

# **Road Accident Analysis and Classification System**

SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENT OF THE AWARD  
OF DEGREE OF

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE**



Submitted by

AYUSH PRATAP SINGH (2000290120054)

ARTH SRIVASTAVA (2000290120041)

ANUJ JAIN (2000290120035)

Supervised by

HARSH KHATTER

**Session 2023-24**

DEPARTMENT OF COMPUTER SCIENCE

**KIET GROUP OF INSTITUTIONS, GHAZIABAD**

**(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India)**

## DECLARATION

I/We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Date : 06/05/2024

Signature :

Name : Ayush Pratap Singh

Roll No : 2000290120054

Signature :

Name : Arth Srivastava

Roll No : 2000290120041

Signature :

Name : Anuj Jain

Roll No : 2000290120035

## **CERTIFICATE**

This is to certify that Project Report entitled “Road Accident Analysis and Classification System” which is submitted by Ayush Pratap Singh, Arth Srivastava and Anuj Jain in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

**Date: 06/04/2024**

**Supervisor**

Dr. Harsh Khatter

Associate Professor

## ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Professor Harsh Khatter, Department of Computer Science, KIET, Ghaziabad, for his/her constant support and guidance throughout the course of our work. His/Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Ajay Kumar Shrivastava, Head of the Department of Computer Science, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date : 06/05/2024

Signature :

Name : Ayush Pratap Singh

Roll No : 2000290120054

Signature :

Name: Arth Srivastava

Roll No : 2000290120041

Signature :

Name : Anuj Jain

Roll No : 2000290120035

## **ABSTRACT**

Road accidents are a significant concern globally, leading to loss of life, injuries, and economic burdens. This project introduces a Road Accident Analysis and Classification System that leverages machine learning techniques to predict accident severity and identify high-risk zones. Using data from the RTA Dataset UK (2012-2014), the system integrates various parameters such as weather conditions, road surface conditions, vehicle characteristics, and driver attributes. Supervised learning algorithms, including Decision Trees, Random Forest, Naive Bayes, and Logistic Regression, are employed for classification, with a voting classifier maximizing model accuracy. Integration with a user-friendly web application, along with Google Maps and weather APIs, enhances accessibility and usability. Results demonstrate a significant improvement in accident classification accuracy, with an overall model accuracy of 86.73%. The system's live severity feature enables on-the-go assessment of accident severity, contributing to proactive response measures. This project represents a proactive approach towards mitigating road accidents and enhancing traffic safety through data-driven solutions.

# TABLE OF CONTENTS

	<b>Page No.</b>
DECLARATION.....	ii
CERTIFICATE.....	iii
ACKNOWLEDGEMENT.....	iv
ABSTRACT.....	v
LIST OF FIGURES.....	xiii
LIST OF TABLES.....	ix
LIST OF ABBREVIATIONS.....	x
 CHAPTER 1 INTRODUCTION	
1.1 INTRODUCTION TO PROJECT	11
1.2 PROJECT CATEGORY	12
1.3 OBJECTIVES	13
1.4 PROBLEM FORMULATION	15
1.5 PROPOSED SYSTEM	16
1.6 UNIQUE FEATURES OF THE SYSTEM	18
 CHAPTER 2. REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION	
2.1 FEASIBILITY STUDY	19
2.2 SOFTWARE REQUIREMENT SPECIFICATION	21
2.3 SDLC MODEL TO BE USED	24

CHAPTER 3. SYSTEM DESIGN	
3.1	DETAIL DESIGN 26
3.2	SYSTEM DESIGN USING DFD LEVEL 0, LEVEL 1 AND LEVEL 2 28
3.4	MODEL FLOWCHART 30
CHAPTER 4. IMPLEMENTATION, TESTING, AND MAINTENANCE	
4.1	INTRODUCTION TO LANGUAGES,TOOLS AND TECHNOLOGIES USED FOR IMPLEMENTATION 31
4.2	TESTING TECHNIQUES AND TEST CASES USED 34
CHAPTER 5. RESULTS AND DISCUSSIONS	
5.1	USER INTERFACE REPRESENTATION (OF RESPECTIVE PROJECT) 43
5.1.1	BRIEF DESCRIPTION OF VARIOUS MODULES OF THE SYSTEM 45
5.2	SNAPSHOTS OF SYSTEM WITH BRIEF DETAIL OF EACH 47
5.3	BACK ENDS REPRESENTATION 51
CHAPTER 6. CONCLUSION AND FUTURE SCOPE 53	
REFERENCES 54	

## LIST OF FIGURES

<b>Fig no.</b>		<b>Page no.</b>
Fig. 1.1	System Architecture	16
Fig. 3.1	DFD Level 0	28
Fig. 3.2	DFD Level 1	28
Fig. 3.3	DFD Level 2	29
Fig. 3.4	Model Flowchart	30
Fig. 4.1	Manual Test Case Input (Location)	38
Fig. 4.2	Manual Test Case Input (Other Details)	38
Fig. 4.3	Manual Test Case Results	39
Fig. 4.4	Automation Testing	39
Fig. 4.5	Automation Testing Results	40
Fig. 4.6	Postman	40
Fig. 5.1	Home Page	43
Fig. 5.2	Results & Analysis	44
Fig. 5.3	Download Report	44
Fig. 5.4	Live Severity Simulation	45
Fig. 5.5	Location Selection Module	47
Fig. 5.6	Additional Details Input Module	48
Fig. 5.7	Downloaded Report	49
Fig. 5.8	Live Severity Feature	50



## LIST OF TABLES

<b>Table no.</b>		<b>Page no.</b>
Tab 4.1	Test Case	36
Tab 4.2	Decision Table	37
Tab 4.3	Acronym Table	42

## LIST OF ABBREVIATIONS

WHO	World Health Organization
ML	Machine Learning
GUI	Graphical User Interface
CSV	Comma Separated Values
API	Application Programming Interface
SRS	Software Requirement Specification
RTA	Road Traffic Accident
SDLC	Software Development Life Cycle
DFD	Data Flow Diagram
AUT	Application Under Test
QA	Quality Analyst
UAT	User Acceptance Testing

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 INTRODUCTION**

Road safety is a critical concern worldwide, with road accidents resulting in significant loss of life, injuries, and economic burdens. According to the World Health Organization (WHO), road traffic injuries are one of the leading causes of death globally, with approximately 1.35 million lives lost each year. In addition to the human toll, road accidents impose substantial economic costs, including medical expenses, property damage, and productivity losses.

The complexity of road accidents necessitates a multifaceted approach to address the underlying causes and mitigate their impact. One key aspect of this approach is the analysis of road accident data to identify patterns, trends, and risk factors. By understanding the factors contributing to accidents, policymakers, transportation authorities, and law enforcement agencies can develop targeted interventions and preventive measures to improve road safety.

In recent years, advancements in technology, particularly in the field of machine learning (ML), have revolutionized the way we analyze and interpret data. ML algorithms, capable of processing large volumes of complex data, have shown promise in various domains, including healthcare, finance, and transportation. In the context of road safety, ML techniques offer a powerful tool for predicting accident severity, identifying high-risk areas, and informing decision-making processes.

The objective of this project is to develop a Road Accident Analysis and Classification System that harnesses the potential of ML algorithms to enhance road safety. By leveraging real-time data from sources such as traffic sensors, weather stations, and vehicle telematics, the system aims to accurately predict accident severity and identify areas prone to accidents. Through the integration of user-friendly interfaces and interactive visualization tools, stakeholders can access actionable insights and make informed decisions to prevent accidents and improve traffic management.

## 1.2 PROJECT CATEGORY

In the context of project categorization, the endeavor aligns with "System Development." This classification denotes a project centered on designing, implementing, and integrating a functional system to address a specific need or problem. In this case, the focus lies on developing a comprehensive Road Accident Analysis and Classification System. This entails the creation of a software solution capable of analyzing road accident data, leveraging machine learning algorithms to classify accidents based on severity, and integrating real-time data sources such as traffic sensors, weather stations, and vehicle telematics.

The scope of this project involves several key components of system development:

1. **Requirement Analysis:** Understanding the needs and objectives of the project, including the desired functionalities and features of the road accident analysis system.
2. **Design and Architecture:** Creating a robust system architecture that outlines the structure, components, and interactions of the software solution. This involves designing the user interface, data processing modules, and integration with external data sources.
3. **Implementation:** Developing the software solution according to the defined design and architecture. This includes coding the algorithms for accident analysis, integrating APIs for real-time data retrieval, and building the user interface for interaction.
4. **Testing and Quality Assurance:** Conducting thorough testing to ensure the functionality, reliability, and performance of the system. This involves both manual and automated testing to identify and address any issues or bugs.
5. **Deployment and Integration:** Deploying the developed system in a production environment and integrating it with existing infrastructure or platforms. This may involve setting up servers, configuring databases, and ensuring seamless communication between different components.
6. **Maintenance and Support:** Providing ongoing maintenance and support services to ensure the continued functionality and efficiency of the system. This includes addressing user feedback, implementing updates or enhancements, and resolving any issues that may arise post-deployment.

## 1.3 OBJECTIVES

The objectives of the Road Accident Analysis and Classification System project are as follows:

1. **Enhance Road Safety:** Develop a system that contributes to improving road safety by analyzing historical road accident data and identifying patterns and trends that can inform preventive measures and interventions.
2. **Accurate Accident Classification:** Implement machine learning algorithms to classify road accidents based on severity, enabling stakeholders to prioritize resources and interventions for high-risk areas.
3. **Real-time Data Integration:** Integrate real-time data sources, including traffic sensors, weather stations, and vehicle telematics, to provide up-to-date information on road conditions and accident risk factors.
4. **Optimize Emergency Response:** Enable faster and more effective emergency response by predicting accident severity and providing actionable insights to emergency services personnel.
5. **User-Friendly Interface:** Design a user-friendly interface for the system, allowing stakeholders such as road authorities, emergency services, and policymakers to easily access and interpret the analysis results.
6. **Integration with Web Application:** Develop a web-based application to host the system, providing accessibility and usability across different devices and platforms.
7. **Utilize Google Maps and APIs:** Integrate Google Maps and other APIs, such as weather APIs, to enhance the visualization of accident data and provide additional contextual information for analysis.
8. **Provide Detailed Reports:** Generate detailed reports on accident severity zones, including current weather conditions, light conditions, road surface conditions, and other relevant factors, to facilitate informed decision-making by stakeholders.
9. **Implement Live Severity Feature:** Implement a live severity feature that calculates accident severity en route, enabling drivers to make informed decisions about their travel routes and driving behavior.

**10. Ensure Data Privacy and Security:** Implement measures to safeguard the privacy and security of the collected data, ensuring compliance with relevant regulations and standards.

By achieving these objectives, the project aims to contribute to the reduction of road accidents, improvement of traffic safety measures, and enhancement of emergency response systems, ultimately leading to safer roads and communities.

## 1.4 PROBLEM FORMULATION

The problem formulation for the Road Accident Analysis and Classification System project revolves around addressing the following challenges:

1. **Road Accident Severity Prediction:** Develop a predictive model capable of accurately assessing the severity of road accidents based on various contributing factors, such as weather conditions, road surface conditions, and driver behavior.
2. **Data Integration and Analysis:** Integrate diverse datasets, including historical road accident data, real-time traffic information, weather data, and vehicle telemetry, to perform comprehensive analysis and identify patterns or trends associated with road accidents.
3. **Machine Learning Model Selection:** Select appropriate machine learning algorithms and techniques to train classification models capable of effectively categorizing road accidents into different severity levels.
4. **Real-time Prediction and Response:** Design a system capable of providing real-time predictions of accident severity to enable prompt response by emergency services and stakeholders, thereby minimizing the impact of accidents on traffic safety.
5. **User Interface and Accessibility:** Develop a user-friendly interface for the system that allows stakeholders, including road authorities, emergency services, and policymakers, to easily access and interpret the analysis results for informed decision-making.
6. **Integration with Web Application and APIs:** Integrate the analysis system with a web-based application, leveraging APIs such as Google Maps and weather APIs to enhance data visualization and provide contextual information for analysis.
7. **Privacy and Security:** Ensure the privacy and security of the collected data through appropriate measures, including data encryption, access controls, and compliance with relevant regulations and standards.

By formulating and addressing these challenges, the project aims to develop a robust and effective system for analyzing road accidents, predicting severity levels, and facilitating timely response and decision-making to enhance road safety and mitigate the impact of accidents on communities.

## 1.5 PROPOSED SYSTEM

### 1.5.1 ARCHITECTURE

Before delving into the detailed architecture of the Road Accident Analysis and Classification System, it's crucial to understand the underlying framework that supports its functionality. The architecture of the system is designed to seamlessly integrate various components, including data collection, preprocessing, machine learning model training, real-time prediction, and user interface. This comprehensive framework enables the system to analyze diverse data sources, derive meaningful insights, and provide actionable information to stakeholders. Now, let's explore the architectural layout in detail to gain a deeper understanding of how the system operates.

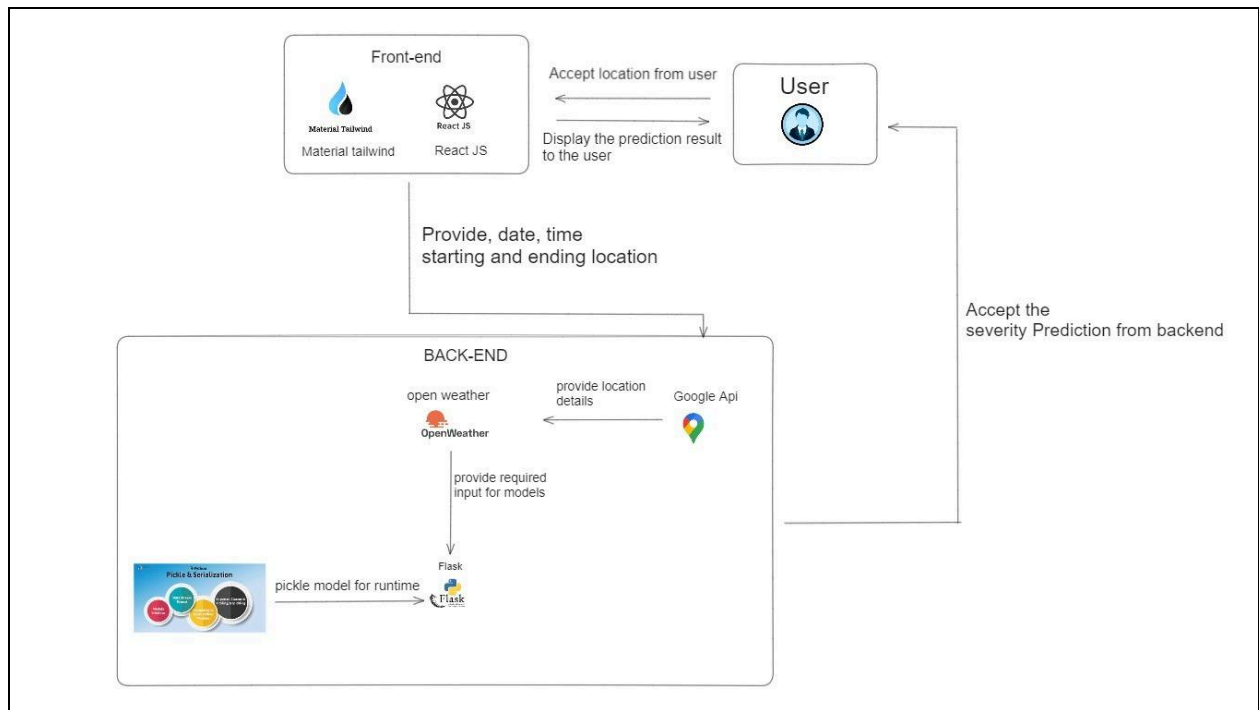


Fig 1.1 System Architecture



### **1.5.1.1 Front-End**

The frontend of the Road Accident Analysis and Classification System serves as the user-facing component, providing an intuitive and interactive interface for users to interact with the system. Built using cutting-edge technologies such as React.js along with Material Tailwind, the frontend offers a seamless and visually appealing experience.

With React.js at its core, the frontend ensures high performance and responsiveness, enabling smooth navigation and interactions. The use of Material Tailwind adds a layer of aesthetic appeal and consistency to the user interface, enhancing the overall user experience.

Through the frontend interface, users can access a range of functionalities, including real-time accident prediction, visualization of accident severity zones on maps, and detailed reports on high-risk areas. The interface is designed to be user-centric, with intuitive navigation and clear presentation of information.

### **1.5.1.2 Back-End**

The backend of the Road Accident Analysis and Classification System acts as the backbone of the entire application, responsible for processing data received from the frontend and external APIs, as well as hosting the machine learning model for accident prediction.

Built using Flask, a lightweight and flexible Python web framework, the backend provides a robust and scalable infrastructure for handling data processing tasks and serving predictive functionalities. Flask's simplicity and extensibility make it an ideal choice for implementing the backend of our system.

The backend receives data from the frontend interface, including user inputs and requests for accident prediction and analysis. Additionally, it interfaces with external APIs such as Google Maps and weather APIs to fetch real-time data on road conditions, weather forecasts, and other relevant parameters. This data is then processed and integrated into the predictive model to enhance its accuracy and reliability.

## 1.6 UNIQUE FEATURES OF THE SYSTEM

The Road Accident Analysis and Classification System boasts several unique features that set it apart from traditional approaches to road safety management:

1. **Real-time Accident Prediction:** Leveraging machine learning algorithms and real-time data integration, the system can accurately predict accident severity, enabling proactive measures to be taken to mitigate risks and improve emergency response times.
2. **Integration with External APIs:** By interfacing with external APIs such as Google Maps and weather APIs, the system enriches its predictive capabilities with up-to-date information on road conditions, weather forecasts, and other relevant factors influencing accident occurrence.
3. **User-friendly Web Application:** The system features a user-friendly web application with an intuitive graphical user interface (GUI), allowing road users and authorities to easily access and interact with accident data, severity zones, and detailed reports.
4. **Detailed Accident Reports:** In addition to predicting accident severity, the system provides detailed reports on high-risk areas, including current weather conditions, light conditions, road surface conditions, and other crucial factors, aiding in informed decision-making and resource allocation.
5. **Exportable Data:** Users have the option to export detailed accident reports in CSV format, facilitating further analysis and sharing of insights with stakeholders and relevant authorities.
6. **Live Severity Feature:** The system includes a live severity feature that calculates accident severity en route, providing real-time updates on potential risks and hazards to road users.
7. **Machine Learning Model Integration:** With a Flask backend hosting the machine learning model, the system seamlessly integrates predictive functionalities into its architecture, ensuring efficient accident prediction and analysis.

These unique features collectively enhance road safety management efforts, empowering stakeholders with actionable insights and tools to address the pervasive issue of road accidents effectively.

## CHAPTER 2

### REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

#### 2.1 FEASIBILITY STUDY

##### 1. Technical Feasibility:

- The project leverages modern technologies such as React for the frontend, Flask for the backend, and various APIs for data integration. These technologies are widely supported and offer robust development frameworks, ensuring technical feasibility.
- Machine learning models for accident severity prediction require significant computational resources for training and inference. However, with advancements in cloud computing and the availability of scalable infrastructure, the technical requirements can be met efficiently.

##### 2. Economic Feasibility:

- The project's economic feasibility is dependent on factors such as development costs, maintenance expenses, and potential returns on investment.
- Open-source tools and libraries are utilized wherever possible to minimize development costs. Additionally, the modular architecture allows for scalability and incremental updates, reducing long-term maintenance expenses.
- The potential benefits of the system, including improved road safety, reduced accident-related costs, and enhanced emergency response efficiency, justify the investment in developing and deploying the system.

##### 3. Operational Feasibility:

- The system is designed with user-friendliness in mind, featuring an intuitive web interface accessible to both road users and authorities. Training requirements for end-users are minimal, ensuring smooth adoption and operation.
- Integration with external APIs such as Google Maps and weather APIs streamlines data acquisition processes, enhancing operational efficiency.

- Regular updates and maintenance ensure that the system remains functional and up-to-date with evolving road conditions and user needs, further bolstering operational feasibility.

#### **4. Legal and Regulatory Feasibility:**

- Compliance with data protection regulations, privacy laws, and road safety standards is crucial for the system's deployment and operation.
- Measures are in place to ensure that user data is handled securely and in accordance with applicable regulations, mitigating legal and regulatory risks.
- Collaboration with relevant authorities and stakeholders helps align the system's functionality with existing legal frameworks and regulatory requirements, ensuring compliance at all levels.

#### **5. Schedule Feasibility:**

- A well-defined project plan with clear milestones and deliverables ensures that the development and deployment schedule remains feasible.
- Agile development methodologies allow for iterative development and rapid prototyping, enabling adjustments to be made based on feedback and changing requirements.
- Close collaboration between development teams, stakeholders, and end-users helps identify potential bottlenecks and mitigate risks, ensuring that the project progresses according to schedule.

Overall, the feasibility study indicates that the Road Accident Analysis and Classification System is viable from technical, economic, operational, legal, regulatory, and schedule perspectives, paving the way for its successful implementation and deployment.

## 2.2 SOFTWARE REQUIREMENT SPECIFICATION

### 1. Introduction:

- The Software Requirement Specification (SRS) outlines the functional and non-functional requirements of the Road Accident Analysis and Classification System. It serves as a blueprint for the development team and provides a clear understanding of the system's capabilities and constraints.

### 2. Functional Requirements:

- Data Acquisition:
  - The system shall collect real-time data from various sources, including the RTA (Road Traffic Accident) dataset spanning from 2012 to 2014, sourced from the UK.
  - It shall integrate historical data from the RTA dataset with live inputs from external APIs such as traffic sensors, weather stations, and vehicle telematics.
- Accident Severity Prediction:
  - The system shall employ machine learning algorithms to predict the severity of road accidents based on input parameters such as weather conditions, road surface conditions, and vehicle characteristics.
  - It shall classify accidents into severity levels (e.g., minor, moderate, severe) for efficient emergency response.
- Data Visualization:
  - The system shall provide visualizations of accident severity zones on a map interface, indicating high-risk areas.
  - It shall offer detailed reports on accident-prone areas, including current weather conditions, light conditions, and road surface conditions.
- Web Application:
  - The system shall feature a user-friendly web interface accessible to road users and authorities.
  - It shall allow users to view severity zones, access detailed reports, and export data for analysis.
- Real-Time Updates:

- The system shall continuously update accident severity predictions based on incoming data, ensuring up-to-date information for users.

### 3. Non-Functional Requirements:

- Performance:
  - The system shall be capable of handling a large volume of data and concurrent user requests without significant latency.
  - It shall provide accurate predictions with a high level of confidence, minimizing false positives and false negatives.
- Security:
  - User data shall be encrypted during transmission and storage to protect privacy and prevent unauthorized access.
  - Access to sensitive features such as data export and administrative controls shall be restricted to authorized users.
- Reliability:
  - The system shall operate reliably under varying environmental conditions and network disruptions.
  - It shall include failover mechanisms and data backup procedures to ensure continuous availability and data integrity.
- Scalability:
  - The system architecture shall be scalable to accommodate future growth in data volume and user traffic.
  - It shall support horizontal and vertical scaling strategies to handle increased demand without sacrificing performance.
- Usability:
  - The system interface shall be intuitive and easy to navigate, requiring minimal training for end-users.
  - It shall adhere to accessibility standards to ensure inclusivity and usability for users with diverse needs.

### 4. System Constraints:

- Compatibility:

- The system shall be compatible with modern web browsers and mobile devices to ensure broad accessibility.
  - It shall support multiple operating systems and screen resolutions to accommodate diverse user preferences.
  - Data Privacy:
    - The system shall comply with data protection regulations and industry standards for handling sensitive user information.
    - It shall implement data anonymization techniques to safeguard user privacy while still providing valuable insights.
5. Assumptions and Dependencies:
- The system assumes the availability of reliable data sources for accident-related information, including the RTA dataset and external APIs.
  - It depends on the continuous availability of external APIs such as Google Maps and weather APIs for real-time data integration.

The Software Requirement Specification provides a detailed overview of the functional and non-functional requirements, constraints, assumptions, and dependencies of the Road Accident Analysis and Classification System, laying the foundation for its successful development and deployment.

## 2.3 SDLC MODEL TO BE USED

The Waterfall Model is selected as the Software Development Life Cycle (SDLC) model for the development of the Road Accident Analysis and Classification System. The Waterfall Model follows a linear and sequential approach, where each phase is completed before moving on to the next one. The phases of the Waterfall Model used in this project are as follows:

1. Requirements Analysis:
  - In this phase, the project requirements are gathered and analyzed in detail. This includes understanding the needs of stakeholders, defining system functionalities, and documenting the requirements in a comprehensive manner.
2. System Design:
  - The System Design phase focuses on designing the architecture of the system based on the requirements gathered in the previous phase. This involves creating high-level and low-level design documents, defining system components, and outlining the interaction between different modules.
3. Implementation:
  - During the Implementation phase, the actual development of the system takes place. The design specifications are translated into code, and the various modules of the system are implemented according to the defined architecture. This phase also involves unit testing to ensure the correctness of individual components.
4. Testing:
  - The Testing phase involves verifying and validating the developed system to ensure that it meets the specified requirements and quality standards. This includes various testing activities such as functional testing, integration testing, performance testing, and user acceptance testing.
5. Deployment:
  - In the Deployment phase, the developed system is deployed to the production environment or made available for users. This involves installing the system, configuring it for production use, and ensuring that it operates correctly in the intended environment.
6. Maintenance:



- The Maintenance phase involves ongoing support and maintenance of the deployed system. This includes addressing any issues or defects identified post-deployment, implementing enhancements or updates, and providing user support as needed.

## **CHAPTER 3**

### **REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION**

#### **3.1 DETAIL DESIGN**

1. System Architecture:

- Describe the overall architecture of the system, including its high-level components and their interactions.
- Highlight the frontend and backend components, as well as any external APIs or services integrated into the system.
- Discuss the architectural patterns or design principles used to ensure scalability, maintainability, and reliability.

2. Component Design:

- Provide a detailed breakdown of the individual components of the system, including their functionalities and interfaces.
- Describe the purpose and role of each component within the overall system architecture.
- Discuss any design patterns or best practices followed in designing the components.

3. Technical Considerations:

- Discuss any technical considerations or constraints that influenced the design decisions.
- Address factors such as performance requirements, scalability, security, and compatibility with existing systems or technologies.
- Explain how these considerations were incorporated into the design of the system architecture and components.

4. Data Management:

- Outline the data management strategy employed in the system, including data storage, retrieval, and manipulation.
- Describe the database schema or data model used to organize and represent the data.

- Discuss any data processing or transformation techniques applied to ensure data integrity and consistency.
5. Integration with External Systems:
- Explain how the system integrates with external APIs, services, or third-party platforms.
  - Discuss the protocols or standards used for communication with external systems.
  - Describe any challenges or considerations related to interoperability and data exchange between the system and external systems.
6. Security Considerations:
- Address the security measures implemented to protect the system against potential threats or vulnerabilities.
  - Discuss authentication, authorization, data encryption, and other security mechanisms employed to safeguard sensitive information.
  - Explain how security considerations were integrated into the design of the system architecture and components.
7. User Interface Design:
- Provide an overview of the user interface design, including the layout, navigation, and visual elements.
  - Describe the user interactions supported by the system and the workflow for accessing different features or functionalities.
  - Discuss any usability testing or user feedback incorporated into the design process to improve the user experience.
8. Error Handling and Logging:
- Explain the approach to error handling and logging implemented in the system.
  - Describe how errors, exceptions, and system events are captured, logged, and reported to administrators or users.
  - Discuss any mechanisms in place for monitoring system performance and identifying potential issues proactively.

### 3.2 SYSTEM DESIGN USING DFD LEVEL 0, LEVEL 1 AND LEVEL 2

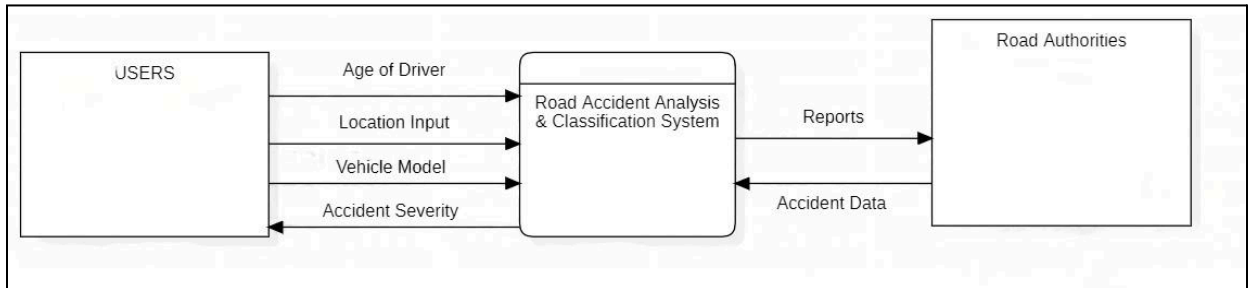


Fig. 3.1 DFD Level 0

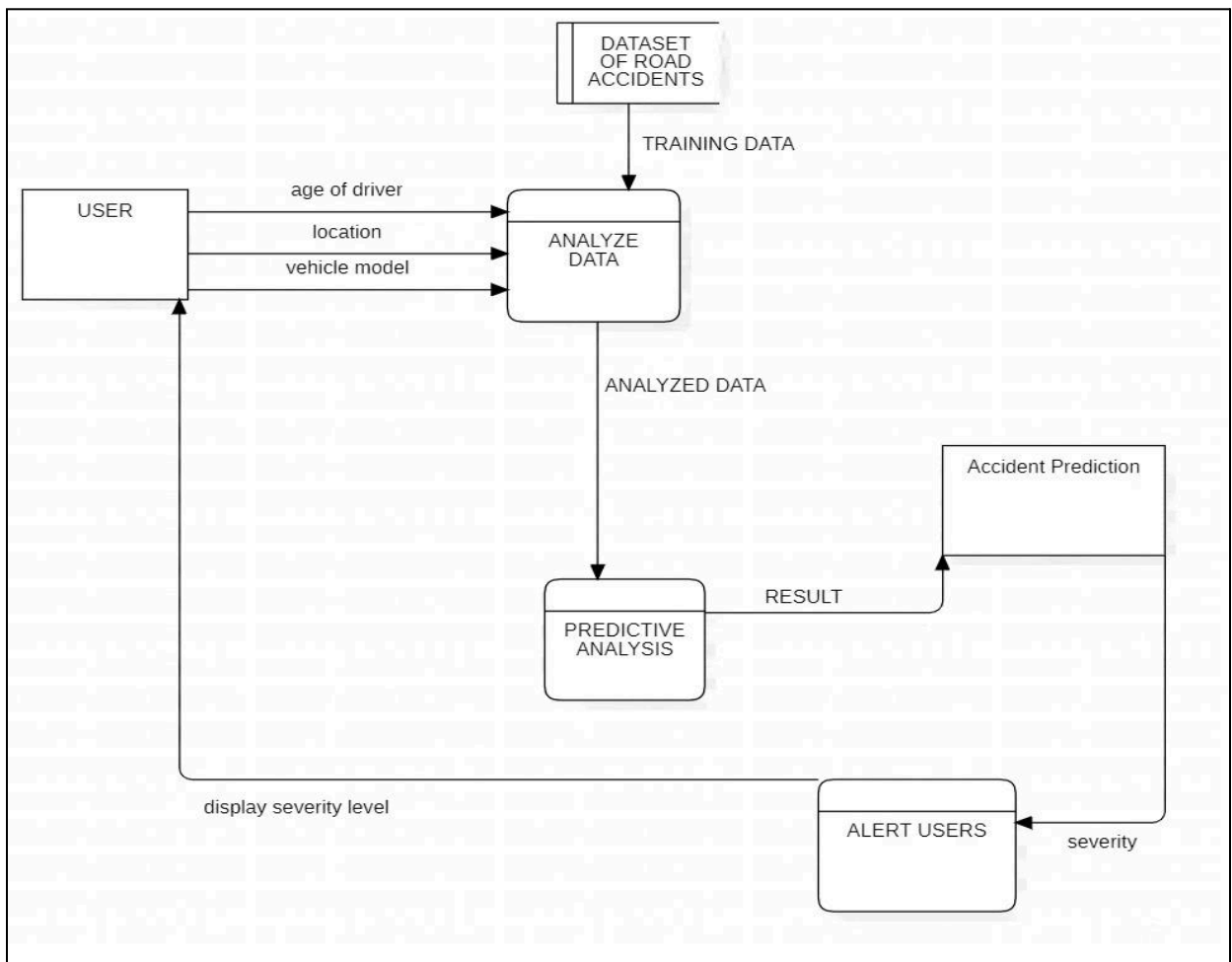


Fig. 3.2 DFD Level 1

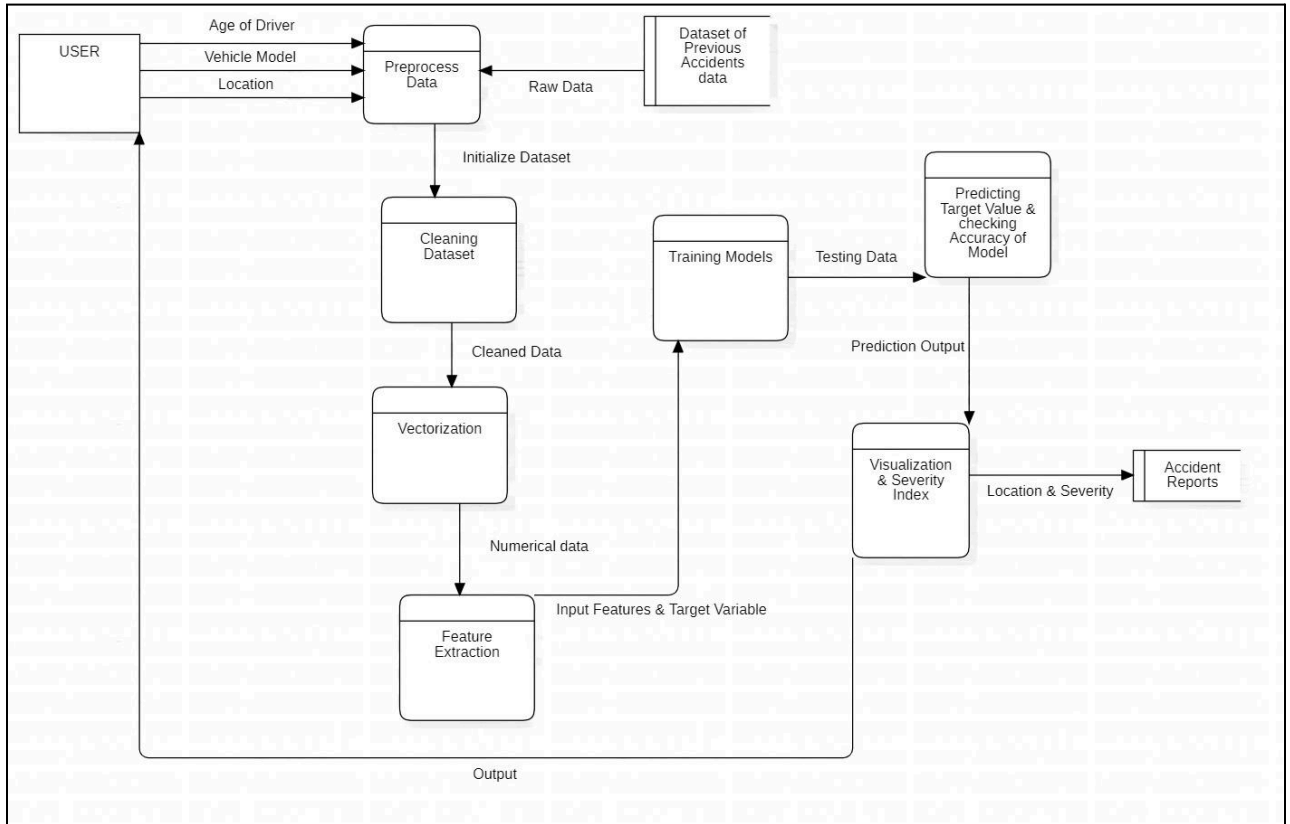


Fig. 3.3 DFD Level 2

### 3.3 MODEL FLOWCHART

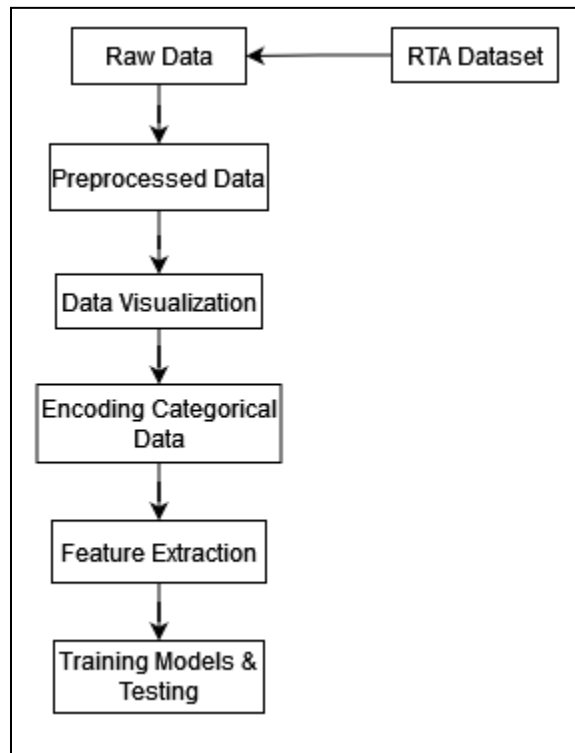


Fig 3.4 Model Flowchart

## **CHAPTER 4**

### **IMPLEMENTATION, TESTING, AND MAINTENANCE**

#### **Introduction**

Road accidents are a significant global concern, leading to immense human suffering and economic losses. To address this issue, our project focuses on developing a real-time road accident severity prediction system that leverages location data, vehicle information, and machine learning. This system aims to provide valuable insights to both drivers and traffic management authorities, helping them make informed decisions to enhance road safety.

In this testing report, we will delve into the details of our project, outlining its objectives, methodology, and the technology stack used. We will also present the results of our testing efforts, including the system's accuracy, performance, and user experience. This report serves as a comprehensive evaluation of the road accident prediction system, highlighting its potential to reduce accidents and their severity, ultimately making our roads safer for all.

#### **Scope**

##### **In Scope**

##### **Functional Requirement: -**

**Real-time Location Data Collection:** The system must collect real-time location data from various sources, such as GPS-enabled devices and traffic cameras.

**Accident Data Analysis:** The project must analyze data related to previous accidents, such as the accident location, weather conditions, and time of day, to predict accident severity.

**Severity Prediction:** The core functionality of the system is to predict the severity of road accidents in real-time, indicating whether an accident is likely to be minor or severe.

User Interface: The system should provide a user-friendly interface for both drivers and traffic authorities to access accident severity information.

Notifications and Alerts: Users should receive timely notifications and alerts about predicted accident severity to help them make informed decisions.

**Non-Functional requirement: -**

Scalability: The system should be capable of handling a high volume of real-time data and user requests.

Accuracy: The severity predictions must be accurate to ensure their reliability for decision-making.

Response Time: The system should provide real-time predictions with minimal latency to support quick decision-making by users.

Usability: The user interface should be intuitive and easy to use, catering to both drivers and traffic management authorities.

Performance: The system's performance, in terms of prediction speed and data processing, should meet or exceed predefined benchmarks.

**Out of Scope**

Equally important to defining what is within the project's scope is understanding what is explicitly excluded from the project. This section outlines the elements and functionalities that are out of scope for the Road Accident Severity Prediction System.

Traffic Control Systems: The project will not be responsible for controlling traffic signals, cameras, or any physical traffic management systems.

Vehicle Hardware Development: The development of hardware components for vehicles, such as sensors or cameras, is not within the scope.

Legal or Regulatory Changes: The system will not make or enforce changes to traffic laws or regulations. It will operate within the framework of existing traffic laws.



## Quality Objective

Ensuring the Road Accident Severity Prediction System operates effectively and reliably is fundamental to its success. The following quality objectives define the expected performance and operational characteristics of the system:

- The system must provide accurate predictions of accident severity based on real-time and historical data, with a reliability rate exceeding 95%. Users should have confidence in the system's recommendations. Ensure the AUT meets the quality specifications defined by the client.
- The system should respond to user inputs and incoming data in real-time, with a latency of no more than 2 seconds. Quick responses are critical for timely accident management.
- The architecture of the system must be scalable to handle increased data and user loads as the user base grows. It should easily accommodate higher traffic volumes.
- The user interface should be intuitive, with clear and user-friendly features. Users of varying technical expertise should be able to interact with the system with minimal training.

## Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

- QA Analyst- Anuj Jain
- Test Manager- Dr. Harsh Khatter
- Configuration Manager- Prof. Akanksha
- Developers- Ayush Pratap Singh, Arth Srivastava
- Installation Team- Ayush Pratap Singh, Arth Srivastava

# **Test Methodology**

## **Overview**

The primary purpose of the Road Accident Severity Prediction System is to predict the severity of road accidents in real-time. By analyzing a variety of data sources, including weather conditions, road type, vehicle count, and historical accident data, the system can provide valuable insights to both authorities and drivers.

## **Test Levels**

Testing is a critical component of our road accident analysis project, ensuring that our system functions as expected and meets the defined requirements. We have organized our testing efforts into multiple levels to systematically evaluate different aspects of the project. These test levels help us identify and rectify issues at various stages of development. Below are the key test levels applied to our project:

### **Unit Testing:**

Scope: Unit testing focuses on the smallest components of the project, such as individual functions, modules, or algorithms. In the context of the road accident analysis project, this level ensures that each component, like data collection scripts or analysis algorithms, performs as intended.

Objective: The primary goal of unit testing is to validate the correctness of individual components and to identify and rectify any defects at an early stage. This level aims to ensure that each piece of code works in isolation.

Testing Approach: Developers and testers collaborate to create test cases that assess the functionality of specific code units. Test data is designed to cover normal and boundary conditions. Automated testing tools can aid in the execution of a large number of tests quickly.

### **Integration Testing:**

Scope: Integration testing assesses the interactions between various components or modules of the system. In the road accident analysis project, it verifies that the data collection process effectively feeds data into the analysis modules.

Objective: The objective of integration testing is to expose issues related to data flow, communication between modules, and any inconsistencies in the system's overall behavior when multiple components interact.

Testing Approach: Test cases are designed to evaluate how different components work together. This may involve testing data transfers between components and assessing the system's ability to handle and process the data as it flows through various stages.

### **System Testing:**

Scope: System testing evaluates the entire road accident analysis system as a whole. It tests the system's compliance with the project requirements, ensuring that it meets the defined objectives and functions as expected.

Objective: The goal of system testing is to validate that the complete system fulfills the project's objectives and works seamlessly in a real-world context. It covers functionality, performance, security, and usability aspects.

Testing Approach: Test scenarios are developed to simulate real-world use cases. This testing phase assesses the system against predefined requirements and evaluates its performance under typical and stress conditions.

### **User Acceptance Testing (UAT):**

Scope: UAT is the final testing phase, focusing on ensuring that the road accident analysis system meets the needs and expectations of its intended users. This testing level typically involves real end-users.

Objective: The primary objective of UAT is to validate that the system aligns with user requirements and that it is user-friendly. It confirms that the system is ready for production use.

Testing Approach: End-users and stakeholders participate in UAT. They execute test scenarios representing real-world tasks, and feedback is collected. Defects, if any, are addressed before system deployment.

## Test Completeness

In software testing, "test completeness" refers to the extent to which a system or software application has been tested in terms of various aspects such as functionality, requirements coverage, and potential use cases. It is a critical aspect of the testing process that ensures that the system is rigorously evaluated to identify defects and ensure that it meets its intended objectives.

For instance, a few criteria to check Test Completeness would be

- 100% test coverage
- All Manual & Automated Test cases executed
- All open bugs are fixed or will be fixed in next release

## Test Deliverables

### Test Cases :-

Test Case	Test Objective	Test Data	Expected Result	Actual Result	Pass/Fail
1	Verify Input for start and end location	Start & end location outside UK	Only UK Locations allowed	Invalid location	Pass
2	Verify Number of vehicles	Number of vehicles $\leq 3$	Allowed	Valid Input	Pass
3	Verify route calculation	Start & end location inside UK	Should calculate route correctly	Map displaying highlighted route	Pass

4	Verify API Data Fetching	Start & end location inside UK	Fetch location-based data from API & feed it to model	Correctly fetching location-based data from API	Pass
5	Verify easting and northing	Start & end location inside UK	Correctly calculate location easting & northing	Model working correctly	Pass

Tab 4.1 Test Cases

### Decision Table:-

Conditions	Rules							
Traffic congestion is high	True				False			
Road condition is poor	True		False		True		False	
Number of vehicles > 3	True	False	True	False	True	False	True	False
Actions	Entries							
Alert The User	Yes	Yes	Yes	Yes	No	No	No	No
Record Data	Yes	Yes	Yes	No	Yes	Yes	No	No
Notify Authorities	Yes	No	No	No	Yes	No	Yes	No

Tab 4.2 Decision Table

# Test Case Output Images

## 1. Manual Testing:

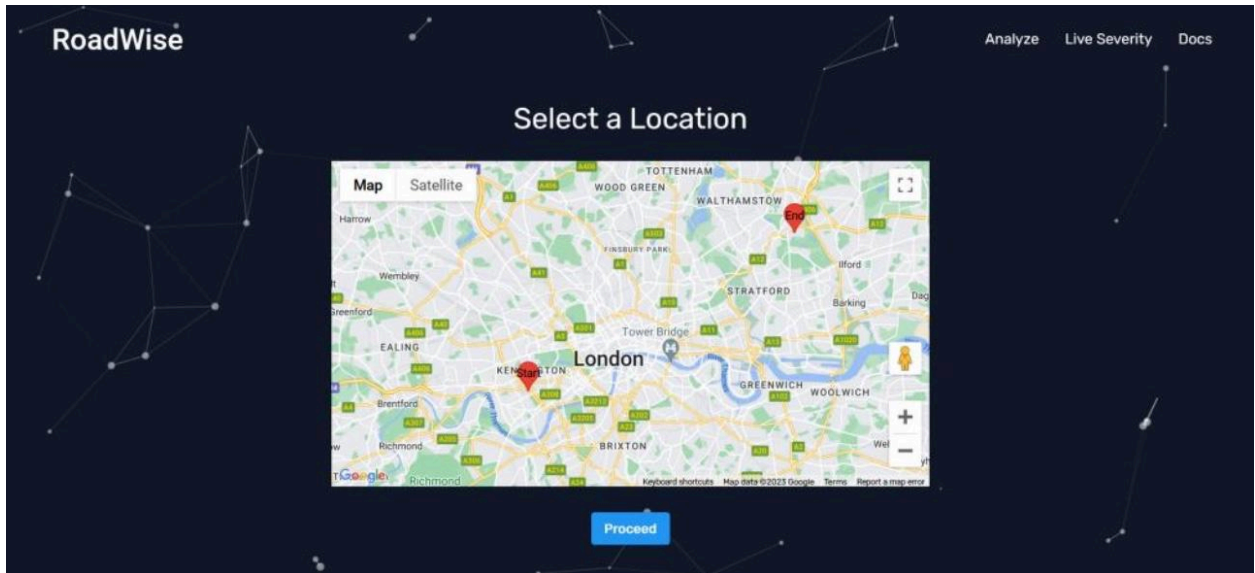


Fig. 4.1 Manual Test Case Input (Location)

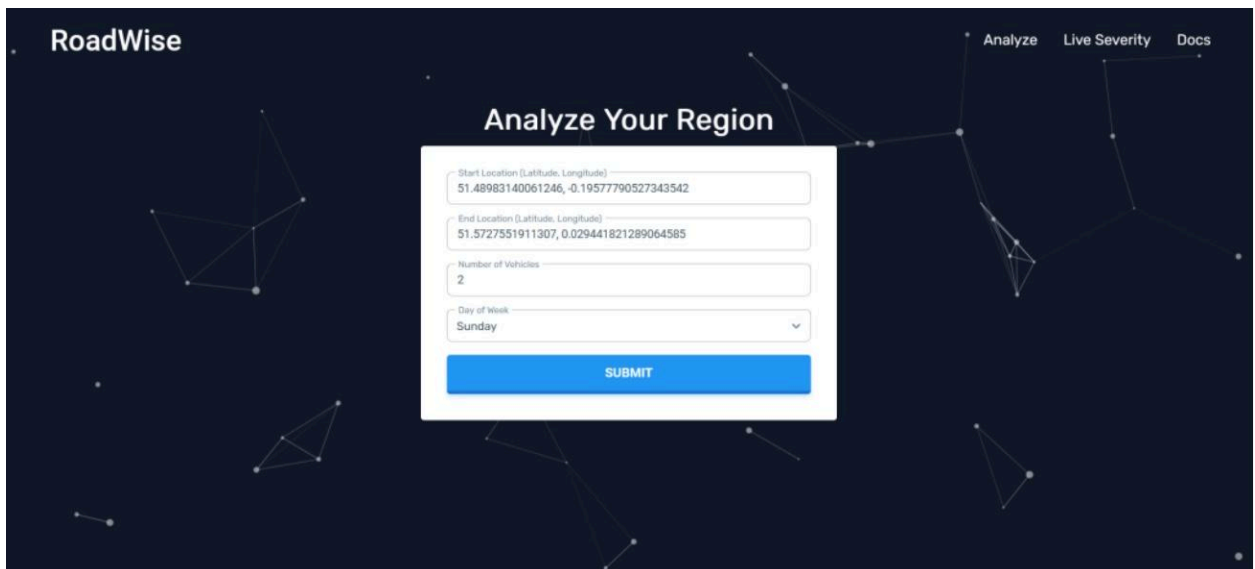


Fig. 4.2 Manual Test Case Input (Other Details)

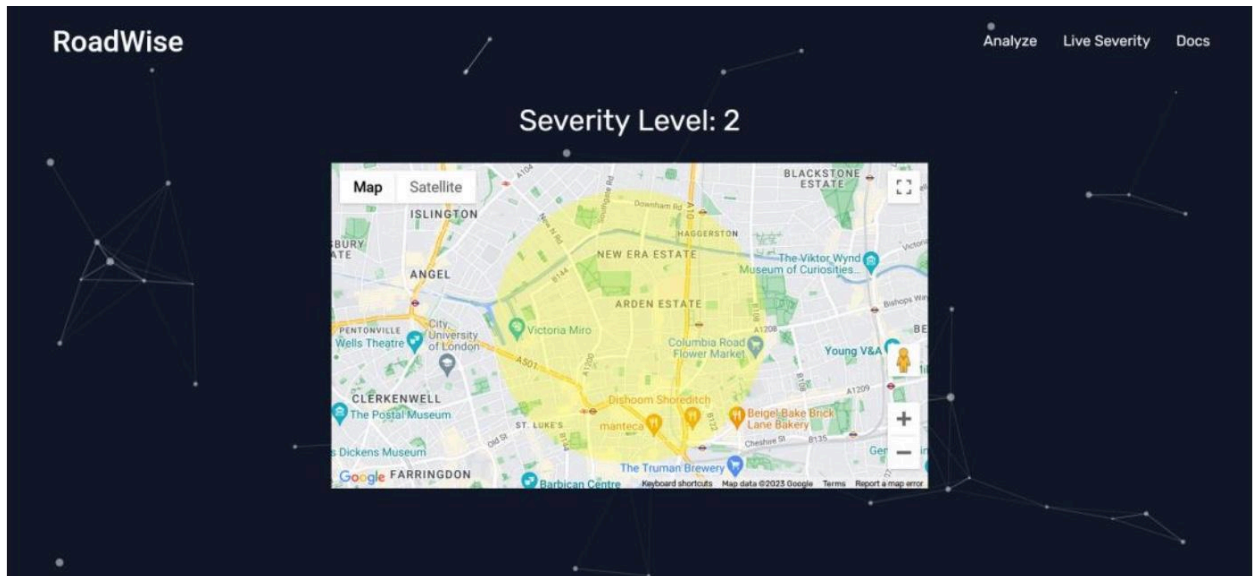


Fig. 4.3 Manual Test Case Results

## 2. Automation Testing

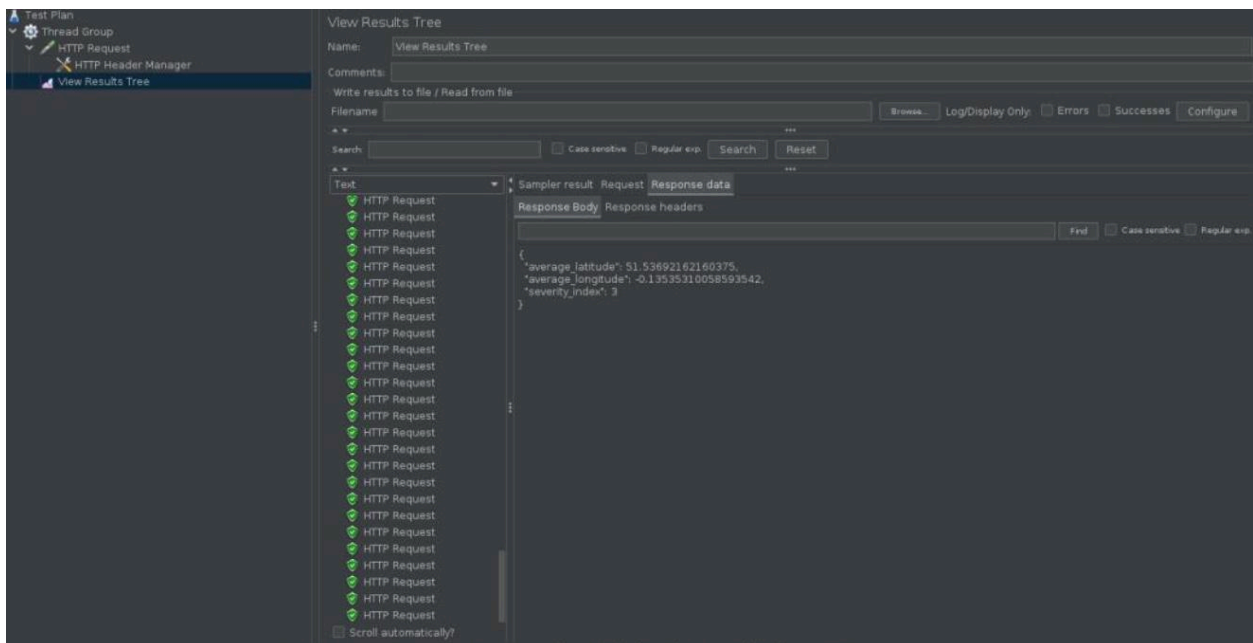


Fig. 4.4 Automation Testing

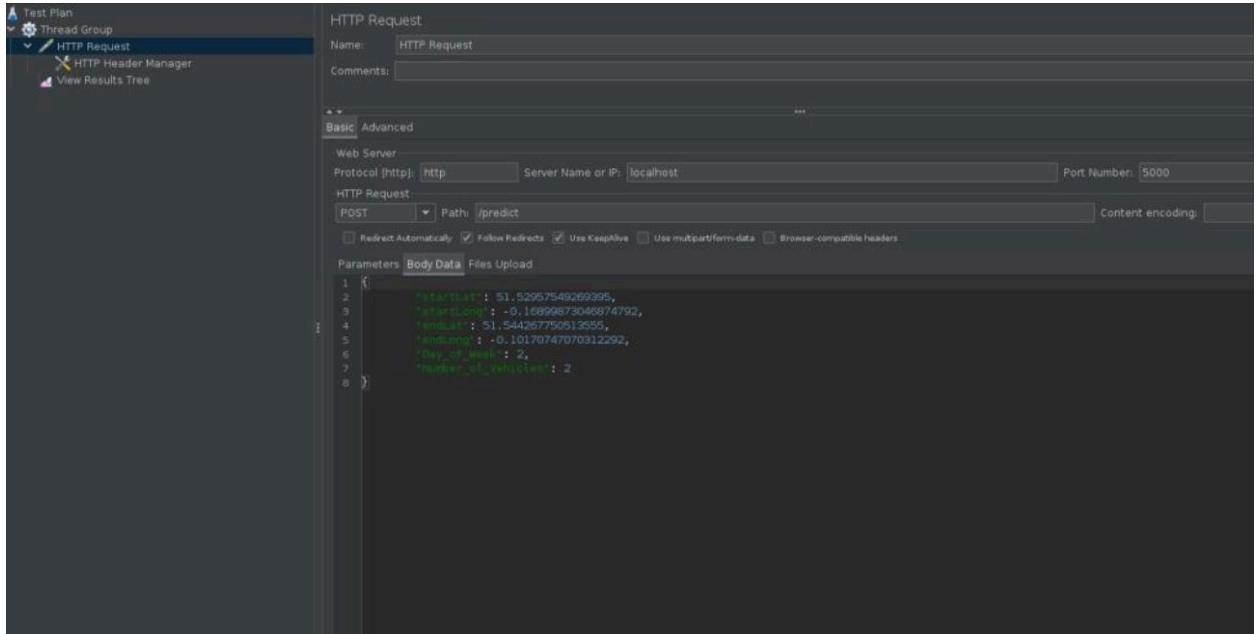


Fig. 4.5 Automation Testing Results

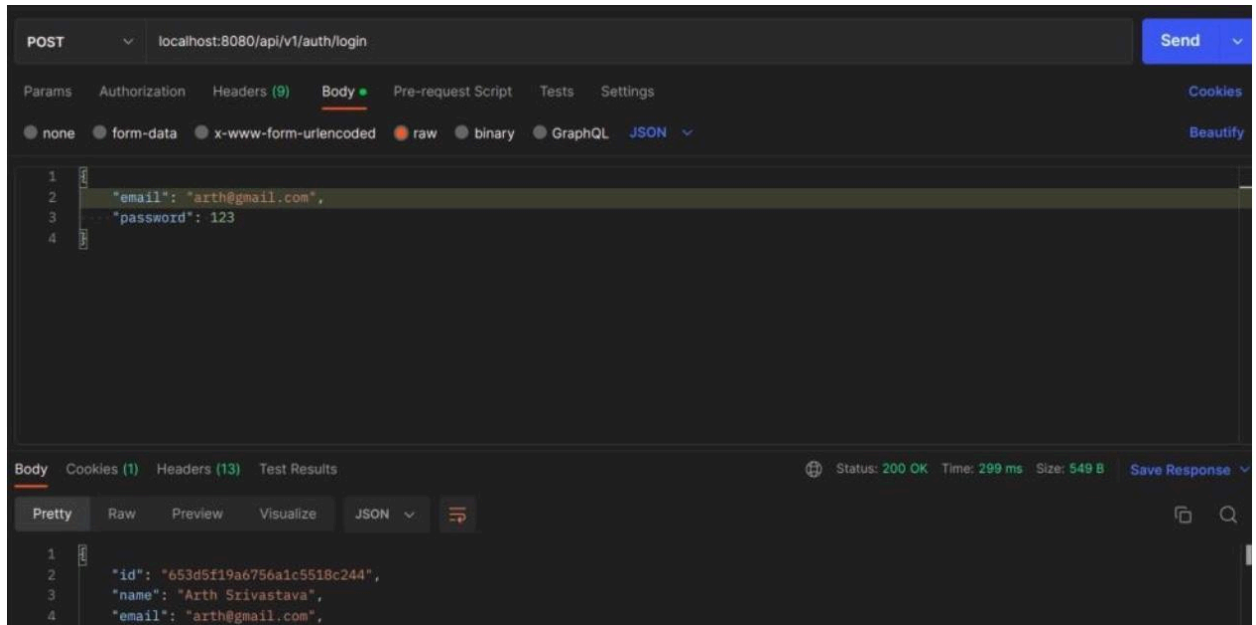


Fig. 4.6 Postman



## Resource & Environment Needs

### Testing Tools

In your road accident analysis project, various testing tools can be employed to ensure the efficiency, accuracy, and reliability of your application. These tools assist in automating tests, managing test cases, tracking defects, and more. Here are some of the testing tools that can be beneficial for your project:

1. Postman:
  - Scope: Postman is a popular API testing tool. It can be used to test the RESTful APIs integrated into your project, such as those fetching accident data.
  - Objective: To validate API endpoints, check responses, and ensure that the API requests and data retrieval function correctly.
  - Testing Approach: Develop API test collections and scripts to send requests, assess responses, and verify the accuracy of data retrieved from external sources.
2. Load Testing Tools (e.g., Apache JMeter):
  - Scope: Load testing tools help assess how your application performs under various levels of concurrent user activity.
  - Objective: To identify performance bottlenecks, server capacity, and response times.
  - Testing Approach: Design test scenarios that simulate real-world traffic and evaluate the application's stability under load.

### Test Environment

Following software is required in addition to client-specific software.

- Web browsers (e.g., Chrome, Firefox) for testing the web interface.
- Internet access to retrieve real-time data.
- Testing tools (e.g., Selenium, Postman, load testing tools) for test automation and API testing.
- Database management systems (e.g., MySQL) for data storage and retrieval testing.
- Development environments (e.g., Visual Studio Code) for scripting automated tests.

- Continuous Integration/Continuous Deployment (CI/CD) tools (e.g., Jenkins) for automating build and deployment.
- Test data generation tools for creating realistic test scenarios.
- Source code repositories (e.g., GitHub) for version control.
- Collaboration and project management tools (e.g., JIRA) for issue tracking and task management.

## Terms/Acronyms

TERM/ACRONYM	DEFINITION
AUT	Application Under Test
API	Application Program Interface

Tab 4.3 Acronym Table

## CHAPTER 5

### RESULTS AND DISCUSSIONS

#### 5.1 USER INTERFACE REPRESENTATION

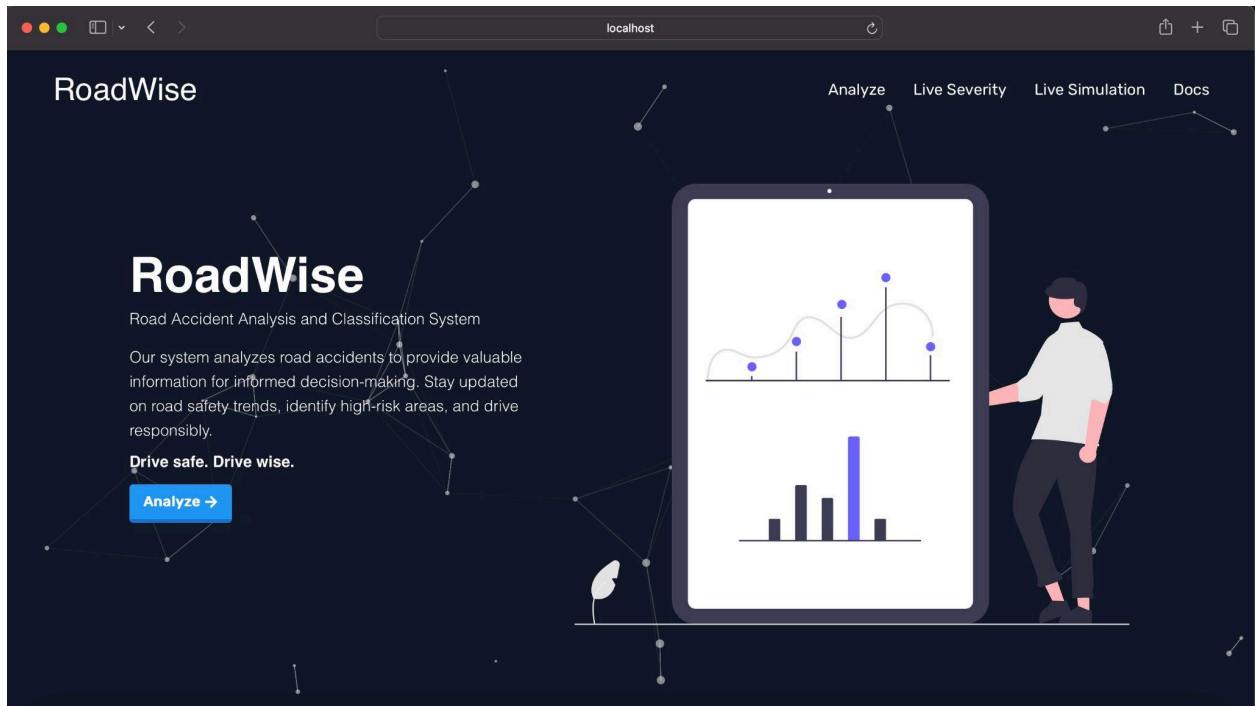


Fig. 5.1 Home Page

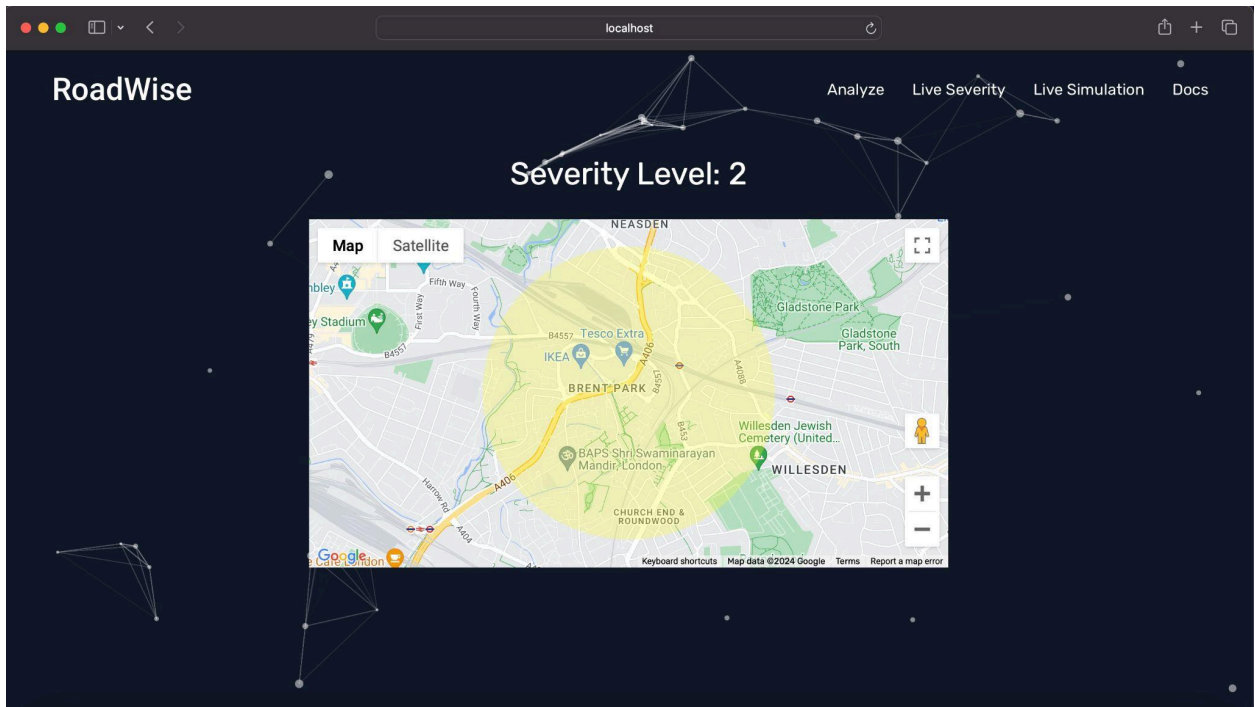


Fig. 5.2 Results & Analysis

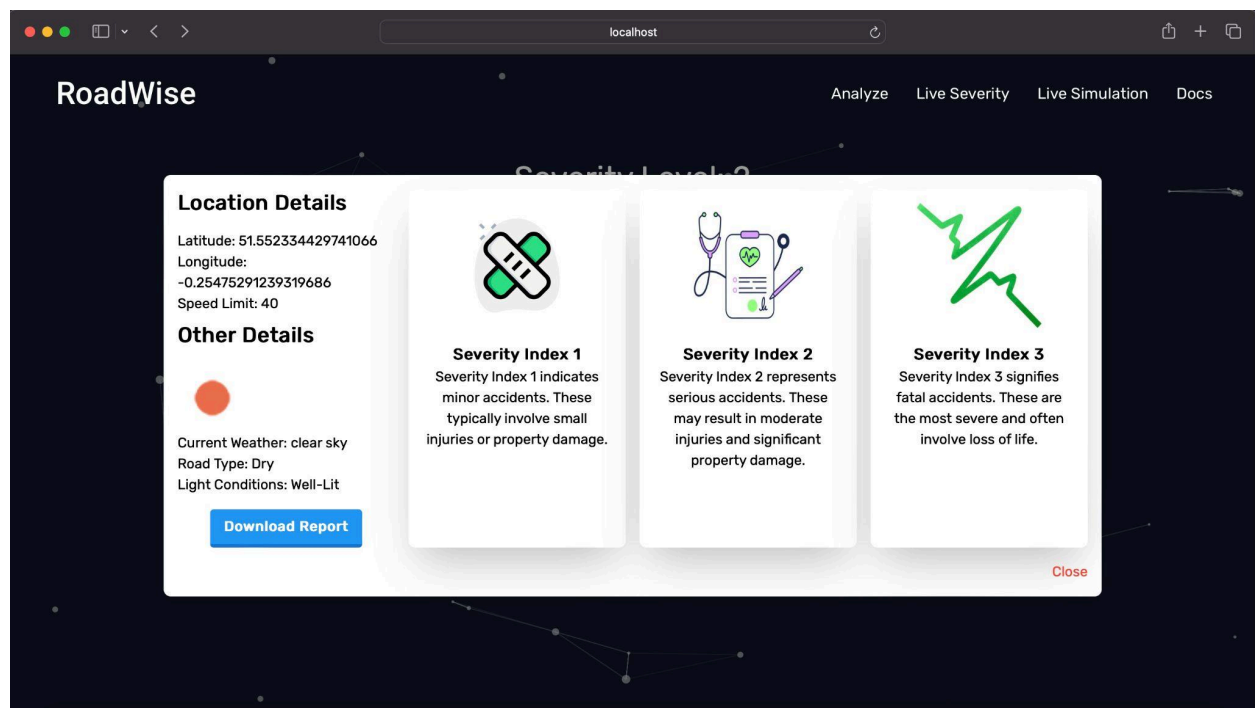


Fig 5.3 Download Report

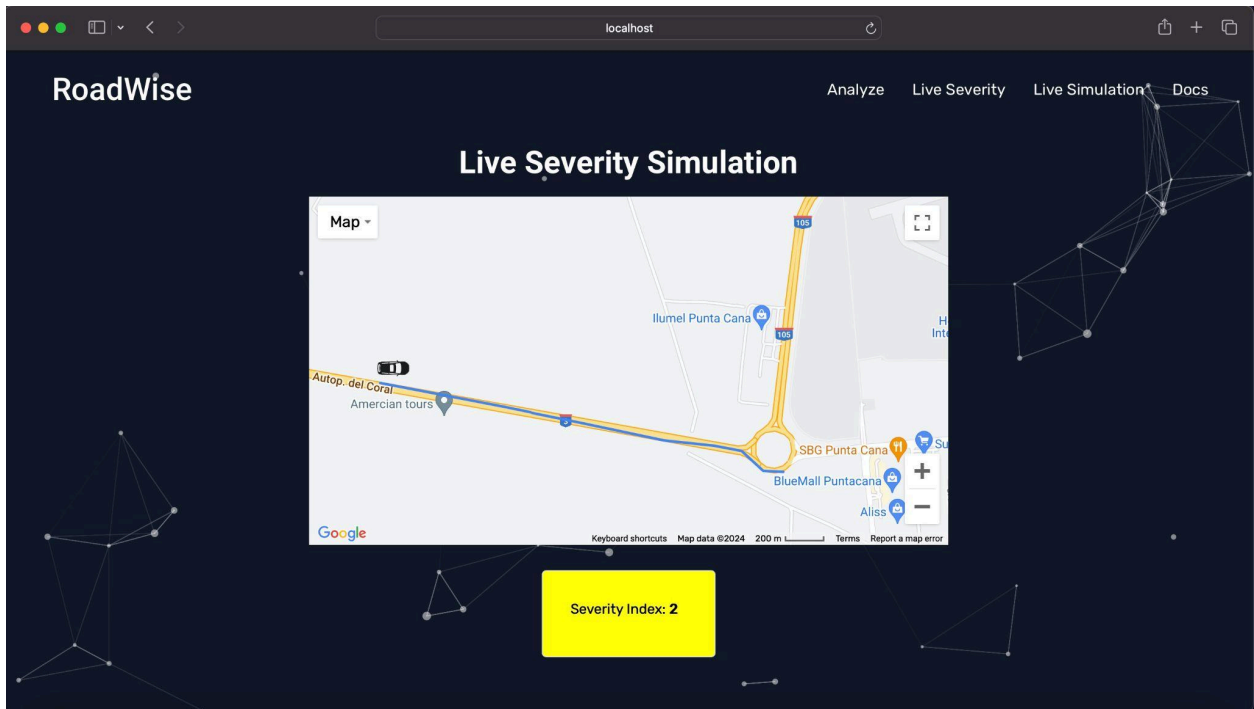


Fig. 5.4 Live Severity Simulation

### 5.1.1 BRIEF DESCRIPTION OF VARIOUS MODULES OF THE SYSTEM

#### 1. Location Selection Module:

- This module allows users to select their start and end locations on a map interface.
- Users can visually pinpoint their desired locations, enabling accurate route selection.

#### 2. Additional Details Input Module:

- After selecting the locations, users are directed to a page where they can input additional details such as the number of vehicles involved and the day of the week.
- This module collects relevant information to refine the accident severity analysis.

#### 3. Results and Analysis Module:

- In this module, users are presented with a map displaying accident severity zones, categorized into three levels: green for minor accidents, yellow for serious accidents, and red for fatal accidents.

- Clicking on each zone provides a detailed view, including current weather, road type, light conditions, and a download report button for exporting data.
4. Detailed Information Display Module:
- Upon clicking on a severity zone, users are directed to a detailed information display module.
  - This module presents additional information about the selected zone, such as specific weather conditions, road types, and light conditions.
  - It also includes a download report button for exporting detailed accident information into an Excel file.
5. Live Severity Feature:
- This module provides a real-time display of accident severity en route.
  - Users can view the current severity level along their selected route, helping them make informed decisions while traveling.
6. Simulation Page:
- The simulation page allows users to visualize how accident severity will be displayed en route.
  - A car logo moves along the route every second, providing a dynamic representation of the severity levels as the user progresses.

## 5.2 SNAPSHOTS OF SYSTEM WITH BRIEF DETAIL OF EACH

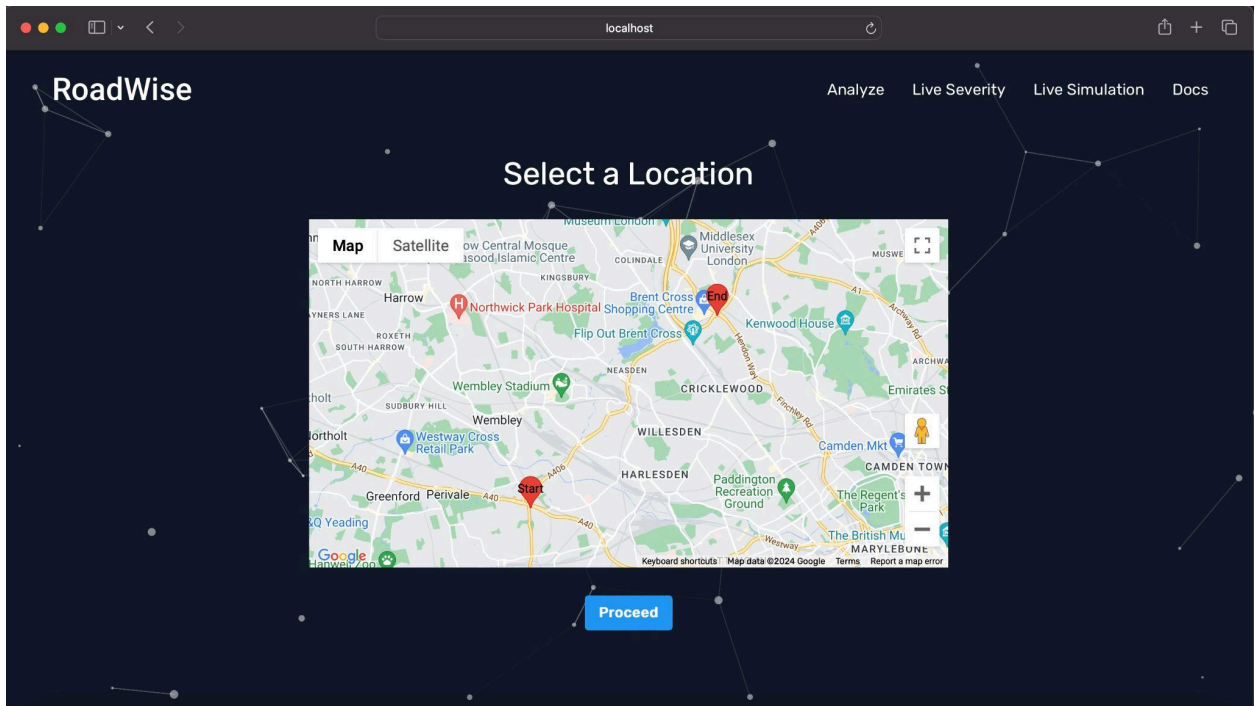


Fig 5.5 Location Selection Module

Upon redirection from the homepage, users access the Location Selection Module. Here, they use an intuitive map interface to select their start and end locations. This user-friendly feature allows precise location pinpointing through zooming, dragging, and clicking. It serves as the first step in the user journey, facilitating subsequent data input and analysis.

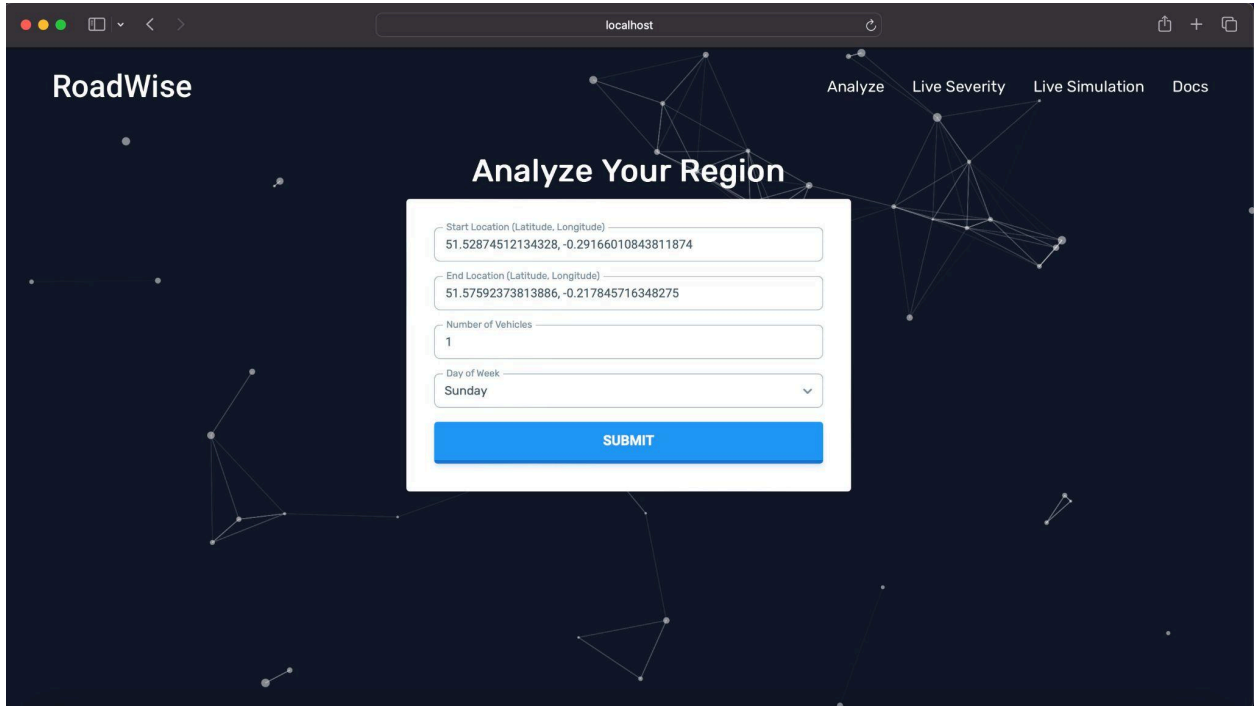


Fig. 5.6 Additional Details Input Module

Following location selection, users are seamlessly directed to the Additional Details Input Module. This module streamlines data entry by automatically populating latitude and longitude coordinates based on the chosen location. Users then provide supplementary information, such as the number of vehicles involved and the day of the week. This streamlined process ensures accurate and efficient data input, enhancing the overall user experience and facilitating comprehensive analysis.



report-2

Latitude	Longitude	Severity Index	Speed Limit	Current Weather	Road Type	Light Conditions
51.552334429741100	-0.25475291239319700	2	40	clear sky	Dry	Well-Lit

Fig 5.7 Downloaded Report

The Downloaded Report Module offers users comprehensive insights into accident data, facilitating informed decision-making and strategic planning. The downloadable report, presented in Excel format, encompasses crucial parameters essential for thorough analysis. These include latitude and longitude coordinates, facilitating geographical mapping and spatial visualization of accident hotspots. Additionally, the severity index provides a clear indication of accident severity levels, enabling prioritization of intervention strategies. Other pertinent details such as road type, light conditions, and speed limits offer contextual information to better understand contributing factors to accidents. Furthermore, real-time weather data enhances situational awareness, allowing stakeholders to account for weather-related risks and implement preemptive measures. This module empowers users with actionable data, fostering proactive measures to improve road safety and mitigate accident risks.

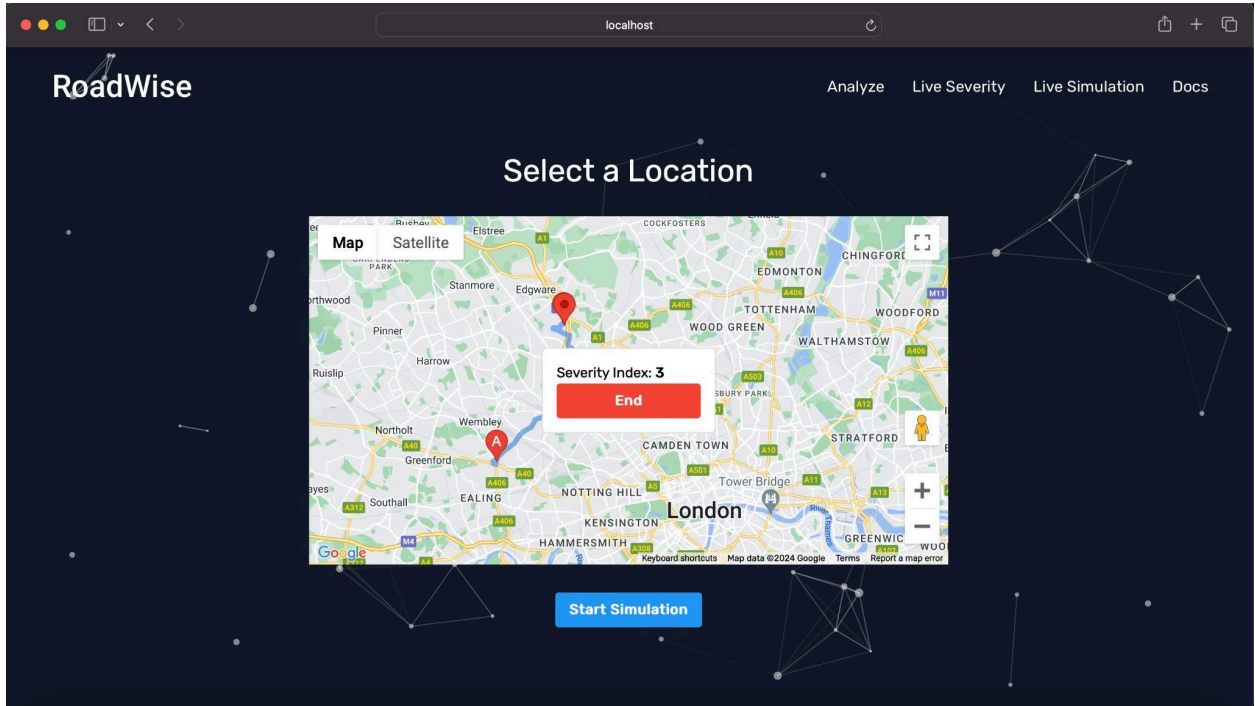


Fig. 5.8 Live Severity Feature

The Live Severity Feature empowers users to access real-time accident severity information en route, enhancing situational awareness and informed decision-making. Users can effortlessly mark their desired destination, allowing the system to retrieve their current location automatically. As users navigate their journey, the system dynamically generates a path marked with a distinctive blue color, providing visual guidance along the chosen route. Alongside the path, users receive continuous updates on accident severity, enabling them to anticipate potential risks and adjust their travel plans accordingly. To conclude the live severity feature, users can simply click the "End" button, seamlessly transitioning back to the main interface while retaining valuable insights gathered during their journey.

## 5.3 BACK ENDS REPRESENTATION

The back-end of our system is designed to handle data processing, machine learning model inference, and interaction with external APIs. Leveraging Flask, a lightweight web framework, our back-end architecture follows a microservices approach, allowing for modularity and scalability.

### **Components:**

- **Flask Backend:** The Flask backend serves as the core component of our system, responsible for handling incoming requests from the front-end application and coordinating data processing tasks.
- **Machine Learning Model (Pickle):** A machine learning model, trained to predict road accident severity, is integrated into the Flask backend using Pickle serialization. This model processes input data received from the front end and generates predictions in real-time.
- **API Integration:** The back-end interacts with various external APIs to fetch real-time data essential for accident severity prediction. APIs such as Google Maps API for location services and Weather API for current weather conditions are seamlessly integrated into the system to enrich the prediction process.

### **Technologies:**

- **Flask:** Flask is chosen as the web framework for its simplicity and flexibility, allowing for rapid development of RESTful APIs and easy integration with machine learning models.
- **Pickle:** Pickle serialization is utilized to persist the trained machine learning model, enabling efficient model loading and inference within the Flask application.
- **APIs:** Various APIs, including Google Maps API and Weather API, are incorporated into the back-end to access relevant data for accident severity prediction. These APIs provide essential information such as geographical coordinates, road conditions, and weather forecasts.

**Functionality:**

- **Data Processing:** Upon receiving input data from the front end, the Flask backend performs data preprocessing tasks, such as feature extraction and normalization, to prepare the data for model inference.
- **Model Inference:** The serialized machine learning model is loaded into memory, and the input data is fed into the model for prediction. The backend then returns the predicted accident severity to the front end for display.
- **API Interaction:** The back-end interacts with external APIs to fetch additional data required for accident severity prediction, such as real-time weather conditions and road traffic information. This data is seamlessly integrated into the prediction process to enhance accuracy.

## **CHAPTER 6**

### **CONCLUSION AND FUTURE SCOPE**

In conclusion, our road accident analysis and classification system represent a significant advancement in leveraging machine learning and real-time data integration to enhance road safety. Through comprehensive data preprocessing, machine learning model training, and seamless integration with external APIs, we have developed a robust system capable of predicting accident severity with an accuracy of 86.73%. By providing users with actionable insights into accident-prone areas and real-time severity information, our system aims to mitigate the occurrence of road accidents and improve overall safety on the roads.

Looking ahead, there are several avenues for future exploration and enhancement of our system. One potential direction is to further refine the machine learning models by incorporating more extensive and diverse datasets, including data from different geographical regions and time periods. Additionally, exploring advanced techniques such as ensemble learning and deep learning could lead to even higher prediction accuracy and reliability.

Furthermore, there is scope for expanding the functionality of our system to include features such as predictive analytics for proactive accident prevention, integration with intelligent transportation systems for real-time traffic management, and adaptive user interfaces for personalized safety recommendations.

Overall, our road accident analysis and classification system lay the foundation for a data-driven approach to road safety management, with the potential to make significant strides in reducing accidents and saving lives on the roads. Through continued research, innovation, and collaboration, we envision a future where road accidents are minimized, and transportation systems are safer and more efficient for all.

## REFERENCES

- [1] A. Pakgohar, Reza Tabrizi, Mohadeseh Khalili, and Alireza Esmaeili, "The role of human factor in incidence and severity of road crashes based on the CART and LR regression: A data mining approach," *Procedia Computer Science*, vol. 3, pp. 764-769, December 2011.
- [2] Mohad Fedder Musa, Sitti Asmah Hassan, Nordiana Mashros, "The impact of roadway conditions towards accident severity on federal roads in Malaysia," *PLoS One*, 2020 Jul 6. DOI: doi.org/10.1371/journal.pone.0235564.
- [3] Mouyid Bin Islam, Kunnawee Kanitpong, "Identification of factors in road accidents through in-depth accident analysis," *IATSS research*, vol. 32, no. 2, pp. 58-67, 2008. Publisher: Elsevier.
- [4] Nejdett Dogru, A. Subasi, "Traffic accident detection using random forest classifier," 15th Learning and Technology Conference (L&T), 1 February 2018.
- [5] Liling Li, Sharad Shrestha, Gongzhu Hu, "Analysis of road traffic fatal accidents using data mining techniques," *International Conference on Software Engineering Research and Applications*, 7 June 2017.
- [6] Mohamed Nour, Atif Naseer, Basem Alkazemi, Muhammad Abid Jamil, "Road Traffic Accidents Injury Data Analytics," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 12, January 2020. DOI: 10.14569/IJACSA.2020.0111287.
- [7] Jayesh Patil, Vaibhav Patil, Dhaval Walavalkar, Vivian Brian Lobo, "Road Accident Analysis and Hotspot Prediction using Clustering," 2021 6th International Conference, Publisher: IEEE, Date of Conference: 08- 10 July 2021.
- [8] "Classification Algorithms in Machine Learning," online. Available: [medium.datadriveninvestor.com/classification-algorithms-in-machine-learning-85c0ab65ff4](https://medium.datadriveninvestor.com/classification-algorithms-in-machine-learning-85c0ab65ff4). Accessed on October 26, 2023.
- [9] "UK Road Safety: Traffic Accidents and Vehicles," online. <https://www.kaggle.com/datasets/tsiaras/uk-road-safety-accidents-and-vehicles>. Accessed on October 26, 2023

- [10] J. Patil, M. Prabhu, D. Walavalkar, and V.B. Lobo, “Road accident analysis using machine learning,” In 2020 IEEE Pune Section International Conference (PuneCon), IEEE, pp. 108–112, 2020.
- [11] Y. Wang and W. Zhang, “Analysis of roadway and environmental factors affecting traffic crash severities,” *Transportation Research Procedia*, vol. 25, pp. 2119–2125, 2017.
- [12] Harsh Khatter, Amrita Jyoti, Rashmi Sharma, Pooja Malik, Rashmi Mishra “Enhancing Network Efficiency and Extending Lifetime through Delay Optimization and Energy Balancing Techniques”, *Wireless Personal Communications* (2023).  
<https://doi.org/10.1007/s11277-023-10812-7>
- [13] Xue Liu, Xiaowei Wang, Feng Ren, Ming Zhang, Harsh Khatter, Afroj Alam, “Multi-objective optimization scheduling of sequential charging software for Networked Electric Vehicles”, *Journal of Sensors* (2022), Hindawi, Vol 2022(6968470), pp. 1-8, 29 July 2022, <https://doi.org/10.1155/2022/6968470>
- [14] Amit Kumar Gupta, Ruchi Rani Garg, Arti Sharma, Saurabh and Harsh Khatter, “Posture Identification Using Artificial Neural Network”, Presented in 3rd International Conference on Smart Computing and Cyber Security SMARTCYBER, 28-29 June 2023, Kyungdong University, South Korea.