TRANSPORT MANAGEMENT SYSTEM

A

MAJOR PROJECT-II REPORT

Submitted in partial fulfillment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

GROUP NO. 18

Kartik Sahu	**0187CS211085**
Jatin Sahu	**0187CS211078**
Praveen Pandey	**0187CS211124**
Kartik Jaiswal	**0187CS211083**

Under the guidance of

Prof. Jai Mungi

(Assistant Professor)

Department of Computer Science & Engineering

Sagar Institute of Science &Technology (SISTec), Bhopal (M.P)

Approved by AICTE, New Delhi & Govt. of M.P.

Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)

June -2025

Sagar Institute of Science & Technology (SISTec), Bhopal (M.P)

Department of Computer Science & Engineering

CERTIFICATE

We hereby certify that the work which is being presented in the B.Tech. Major Project-II Report entitled **TRANSPORT MANAGEMENT SYSTEM,** in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology,*** submitted to the Department of **Computer Science & Engineering**, Sagar Institute of Science & Technology (SISTec), Bhopal (M.P.) is an authentic record of our own work carried out during the period from Jan-2025 to Jun-2025 under the supervision of **Prof. Jai Mungi.**

The content presented in this project has not been submitted by me for the award of any other degree elsewhere.

Kartik Sahu	**Jatin Sahu**	**Praveen Pandey**	**Kartik Jaiswal**
0187CS211085	**0187CS21078**	**0187CS211124**	**0187CS211083**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Prof. Jai Mungi	**Dr. Amit Kumar Mishra**	**Dr. D.K. Rajoriya**
Project Guide	**HOD, CSE**	**Principal**

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. D. K. Rajoriya**, **Principal, SISTec and Dr. Swati Saxena, Vice Principal SISTec** Gandhi Nagar, Bhopal for giving us an opportunity to undertake this project.

We also take this opportunity to express a deep sense of gratitude to **Dr. Amit Kumar Mishra, HOD, Department of Computer Science & Engineering** for his kindhearted support.

We extend our sincere and heartfelt thanks to our guide, **Prof. Jai Mungi,** for providing us with the right guidance and advice at crucial junctures and for showing us the right way.

I am thankful to the **Project Coordinator, Prof. Deepti Jain,** who devoted her precious time in giving us the information about various aspects and gave support and guidance at every point of time.

I would like to thank all those people who helped me directly or indirectly to complete my project whenever I found myself in any issue.

TABLE OF CONTENTS

ABSTRACT

This project is an integrated web-based bus tracking and passenger support system that allows users to track public transportation vehicles in real-time and access essential services online. The system enables passengers to register, log in, and view the live location of buses, helping them plan their journeys more efficiently. Additionally, users can file complaints and raise SOS alerts in case of emergencies directly through the platform. By automating traditional manual processes such as location tracking, issue reporting, and emergency response, it improves commuter convenience, safety, and reliability. The goal of this system is to deliver a user-friendly solution that empowers passengers with real-time transit information and streamlined support services.

The architecture of the system is built using JSP and Servlet technologies, ensuring a responsive user interface and efficient server-side processing. The integration of GPS tracking allows users to access accurate, real-time bus location updates, while the backend database maintains user information, complaint logs, and emergency reports. By combining real-time data processing with user interaction features, the system not only modernizes the public transport experience but also sets a foundation for future enhancements such as predictive arrival times and mobile app integration.

LIST OF ABBREVIATIONS

ACRONYM	FULL FORM
SDLC	Software Development Life Cycle
SQL	Structured Query Language
HTML	Hyper Text Markup Language
UML	Unified Modeling Language

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1
INTRODUCTION

1.1 Current Scenario and Challenges

Public transportation, particularly bus transit systems, serves as the backbone of urban and suburban mobility, offering an affordable and accessible mode of transport for a large segment of the population. Despite its critical role, the current state of many bus systems is far from optimal. Passengers frequently experience uncertainty regarding bus arrival times, which often results in prolonged waiting periods at bus stops without any clear indication of when the next bus will arrive[1]. This unpredictability not only causes inconvenience but also leads to a loss of time and decreased trust in public transit systems.

A major contributing factor to these issues is the lack of real-time tracking features accessible to passengers. While a few transportation providers have implemented GPS technology, the benefits are mostly limited to internal monitoring and not shared with commuters. Consequently, travelers are unable to plan their journeys efficiently, especially during peak traffic hours or unexpected delays. Furthermore, these systems often do not adapt dynamically to real-time road conditions, such as congestion, road closures, or weather disruptions.

Most traditional bus systems still rely on fixed schedules and manual monitoring, which are inadequate in today's fast-paced, tech-driven environment. They lack automation and depend heavily on outdated operational methods, making it difficult to respond promptly to route changes or delays. This gap in automation and real-time intelligence results in inefficiencies that not only affect passengers but also place a heavy burden on transport operators who struggle with fleet coordination and timely service delivery.

In many cities, there is an evident absence of a robust and centralized system for bus tracking and passenger complaint management. Even where such systems exist, they are often fragmented or poorly maintained, rendering them ineffective. As a result, commuter grievances often go unresolved, and their concerns related to punctuality, driver behavior, or service disruptions are not addressed in a timely manner[2]. Moreover, in emergency situations, the lack of direct communication channels or integrated safety features makes it difficult to provide immediate assistance to passengers.

These issues collectively result in:
- Passenger dissatisfaction due to inconsistent and unpredictable wait times.
- Inadequate safety infrastructure, especially in emergency scenarios where rapid response is crucial.
- Operational challenges for service providers who lack real-time data to make informed decisions.

Although a few transportation services have adopted GPS tracking, most of them fail to extend these benefits to the end-users. The technology is either not made accessible to the public or is limited in scope. Additionally, the high cost of implementing and maintaining comprehensive tracking and emergency management systems makes it unaffordable for many cities, particularly those with budget constraints. Most of the existing solutions are either expensive to scale or lack crucial features such as user-friendly interfaces and integrated support for emergency scenarios, making them unsuitable for widespread adoption.

1.2 Need for Automation

In recent years, rapid advancements in technologies such as GPS, cloud computing, Internet of Things (IoT), and mobile applications have created new opportunities to revolutionize the public transport ecosystem. These innovations make it not only feasible but also practical to automate several critical aspects of public transportation, especially bus tracking and complaint resolution systems.

Automation can bridge the gap between passenger expectations and current service limitations. By deploying an integrated digital platform, public transit systems can dramatically enhance both the user experience and operational performance. Real-time data collection and processing enable buses to be tracked live, allowing passengers to receive accurate location updates and estimated arrival times. This level of transparency can significantly reduce uncertainty, boost commuter confidence, and make public transit more appealing.

Moreover, automation helps optimize routes and fleet operations. By analyzing traffic data and commuter demand in real-time, transport operators can dynamically adjust routes, schedules, and bus frequency. This improves fuel efficiency, reduces idle times, and ensures better utilization of resources. Additionally, with mobile-enabled complaint and feedback systems, passengers can easily report issues, helping operators take timely action and continuously improve service quality.

Integrated automation can also provide critical safety features. For instance, emergency alerts sent from passengers can immediately notify authorities or transport officials, allowing them to intervene quickly in case of medical emergencies, harassment, or accidents. Such features not only protect passengers but also create a sense of safety and trust in public transportation systems.

Key benefits of automation include:

- Real-time location tracking and live updates for passengers.
- Intelligent route optimization to reduce delays and traffic congestion.
- Emergency alert systems integrated into mobile apps for quick response.
- Data-driven decision making for better service delivery.
- A digital channel for feedback and grievance redressal.

By embracing automation, cities can make their transportation systems smarter, safer, and more efficient, addressing the core issues faced by both passengers and operators.

1.3 Objectives

The SISTRaahi project is designed to tackle the fundamental challenges in today's public transportation systems by leveraging modern technologies to create a smart, reliable, and passenger-friendly solution. The primary goal is to automate and improve the experience of both commuters and operators through a robust and scalable digital system.

The key objectives of SISTRaahi are:

To develop and deploy a real-time bus tracking system using advanced GPS modules and cloud-based infrastructure, ensuring continuous monitoring of bus movements and availability of live updates for passengers via a mobile-friendly platform.

To provide accurate estimated time of arrivals (ETAs) based on real-time data analytics, thus significantly reducing the average wait time for passengers and improving the overall efficiency of travel planning.

To establish an efficient and user-centric complaint management system, enabling commuters to submit feedback, complaints, or suggestions directly through a mobile app or web portal. This ensures transparency, accountability, and faster issue resolution.

To integrate an emergency response feature within the system, allowing passengers to trigger alerts in case of safety threats or medical emergencies. These alerts can be routed to both transport authorities and emergency services for a prompt response.

To enhance operational efficiency for transport providers by automating route management, fleet tracking, and scheduling through data-driven insights. This will result in better fuel management, lower operational costs, and improved service delivery.

In the long run, by meeting these objectives, SISTRaahi aims to revolutionize the public transportation landscape. The system not only enhances the convenience and safety of passengers but also empowers operators with tools to manage their services more efficiently. Ultimately, it envisions a future where public transport is more accessible, punctual, responsive, and safe—ensuring a smoother and smarter commuting experience for everyone.

CHAPTER – 2
SOFTWARE AND HARDWARE REQUIREMENTS

2.1 SOFTWARE REQUIREMENTS

To build and maintain a robust, scalable, and responsive public transportation automation system, a well-defined set of software tools, frameworks, and platforms is essential. These components will support development, testing, deployment, and real-time functionality.

2.1.1 DEVELOPMENT TOOL

Backend:

- Java 8+: Core programming language used to implement server-side logic and handle business operations efficiently.
- Servlets: Used to handle HTTP requests and responses, serving as the backbone for dynamic web content generation.
- JSP (JavaServer Pages): Enables embedding Java directly into HTML pages to dynamically generate content on the server side.
- JDBC (Java Database Connectivity): Allows interaction with the relational database using SQL queries from Java code.
- Apache Tomcat: A lightweight web server and servlet container used for deploying and running the application.
- Maven: Project management tool used for managing dependencies and building the application in a standardized way.

Frontend:

- HTML, CSS, JavaScript: Core web technologies used to structure, style, and add interactivity to the user interface.
- Bootstrap: A responsive front-end framework used to speed up UI development and ensure consistent styling.
- JSP: Also plays a role on the front end, rendering dynamic content sent from the server.

2.1.2 DATABASE MANAGEMENT

- MySQL 8.0+: A reliable, open-source RDBMS to store structured data such as user profiles, bus details, routes, tracking logs, and complaint records.
- MySQL Workbench: Graphical interface for designing, querying, and managing the MySQL database effectively, also used for ER modeling and schema design.

2.1.3 REAL-TIME TRACKING & APIS

- Google Maps API: Enables mapping and geolocation services, such as displaying buses on a map, route plotting, and providing real-time updates to users.
- GPS Module: Hardware-integrated with buses to fetch accurate latitude and longitude coordinates in real-time for tracking and location updates.

2.1.4 TESTING TOOLS

- JUnit & Mockito: Used for unit testing backend logic and simulating dependent components, ensuring correctness and robustness of business logic.
- Postman: API testing tool that allows developers to send requests, inspect responses, and debug RESTful services with ease.
- Selenium: Automates UI testing by simulating user interactions in the browser, ensuring smooth and bug-free user experiences.

2.1.5 VERSION CONTROL & DEPLOYMENT

- Git & GitHub: Essential for collaborative development, version history, branching, and continuous integration workflows.
- Apache Tomcat 9+: Java-based web server for deploying Spring Boot applications locally or on-premise.
- AWS / DigitalOcean / Heroku: Cloud hosting providers for deploying backend and frontend services with scalability, high availability, and real-time monitoring capabilities. They offer continuous integration pipelines, storage, and runtime environments.

2.2 HARDWARE REQUIREMENTS

A reliable hardware setup is essential for the development, testing, deployment, and real-time operations of the Transport Management system. These requirements include devices for both the development team and the hardware that must be installed on buses for tracking.

2.2.1 DEVELOPMENT AND DEPLOYMENT ENVIRONMENT

2.2.1.1 Developer Workstations:

- Processor: Intel i5 10th Gen or AMD Ryzen 5 (or higher)
- RAM: Minimum 8 GB (Recommended: 16 GB for faster compilation and multitasking)
- Storage: 512 GB SSD or more (for fast read/write performance)
- OS: Windows 10/11, Ubuntu 20.04+, or macOS
- Display: 1080p monitor for optimal code and UI layout visualization
- Network: High-speed internet for cloning repositories, deploying code, and testing APIs
- Server Hardware (if deploying on-premise instead of cloud):
- Processor: Intel Xeon / AMD EPYC (Multi-core)
- RAM: Minimum 32 GB (for handling multiple concurrent user sessions)
- Storage: 1 TB SSD or RAID setup (for large-scale data storage, logs, and backups)
- Network: High-throughput bandwidth with redundant internet connection
- Power Backup: UPS with at least 2-hour backup time for server continuity

2.2.2 GPS & BUS-SIDE HARDWARE

2.2.2.1 GPS Tracker Module:

- Module Type: SIM-enabled GPS module (e.g., SIM800L + NEO-6M or Quectel L89)
- Accuracy: Up to 2.5 meters
- Power Source: Bus battery connection with voltage regulators
- Connectivity: GSM/4G SIM for data transmission to cloud server
- Microcontroller: ESP32 / Raspberry Pi for data handling and transmission

2.2.2.2 In-Vehicle Display (Optional):

- LED/LCD screen to show route info, ETA, or alerts
- Controlled via Raspberry Pi or other embedded devices
- Emergency Alert Button:
- Panic button installed near the driver and/or passenger seats
- Sends immediate alert to the system upon activation

2.2.2.3 Installation Accessories:

- Wiring kits, mounting brackets, antenna, and power regulators to support GPS and IoT components.

2.2.3 MOBILE DEVICES (FOR TESTING & USER INTERACTION)

- Android Phones (API Level 26+) and iOS Devices for testing mobile responsiveness and real-time tracking experience.
- Tablets (optional) for testing complaint management interface and real-time bus dashboards.

CHAPTER – 3
PROBLEM DESCRIPTION

3.1 ISSUES IN THE PUBLIC TRANSPORTATION SYSTEM

Public transportation, especially bus services, serves as the primary mode of commuting for millions across urban and rural areas. Despite its widespread usage and importance, the current system faces multiple inefficiencies that hinder both passenger satisfaction and operational performance. One of the most pressing issues is the absence of real-time bus tracking. Commuters are often unaware of the current location or delay status of buses, leading to long waiting times and increased uncertainty. Since most systems are based on fixed schedules that do not account for traffic congestion, breakdowns, or unplanned delays, passengers end up with a frustrating and unreliable commuting experience. In many cities, especially smaller ones, there is a complete lack of infrastructure to support automated tracking and live updates, and where such systems do exist, they are often limited to the transport authorities without offering access to the public.

Another critical challenge is the unpredictability of bus arrival times. Passengers have no accurate means of knowing when a bus will arrive, making it difficult to plan their journeys efficiently. Most bus stops lack any digital display boards or mobile app integration that can offer estimated arrival times, forcing commuters to wait for extended periods at stops. This waiting is not only inconvenient but also potentially unsafe, particularly in low-light conditions or isolated locations. The lack of communication between operators and passengers contributes to the frustration and discourages the use of public transport altogether.

Furthermore, there is a major gap in complaint management. Many transportation systems do not offer passengers a structured way to report grievances, whether it's about rude behavior by the driver, poor maintenance of the buses, or incidents involving lost items. The absence of a centralized platform to log and track complaints means that feedback often goes unheard, and issues remain unresolved. Passengers find it challenging to make their voices heard, and when they do, there is usually no follow-up or visibility into whether their concerns were addressed. This lack of accountability weakens public trust in the transportation system.

Emergency support is another area where existing bus services fall short. Women, children, and elderly passengers frequently report feeling unsafe, particularly during night-time travel or when buses are sparsely occupied. In the event of medical emergencies, harassment, or accidents, there is usually no rapid response system in place. Buses often lack emergency alert features such as panic buttons or direct communication channels with emergency services, which delays assistance and puts passengers at risk. The inability to call for help instantly in such scenarios is a serious shortcoming in today’s fast-paced and security-conscious world.

Lastly, inefficiencies in route and fleet management add to the operational burdens of bus operators. Most services are managed manually without any technological support to optimize routes or schedules. This results in poor utilization of buses, missed opportunities to serve high-demand areas, and difficulty in adjusting operations dynamically based on real-time traffic or passenger volume. The systems in place fail to capture essential data like passenger load, vehicle emissions, or maintenance cycles, which are crucial for planning and environmental compliance. With no digital tools to manage or analyze these aspects, fleet operators are often forced to rely on outdated methods that lack accuracy and responsiveness.

3.2 NEED FOR AUTOMATION

Considering the numerous challenges that plague the current public transportation systems, the need for automation has become increasingly evident. An intelligent solution like SISTRaahi can address these issues by transforming the traditional transport framework into a smart, responsive, and commuter-friendly system. By integrating real-time tracking through GPS technology, the system can provide passengers with up-to-date information about bus locations and estimated arrival times, allowing them to plan their journeys more efficiently and reduce idle waiting. This would not only improve the convenience of public travel but also increase trust in the system.

Automation would also enable dynamic route management by analyzing real-time traffic and passenger demand data, thereby ensuring optimal fleet utilization. This helps in reducing operational costs and improving coverage in areas that need better access to transport. Through an integrated digital platform, passengers will be able to submit complaints and track their resolution status, ensuring that feedback is not lost and accountability is maintained. Such a system would bridge the gap between passengers and authorities, making the transportation system more interactive and responsive.

Moreover, incorporating an SOS feature for emergency assistance can significantly enhance passenger safety. A dedicated panic button in the mobile app or within the bus itself would allow passengers to alert authorities instantly in case of emergencies, ensuring timely intervention. This is particularly important for ensuring the safety of vulnerable groups and would encourage more people to use public transport with confidence.

Finally, the system would support data-driven decision-making by providing detailed analytics on bus usage patterns, complaint trends, maintenance schedules, and emissions. This would help operators and authorities to continuously improve services, adopt sustainable practices, and respond quickly to evolving commuter needs. In essence, automation through SISTRaahi would elevate the overall standard of public transportation, making it more reliable, efficient, and safe for everyone.

CHAPTER – 4
LITRATURE SURVEY

4.1 UNDERSTANDING THE PUBLIC TRANSPORTATION DOMAIN

Public transportation, especially bus transit systems, plays a vital role in ensuring mobility for large populations across urban, semi-urban, and rural areas. It serves as a cost-effective and accessible mode of transportation for daily commuters including students, office-goers, and the elderly. Efficiently managing these bus systems is a complex task that involves tracking the location of each vehicle, managing routes dynamically, ensuring timely arrival and departure, and maintaining constant communication with passengers. However, in many regions, the infrastructure supporting these operations is outdated or managed manually. This leads to a host of issues such as delayed buses, inefficient route utilization, overcrowding, and dissatisfaction among commuters. The absence of live tracking systems means passengers are often left guessing when their bus will arrive, which results in long wait times and a lack of trust in public transit.

Moreover, manual management makes it difficult to adapt to real-time challenges such as sudden traffic congestion, roadblocks, or vehicle breakdowns. Fleet operators often have no immediate way to update schedules or reroute vehicles, further aggravating delays. Additionally, without proper communication tools, passengers are unable to receive updates about delays, changes in routes, or emergency situations. As urban populations continue to grow, the strain on traditional systems becomes more evident. Fortunately, the integration of advanced technologies such as GPS-based tracking, cloud computing for data management, and mobile applications for passenger engagement is revolutionizing the way public transport operates. These digital solutions offer real-time visibility, enable data-driven decision-making, and provide better interaction between transport operators and commuters. Embracing these technologies is essential for modernizing public transportation systems and offering a reliable, safe, and commuter-friendly experience.

4.2 KEY TERMINOLOGIES IN THE DOMAIN

In order to fully grasp the technical and functional aspects of automated transportation systems, it is important to understand certain key terminologies used within this domain. GPS Tracking refers to the use of satellite-based navigation systems to pinpoint the real-time geographical location of vehicles such as buses. This information can be shared with both operators and passengers to improve transparency and planning. ETA, or Estimated Time of Arrival, is a calculated value that determines when a bus is expected to arrive at a specific stop, taking into consideration its current location, speed, and real-time traffic conditions.

Fleet Management encompasses the overall supervision, coordination, and optimization of a collection of buses. It includes monitoring fuel usage, maintenance schedules, driver behavior, and passenger loads to enhance operational efficiency. Automated Ticketing is a system that allows passengers to book tickets through digital platforms like mobile apps or websites, thereby eliminating the need for cash payments and reducing wait times at counters. Another significant component is the Public Transport Complaint System. This is a dedicated platform where passengers can submit grievances, feedback, or suggestions regarding bus services. An effective complaint system ensures that transport authorities are able to receive, process, and resolve issues in a timely and structured manner. Understanding these terms is essential for both the implementation and management of any intelligent transportation system.

4.3 REAL-TIME WORKING OF BUS TRANSIT SYSTEMS

A real-time bus transit system functions through a sequence of interconnected steps that begin long before a bus hits the road and continue throughout its journey. The process starts with Route Planning, where transport authorities design fixed or flexible routes based on population density, commuter demand, road infrastructure, and connectivity needs. These routes are accompanied by predefined schedules that specify bus frequency and timing. Next is Vehicle Assignment, where buses are allocated to specific routes depending on the number of buses available, peak travel hours, and expected commuter volume. This step ensures that no route is underserved and helps balance fleet distribution efficiently.

Once assigned, buses begin operating on their respective routes, picking up and dropping off passengers at designated stops. However, many systems lack real-time tracking capabilities, meaning passengers are unaware of the bus's exact location or delay status. As a result, they wait at stops without knowing when the next bus will arrive, often leading to frustration and potential safety risks. Furthermore, during Passenger Boarding and Travel, any service issues such as driver misconduct, overcrowding, or mechanical problems cannot be reported instantly due to the absence of an integrated complaint mechanism. Complaints are typically registered manually through phone calls or office visits, which significantly slows down the resolution process.

In addition to routine operations, Emergency Handling is a crucial part of the system. Unfortunately, in most conventional setups, there are no provisions for real-time emergency communication. In case of an accident, health emergency, or security concern, passengers must rely on bystanders or external help to alert authorities. This delay in communication can have serious consequences, especially in urgent situations. A modern and automated transit system addresses all these limitations by incorporating real-time GPS tracking, dynamic scheduling, a centralized complaint system, and emergency alert mechanisms that allow passengers to travel more confidently and safely. These systems ensure better coordination among all stakeholders—passengers, drivers, and transport authorities—resulting in a significantly improved public transportation experience.

CHAPTER – 5
SOFTWARE REQUIREMENTS SPECIFICATION

5.1 FUNCTIONAL REQUIREMENTS

These are the essential operations and features that the SISTRaahi public transportation management system must support in order to fulfill its intended purpose effectively and efficiently.

1. User Registration and Authentication:
The system must allow users, including passengers and administrators, to register and authenticate securely. New users should be able to create accounts using their email, phone number, or social media credentials. Once registered, they should be able to log in and log out using a secure authentication mechanism. Admins will be provided with elevated privileges to manage and oversee all system functions, such as bus information updates, user management, and complaint monitoring. Additionally, the system should support session timeouts, password encryption, and account recovery features through email or OTP verification.

2. Live Bus Tracking:
The system should provide users with the ability to track buses in real time using GPS modules installed in each bus. This tracking feature must refresh frequently (every few seconds) and offer smooth visualization of the bus movement on an interactive map interface, such as Google Maps. Passengers should be able to see the current position of a bus, its route progress, estimated time to arrival, and any delays due to traffic congestion or breakdowns. The tracking should be dynamic and responsive across all platforms including web and mobile applications.

3. Route and Bus Search:
Passengers should be able to search for available buses based on multiple input criteria like bus number, route number, origin or destination stop, and stop names. The system must instantly return relevant bus details such as the estimated arrival time at a particular stop, route map, bus capacity status (e.g., crowded or empty), and real-time tracking option. The search mechanism should be fast and accurate, offering predictive suggestions and filters to enhance the search experience.

4. Estimated Arrival Time Calculation:
The system must provide passengers with accurate ETAs (Estimated Time of Arrival) when they input their destination or select a bus. The calculation should take into account real-time traffic data, bus speed, route deviations, and known delays. This information should be displayed clearly on the user interface, helping passengers plan their journey effectively. Additionally, the system can notify users of expected arrival times via push notifications or SMS alerts if they subscribe.

5. Admin Panel for Bus Management:
Administrators should be provided with a secure and feature-rich admin dashboard to monitor and manage the entire fleet. This panel must support the addition, update, and deletion of bus information including bus number, driver details, route assignments, and operational timings. It

should allow real-time status tracking of each bus, receive and review passenger complaints, monitor traffic patterns, and access analytics related to passenger usage and route efficiency.

6. Emergency Assistance & Safety Features:

The system should prioritize commuter safety by providing built-in emergency assistance functionalities. A chatbot or complaint bot must be available for users, particularly women and vulnerable groups, to instantly report incidents like harassment, theft, or health emergencies. The application should have an SOS feature that allows users to share their live location with emergency contacts or alert transport authorities. The system should display emergency helpline numbers and support direct contact to nearby police or ambulance services in a single tap.

7. User Feedback System:

To maintain service quality, the platform should offer a structured user feedback system where passengers can submit complaints, suggestions, and feedback. These submissions should be categorized (e.g., service delay, safety, cleanliness, driver behavior) and automatically routed to the concerned admin for review. Users should receive updates on complaint status (e.g., under review, resolved), and admins should be able to respond directly within the app or portal. Feedback analytics should help in service improvement.

8. Emission Compliance Check:

Environmental sustainability is a key component of modern transportation. The system should display emission standards and compliance information for each bus such as whether it adheres to Bharat Stage IV, V, or VI norms. Passengers should be able to view these details in the app, and authorities should be notified if a bus falls out of compliance. This will ensure eco-friendly fleet operation and help in regulatory audits and awareness among passengers.

5.2 NON-FUNCTIONAL REQUIREMENTS

These define the quality attributes of the SISTRaahi system and how well it performs under various conditions.

1. Performance Requirements:

The system must be optimized for speed and responsiveness. It should process all user requests, including bus search and tracking updates, within 2 seconds under normal network conditions. GPS data for live bus tracking should refresh every 10 seconds to ensure real-time accuracy. Page load time should remain below 3 seconds even with high user activity. System response must remain smooth during both peak and non-peak hours.

2. Scalability:

The application should be designed to scale both vertically and horizontally. It must handle increasing user traffic efficiently, supporting up to 10,000 concurrent users without significant performance degradation. It should also allow for the seamless addition of new buses, routes, stops, and admin accounts. The system architecture should support load balancing, efficient database indexing, and auto-scaling of cloud resources when traffic spikes occur.

3. Security Requirements:

Security is crucial, especially for protecting user credentials and location data. All user authentication processes should implement JWT (JSON Web Token) for secure, stateless sessions. Sensitive data like passwords and personal information must be encrypted using AES-256. The system must include role-based access control (RBAC), ensuring that only authorized personnel can view or modify critical data such as bus routes, user complaints, or emergency reports. The system should also maintain logs of admin activities for audit purposes.

4. Usability & Accessibility:

The system should provide an intuitive and user-friendly interface accessible to users of varying digital literacy. It must be responsive, adapting to different screen sizes including desktops, laptops, tablets, and smartphones. The UI/UX should be designed with clear navigation, accessibility options like large text mode and screen reader support, and language localization features to serve users from diverse backgrounds. Minimal training should be required to use the system effectively.

5. Availability & Reliability:

The SISTRaahi system must ensure a minimum uptime of 99.9%, enabling continuous and uninterrupted service. It should be deployed on a cloud platform that supports failover mechanisms and auto-recovery features. In case of unexpected server failure or crash, the system should be capable of recovering operations and restoring full functionality within 5 minutes. A notification mechanism must be in place to alert administrators of system failures.

6. Maintainability & Modularity:

To support long-term upgrades and troubleshooting, the system should be built using modular architecture following MVC (Model-View-Controller) principles. This modularity will allow developers to modify individual components without affecting the entire system. Code documentation and version control should be maintained strictly. The system must support plugin-based integration with third-party APIs for features like payment gateways, location services, and SMS alerts.

7. Compliance & Legal Requirements:

The application must comply with legal frameworks such as GDPR (General Data Protection Regulation) to protect user data privacy and ensure lawful data processing. It should also adhere to national transportation safety and digital service regulations. Any user data collected must be done with consent and used transparently. Logs of data access, consent history, and privacy policies must be clearly maintained and shared with users upon request.

CHAPTER – 6
SOFTWARE DESIGN

6.1 USE CASE DIAGRAM

A Use Case Diagram represents the interaction between users (actors) and the system. It includes the functionalities of the system and how users interact with them.

6.1.1 ACTORS IN THE SYSTEM:

Passenger (User): Can search for buses, track buses, check estimated arrival times, report complaints, and request emergency assistance.
Admin: Manages buses, handles complaints, and oversees system functionality.
GPS Device: Sends real-time bus location data to the system.

Use Case Diagram:

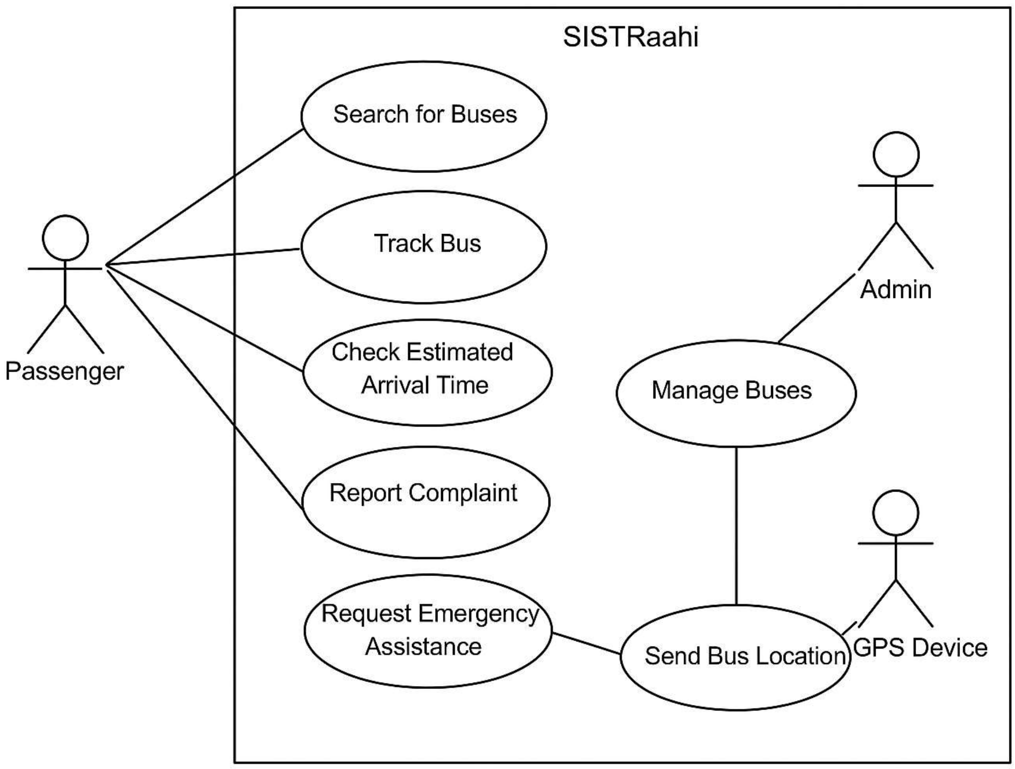

Figure 6.1 : Use Case

6.2 ER – DIAGRAM

The ER (Entity-Relationship) diagram of the Bus Management System represents the logical structure of the database. It includes key entities such as User, Bus, Route, Complaint, and Emergency Alert. Relationships are established to reflect real-world interactions, such as a user registering complaints, or a bus being assigned to a specific route. Primary and foreign keys maintain referential integrity between tables. This design ensures efficient data organization, reduces redundancy, and enhances system scalability. The ER diagram plays a critical role in visualizing data flow and supports the development of a robust and well-structured database system.

Figure 6.2: ER Diagram

6.3 TABLE STRUCTURE

Table 6.1: User table

Field Name	Data Type	Constraints	Purpose
user_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique ID for each user
name	VARCHAR(100)	NOT NULL	Full name of the user
email	VARCHAR(100)	UNIQUE, NOT NULL	User's email address
password	VARCHAR(255)	NOT NULL	Encrypted password for login
role	VARCHAR(20)	DEFAULT 'passenger'	Role of the user (admin/passenger)

Table 6.2: Bus table

Field Name	Data Type	Constraints	Purpose
bus_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique ID for each bus
bus_number	VARCHAR(50)	UNIQUE, NOT NULL	Unique identification number of the bus
driver_name	VARCHAR(100)	NOT NULL	Name of the bus driver
route_id	INT	FOREIGN KEY	Link to the route on which bus operates
status	VARCHAR(20)	DEFAULT 'active'	Operational status of the bus

Table 6.3: Route Table

Field Name	Data Type	Constraints	Purpose
route_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique ID for each route
source	VARCHAR(100)	NOT NULL	Starting location of the route
destination	VARCHAR(100)	NOT NULL	Ending location of the route
distance_km	FLOAT		Distance in kilometers
duration	VARCHAR(50)		Estimated duration

Table 6.4: Complaint Table

Field Name	Data Type	Constraints	Purpose
complaint_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique ID for each complaint
user_id	INT	FOREIGN KEY	ID of the user filing the complaint
description	TEXT	NOT NULL	Complaint details
date_time	DATETIME	DEFAULT CURRENT_TIMESTAMP	Date and time of complaint submission
status	VARCHAR(20)	DEFAULT 'pending'	Status of the complaint (pending/resolved)

Table 6.5: Emergency Alert Table

Field Name	Data Type	Constraints	Purpose
alert_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique ID for each alert
user_id	INT	FOREIGN KEY	ID of the user who sent the alert
bus_id	INT	FOREIGN KEY	ID of the bus during alert
message	TEXT	NOT NULL	Alert message
date_time	DATETIME	DEFAULT CURRENT_TIMESTAMP	Time of emergency alert

CHAPTER – 7
OUTPUT SCREEN

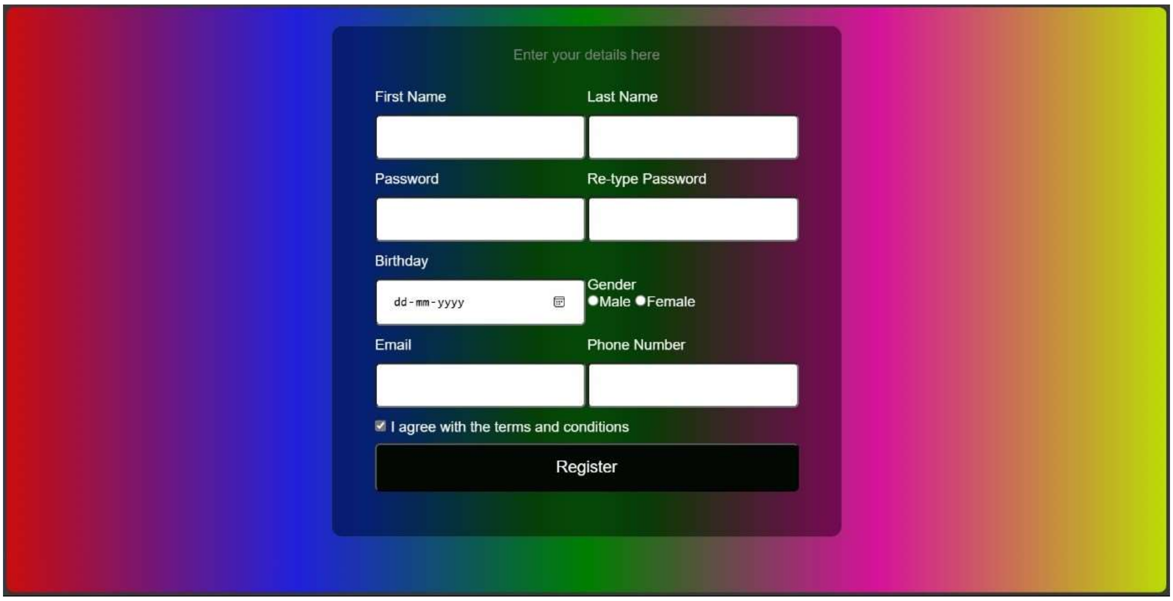

Figure 7.1: Registration

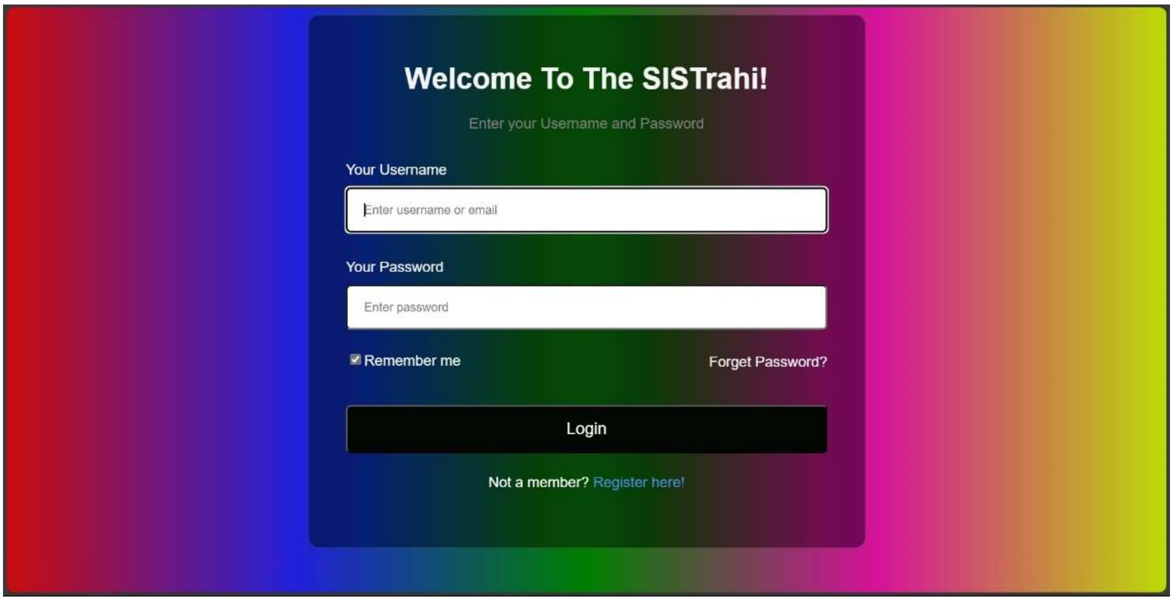

Figure 7.2 : Login

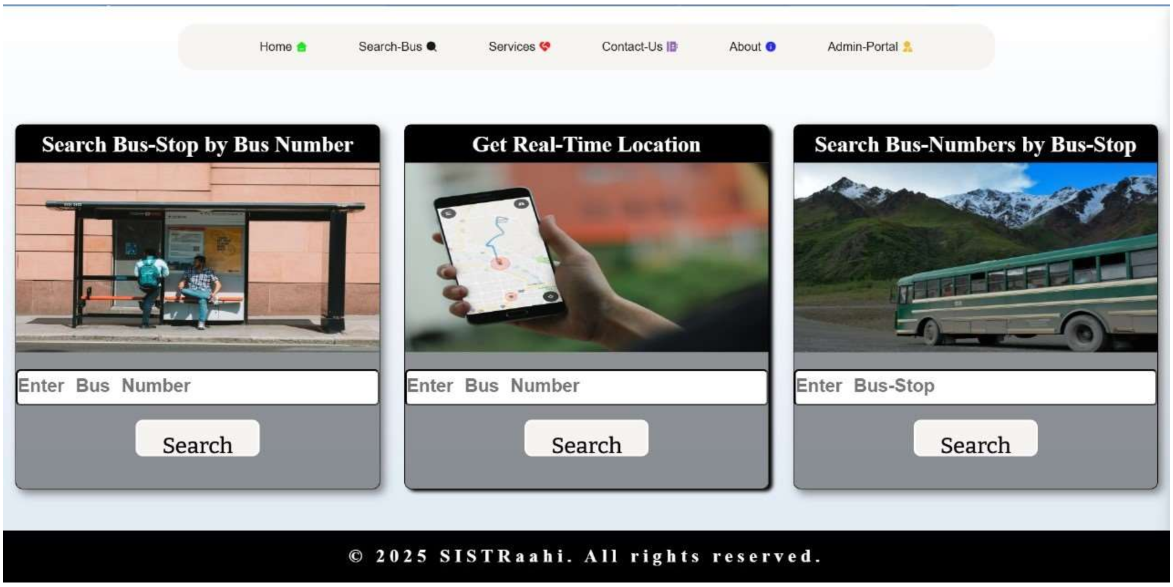

Figure 7.3 : Home Page

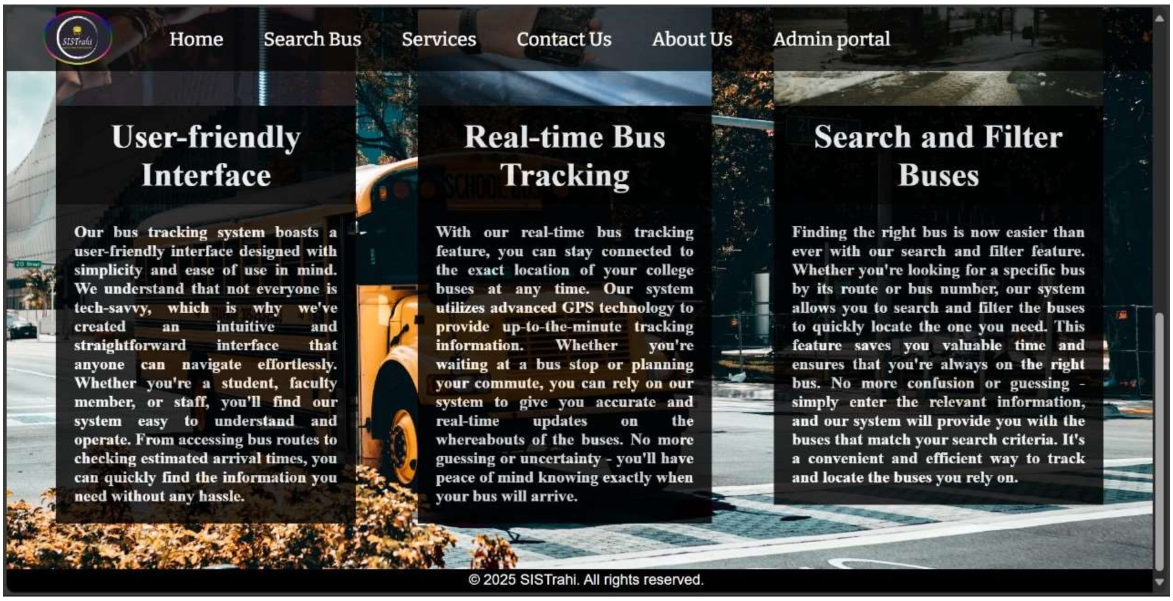

Figure 7.4: Dashboard

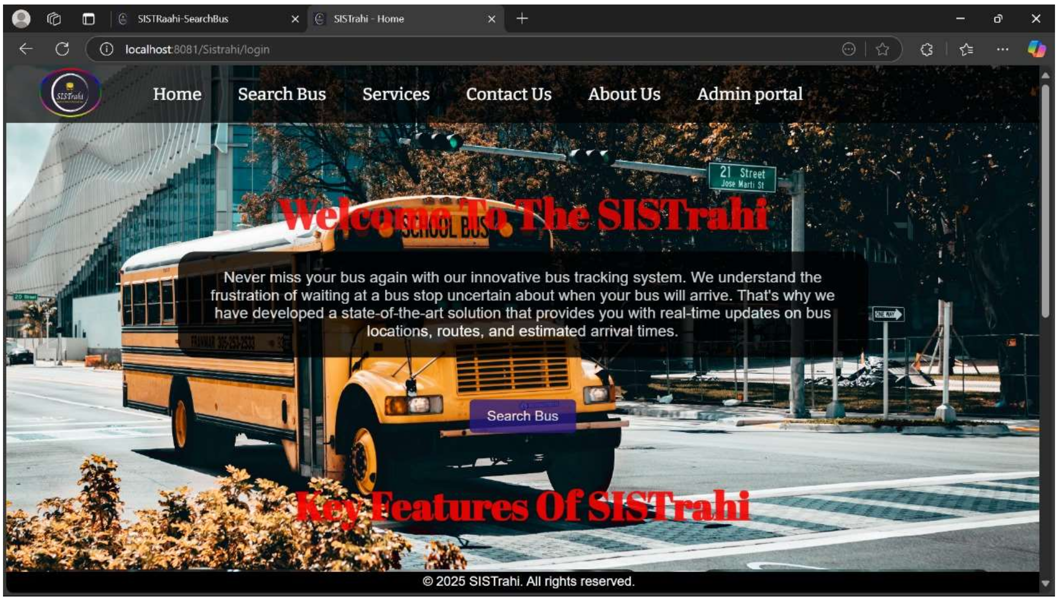

Figure 7.5: Home Page

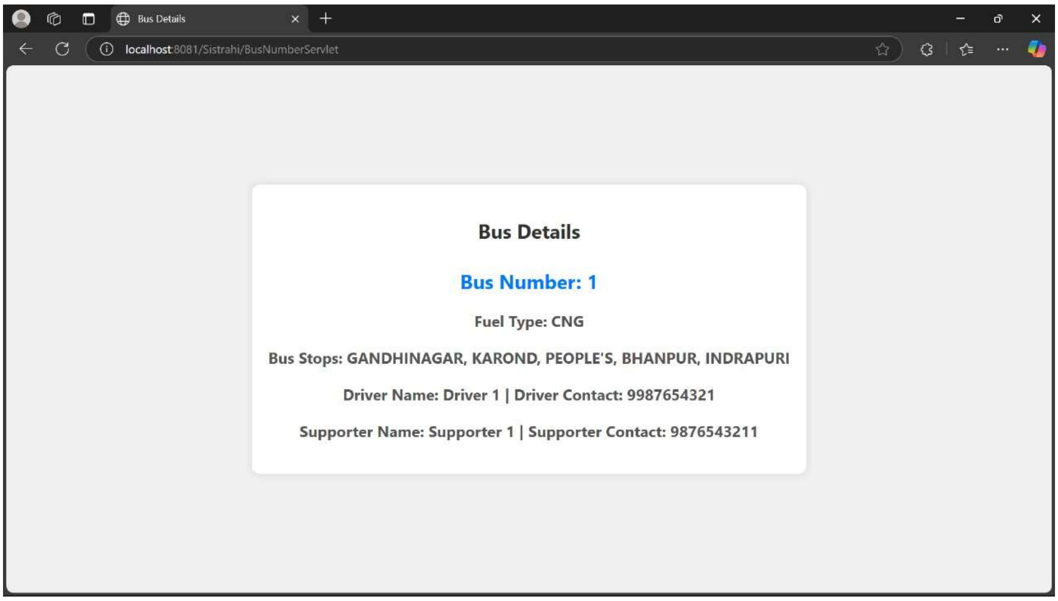

Figure 7.6 : Bus Detail

CHAPTER – 8
DEPLOYMENT

This chapter outlines the steps required to install, configure, and deploy the Bus Management System to bring it into a fully functional, ready-to-use state. The application has been developed using Java technologies including Servlets, JSP, and JDBC, with MySQL as the backend database and Apache Tomcat as the web server.

8.1 PREREQUISITES

Before proceeding with deployment, the following software and configurations must be available:
Software Requirements:

- Java Development Kit (JDK) 8 or higher
- Apache Tomcat Server version 9 or above
- MySQL Server version 8.0+
- MySQL Workbench (for database operations)
- IDE (Eclipse / IntelliJ IDEA) – optional for further development or testing
- Web Browser (Chrome / Firefox)
- Google Maps API Key (for location-based services)

8.2 DATABASE CONFIGURATION

- Install and launch MySQL Server and open MySQL Workbench.
- Create a new schema/database named sistrahi_db.
- Execute the SQL script (sistrahi_db.sql) provided with the project to create necessary tables and insert sample data.
- Update the database connection details in the Java code (typically within the DAO layer or a configuration class):
- String url = "jdbc:mysql://localhost:3306/sistrahi_db";
- String username = "root";
- String password = "root";

8.3 APPLICATION DEPLOYMENT ON APACHE TOMCAT

- Copy the compiled project (WAR file or project directory) to the webapps folder inside the Apache Tomcat installation directory.
- Start the Apache Tomcat server:

    Windows: Run startup.bat located in tomcat/bin

    Linux/Mac: Run ./startup.sh in the same directory

- Once the server starts successfully, open a web browser and enter the following URL: http://localhost:8080/SISTRaahi
- The application should now be live and accessible. Users can register, log in, track buses in real-time, raise complaints, and access emergency features.

8.4 DEPLOYMENT TO CLOUD ENVIRONMENT

For wider accessibility and real-time usage, the application can also be hosted on cloud platforms such as:

- AWS EC2 instance running Apache Tomcat and MySQL
- Steps for cloud deployment:
- Upload the WAR file to the cloud server's Tomcat webapps folder.
- Install and configure MySQL on the cloud server or connect to a remote RDS instance
- Open necessary ports (like 8080) in the firewall to allow HTTP access.
- Update the base URL and database connection string if required.

8.5 POST-DEPLOYMENT CONSIDERATIONS

- Ensure database credentials are stored securely and not exposed in the source code.
- Monitor the application logs regularly using Tomcat's logs directory to handle unexpected errors or crashes.
- Backup the database periodically to avoid data loss.
- Regular maintenance and updates are recommended to ensure application security and performance.