SADAK SURAKSHA

[ROAD SAFTY & POTHOL HAZARD SYSTEM]

A PROJECT REPORT

Submitted by

PRASANNA RAMAKANTA PRUSTY [22BECE30388]

PURANI HARSHAD PRADIPKUMAR [22BECE30389]

RAJPUT SHIVAM SUBHASHBHAI [22BECE30393]

RANA DHRUVESHKUMAR SUBHASHBHAI [22BECE30396]

In fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH,

KADI SARVA VISHWAVIDYALAYA,

GANDHINAGAR 2025-26.

January-2025

LDRP Institute of Technology and Research

Computer Engineering Department

CERTIFICATE

This is to certify that the Project Work entitled "**SADAK SURAKSHA [ROAD SAFTY AND POTHOLE HAZARDS SYSTEM]"** has been carried out by **RANA DHRUVESHKUMAR SUBHASHBHAI [22BECE30396]** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering (7th Semester) of **Kadi Sarva VishwaVidyalaya University, Gandhinagar** during the academic year 2025-26.

NAME OF GUIDE :-
Prof. Ankita Shah
INTERNAL GUIDE
LDRP-ITR.

NAME OF HOD:-
Dr. Ashish Patel
LDRP-ITR.

January-2025

LDRP Institute of Technology and Research

Computer Engineering Department

CERTIFICATE

This is to certify that the Project Work entitled **"SADAK SURAKSHA [ROAD SAFTY AND POTHOLE HAZARDS SYSTEM]"** has been carried out by **PURANI HARSHAD PRADIPKUMAR [22BECE30389]** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering (7th Semester) of **Kadi Sarva VishwaVidyalaya University, Gandhinagar** during the academic year 2025-26.

NAME OF GUIDE :-
Prof. Ankita Shah
INTERNAL GUIDE
LDRP-ITR.

NAME OF HOD:-
Dr. Ashish Patel
LDRP-ITR.

January-2025

LDRP Institute of Technology and Research

Computer Engineering Department

CERTIFICATE

This is to certify that the Project Work entitled **"SADAK SURAKSHA [ROAD SAFTY AND POTHOLE HAZARDS SYSTEM]"** has been carried out by **PRASANNA RAMAKANTA PRUSTY [22BECE30388]** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering (7^{th} Semester) of **Kadi Sarva VishwaVidyalaya University, Gandhinagar** during the academic year 2025-26.

NAME OF GUIDE :-
Prof. Ankita Shah
INTERNAL GUIDE
LDRP-ITR.

NAME OF HOD:-
Dr. Ashish Patel
LDRP-ITR.

January-2025

LDRP Institute of Technology and Research

Computer Engineering Department

CERTIFICATE

This is to certify that the Project Work entitled "**SADAK SURAKSHA [ROAD SAFTY AND POTHOLE HAZARDS SYSTEM]"** has been carried out by **RAJPUT SHIVAM SUBHASHBHAI [22BECE30393]** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering (7th Semester) of **Kadi Sarva VishwaVidyalaya University, Gandhinagar** during the academic year 2025-26.

NAME OF GUIDE :-
Prof. Ankita Shah
INTERNAL GUIDE
LDRP-ITR.

NAME OF HOD:-
Dr. Ashish Patel
LDRP-ITR.

TABLE OF CONTENT

1. Introduction

1.1 Introduction

The Sadak Suraksha platform is a comprehensive web solution designed to streamline the reporting and resolution of road hazards for both citizens and municipal authorities. It simplifies the complaint process by allowing users to submit pothole or hazard reports—with location, category, priority, and optional photo—via an intuitive online form, while instantly plotting each issue on an interactive map for real-time visibility. This system integrates key functionalities, including user-friendly issue submission, live status tracking, an admin dashboard for task assignment and progress updates, and analytics to monitor repair performance. By automating these workflows, Sadak Suraksha empowers citizens to participate in roadway maintenance, accelerates municipal response times, and enhances overall road safety and infrastructure reliability.

1.2 Scope

Sadak Suraksha is designed to be used by citizens to report potholes and by municipal authorities to manage and track those reports. The system supports real-time mapping, priority handling, complaint status updates, and administrative dashboards for efficient decision-making. It is scalable to multiple cities and regions and adaptable for future enhancements such as AI-based image analysis or mobile app extensions.

1.3 Project Summary and Purpose

The primary purpose of Sadak Suraksha is to enhance urban road safety by providing an accessible, user-driven platform for reporting road hazards. By bridging the communication gap between citizens and road maintenance teams, the system promotes timely issue resolution, helps prevent accidents, and supports better infrastructure planning.

1.4 Overview of the Project

Sadak Suraksha is developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js). The platform includes a user-friendly reporting interface, an admin panel with complaint management tools, and a mapping system to visualize reported locations. It also provides analytics features to help authorities prioritize tasks based on severity and area importance.

1.5 Problem Definition

The challenge in urban road maintenance lies in enabling a fast, reliable, and transparent mechanism for citizens to report potholes and hazards—and for authorities to track and resolve them efficiently. Current reporting methods often suffer from long response times, lack of status visibility, and fragmented data collection. Sadak Suraksha addresses these problems by providing a seamless web-based platform that connects road users with municipal maintenance teams instantly—ensuring immediate location-aware reporting.

2. Technology and Literature Review

2.1 About Tools and Technology

Sadak Suraksha is developed using the MERN stack—a combination of MongoDB, Express.js, React.js, and Node.js. MongoDB is used as the NoSQL database for storing user reports and complaint data. Express.js and Node.js power the backend server, handling APIs and request routing, while React.js is used for building the interactive and responsive user interface.

Other technologies used include:
- Leaflet.js: For map integration and visualization of geotagged pothole reports.
- Multer and Cloudinary: For image upload and storage handling.
- JWT (JSON Web Tokens): For user authentication and secure login sessions.
- Tailwind CSS or Bootstrap: For styling the frontend components and layout.

2.2 Brief History of Work Done

Urban infrastructure monitoring has increasingly leveraged digital technologies to empower citizens and streamline maintenance workflows. Pothole reporting systems and civic engagement platforms have evolved to provide rapid, location-aware issue tracking and resolution.

Crowdsourced Civic Reporting Platforms: Research shows that citizen-driven reporting apps harness smartphones' GPS and camera features to simplify fault submission directly from the road surface. When residents can instantly capture and send geotagged images, community participation increases significantly, leading to faster municipal response times.

Real-Time Mapping and Geo-Visualization: GIS-enabled solutions allow maintenance teams to visualize clusters of reports and prioritize repairs more effectively. These platforms help improve operational efficiency by enabling location-based task assignment.

User Experience and Accessibility: Successful road-hazard reporting platforms focus on intuitive, minimal-step submission workflows. Streamlined, responsive web or mobile internet.

Administrative Task Management: Backend dashboards provide tools for managing reports, assigning tasks to field teams, and sending updates to users. Real-time notifications and transparentsy.

Security and Data Integrity: Systems implement measures such as user authentication and secure uploads to maintain data reliability and user privacy. Robust validation mechanisms ensure only legitimate reports enter the system.

3. System Requirements Study

3.1 User Characteristics

The users of the Sadak Suraksha system can be broadly categorized into:
- Citizens: Individuals who report potholes and road hazards. They are not required to have technical knowledge and should be able to navigate the platform with minimal guidance.
- Admins: Municipal officers responsible for reviewing reports, assigning resolution tasks, and updating complaint statuses. They are expected to be familiar with web-based dashboards and administrative tools.

3.2 Hardware and Software Requirements

Hardware Requirements:

- For Users: Any smartphone or computer with internet connectivity.

- For Admins: Desktop or laptop with internet connectivity.

Software Requirements:

- Frontend: React.js (with HTML, CSS, JavaScript)

- Backend: Node.js with Express.js

- Database: MongoDB Atlas

- Hosting Platforms: Vercel (frontend), Render or Railway (backend)

- Browser: Chrome, Firefox, or any modern web browser

3.3 Constraints

3.3.1 Regulatory Policies

The application must comply with municipal data sharing guidelines and digital privacy laws (such as the IT Act in India). Any personal data (e.g., user email, location) must be securely stored and never shared with third parties without consent.

3.3.2 Hardware Limitations

The platform is optimized for standard desktop browsers and recent mobile devices. However, users on low-end hardware or with outdated browsers may face performance issues. Map loading and image rendering can be resource-intensive on older phones.

3.3.3 Interfaces to Other Applications

Sadak Suraksha integrates with:
- Google Maps or Leaflet.js for geolocation and mapping.

- Cloudinary or Firebase for image hosting.
- JWT-based authentication libraries.

Future versions may include API interfaces for city-wide GIS databases or public works systems.

3.3.4 Parallel Operations

The system supports multiple concurrent user sessions and simultaneous complaint submissions. Backend APIs and the database are designed to handle asynchronous operations using non-blocking I/O and efficient query patterns to ensure scalability.

3.3.5 Higher Order Language Requirements

The system is primarily built using JavaScript (Node.js, React.js), with HTML/CSS for the frontend. MongoDB handles NoSQL data storage. No assembly or low-level programming languages are used. Compatibility with JSON-based REST APIs is critical.

3.3.6 Reliability Requirements

System must maintain >99% uptime for complaint submission and dashboard visibility. Auto-restart mechanisms, logging tools, and monitoring dashboards will ensure high availability. Data integrity is ensured using atomic writes and validation middleware.

3.3.7 Criticality of the Application

Sadak Suraksha is not life-critical but directly impacts civic engagement and urban safety. Delays or downtime could reduce public trust and responsiveness from authorities. Therefore, consistent availability and accurate data handling are essential.

3.3.8 Safety and Security Considerations

Security implementations include:
- Encrypted passwords (bcrypt).
- Token-based authentication (JWT).
- Input validation and sanitization to prevent XSS and SQL/NoSQL injections.
- Role-based access control to restrict admin functions.
- Secure API endpoints and database connection handling.

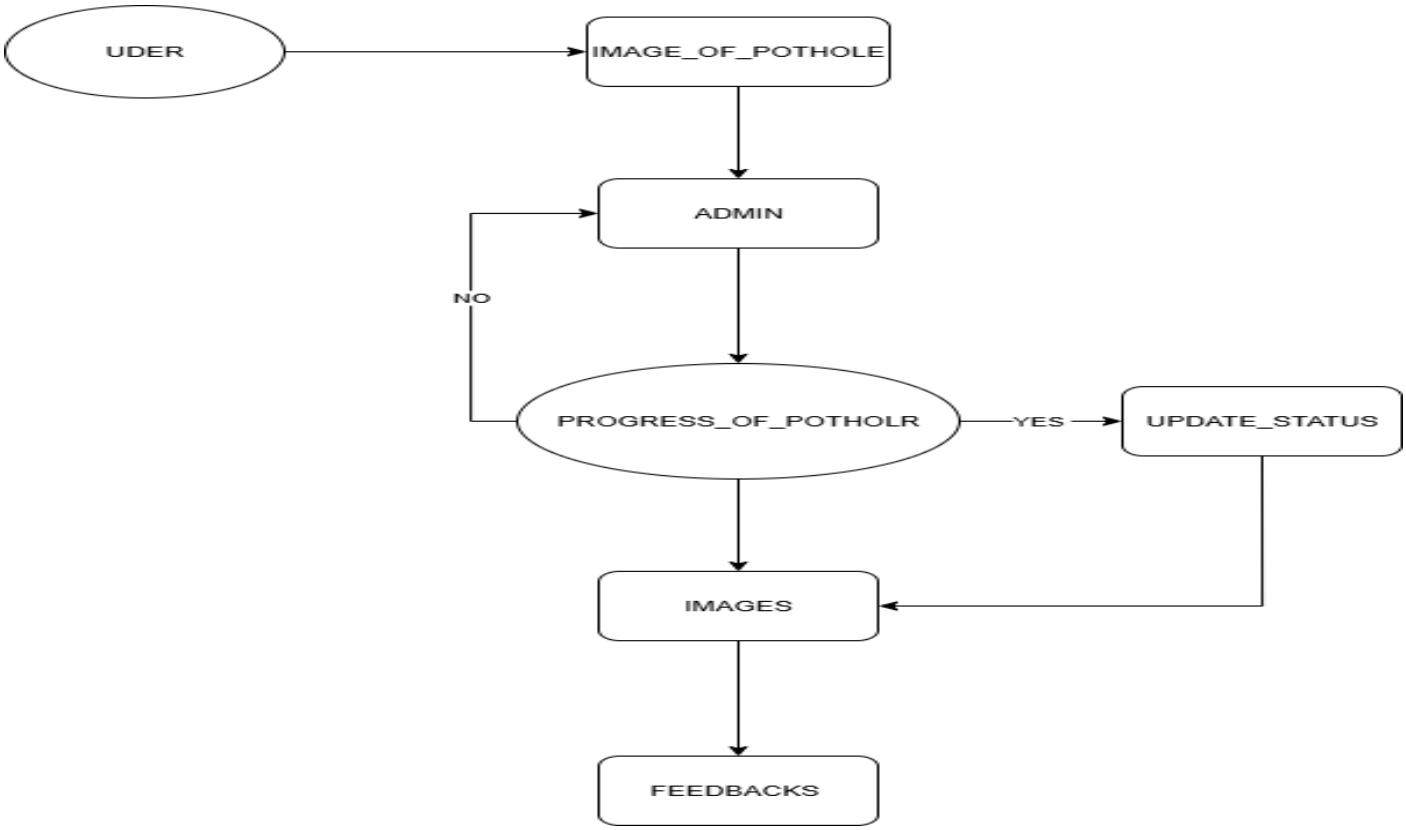
3.4 Assumptions and Dependencies

- Users have access to a GPS-enabled device to submit location-based complaints.
- Admins are responsive and able to manage complaints regularly
- Hosting and database services remain available during operation.
- System relies on third-party services such as map APIs and image storage.

4. System Analysis

4.1 Study of Current System

The current process for reporting potholes and road hazards is typically offline or based on outdated portals. Users often call municipal helplines or send emails, which lack location tagging, structured complaint formats, and status tracking. As a result, maintenance workflows are slow, uncoordinated, and opaque to the public.



4.1 system architecture

4.2 Problem and Weaknesses of Current System

- No real-time status updates or complaint tracking for users
- Manual and delayed report intake
- Lack of geolocation and multimedia data with reports
- Limited administrative insight into hotspots or unresolved cases
- Low civic engagement due to poor user experience

4.3 Requirements of New System

4.3.1 User Requirements

The new system must fulfill the following user expectations:

- Simple, intuitive interface for submitting pothole and hazard complaints.
- GPS-enabled location tagging for accurate report submission.
- Image upload functionality with optional description.
- Complaint history and status tracking.
- Multilingual interface (optional in future).

4.3.2 System Requirements

- Frontend: React.js-based SPA (Single Page Application).
- Backend: Node.js with Express framework.
- Database: MongoDB for non-relational document storage.
- Deployment: Cloud platforms like Vercel (frontend) and Render (backend).
- Other Dependencies: JWT for authentication, Cloudinary for image storage, Leaflet for map rendering.

4.4 Feasibility Study (Extended)

4.4.1 Contribution to Organization Objectives

Sadak Suraksha aligns with the objectives of municipal and public safety bodies by:
- Streamlining the road maintenance complaint process.
- Increasing public engagement in civic management.
- Providing authorities with real-time geotagged data to act swiftly.

4.4.2 Implementation within Tech, Cost & Schedule Constraints

The system is designed using open-source tools and cloud services with free-tier deployment options, making it affordable. The MERN stack ensures development can proceed within an academic semester using agile sprints and parallel module development.

4.4.3 Integration with Other Systems

The architecture supports modular API integration, enabling future extensions with:
- GIS-based city planning dashboards.
- Smart city IoT frameworks for sensor-based road monitoring.
- Municipal portals for centralized complaint handling.

4.6 Activity/Process in New System

1. User submits a pothole or hazard report with description, photo, and location.
2. The report is stored in the database and visible on the live map.
3. Admin reviews the report and assigns it to the relevant department.
4. Status is updated through admin actions (e.g., In Progress, Resolved).
5. User receives status updates and can view report history.

4.7 Features of New System

- Web-based reporting portal for citizens
- Interactive map to view complaint clusters
- Admin dashboard to manage and track reports
- Image uploads for visual evidence
- Priority tagging and automated complaint sorting
- Analytics dashboard for authorities

5. System Design

The design of Sadak Suraksha is modular and follows the MVC (Model-View-Controller) architecture pattern. The frontend is developed using React.js, while the backend API is managed by Express.js and Node.js. The database layer uses MongoDB for storing user and complaint data.

Key design components include:
- User Interface: Web form for submitting reports, complaint status tracking, and map view.
- Admin Interface: Dashboard for viewing complaints, updating status, and assigning tasks.
- RESTful APIs: Interfaces to connect the frontend with backend services.
- Data Model: MongoDB collections to store reports, user details, and status logs.

5.1 System Application Design

5.1.1 Method Pseudo Code

Below is a simplified pseudo code for the complaint submission and categorization logic:

```
Method: submitComplaint(userInput)
   Validate(userInput)
   location ← getLocation(userInput.coordinates)
   priority ← calculatePriority(userInput.image, userInput.severity, locationData)
   SaveToDatabase({userID, location, description, priority, image})
   NotifyAuthorities(priority, location)
   Return confirmation

Method: calculatePriority(image, severity, locationData)
   if severity == 'High' or imageAnalysis(image) indicates large pothole
      return 'Critical'
   else if locationData = High Traffic Zone
      return 'High'
   else
      return 'Moderate'
```

5.2 Database Design / Data Structure Design

5.2.1 Table and Relationship

Main Tables:
- Users (UserID, Name, Email, Role)
- Complaints (ComplaintID, UserID, Location, ImageURL, Description, Priority, Status, Timestamp)
- Feedback (FeedbackID, ComplaintID, UserID, Comment, Timestamp)

Relationships:
- One-to-Many: User → Complaints
- One-to-Many: Complaint → Feedback

5.2.2 Logical Description of Data

- ComplaintID: Auto-generated unique identifier for each report.
- UserID: Links the report to the reporter.
- Priority: Derived from severity, image and traffic sensitivity.
- Location: Stored as latitude and longitude.
- ImageURL: Stored via Cloudinary/Firebase and linked in database.
- Timestamp: Date and time of submission.
- Status: Open, In-Progress, Resolved.

5.3 Input/Output and Interface Design

5.3.1 State Transition / UML Diagram (Description)

State transitions:
- Initial State: User logs in
- Report Creation: User selects location, uploads image, enters description
- Submitted State: Complaint is saved to DB and shown in history
- Assigned State: Admin assigns it to maintenance crew
- Resolved State: Marked resolved with a reply

UML Diagram representation involves: User → Form UI → Submission → Backend API → Database
→ Admin Dashboard → Update/Resolve
→ User View Complaint Status

5.3.2 Samples of Forms, Reports and Interface (Description)

Forms:
- Complaint Submission Form: Location picker (map), text input for description, photo upload button.
- Login/Register Form: Standard fields with email and password.

Reports:
- Admin View: Complaint list sorted by priority with status update options.

Interfaces:
- User Dashboard: Submit complaint, view history.
- Admin Dashboard: View all complaints, filter by status/priority, update complaint state.

6. System Testing

The system was tested for functionality, usability, and responsiveness across multiple devices and browsers. Testing included the following types:

- Unit Testing: Ensured individual components like form submission, API requests, and authentication worked as intended.
- Integration Testing: Validated that frontend forms correctly connected to backend services and database.
- UI/UX Testing: Verified that the interface was intuitive, mobile responsive, and user-friendly.
- System Testing: Full end-to-end tests for report submission, admin update, and status feedback.
- Performance Testing: Checked the time required to load maps and handle multiple simultaneous submissions.

All major test cases passed and no critical issues were found during evaluation.

7. System Diagrams

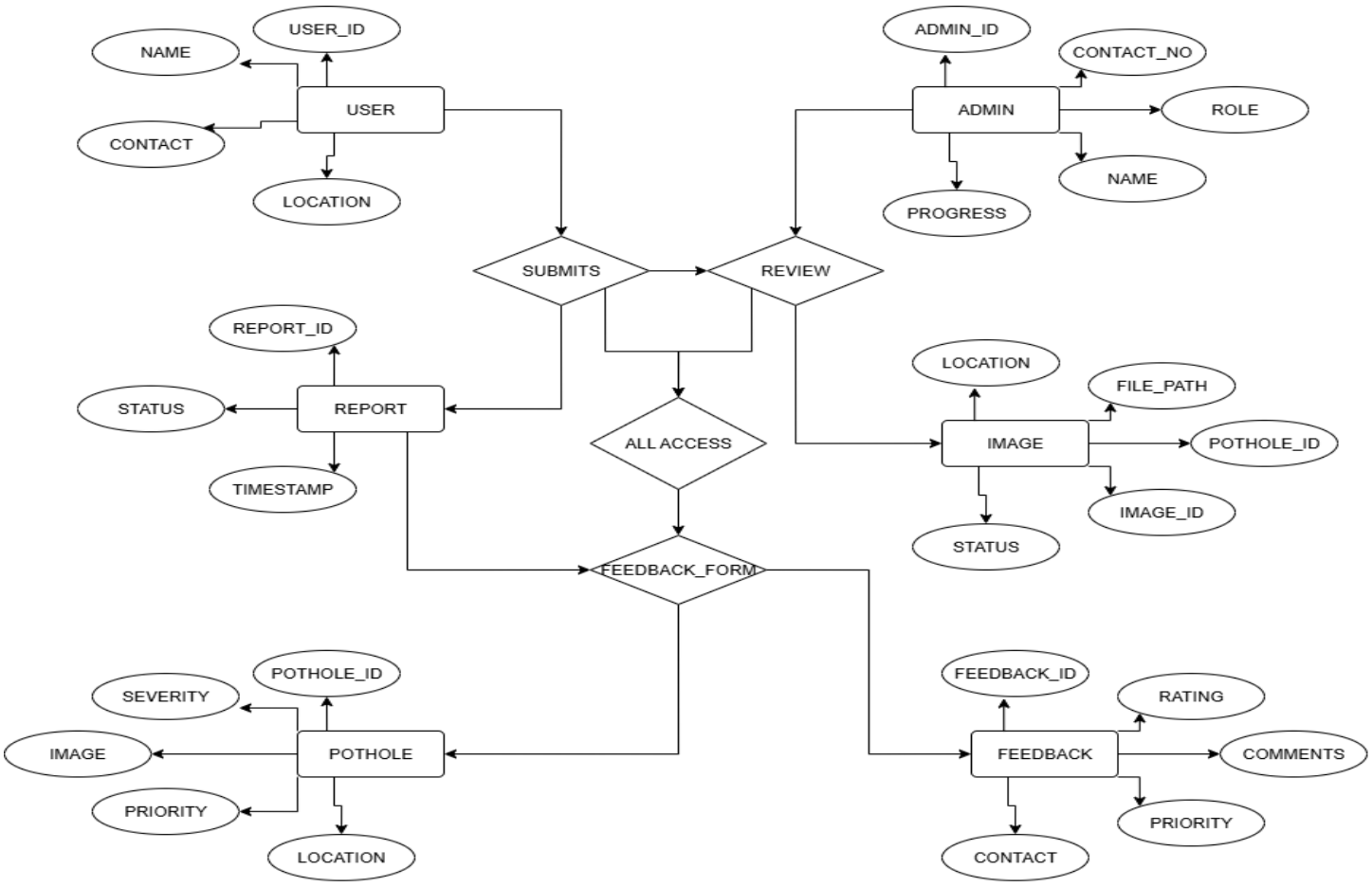
7.1 E-R DAIGRAM

Key Entities:

- User (user_id, name, contact)
- Report (report_id, timestamp, status)
- Pothole (pothole_id, severity, dimensions)
- Admin (admin_id, role, contact)
- Image (image_id, file_path, pothole_id)
- Feedback (feedback_id, rating, comments)

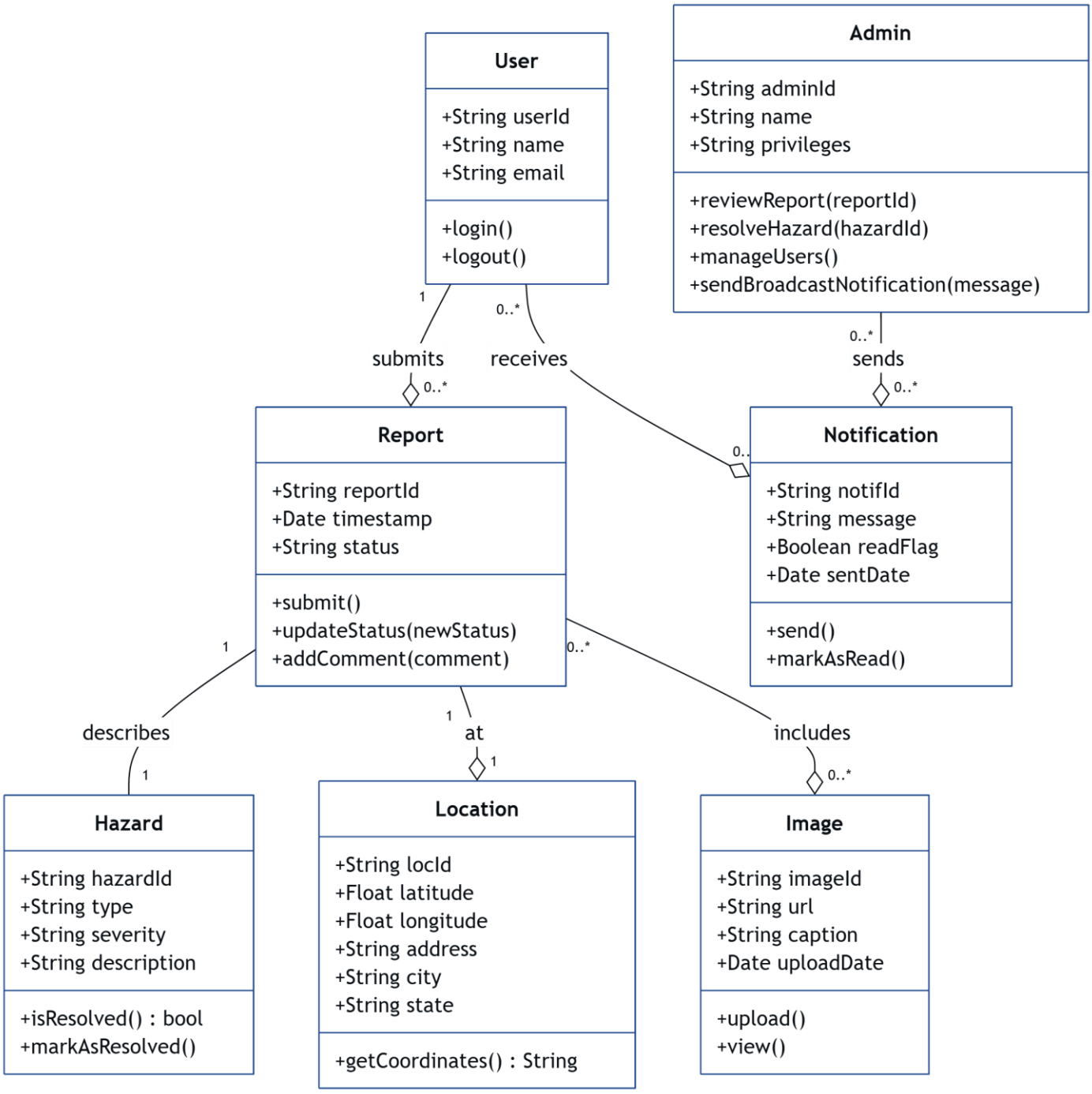
Relationships:

- User submits Report
- Report references Location and Pothole
- Admin reviews Report
- Pothole has Image
- User gives Feedback



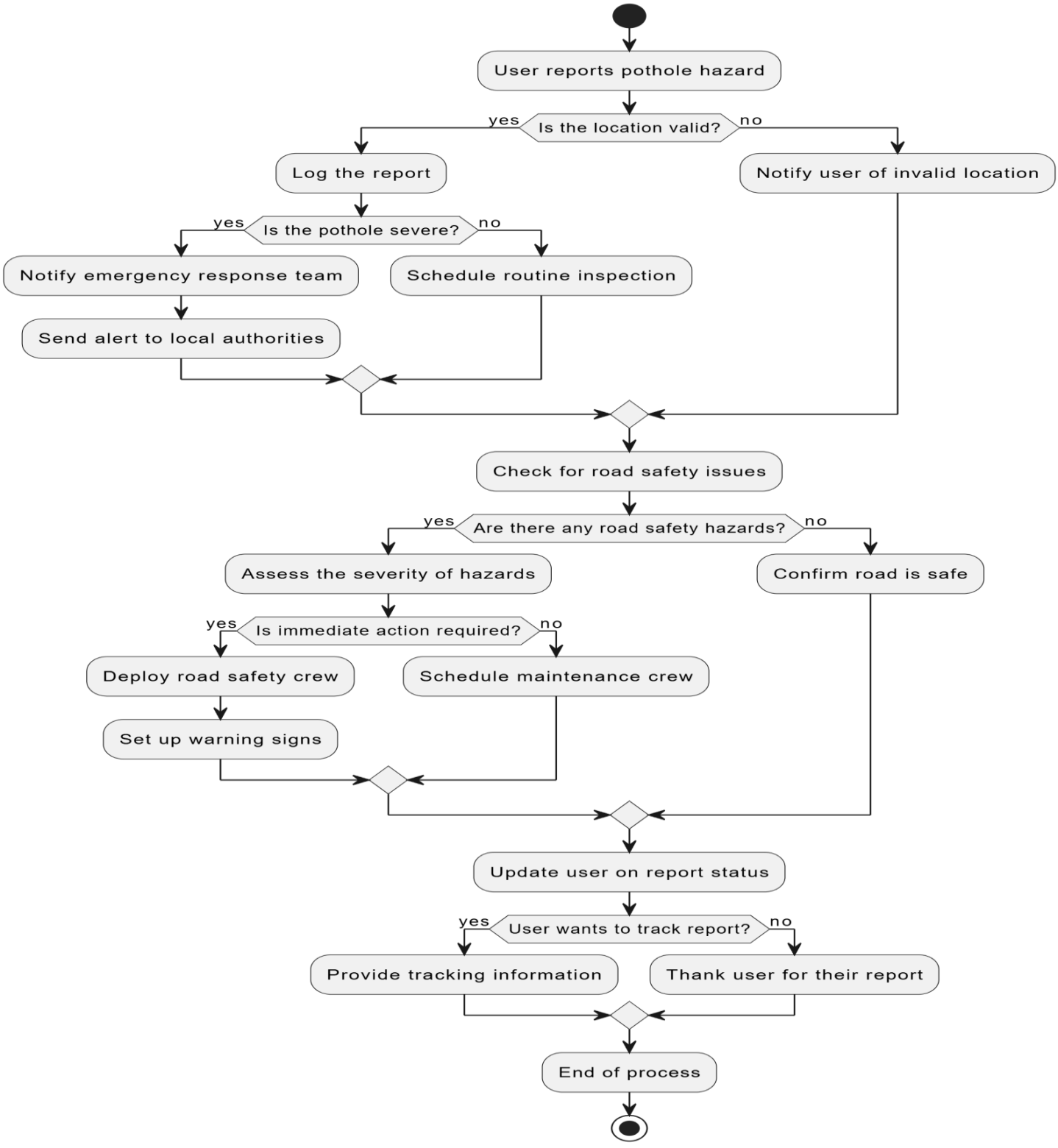
7.1 ER DAIGRAM

7.2 CLASS DIAGRAM



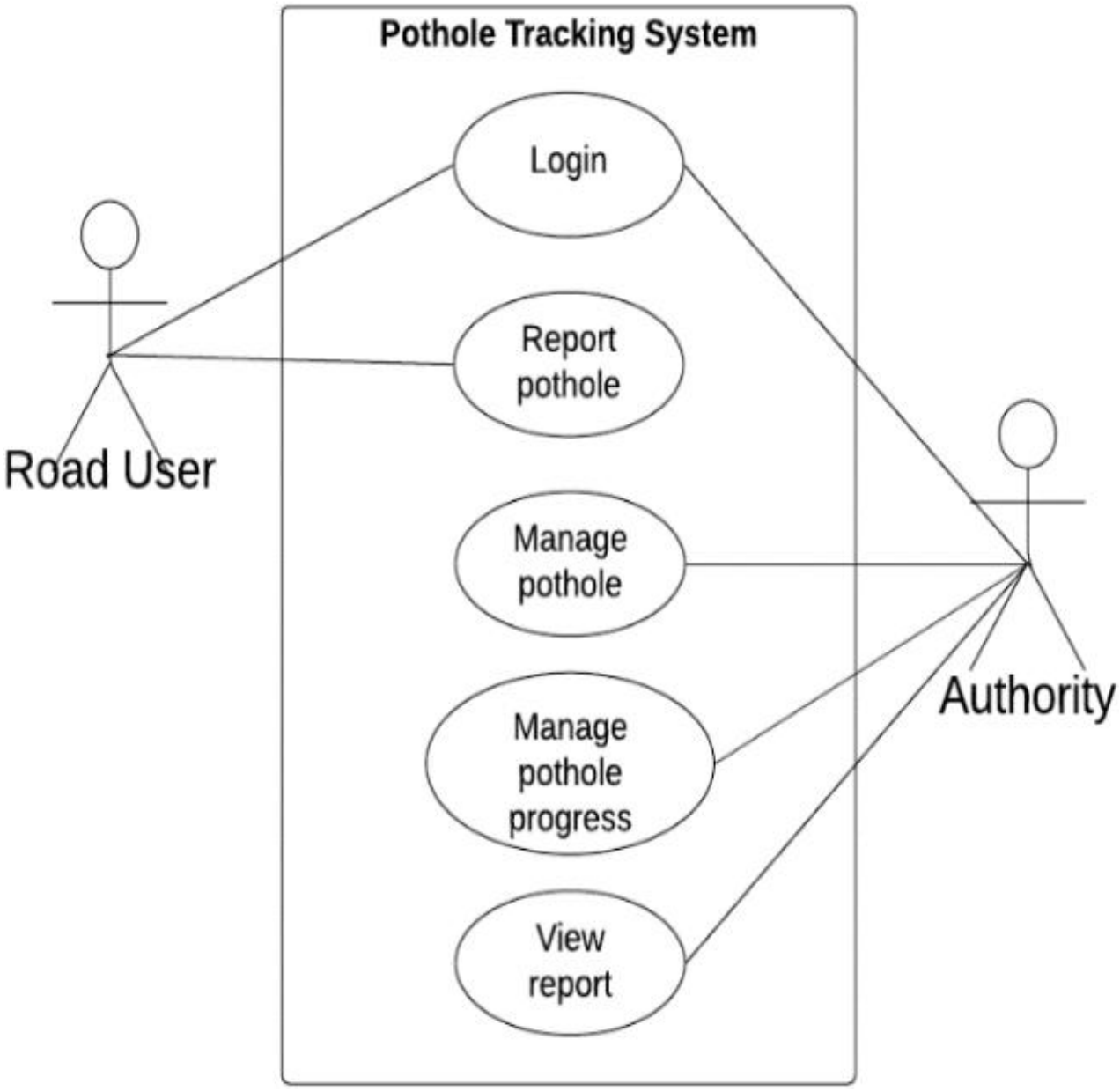
7.2 CLASS DAIGRAM

7.3 ACTIVITY DIAGRAM



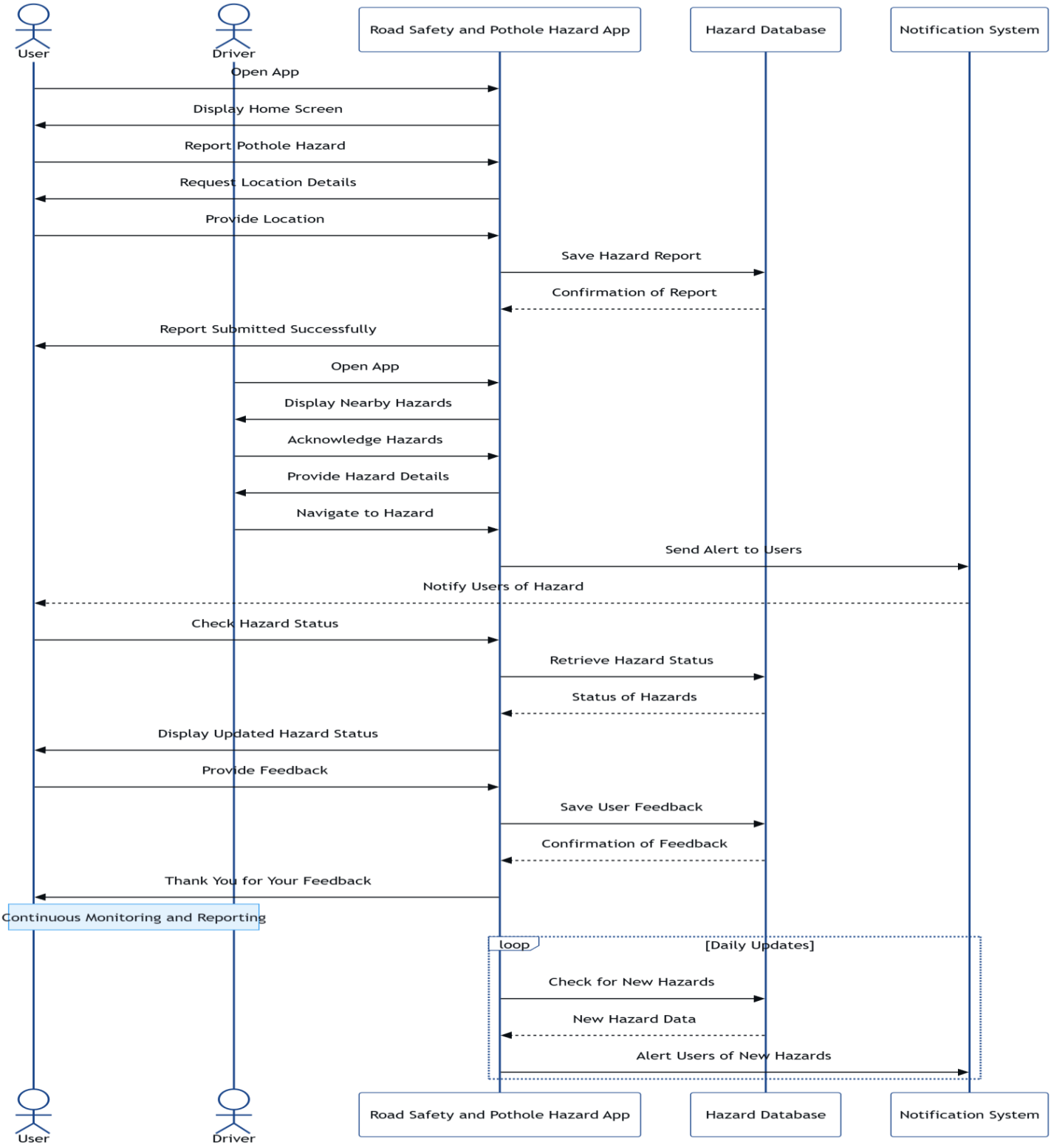
7.3 ACTIVITY DAIGRAM

7.4 USECASE DIAGRAM



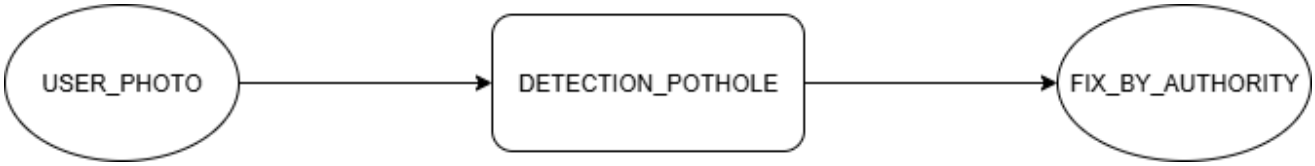
7.4 USE CASE DAIGRAM

7.5 SEQUNCIAL DIAGRAM

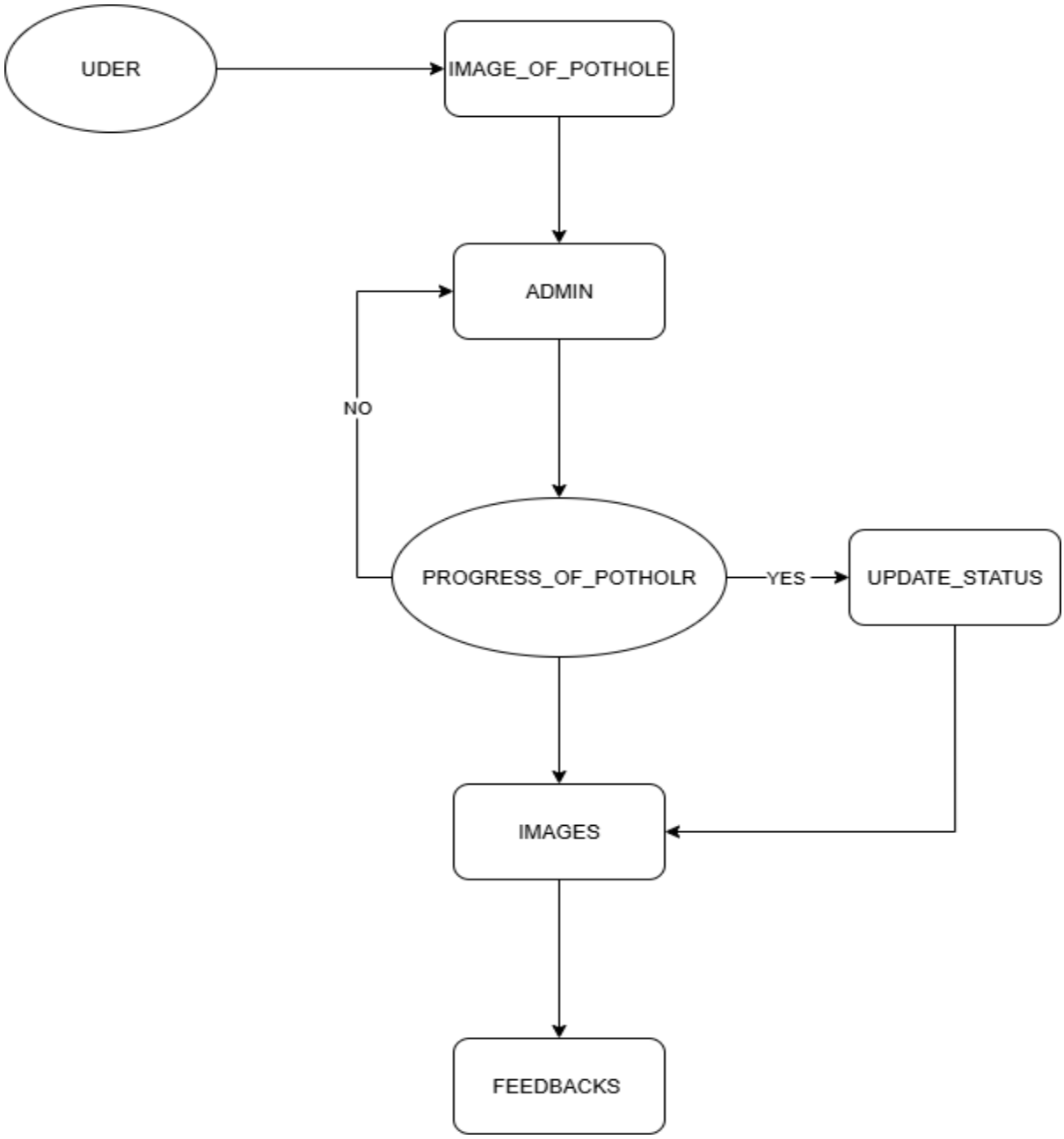


7.5 SEQUNCIAL DIAGRAM

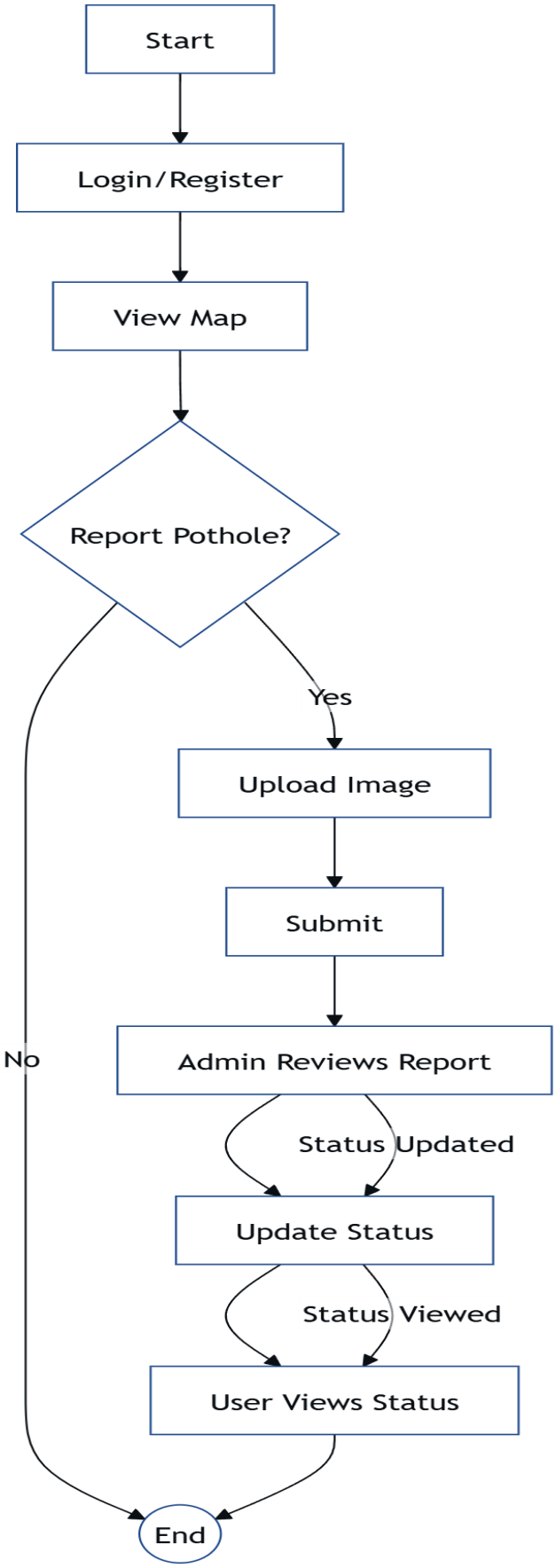
7.6 DATA FLOW DIAGRAM



7.6.1 DFD 0



7.6.2 DFD 2


Start
Login/Register
View Map
Report Pothole?
Yes
No
Upload Image
Submit
Admin Reviews Report
Status Updated
Update Status
Status Viewed
User Views Status
End

7.6.3 DFD 3

8. Implementation Standards

8.1 Environment

- Frontend: React.js
- Backend: Node.js and Express
- Database: MongoDB
- Deployment: Vercel (Frontend), Render or Railway (Backend)
- Authentication: JWT and Bcrypt.js

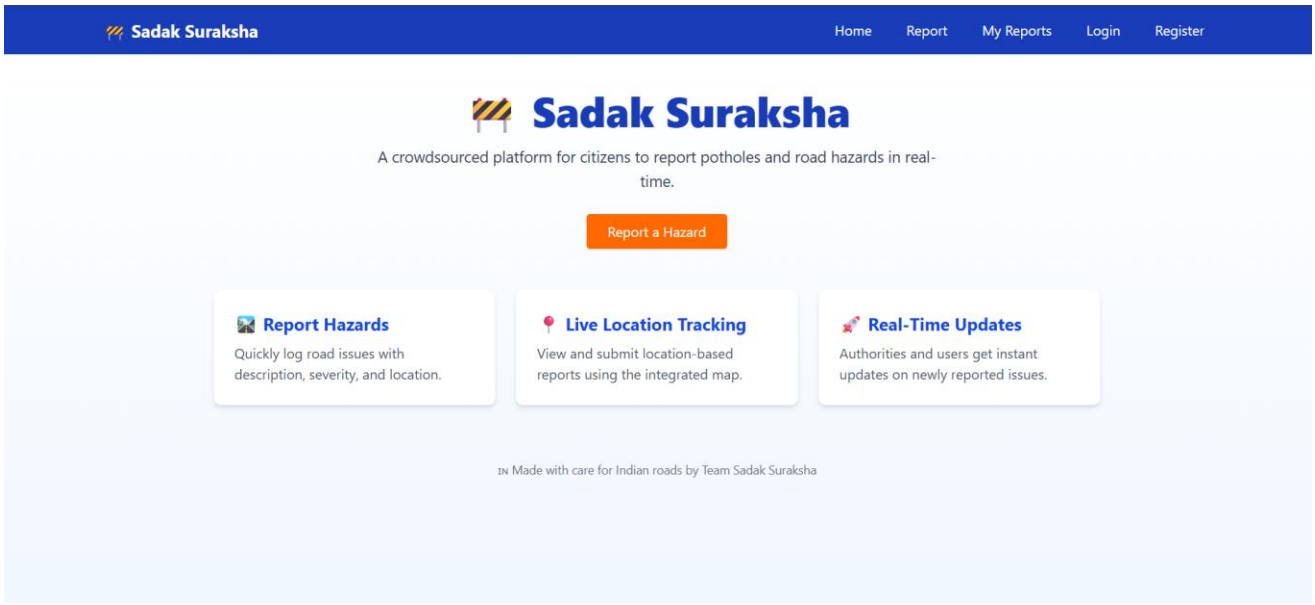
8.2 Security Features

- Role-based access control for Admin vs Users
- Encrypted credentials using Bcrypt
- API access protected via token-based middleware
- Input validation and sanitization for uploaded content
- Database backups and secure cloud-hosted services

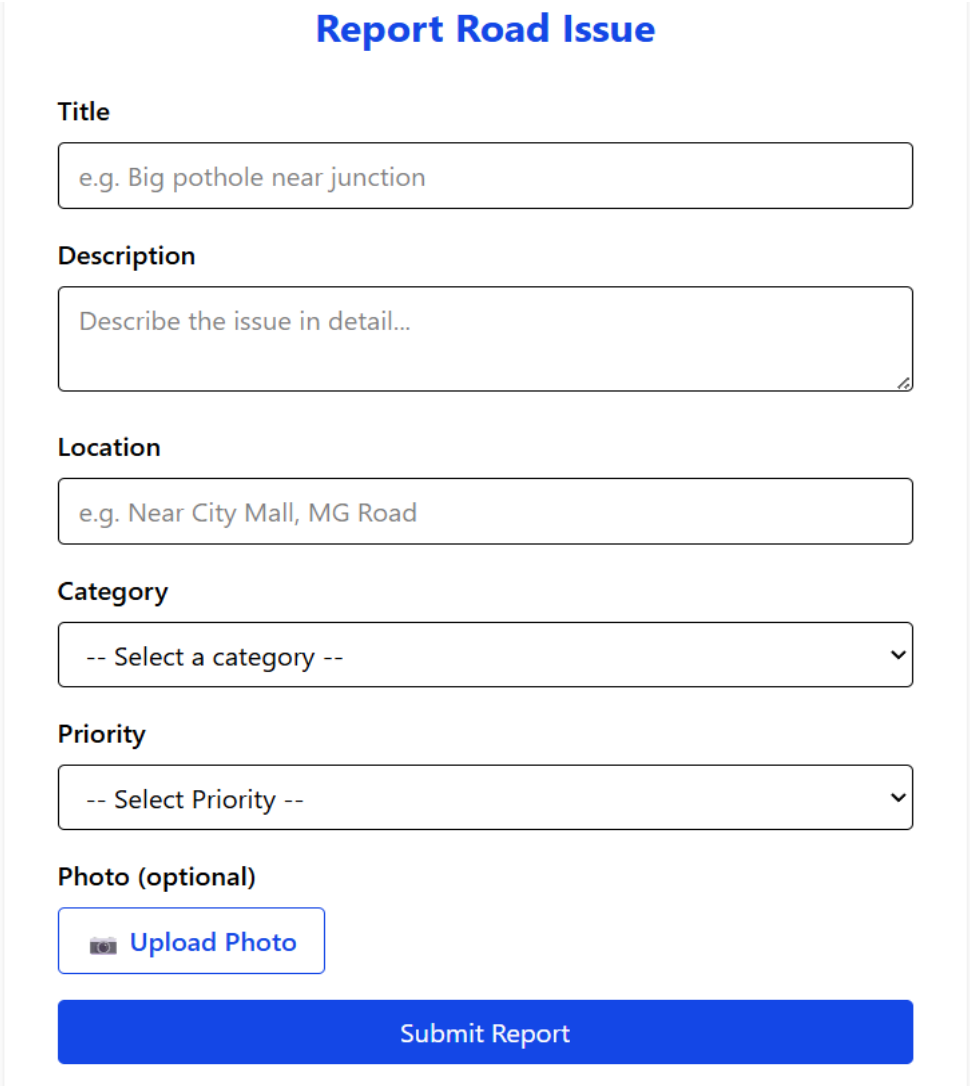
8.3 Coding Standards

- ES6 JavaScript conventions followed
- MVC separation of logic
- Git-based version control with structured commits
- Modular codebase for scalable deployment
- Clear in-code documentation using JSDoc

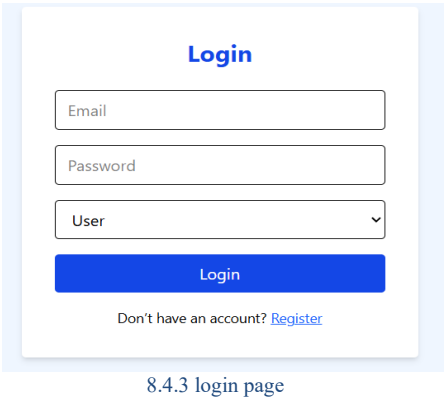
8.4 SCREENSHOTS



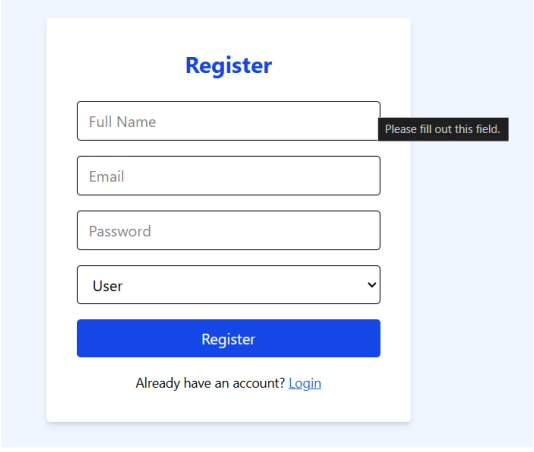
8.4 HOME PAGE



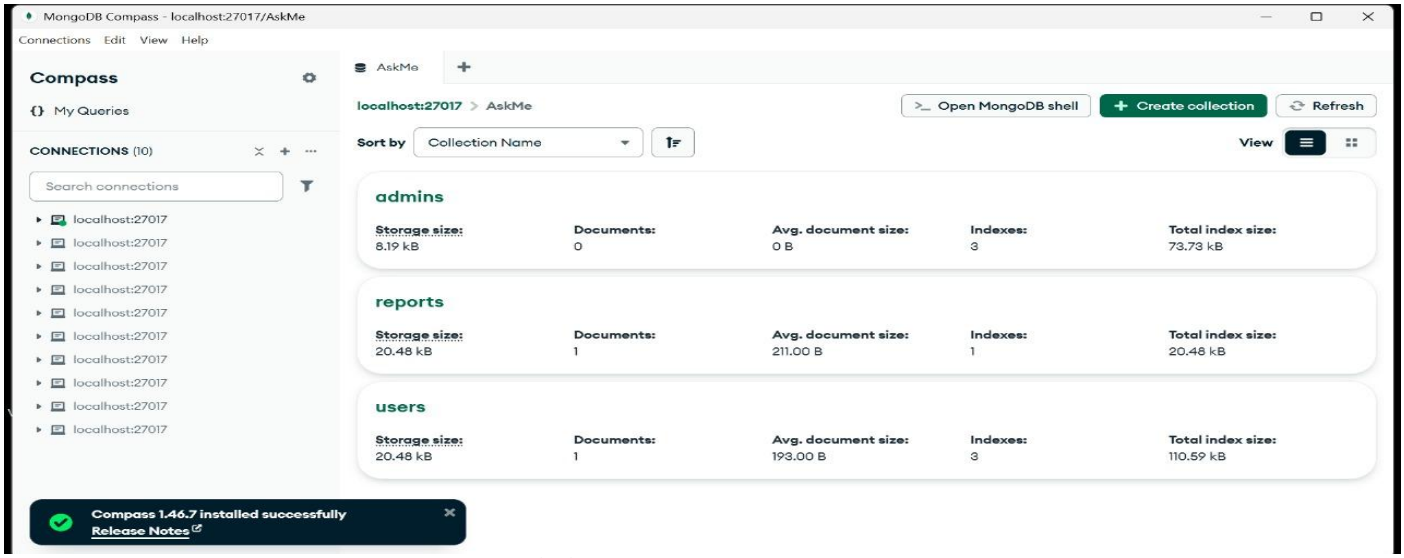
8.4.2 REPORT PAGE



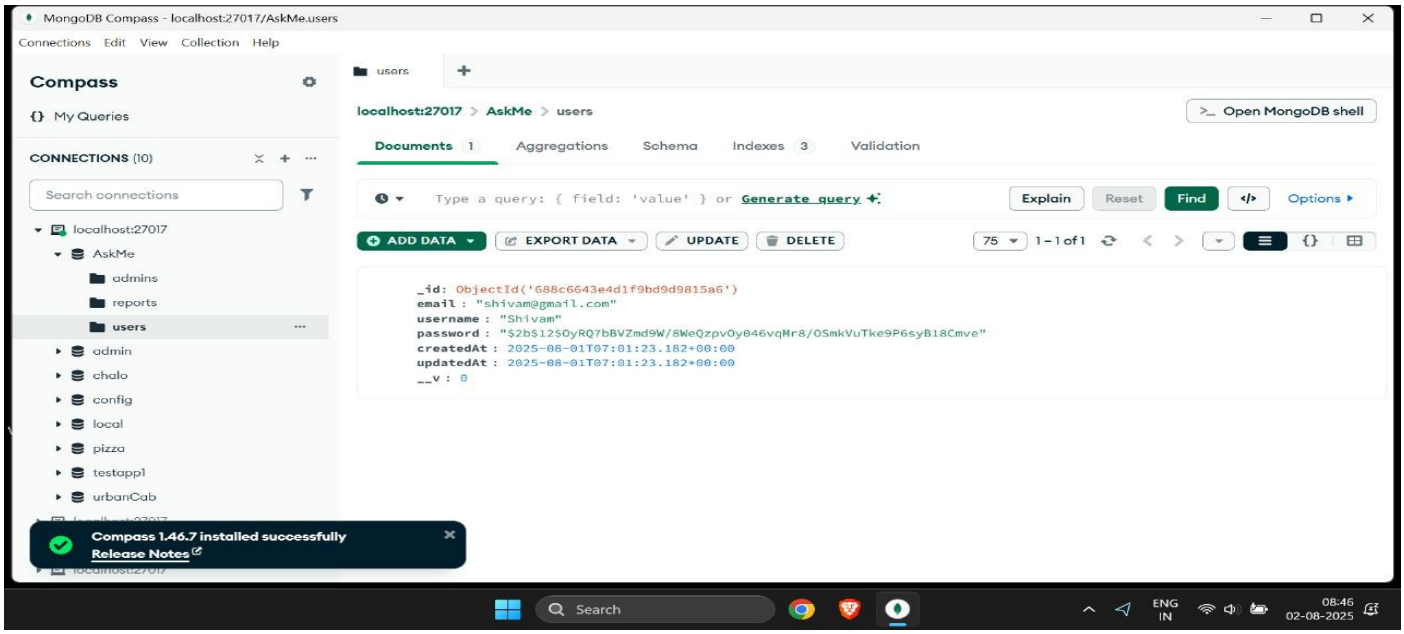
8.4.3 login page

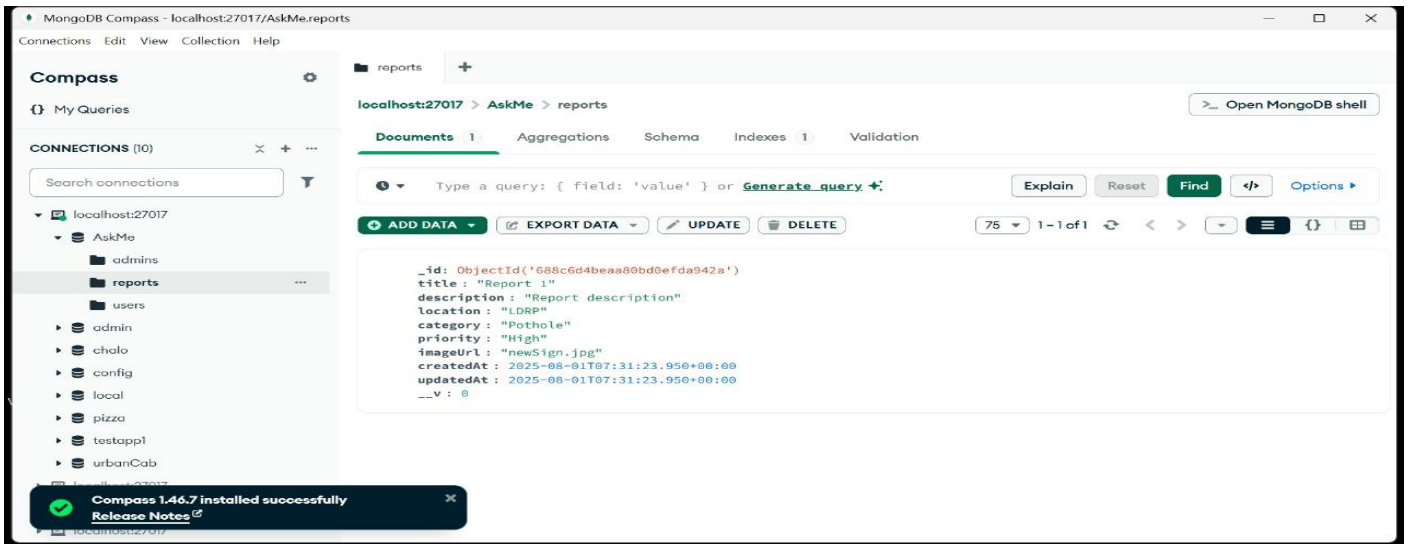


8.4.4 registration page



8.4.5 database





8.4.5 database

9. Conclusion

Sadak Suraksha successfully addresses the gap between citizens and municipal authorities in the context of road maintenance. By offering a real-time, location-based reporting tool with an intuitive interface, it empowers the public to actively contribute to road safety. On the administrative side, the system helps authorities prioritize and manage complaints efficiently through dashboards and analytics. The project demonstrates how modern web technologies can be leveraged to create civic impact and improve urban infrastructure. Future improvements may include predictive analytics, AI-based pothole detection, and a mobile application for even wider accessibility.

10. Bibliography

Suggested references:

- MongoDB Documentation
- React.js Official Docs
- Express.js and Node.js APIs
- Leaflet.js Map Library
- Government portals and urban planning resources
- Academic journals on civic technology and smart cities