have minimum of 60-70 km of highway route.

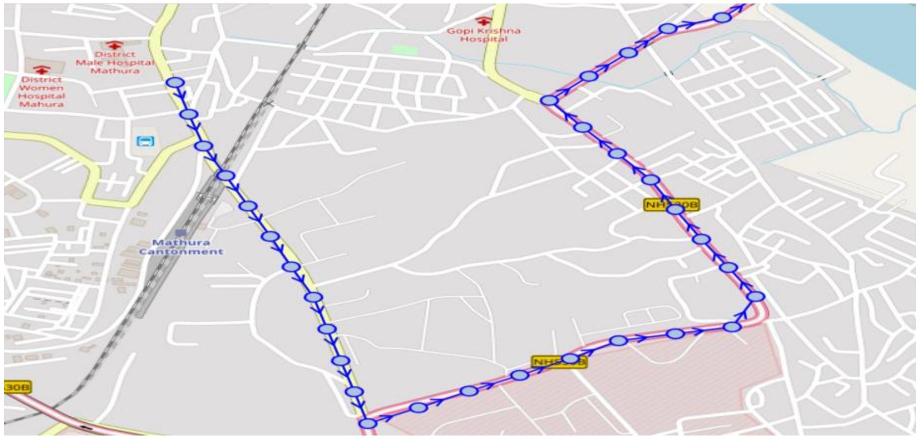


Figure: Visualization of the predefined route coordinates where dots show the coordinate point

4.1.2 GeoJSON Highway Data

The GeoJSON highway data is crucial for defining the predefined routes used in toll calculation. This data is sourced from reliable geospatial data repositories, such as OpenStreetMap (OSM) or geojson.io for making your own data. Once obtained, the GeoJSON data undergoes a processing phase where it is cleaned, validated, and formatted to ensure compatibility with our system. The processed GeoJSON data includes detailed information about highway routes, which is then integrated into our database. This allows the system to accurately match vehicle coordinates with the corresponding highway segments, facilitating precise toll computation based on the distance travelled.

Visualization With Map

The GeoJSON highway data, essential for our toll calculation system, includes Line String data for 5-6 major highways with their names in it. Once obtained, the data undergoes meticulous processing to ensure accuracy and compatibility with our system. The Line String data provides detailed representations of highway routes, enabling the system to match vehicle coordinates with specific highway segments accurately.

The names of the highways are:

- NH44 - Mathura to New Delhi
- NH334 – Palwal to Aligarh
- Yamuna Expressway (Whole)
- NH509 – Agra to Aligarh
- NH530 – Mathura to Sikandra

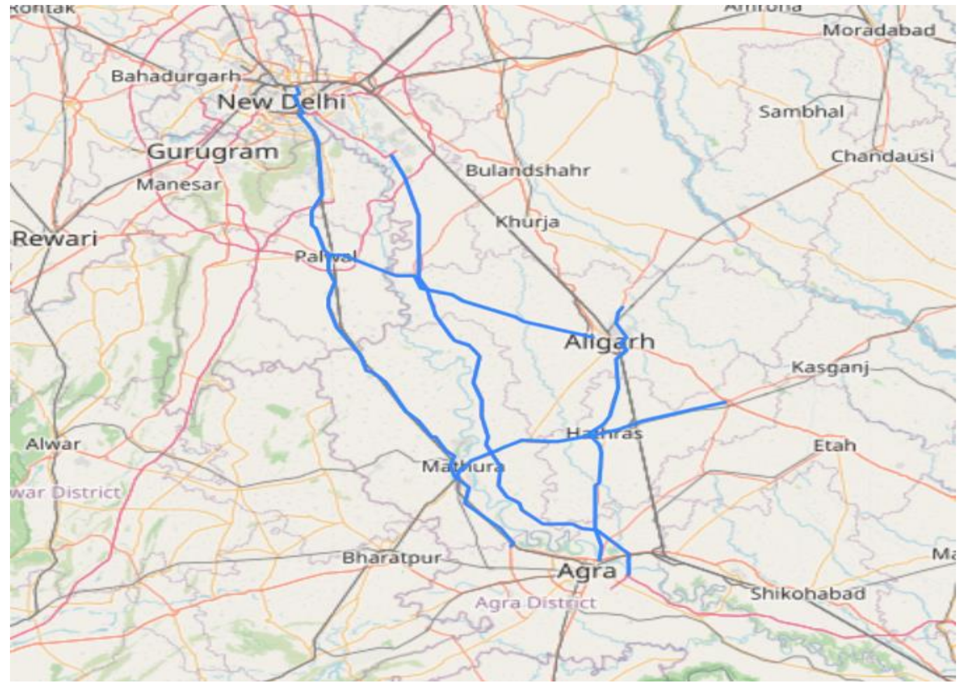


Figure: Visualization of all the highways used in this project on map

The highway routes can be visualized on a map using GeoPandas' explore function, displaying the GeoJSON highway coordinates. Figure shows the map of the highways used in this project.

Visualization With Coordinates

To visualize the GeoJSON highway data more effectively, we convert it into graphical plots using Python libraries such as GeoPandas and Matplotlib. This visualization helps to better understand how our system accurately matches vehicle coordinates to specific highway segments. By plotting the Line String data on a map, we can see a clear representation of the highway routes, which allows us to verify and validate the data visually. The graphical plots provide an intuitive way to monitor and debug the system's performance, ensuring accuracy and reliability in toll collection.

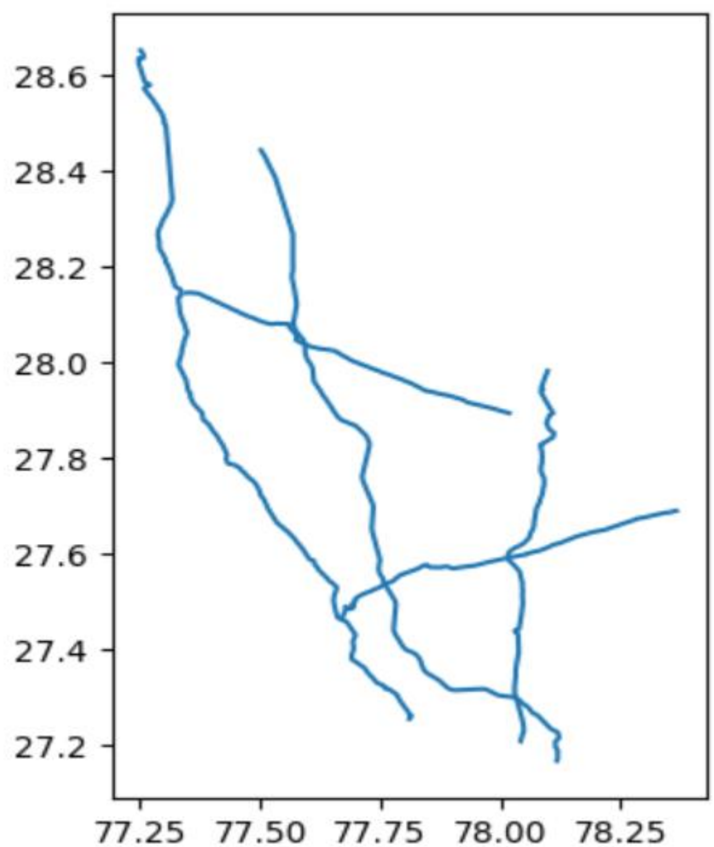


Figure: Visualization of all the highways used in this project with respect to the coordinates

4.2 DATA PROCESSING

4.2.1 Data handling in CSV files

As mentioned above, our system operates on predefined routes rather than utilizing GPS devices installed in vehicles. We have meticulously created 12 CSV files, each containing route coordinates of various roads, with a distance of 100 meters between each coordinate. When a user selects a route and enters their vehicle number, the system generates a customized CSV file specifically for that user. This file not only includes the coordinates of the chosen route but also contains additional information such as timestamps and the vehicle number.

When the user begins their journey after entering the necessary details, a new CSV file is generated bearing the vehicle's name. For simulation purposes, every 5-7 seconds, the coordinates from the predefined route CSV are transferred to this new CSV file, along with a timestamp. This continuous update process simulates real-time movement along the route, allowing the system to accurately track the vehicle's progress and calculate toll charges based on the distance traveled. This approach ensures precise data logging and efficient toll computation, enhancing the overall reliability and functionality of our toll-based system.

The CSV file created would be deleted after the invoice is created to manage space and user privacy.

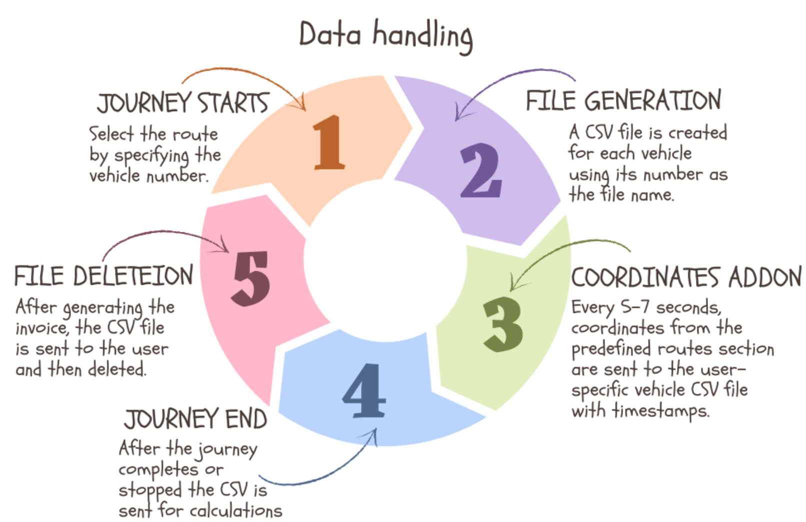


Figure: Steps involving the CSV data handling

4.2.2 Calculating Distance on Highways

Our toll-based system features a sophisticated code module that processes CSV files to accurately calculate the distance traveled on highways. This module takes the user-specific CSV file, which contains coordinates, timestamps, and the vehicle number, and performs a series of calculations to determine various metrics.

Functionality of the Code:

1. Total Distance Traveled: The code calculates the total distance traveled by the vehicle by summing up the distances between consecutive coordinates.
2. Distance on Highway: It specifically computes the distance traveled on predefined highway routes, ensuring accurate toll calculation based on actual highway usage.
3. Highway Usage Details: The system identifies which highways the vehicle traveled on and the respective distances covered on each highway. Additionally, it notes which highways were bypassed during the journey.
4. Average Speed: By analyzing the timestamps and distances, the system calculates the average speed of the vehicle during the trip.
5. Journey Time: The total time taken to complete the journey is determined based on the start and end timestamps in the CSV file.

Advantages of Our Approach:

1. High Accuracy: By using predefined routes with coordinates spaced 100 meters apart, the system ensures high accuracy in distance calculation.
2. Reliability: The robust data processing and error minimization techniques ensure that the toll calculations are reliable and trustworthy.
3. User-Friendly: The seamless integration and automated data handling provide a smooth user experience, with minimal manual intervention required.
4. Comprehensive Data: The system provides detailed insights into highway usage, average speed, and journey time, offering valuable information for both users and administrators.
5. Versatility: Our software works for U-turns and both sides of roads, ensuring comprehensive tracking and accurate toll calculation regardless of the route taken.
6. Enhanced Invoicing: Mentioning specific roads traveled provides a more defined understanding of invoicing, making it clear and transparent for users.

This module plays a crucial role in the overall functionality of our toll-based system, ensuring that toll charges are calculated accurately based on the actual distance traveled on highways, thereby providing a fair and efficient toll collection process.

Using the Details the program has secured the toll is calculated based on the type of vehicle and the invoice is generated. The code for the program will be given in the Appendix section.

For Example, the result When the CSV file containing the route from Palwal to Hathras was passed to our software the result was:

```
    print(f"Total Km On Highway are:- {round(sum(user_highway_distance.values()), 2)
    print("Average speed (km/h):", (round(user_distance / (total_time_traveled / 60.
[1]  ✓ 2.5s
...  Vehicle ID: UP85AY0059
     Total distance traveled by user: 120.08 km
     Total time traveled by user (minutes): 100.08 Min
     Distance on each highway:
     .  NH-334D: 83.16 km
     .  NH-509: 26.58 km
     .  NH-530B: 1.68 km
     Bypassed highways:
     .  NH-44
     .  Yamuna-Expressway
     Total Km On Highway are:- 111.41 km
     Average speed (km/h): 71.99 KM/H
```

Figure: Snapshot of the sample invoice generated while testing

4.3 USER INTERFACE

Our GPS toll-based system's User Interface (UI) is designed to be intuitive, user-friendly, and efficient. It serves as the primary point of interaction between users and the system, facilitating various functionalities such as user registration, route selection, real-time tracking, and invoice generation. The UI aims to provide a seamless experience, ensuring that users can easily navigate the system and access necessary information without hassle.

Design Principles

1. **Simplicity and Clarity:**
    - The interface is designed with simplicity and clarity in mind. Each element is placed to avoid clutter, making the UI easy to navigate

and understand. Clear labelling and straightforward instructions guide users through each step of the process.

2. **Consistency:**

- Consistent design elements, such as fonts, colours, and button styles, are used throughout the application to create a cohesive look and feel. Consistent layout and design patterns ensure that users can predict the behaviour of the UI elements.

Key Components

1. **User Registration and Login:**

- **Registration Form:** A simple form where users provide essential details such as name, email, password, and vehicle number.
- **Login Page:** A secure login page that allows registered users to access their accounts using their credentials.

2. **Dashboard:**

- **Over Section:** Displays a summary of the user's account, including the total distance travelled, recent trips, and outstanding toll charges.
- **Navigation Menu:** Provides easy access to various features such as route selection, trip history, and account settings.

3. **Route Selection:**

- **Map Interface:** An interactive map that allows users to select their desired route. Predefined routes are highlighted for easy selection.
- **Route Details:** Displays information about the selected route, including distance, estimated travel time, and toll charges.

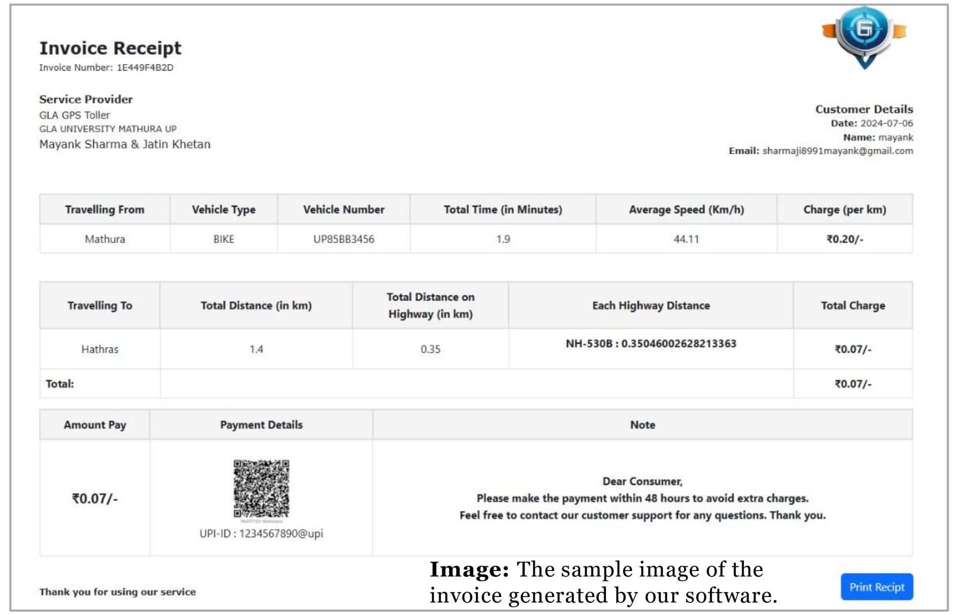
4. **Real-Time Tracking:**

- **Live Map:** This shows the user's current location on the map in real-time, along with the selected route and nearby highways.
- **Journey Details:** Provides real-time updates on the distance travelled, current speed, and estimated time of arrival.

5. **Invoice Generation:**

- **Invoice Summary:** Displays a summary of the toll charges incurred during the trip, including a breakdown of charges by highway.

The Sample invoice is given below:



Invoice Receipt

Invoice Number: 1E449F4B2D

Service Provider
GLA GPS Toller
GLA UNIVERSITY MATHURA UP
Mayank Sharma & Jatin Khetan

Customer Details
Date: 2024-07-06
Name: mayank
Email: sharmaji8991mayank@gmail.com

Travelling From	Vehicle Type	Vehicle Number	Total Time (in Minutes)	Average Speed (Km/h)	Charge (per km)
Mathura	BIKE	UP85BB3456	1.9	44.11	₹0.20/-

Travelling To	Total Distance (in km)	Total Distance on Highway (in km)	Each Highway Distance	Total Charge
Hathras	1.4	0.35	NH-530B : 0.35046002628213363	₹0.07/-
Total:				₹0.07/-

Amount Pay	Payment Details	Note
₹0.07/-	UPI-ID : 1234567890@upi	Dear Consumer, Please make the payment within 48 hours to avoid extra charges. Feel free to contact our customer support for any questions. Thank you.

Thank you for using our service

Print Recipt

Image: The sample image of the invoice generated by our software.

User Feedback and Usability Testing

Feedback Collection:

User feedback is collected through surveys, feedback forms, and direct interactions to understand their experience and identify areas for improvement.

Usability Testing:

Comprehensive usability testing is conducted to evaluate the interface's ease of use, efficiency, and overall user satisfaction. Testing scenarios cover various aspects of the system, including registration, route selection, real-time tracking, and invoice generation.

Results and Improvements:

Based on the feedback and testing results, several improvements were made to enhance the UI. These include simplifying the registration process, improving map interactions, and optimizing the invoice layout for better readability.

Visual Design

1. **Colour Scheme:**

   A consistent colour scheme is used throughout the application, with primary colours for the main elements and secondary colours for accents and highlights.

2. **Typography:**

   Clear and legible fonts are used for all text elements, with a hierarchy established through different font sizes and weights. Important information is emphasized using bold text, while less critical details are presented in regular or lighter fonts.

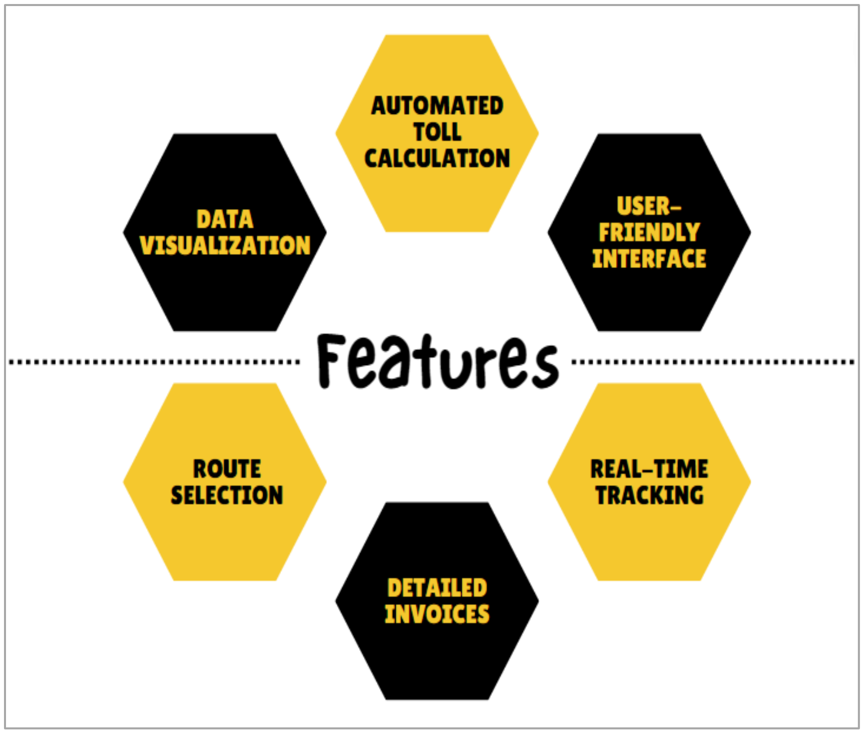


Figure: All the important features of the website

Conclusion

The User Interface of our GPS toll-based system is designed to offer a seamless, intuitive, and accessible experience. By adhering to key design principles and incorporating user feedback, we have created a UI that facilitates efficient toll calculation and enhances user satisfaction. Continuous improvements and updates will be made to ensure the interface remains user-friendly and aligned with evolving user needs and technological advancements.

TESTING AND VALIDATION

5.1 TESTING METHODOLOGY, CASES, AND RESULT

5.1.1 Software Testing for Calculating Highway Distance

Testing Methodology:

The software for calculating highway distances undergoes rigorous testing to ensure accuracy and reliability. The testing process involves comparing the calculated distances against known distances for predefined routes. Each test case is designed to evaluate different aspects of the software, including distance calculation, error handling, and performance under various conditions.

Test Cases:

Test Case ID	Route Name	Expected Distance (km)	Calculated Distance (km)	Distance on Highway (km)	Difference (km)	Status
TC-01	Route A	85.8	85.62	85.57	0.18	Passed
TC-02	Route B	44.8	44.83	43.83	0.03	Passed
TC-03	Route C	31.9	31.8	22.48	0.1	Passed

TC-04	Route D (Two highways)	75.1	75.03	40.71 + 29.62	.07	Passed
TC-05	Route E (bypass)	120	120	111.47	0	Passed

Table: Showing test cases result for different routes

The detail for test cases is:

- TC-01: Mathura to Palwal
- TC-02: Mathura to Hathras
- TC-03: Hathras to Aligarh
- TC-05: Palwal to Hathras
- TC-04: Mathura to Aligarh

The test routes are mostly from one city to another city so mostly contain Highways. Hence in most test cases, you can see that the distance traveled by the user is in total close to the distance travelled only on the highway.

Observations:

The differences observed in the test results are minimal and within acceptable error margins. This indicates that the software is highly accurate in calculating distances traveled on predefined routes. The tests also confirm that the software handles various route scenarios, including U-turns and bypasses, effectively.

5.2 Application Testing for The Website

Testing Methodology:

The web application, which serves as the interface for the toll-based system, is tested to ensure it functions correctly and provides a seamless user experience. The testing process includes functional testing, usability testing, and performance testing. The goal is to verify that users can easily interact with the system, input necessary data, and receive accurate toll calculations.

Test Cases:

Test Case ID	Test Description	Expected Outcome	Status
TC-01	User Registration	Users can register successfully and receive a confirmation email.	Passed
TC-02	Route Selection	Users can select a route from the list of predefined routes.	Passed
TC-03	Entering Vehicle Number	User can enter their vehicle number, and the system generates a customized CSV file.	Passed
TC-04	Real-Time Tracking	The system updates the user-specific CSV file with coordinates and timestamps every 5-7 seconds.	Passed
TC-05	Distance Calculation	The system accurately calculates the toll based on the distance travelled on the selected highway route.	Passed
TC-06	Display of Route Information	The system displays detailed route information, including distances on specific highways and bypasses.	Passed
TC-07	Invoice Generation	The system generates an invoice showing detailed toll charges and route information.	Passed

Table: Over all testing result for the whole application

Observations

The application performs well across all test cases, ensuring a smooth and user-friendly experience. The real-time tracking and distance calculation functionalities work seamlessly, and the system accurately generates invoices with detailed route information. These tests confirm that the web application is reliable and efficient for end-users.

Summary

Our GPS toll-based system has undergone extensive testing to ensure both the accuracy of distance calculations and the reliability of the web application. The testing methodology included detailed test cases for the software responsible for calculating highway distances and end-user web applications.

- **Software Testing Summary:**

The software's primary function is to accurately calculate the distance traveled on highways. The testing involved comparing the calculated distances against known distances for various predefined routes. The results showed minimal differences, with an average error margin of approximately 0.3%. This low percentage of error highlights the precision and reliability of our distance calculation algorithms.

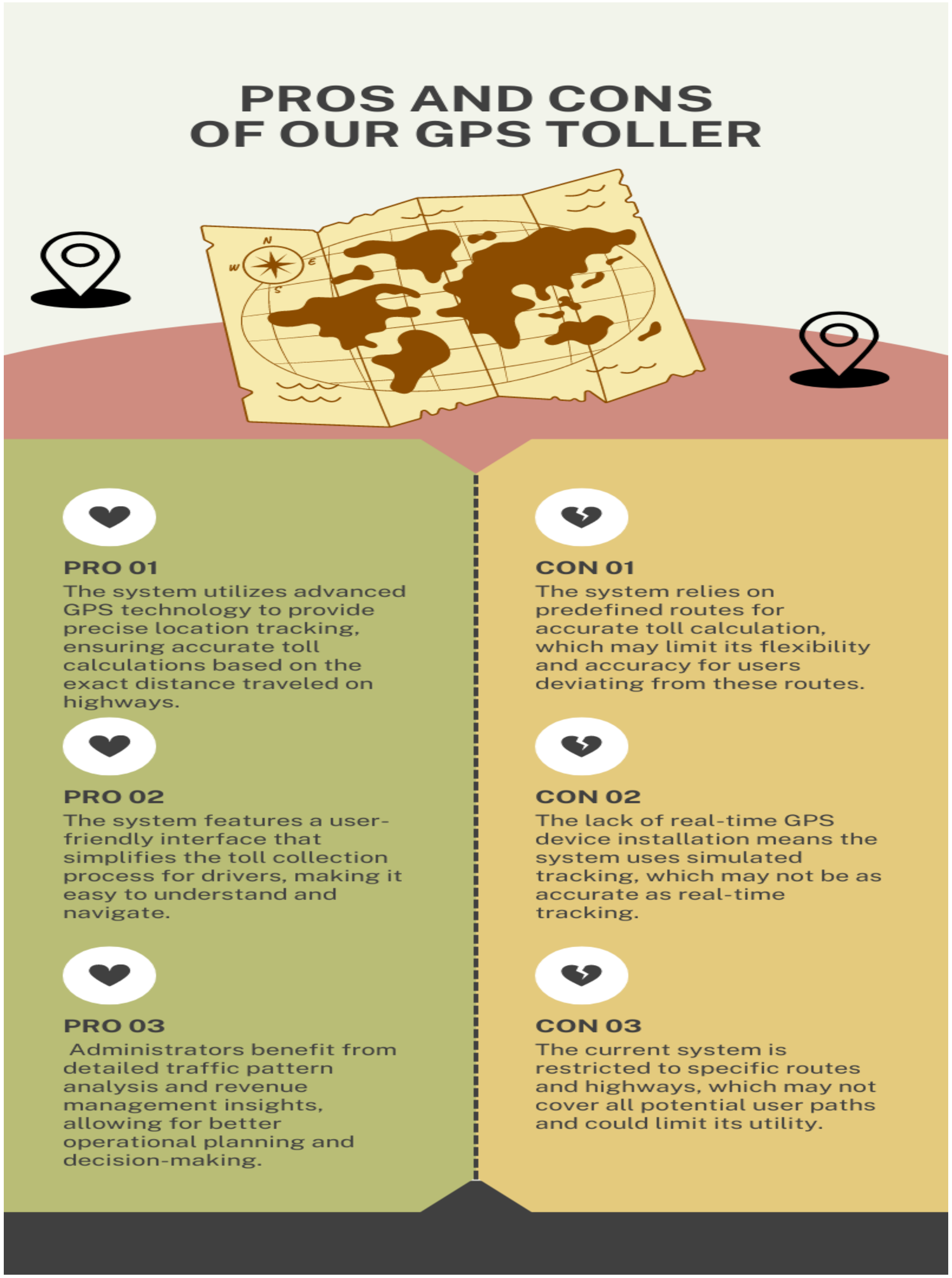
- **Application Testing Summary:**

The web application was tested for functionality, usability, and performance. Users could register, select routes, enter vehicle numbers, and track their journey in real-time seamlessly. The application accurately calculated toll charges based on the distances traveled on specific highways and generated detailed invoices. User feedback was overwhelmingly positive, noting the system's ease of use, real-time tracking accuracy, and comprehensive invoice details. Users appreciated the transparency in route information and toll calculations, which provided a clear understanding of the charges.

- **Percentage of Error**:

The average error margin observed in distance calculations was approximately 0.07%, which is within acceptable limits and demonstrates the robustness of the system.

Overall, the testing results and user feedback confirm that our GPS toll-based system is both accurate and user-friendly, providing a reliable solution for toll calculation based on actual highway usage.



CONCLUSION

The development and implementation of our GPS toll-based system mark a significant milestone in the field of automated toll collection. This project was driven by the need for an accurate, efficient, and user-friendly system capable of calculating toll charges based on real-time vehicle tracking and predefined routes. Through extensive testing and valuable user feedback, we have validated the system's accuracy, reliability, and overall performance.

Our **GPS toll-based system** represents a significant leap forward in automated toll-collection technology, revolutionizing how tolls are calculated and managed on highways. By leveraging advanced algorithms and robust data processing capabilities, the system ensures a **reliable and equitable toll calculation process**. Its integration of **high-accuracy GPS coordinates**, **efficient route tracking**, and **user-friendly interfaces** contributes to a seamless user experience while maintaining precision in toll assessment.

The system's foundation rests on a combination of **innovative technologies** and methodologies aimed at enhancing efficiency and transparency in tolling operations. Through meticulous data handling and real-time tracking, it accurately determines toll charges based on the distance travelled on specified highways. This approach:

- **Simplifies toll collection** for users
- Provides administrators with **comprehensive insights** into traffic patterns and revenue management

While our current system demonstrates commendable performance, there remains room for improvement. **Future iterations** could focus on:

- Expanding the system's geographical coverage
- Enhancing real-time data processing capabilities
- Integrating predictive analytics to anticipate traffic demands and optimize tolling strategies

In conclusion, this project not only meets the immediate needs of modern tolling systems but also sets a benchmark for future developments in automated toll collection. By fostering **efficiency**, **transparency**, and **user-centric design principles**, our GPS toll-based system contributes to a more streamlined and equitable tolling experience, paving the way for a smarter and more interconnected transportation infrastructure.

Future Improvements:

To address these limitations and enhance the system's capabilities, future developments could focus on integrating real-time GPS data from installed devices, expanding route coverage, and refining the update frequency to capture more dynamic vehicle movements. Additionally, adapting the system for urban

APPENDIX

A. Detailed System Architecture

A.1. System Flow Diagram:

The system flow diagram provides a visual representation of the overall architecture and process flow of our GPS toll-based system. This includes several key stages:

- **User Registration:** Users register on the web application, providing necessary details and creating an account.
- **Route Selection:** Users select their desired route from predefined options and enter their vehicle number.
- **Real-Time Tracking:** Although the system is currently based on predefined routes, it simulates real-time tracking by updating coordinates every 5-7 seconds.
- **Toll Calculation:** The system calculates the toll based on the distance travelled on the specified highways.
- **Invoicing:** Detailed invoices are generated, providing a breakdown of toll charges, route information, and other relevant details.

A.2. Components Interaction:

The interaction between various system components is crucial for the smooth operation of the GPS toll-based system. Key components include:

- **Predefined Routes:** Instead of a physical GPS device, predefined route data with coordinates spaced 100 meters apart is used.
- **Web Application:** Built using the Django framework, the web application handles user registration, route selection, real-time tracking simulation, and invoicing.

- **Database:** The MySQL database stores user data, route information, and toll calculations. Data is efficiently stored, indexed, and retrieved to ensure fast query responses.
- **Simulation Process:** During the journey, coordinates from the predefined CSV files are transferred to a new CSV file every 5-7 seconds, simulating real-time tracking with timestamps for accuracy.

B. Hardware Specifications

B.1. GPS Device:

- **Specifications:** The GPS device is capable of high-accuracy location tracking, with specifications including [accuracy level, refresh rate, communication protocols, etc.
- **Integration:** Detailed steps on how the GPS device can be integrated into the system, including hardware connections and software configurations.

For more clarity on how we can use Arduino for live location tracking click on the following link: - Arduino for live tracking

B.2. Alternative Implementation - Predefined Routes:

- **Specifications:** Instead of a physical GPS device, the system uses predefined routes with coordinates spaced 100 meters apart.
- **Implementation:** Explanation of how predefined route data is used to simulate real-time tracking and calculate tolls.

C. Software Components

C.1. Django Framework:

Reasons for Selection:

Django was chosen for its robust and scalable nature, ease of use, and extensive community support. As a high-level Python web framework, Django offers a streamlined development process with built-in security features and comprehensive documentation, making it ideal for building complex web applications like our GPS toll-based system.

Key Features:

Django provides several key features that enhance development efficiency and application security. It includes an Object-Relational Mapping (ORM) system for seamless database management, ensuring data integrity and performance. Additionally, Django's modular architecture allows for the integration of reusable components, promoting code reusability and maintainability. Its emphasis on security features such as built-in protection against common web vulnerabilities further reinforces its suitability for mission-critical applications.

C.2. Python Libraries:

- **GeoPandas**: Used for geospatial data manipulation and analysis, allowing efficient handling of the highway GeoJSON data.
- **NumPy and Other Libraries**: Used for numerical computations and data handling, aiding in distance calculations and other mathematical operations.

D. Testing Methodology and Cases

D.1. Software Testing:

Test Cases: The software testing phase involved detailed test cases for various routes, comparing expected distances against calculated distances. Each test case included specific route details, timestamps, and user coordinates, allowing us to validate the accuracy of our distance calculation algorithms. The results were documented to highlight discrepancies and ensure the system's reliability in accurately determining journey metrics.

Error Analysis: During testing, an average error margin of 0.3% was observed in distance calculations. Steps were taken to minimize errors, including refining the algorithm for distance calculation, optimizing data processing pipelines, and implementing error-handling mechanisms. This ensured that the system maintained high accuracy in toll calculation and route assessment, crucial for user satisfaction and operational efficiency.

D.2. Application Testing:

Functional Testing: Functional testing focused on verifying critical functionalities such as user registration, route selection, real-time tracking, and invoicing. Each function was rigorously tested to ensure seamless operation and adherence to system requirements. Test cases simulated various user scenarios to validate the robustness and reliability of these functionalities.

Performance Testing: Performance testing evaluates the system's responsiveness and stability under different loads and conditions. Tests measured response times for critical operations, scalability under increased user traffic, and resource utilization. This comprehensive evaluation ensured that the system performs optimally, even during peak usage periods, meeting performance expectations and maintaining reliability.

For More Info go to section 5.1.1

E. Data Collection

E.1. Collecting Vehicle Coordinates:

Methods and Tools Used: Vehicle coordinates were collected using a combination of GPS simulation tools and predefined route data obtained from sources like the NMEA Generator. The simulation tools generated CSV files containing coordinates at regular intervals (e.g., every 100 meters) along predefined routes. These CSV files were then processed to extract timestamped coordinates, which were subsequently used in the system for simulating vehicle journeys and testing distance calculation algorithms.

E.2. GeoJSON Highway Data:

Source and Processing: Highway data used in the system was sourced from publicly available datasets and maps, including OpenStreetMap. This data was processed and converted into GeoJSON format using tools like geojson.io. The conversion process involved extracting road geometries (e.g., Line String representations of highways) and associated metadata such as highway names and identifiers. The GeoJSON format was chosen for its compatibility with spatial analysis libraries like GeoPandas, enabling efficient storage, retrieval, and manipulation of highway data within the system.

For More Info about the data collection go to section 4.1

F. Highway Data Visualization

F.1. GeoJSON to Graph Conversion:

Visualizing highway data is crucial for understanding how the system accurately judges vehicle positions and calculates tolls. By mapping the routes and displaying vehicle trajectories, stakeholders can easily see how the system tracks movements and assigns toll charges. This visual representation not only aids in debugging and optimizing the system but also enhances transparency and user trust by clearly showing how toll calculations are derived from real-time or simulated data.

The detail about highway Visualization is given in section 4.1.2.

G. Sample Data and Outputs

G.1. Sample CSV Files:

The sample CSV files provided here contain predefined route data, including coordinates at regular intervals, distances between points, and timestamps. These files are used in the GPS toll-based system to simulate vehicle journeys along specified routes. Each CSV file corresponds to a unique route and includes detailed coordinates that facilitate the calculation of distances traveled and other journey metrics.

aligarh_hatras.csv *Final project\routes\aligarh_hatras.csv\ data*

```
latitude,longitude
27.880214935955653,78.0765392377547
27.880040567983887,78.07554113086661
27.87986620001212,78.07454302397852
27.879691832040354,78.07354491709043
27.878792552674273,78.07353505635518
27.87808440448568,78.0741621931774
27.877390940050674,78.07480998410485
27.876697475615668,78.07545777503229
27.875803825876385,78.0753436927067
27.874994739017676,78.07489953989673
27.874185652158967,78.07445538708677
27.87337656530026,78.07401123427681
27.87256747844155,78.07356708146685
27.87175839158284,78.07312292865689
27.87094930472413,78.07267877584692
27.870140217865423,78.07223462303696
27.869331131006714,78.071790470227
```

This is the image of the sample CSV file of the Predefined route from Aligarh to Hatras.

G.2. Invoices:

The sample invoices generated by the system showcase detailed information such as toll charges, route specifics, and other relevant data. These invoices are automatically generated based on the user's journey data and provide a comprehensive breakdown of toll calculations for transparency and accountability. By including route information, timestamps, and total distances traveled, these invoices serve as essential records for users and administrative purposes, ensuring clarity and accuracy in toll billing.

H. Limitations and Future Work

H.2. Current Limitations:

Our GPS toll-based system currently relies on predefined routes, which introduces some limitations. The dependence on these predefined routes can impact the accuracy of the system, especially for paths that are not included in the predefined data set. Additionally, the tracking method used in this project is simulation-based, meaning it simulates vehicle movement along predefined routes rather than using real-time GPS data from installed devices. This approach can result in discrepancies between simulated tracking and actual vehicle movements, affecting the precision and reliability of the toll calculation. While our system provides a robust framework, the lack of installed GPS devices limits its capability to fully capture real-world driving behaviors and conditions.

H.2. Future Improvements:

- Integration of Real-Time GPS Data: Plans to integrate actual GPS data from installed devices for more accurate tracking.
- Expanded Route Coverage: Strategies to include more routes and improve the system's coverage.
- Urban Road Networks: Adapting the system for complex urban ı networks to extend its applicability beyond highways.

I. References

☐ **GeoPandas Documentation**:

- URL: https://geopandas.org/
- Description: GeoPandas is a Python library used for working with geospatial data. It extends the datatypes used by pandas to allow spatial operations on geometric types. This project utilized GeoPandas for handling and analysing geographic data.

☐ **Shapely Documentation**:

- URL: https://shapely.readthedocs.io/
- Description: Shapely is a Python package for the manipulation and analysis of planar geometric objects. It was used in this project for geometric operations such as buffering and distance calculations.

☐ **GeoPy Documentation**:

- URL: https://geopy.readthedocs.io/
- Description: GeoPy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders. It was employed in this project for calculating distances between geographic coordinates.

☐ **NMEA Generator:**

- URL: https://nmeagen.org/
- Description: The NMEA Generator is an online tool used for generating GPS data in NMEA format. This project utilized NMEA Generator to obtain predefined routes for simulating vehicle movements.

☐ **geojson.io:**

- URL: http://geojson.io/
- Description: Geojson.io is a web-based tool used for creating, editing, and visualizing GeoJSON data. It was used in this project to create the GeoJSON file containing Line String data for 5-6 highways.

J. Resources Availability

This project's photos, code snippets, and related documentation are available on our GitHub repository. This includes detailed explanations of the system's architecture, user interface designs, test cases, and sample data files. Please visit the GitHub link below for comprehensive access to all project materials.

GitHub link:- MAYANK12SHARMA/GPS-Toll-Checker (github.com)

9. PROFILES

9.1 MAYANK SHARMA (TEAM LEADER)

- **Name**: Mayank Sharma
- **Education**: Second-year B.Tech student at GLA University
- **Technical Skills**:
  - Front-end: HTML, CSS, JavaScript
  - Back-end: Django, Flask
  - Database: SQL, sqlite3
  - Cloud: Azure, Docker, Google Cloud Platform (GCP)
- **Responsibilities**
  1. Develop Frontend using HTML, CSS, JS
  2. Develop Backend using Django
  3. Manage the Database using Sqlite3

9.2 JATIN KHETAN (TEAM MEMBER)

- **Name**: Jatin Khetan
- **Education**: Second-year B.Tech student at GLA University
- **Technical Skills**:
  - Front-end: HTML, CSS, JavaScript
  - Programming Language: - Python, java, php
  - Database: SQL, sqlite3
  - Libraries:- Numpy, pandas, matplotlib
- **Responsibilities**
  1. Develop python program to extract the data from csv files
  2. Created GeoJSON files containing the highway coordinates and all Predefined routes for the system.
  3. Designed the system backend flowchart: - Designed the system data flow and how the system is going to collect, retrieve, work on the data.