Design and Implementation of Crime Alert Application for Web and Mobile Platforms

by

Md. Arif Howlader
ID: CSE2102023084

MD. Salman Rahman
ID: CSE2103024102

Suva Guha
ID: CSE2101022090

Jarin Tasnim Mahi
ID: CSE2203027052

Supervised by
Arifur Rahaman

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)
September 2025

APPROVAL

The project titled "**Design and Implementation of Crime Alert Application for Web and Mobile Platforms**" submitted by **Md. Arif Howlader** (CSE2102023084), **MD. Salman Rahman** (CSE2103024102), **Suva Guha (**CSE2101022090) and **Jarin Tasnim Mahi** (CSE2203027052) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Arifur Rahaman — **Supervisor**
Assistant Professor,
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 1**
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 2**
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 3**
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Arifur Rahaman,** Assistant Professor**,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Arifur Rahaman)
Supervisor

Signature

Md. Arif Howlader
ID: CSE2102023084

MD. Salman Rahman
ID: CSE2103024102

Suva Guha
ID: CSE2101022090

Jarin Tasnim Mahi
ID: CSE2203027052

ABSTRACT

The rapid growth of technology has transformed how societies communicate, access information, and ensure public safety. Yet, crime remains one of the biggest threats to individual security and community well-being. In many cases, victims or witnesses are unable to report incidents quickly due to a lack of accessible, reliable, and secure platforms. To address this challenge, we propose the development of a secure and scalable Crime Alert Application that works seamlessly across both web and mobile platforms. The application enables real-time reporting of crimes, sharing of location-based alerts, and quick communication between citizens, law enforcement agencies, and emergency responders. With features like user authentication, data encryption, and cloud scalability, the system ensures confidentiality, reliability, and uninterrupted service even during high traffic situations. Moreover, the platform integrates modern technologies such as push notifications, geolocation services, and role-based access control to create an effective crime management ecosystem. Beyond enabling instant reporting, the system empowers communities to stay informed, increases transparency in law enforcement, and contributes to building safer societies. This paper highlights the design, implementation, and evaluation of the application, along with its potential to reshape modern approaches to crime prevention and response.

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time.

We are auspicious that we had the kind association as well as supervision of **Arifur Rahaman**, Assistant Professor, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We would like to convey our special gratitude to **Bulbul Ahmed**, Head, Department of Computer Science and Engineering for his kind concern and precious suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
DB	Database
DBMS	Database Management System
GPS	Global Positioning System
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
JSON	JavaScript Object Notation
JWT	JSON Web Token
MERN	MongoDB, Express.js, React.js, Node.js
OTP	One-Time Password
RN	React Native
UI	User Interface
UI/UX	User Interface / User Experience
UX	User Experience

TABLE OF CONTENTS

Title				Page No.
DECLARATION ……..……..				iii
ABSTRACT ……...…….				iv
ACKNOWLEDGEMENT …….……				v
LIST OF ABBREVIATION …….…….				vi
CHAPTER 1				1 – 2
INTRODUCTION TO CRIME ALERT APPLICATION				
	1.1	Introduction ……		1
	1.2	Objectives ……...……		1
	1.3	Background ……..……		1
	1.4	Purpose of Crime Alert Application ……..……		2
	1.5	Aim and Scope …….……..….		2
CHAPTER 2				3 – 6
BACKGROUND AND RELATED WORK				
	2.1	Introduction …….		3
	2.2	Language and Tools ……		3 – 6
		2.2.1	Java Script ……...	3 – 4
		2.2.2	React ……...	4
		2.2.3	Tailwind CSS ……..	4
		2.2.4	Node.js ……	4 – 5
		2.2.5	TypeScript ……...	5
		2.2.6	Express.js ……	5
		2.2.7	MongoDB ……...	5
		2.2.8	Visual Studio Code……..	6
		2.2.9	Postman……	6

CHAPTER 3				7 – 8
REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATIONS				
	3.1	Requirements ……...……		7 – 8
		3.1.1	Data Requirements …….…….	7
		3.1.2	Functional Requirements …….	7
		3.1.3	Performance Requirements ……...…..	7 – 8
		3.1.4	Testing and Maintainability Requirements …….	8
CHAPTER 4				13 – 15
SYSTEM DESIGN				
	4.1	Introduction …….…….		9
	4.2	System Architechture Design ……...		9 – 10
	4.3	Database Design ……...		10 – 11
	4.4	Data Flow Design …….		11
	4.5	ER Diagram ……..		12
	4.6	User Interface Design ……...		13
	4.7	Database ……		13 – 15
CHAPTER 5				16 – 18
SOFTWARE OUTCOME AND IMPLEMENTATION				
	5.1	Home Page …….……...		16
	5.2	Sign Up …….		17
	5.3	Sign in …….……..		17 – 18
	5.4	Logout ……...		18
CHAPTER 6				19
CONCLUSION AND FUTURE WORKS				
	6.1	Conclusion ……..…..		19
	6.2	Future Works …….…...		19
REFERENCES …….……..				20

LIST OF TABLES

Table No.	Title	Page No.
Table 1.1	Purpose of Crime Alert Application	2
Table 3.1	Data Requirements	7
Table 4.1	System Architechture Design	10
Table 4.2	Database	15

LIST OF FIGURES

Figure No.	Title	Page No.
Fig.2.1	Front-end and Backened Language	4
Fig.3.1	Website Testing Diagram	8
Fig.4.1	System Architechture Design	10
Fig.4.2	Database Design	11
Fig.4.3	Data Flow Design	11
Fig.4.4	ER Diagram	12
Fig.4.5	User Reports Interface	13
Fig.4.6	User Data in MongoDB	14
Fig.4.7	LocalUser Data in MongoDB	14
Fig.4.8	Report Data in MongoDB	14
Fig.4.9	Central Police Data in MongoDB	15
Fig.4.10	Local-police Data in MongoDB	15
Fig.5.1	Home Page	16
Fig.5.2	Sign Up	17
Fig.5.3	Sign In	18

CHAPTER 1

INTRODUCTION TO CRIME ALERT APPLICATION

1.1 Introduction

The project is a digital platform designed to improve public safety by creating an effective communication link between citizens and law enforcement agencies. Traditional crime reporting methods are often slow and inefficient which can delay responses and increase risks to communities. This application integrates mobile and web technologies to provide real-time crime reporting, evidence submission and instant safety alerts. The mobile app allows users to report incidents with location details, photos and videos while the web platform gives authorities tools to verify reports, monitor crime-prone areas and coordinate quick responses. The system is developed with a strong focus on scalability, security and ease of use while applying technologies such as cloud computing, GPS tracking and push notifications. Beyond reporting incidents, the application builds a proactive safety network where communities and authorities work together to prevent and manage criminal activities and to create safer and more connected societies.

1.2 Objectives

Generally, most of the project objectives is to provide a reliable and user-friendly platform that ensures faster communication and improved safety within communities. This project is designed to achieve the following specific goals:

- Enable users to submit evidence such as photos, videos, and location details to support accurate reporting.
- Provide law enforcement agencies with tools to verify reports, monitor hotspots, and respond effectively.
- Deliver instant safety alerts and notifications to keep citizens informed of nearby threats.
- Ensure data security, privacy, and scalability to support a large number of users and reports.
- Foster community engagement by creating a collaborative safety network between the public and authorities.

1.3 Background

Crime is a major challenge in today's society as traditional reporting methods are often slow and inconvenient. Many incidents remain unreported due to the lack of fast and accessible communication channels. With the rapid growth of smartphone usage and internet availability, there is a need for a modern system that enables instant crime reporting and response. The project addresses this gap by providing a secure platform for real-time reporting, evidence submission, and safety alerts. It empowers citizens to participate in community safety while supporting law enforcement with effective tools for monitoring and quick response. This system ultimately promotes safer and more connected communities.

1.4 Purpose of Crime Alert Application

The purpose of the project is to create a reliable and accessible platform that enables citizens to report crimes instantly, receive timely alerts, and collaborate with law enforcement to enhance public safety. The system aims to overcome the limitations of traditional reporting methods by providing fast, secure, and user-friendly tools for sharing information and evidence. By integrating real-time communication, location tracking, and data management features, the application ensures that incidents are addressed promptly and efficiently. Ultimately, its purpose is to strengthen the relationship between communities and authorities, reduce crime rates, and contribute to the development of a safer and more connected society.

Aspect	Description
Main Purpose	To provide a reliable and accessible platform for instant crime reporting.
User Empowerment	Enables citizens to report incidents with photos, videos, and location details.
Law Enforcement Support	Helps police/authorities verify reports, monitor hotspots, and respond faster.
Real-Time Alerts	Provides timely safety alerts and notifications to nearby users.
Data Security	Ensures confidentiality with encryption, secure storage, and scalable design.
Community Engagement	Builds collaboration between citizens and authorities to reduce crime.
Overall Goal	To create safer, more connected societies through proactive crime management.

Table 1.1: Purpose of Crime Alert Application

1.5 Aim and Scope

The aim of the project is to enhance public safety by providing a fast, reliable, and user-friendly platform for crime reporting and real-time alerts. The system is designed to bridge the communication gap between citizens and law enforcement agencies, ensuring timely intervention and promoting safer communities. The scope of the project includes the following key aspects:

- Real-Time Crime Reporting: Real-time reporting of crimes through web and mobile platforms.
- Evidence Submission: Submission of evidence including photos, videos, and location details.
- Safety Alerts: Instant safety alerts and notifications for users in affected areas.
- Law Enforcement Tools: Tools for law enforcement to verify reports, monitor crime-prone zones, and coordinate rapid responses.
- Data Security and Scalability: Secure storage and management of sensitive data with scalability for large user bases.
- Community Engagement: Fostering community participation in crime prevention and safety initiatives.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Introduction

The increasing rate of crime in modern society and the limitations of traditional reporting methods have created a pressing need for faster, smarter, and more accessible solutions. Over the years, various mobile and web-based crime alert systems have been developed to enable real-time reporting, evidence sharing, and instant notifications. While these systems provide useful features, many face challenges such as delayed response, limited scalability, or weak data security. This chapter explores the evolution of crime reporting technologies, reviews existing platforms, and identifies gaps that the project aims to address. By analyzing prior work, this project builds upon effective strategies while introducing enhanced functionality, stronger security, and improved community engagement to create a comprehensive and reliable crime alert solution.

2.2 Language and Tools

In our proposed project we have used TypeScript, MongoDB, Node.js, Express.js, Tailwind CSS, React, JavaScript as programming language. As software we have used Visual Studio Code, Postman and google chrome. The used languages and applications are described bellow-



Fig.2.1: Front-end and Backend Lanuages

2.2.1 Java Script

JavaScript plays a pivotal role in making the front-end interactive, responsive, and user-friendly. It bridges the gap between the user interface and back-end functionalities, ensuring smooth data handling and real-time updates [5]. Key features powered by JavaScript include:

- Dynamic Content Rendering: Updates crime alerts, notifications, and user reports without reloading the page.
- Form Validation: Ensures accurate user input while reporting incidents, reducing errors before submission.
- Interactive Maps & Geolocation: Displays real-time crime locations and user position using map APIs.
- Event Handling: Handles clicks, form submissions, and navigation efficiently for seamless UX.
- Real-time Alerts & Notifications: Shows immediate alerts using asynchronous requests (AJAX/Fetch API).
- Integration with APIs: Fetches data from Node.js/MongoDB back-end to display crime statistics and reports dynamically [4].

2.2.2 React

React is used to build a fast, modular, and responsive user interface. It allows the creation of reusable components for features like crime reports, alert notifications, and user dashboards, ensuring consistent design and smooth navigation. React's state management enables real-time updates of alerts and user data, while React Router facilitates seamless page transitions. Its integration with APIs and JavaScript ensures dynamic content rendering, interactive maps, and responsive user interactions, making the application highly efficient and user-friendly.Its component-based architecture allows for reusable elements like alert cards, crime reports, and user dashboards, making the development faster and the code easier to maintain. React's state management and virtual DOM ensure that updates happen instantly when new crime data is added or alerts are sent, making the application highly efficient and responsive for both users and police officers [10].

2.2.3 Tailwind CSS

Tailwind CSS is used to design a modern, responsive, and visually appealing user interface. Its utility-first approach allows rapid styling of React components like buttons, forms, alerts, and dashboards without writing custom CSS. Tailwind ensures consistent spacing, colors, and typography across the app, making the UI clean and user-friendly, while fully supporting responsive design for mobile and desktop devices.Its predefined classes make it easy to adjust layouts, colors, spacing, and typography, ensuring that the application is visually appealing on all devices, from desktops to smartphones. By combining Tailwind CSS with React, the UI becomes highly modular and maintainable, allowing rapid updates and consistent design across the platform, which is essential for a professional and user-friendly crime alert system.

2.2.4 Node.js

Node.js serves as the back-end runtime environment, enabling fast and efficient handling of server-side operations. It manages API requests, processes crime reports submitted by users, and

communicates seamlessly with MongoDB to store and retrieve data. Node.js also supports real-time notifications and alert updates, ensuring that the application delivers timely information to users while maintaining high performance and scalability.It enables real-time processing of crime reports and alerts, ensuring that notifications reach users instantly. Node.js's event-driven, non-blocking architecture allows the application to handle multiple simultaneous connections without slowing down, which is crucial for an application where timely alerts and updates are essential for safety and security [8].

2.2.5 TypeScript

TypeScript enhances the front-end and back-end code by adding strong typing, which reduces errors and improves code reliability. It ensures better structure and maintainability of React components, Node.js APIs, and data handling with MongoDB. By catching errors during development, TypeScript makes the application more robust, scalable, and easier to debug, resulting in a smoother and safer user experience. It helps prevent common programming errors, ensures strong type safety, and makes the application easier to scale and debug. For the Crime Alert Application, using TypeScript ensures that data consistency is maintained across different modules, such as alerts, crime records, and user profiles, which is critical for a system where accuracy and reliability are essential.

2.2.6 Express.js

Express.js acts as the back-end web framework that simplifies server-side development. It handles routing, API endpoints, and middleware integration, enabling smooth communication between the front-end (React) and the database (MongoDB). Express.js ensures efficient request handling, manages user submissions and crime report data, and supports real-time notifications, making the application fast, scalable, and reliable. It provides a structured and organized backend, making it easy to handle data flow between the frontend (React) and the database (MongoDB). With Express.js, the application ensures secure and reliable communication, which is vital for a crime alert system where accuracy and responsiveness are critical.

2.2.7 MongoDB

MongoDB serves as the primary database, storing and managing all crime reports, user data, and alert information. Its flexible, document-oriented structure allows easy handling of dynamic data and seamless integration with Node.js and Express.js. MongoDB supports real-time data retrieval and updates, ensuring that users receive the latest crime alerts quickly while maintaining high performance and scalability for the application. Its schema-less design allows the app to adapt quickly to changes, such as adding new types of crime data or alert features. MongoDB supports high-performance queries and real-time data retrieval, which ensures that users and officers can access the latest crime information instantly, making the system reliable and efficient for emergency responses [9].

2.2.8 Visual Studio Code

Visual Studio Code (VS Code) serves as the primary development environment, providing a versatile and efficient platform for coding in React, Node.js, TypeScript, and other technologies. Its rich features like intelligent code completion, debugging tools, extensions for Tailwind CSS and MongoDB, and Git integration streamline development, enhance productivity, and make code management easier, ensuring a smooth and organized workflow.

2.2.9 Postman

Postman is used to test and validate APIs, ensuring smooth communication between the front-end and back-end. It allows developers to send requests, check responses, and debug endpoints in Node.js/Express.js efficiently. By simulating real user interactions with the application, Postman helps verify data flow, troubleshoot errors, and ensure that crime reports, alerts, and notifications work accurately before deployment.

CHAPTER 3

REQUIREMENTS ANALYSIS AND SYSTEM SPECIFICATIONS

3.1 Requirements

We are developing a project that requires real-time location tracking, secure authentication, and fast communication. GPS and network-based services will be used to detect user positions, while push notifications will deliver alerts instantly to nearby users and authorities. The system must support sending crime alerts with text, images, or location data, ensure data security, and provide a smooth experience across both web and mobile platforms.The software Requirements features are as follows-

3.1.1 Data Requiremnets

The project requires structured and reliable data to function effectively. This includes user information, crime reports with details like location, type, and time, as well as alert notifications and historical crime statistics. Data must be stored securely in MongoDB, validated through front-end forms, and processed via APIs to ensure accuracy, real-time updates, and efficient retrieval for analysis and reporting [3].

Data Type	Description	Usage
User Info	Name, email, password, location	Login, profile, communication
Crime Reports	Type, details, media, location, time	Reporting & verification
Alerts	Message, receiver, timestamp	Notify users & authorities
Historical Data	Past reports, statistics	Analysis, crime hotspots
Admin Data	Admin ID, email, privileges	Manage system & reports
Geolocation Data	GPS coordinates	Real-time tracking & mapping

Table 3.1: Data Requirements

3.1.2 Functional Requirements

The project must allow users to register and log in, submit crime reports with details like location and type, and receive real-time alerts. It should display crime statistics, interactive maps, and notifications, while ensuring secure data storage and retrieval via MongoDB [9]. The system must support smooth interaction between the front-end (React, Tailwind CSS, JavaScript) and back-end (Node.js, Express.js, TypeScript), handle API requests efficiently, and provide a responsive, user-friendly interface across devices.

3.1.3 Performance Requirements

It must operate efficiently and respond quickly to user actions. It should load pages and display crime alerts in real-time, handle multiple simultaneous user requests without lag, and process data from MongoDB swiftly. The system must ensure smooth interactions across devices and

browsers, maintain high uptime, and deliver notifications and updates promptly to provide a reliable and responsive user experience.

3.1.4 Testing and Maintainability Requirements

It must operate efficiently and respond quickly to user actions. It should load pages and display crime alerts in real-time, handle multiple simultaneous user requests without lag, and process data from MongoDB swiftly. The system must ensure smooth interactions across devices and browsers, maintain high uptime, and deliver notifications and updates promptly to provide a reliable and responsive user experience [6].



Fig.3.1: Website Testing Diagram

CHAPTER 4

SYSTEM DESIGN

4.1 Introduction

System design is the blueprint of our Crime Alert Application that transforms requirements into a structured solution. It defines how different components such as user interfaces, databases, servers, and communication modules will work together to ensure real-time crime reporting and alert delivery. A clear design not only improves system efficiency and security but also provides scalability, flexibility, and reliability, making the application capable of handling large numbers of users while maintaining smooth performance across web and mobile platforms.

4.2 System Architecture Design

The proposed system follows a client-server architecture with a centralized database. The design integrates both web and mobile clients, which interact with the backend via APIs.

The system is divided into several modules:

1. User Management Module

    - User registration and login
    - Authentication using JWT
    - Profile management

2. Crime Reporting Module

    - Users can submit crime details (type, description, location, time)
    - Option for anonymous reporting.

3. Crime Alert Module

    - Real-time push notifications to nearby users.
    - Alerts displayed on dashboard and mobile app.

4. Crime Map & Visualization Module

    - Interactive map integration using location services.
    - Display of reported incidents with filters (date, type, location).

5. Admin/Authority Module

    - Admin panel for police/authorities to verify reports.
    - Data analytics and reports on crime trends.

6. Data Management Module

- Secure storage of reports in MongoDB.
- APIs for retrieving, updating, and deleting records [1].

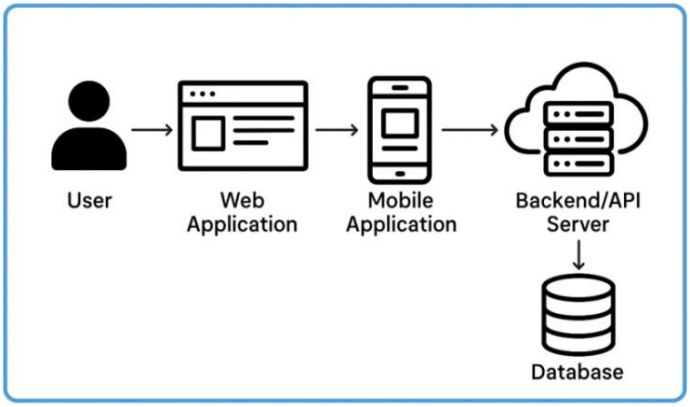


Fig.4.1: System Architecture Design

Module	Key Functions
User Management	Registration, login, authentication (JWT)
Crime Reporting	Submit type, description, media, location, time
Crime Alerts	Real-time notifications to nearby users
Crime Map & Visualization	Interactive maps, filter by date/type/location
Admin/Authority	Verify reports, monitor trends, manage users
Data Management	Store and retrieve data securely in MongoDB

Table 4.1: System Architecture Design

4.3 Database Design

The database is designed using MongoDB (NoSQL) for flexibility.

Collections:

- Users: { user_id, name, email, password_hash, role, location }
- Reports: { report_id, user_id, crime_type, description, location, timestamp, status }
- Alerts: { alert_id, report_id, users_notified, timestamp }
- Admin: { admin_id, email, password_hash, privileges } [2]

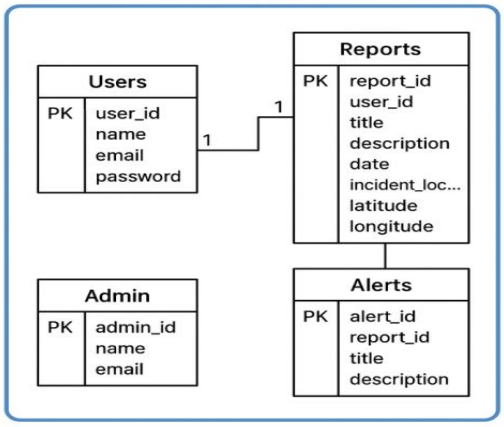


Fig.4.2: Database Design

4.4 Data Flow Design

In this project, the **Level-0 DFD** provides a high-level overview, showing the interaction between the main actors (Users, Admins, and Police) and the system. For example:

- Users provide crime reports as inputs.
- The System processes these reports and stores them in the Database.
- Admins and Police access the data to verify, update, and take necessary actions.
- The system also generates notifications and alerts back to the users.

The **Level-1 DFD** breaks these processes into more detail, such as:

- User Registration/Login
- Report Crime Submission
- Crime Verification by Police
- Alert/Notification Generation
- Admin Management of Users and Reports

By using DFDs, the system's data flow becomes clear, ensuring that no important data processing or communication path is overlooked.

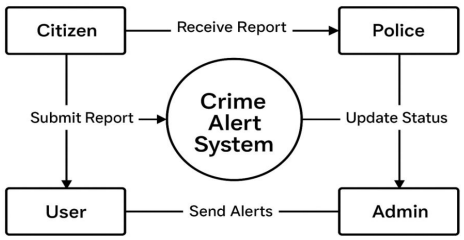


Fig.4.3: Data Flow Design

4.5 ER Diagram

In this project, the key entities include:

- User: Represents citizens who register in the system. Attributes include User_ID, Name, Email, Phone, and Location.
- Admin: Represents system administrators who manage reports and user accounts. Attributes include Admin_ID, Name, and Role.
- Police/Authority: Represents law enforcement officers who verify and act on reported crimes. Attributes include Police_ID, Name, and Station.
- Crime Report: Represents reported incidents. Attributes include Report_ID, Crime_Type, Description, Evidence (photo/video), Location, Date, and Status.
- Notification: Stores alert messages sent to users. Attributes include Notification_ID, Message, Date, and Receiver_ID.

The relationships can be summarized as:

- A User can submit multiple Crime Reports.
- A Police/Authority can verify multiple Crime Reports.
- An Admin manages both Users and Crime Reports.
- A Notification is sent to multiple Users based on crime location.

Thus, the ER diagram provides a blueprint for designing the database schema, ensuring that data is properly organized, linked, and retrievable for efficient system operations.

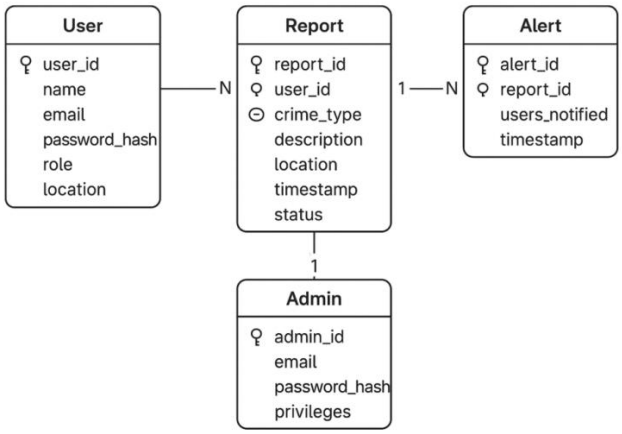


Fig.4.4: ER Diagram

4.6 User Interface Design

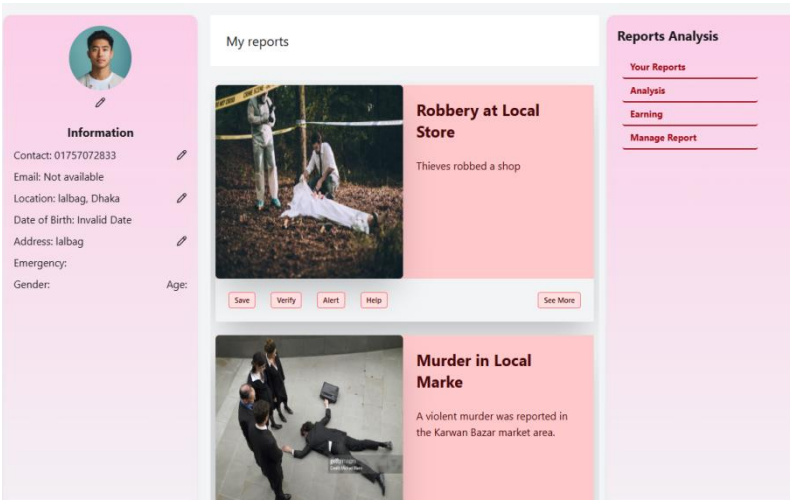


Fig.4.5: User Reports Interface

The Fig.4.5 above displays the "User Reports" section of the Crime Alert system, which serves as a personalized dashboard for individual users. On the left-hand side, the interface presents the user's profile information, including their photo, contact number, location, date of birth, address, and other personal details. This information ensures that reports submitted by the user can be properly authenticated and associated with their identity. The central portion of the interface lists the user's submitted crime reports, each accompanied by a title, description, and an illustrative image to provide better context. Users are given interactive options such as saving, verifying, sending alerts, or requesting help on each report, thereby enhancing community-driven validation and participation. The right-hand panel provides a "Reports Analysis" section, which includes links for viewing reports, analyzing trends, managing reports, and tracking system engagement [7].

4.7 Database

A database is a structured collection of information that allows easy access, management, and updating. MongoDB, a NoSQL database, stores data in flexible JSON-like documents instead of traditional tables, which makes it suitable for handling diverse and dynamic data. In our project, MongoDB is used to store users and admins information such as profile images, contact numbers, emails, passwords, and locations, along with collections for police stations (location), doctors (contact numbers and location), and fire service information. We also maintain a collection for user blogs and experiences. MongoDB provides efficient storage, quick queries, indexing, and scalability, while its flexible schema design makes it easier to update and manage different types of records without rigid predefined structures.

```
_id: ObjectId('68b74e57e418c2a5846ledf1')
name : "John arif"
email : "john@example.com"
contactNumber : "01712345678"
password : "$2b$10$6kVofWYuzHJGCgrnoYONGu8LmBZLv4kJgJt1fUcrjjIQ3tWP6XuAm"
role : "localUser"
isBlocked : false
isDeleted : false
createdAt : 2025-09-02T20:06:47.802+00:00
updatedAt : 2025-09-03T02:48:17.733+00:00
__v : 0
```


Fig.4.6: User Data in MongoDB


```
QUERY RESULTS: 1-13 OF 13

_id: ObjectId('688bc13f893c8066296712ea')
userId : "user12345"
role : "admin"
firstName : "Arif"
middleName : "hossen"
lastName : "Sagor"
profileImage : "https://example.com/images/profile.jpg"
gender : "male"
age : 30
dateOfBirth : 1995-05-15T00:00:00.000+00:00
contactNumber : "+8801712345678"
emergencyContact : "+8801987654321"
address : "123/A, Main Street, Dhaka"
currentLocation : "Dhaka, Bangladesh"
connectedMembers : Array (2)
status : "active"
```


Fig.4.7: LocalUser Data in MongoDB


```
_id: ObjectId('68bf15f3fceb915cfb7d59fc')
reportTitle : "Thif in home"
reportDescription : "A thif try to enter home breaking lock"
reportImage : "https://i.ibb.co/ZpV4XhPW/istockphoto-2217049822-1024x1024.jpg"
reporterId : ObjectId('68bf0cf5fceb915cfb7d59cf')
reportLocation : "lalbag"
reportType : "theft"
informPerson : true
informLocalPolice : true
crimeDate : 2025-09-03T00:00:00.000+00:00
crimeTime : "20:56"
isBlocked : false
isDeleted : true
status : "pending"
reportDate : 2025-09-08T17:44:19.883+00:00
reportTime : 2025-09-08T17:44:19.883+00:00
createdAt : 2025-09-08T17:44:19.889+00:00
updatedAt : 2025-09-10T15:53:31.461+00:00
reportId : "RPT-1757353459891-734"
```


Fig.4.8 Report Data in MongoDB



Fig.4.9 Central-police Data in MongoDB



Fig.4.10 Local-police Data in MongoDB

ID	User/Officer Name	Role	Crime Type	Location	Status	Alert Message
1	John Doe	User	Theft	City Center	Pending	Be aware of theft!
2	Officer Smith	Officer	Assault	Market Street	Investigating	Respond immediately
3	Jane Roe	User	Burglary	Elm Street	Resolved	Crime reported

Table 4.2: Database

CHAPTER 5

SOFTWARE OUTCOME AND IMPLEMENTATION

Software outcomes refer to the result of the software and software implementation stage involves the transformation of the software technical data package (TDP) into one or more fabricated, integrated, and tested software configuration items that are ready for software acceptance testing. Several factors must be considered to appreciate the software implementation workload for the Crime Alert Application:

- The skills and experience of the software implementation personnel to design, code, test, and integrate the structural elements into a complete software product configuration for web and mobile platforms.
- The data manipulation dexterity of the programming language constructs (statements, semantics, evocation, extensibility, etc.) to enable software units of the Crime Alert Application to be effectively and efficiently designed and coded.

5.1 Home Page

The Fig.5.1 below shows the user registration interface of the Crime Alert system, which is designed to collect essential personal details required for creating a secure user account. The form includes input fields for first name, last name, age, date of birth, address, email, and contact number, ensuring that each registered user can be uniquely identified and contacted when necessary. Additionally, users are allowed to upload a profile image, which helps in personalizing their accounts and improving community trust within the system. For security purposes, the form requires users to create a password and confirm it by re-typing, minimizing the risk of errors during account creation. The design follows a simple and user-friendly structure, with clearly labeled fields and validation support to guide users through the process [10].

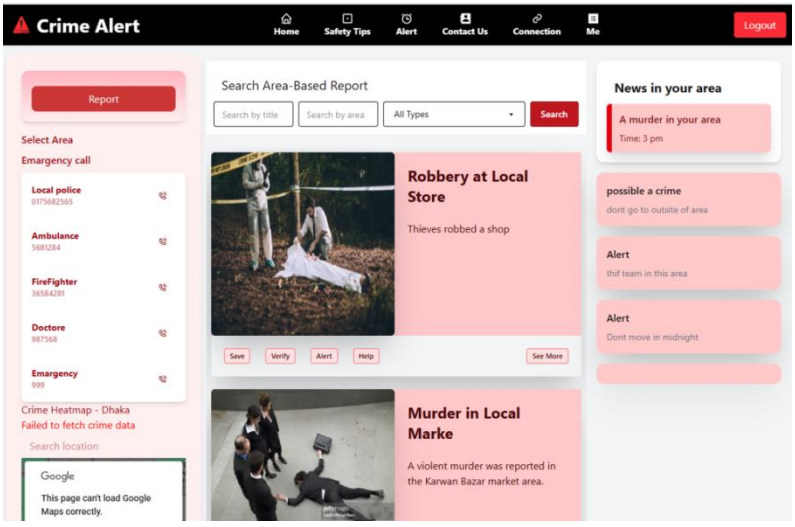


Fig.5.1: Home Page

5.2 Sign Up

The figure 8.8 above shows the user registration interface of the Crime Alert system, which is designed to collect essential personal details required for creating a secure user account. The form includes input fields for first name, last name, age, date of birth, address, email, and contact number, ensuring that each registered user can be uniquely identified and contacted when necessary. Additionally, users are allowed to upload a profile image, which helps in personalizing their accounts and improving community trust within the system [4].

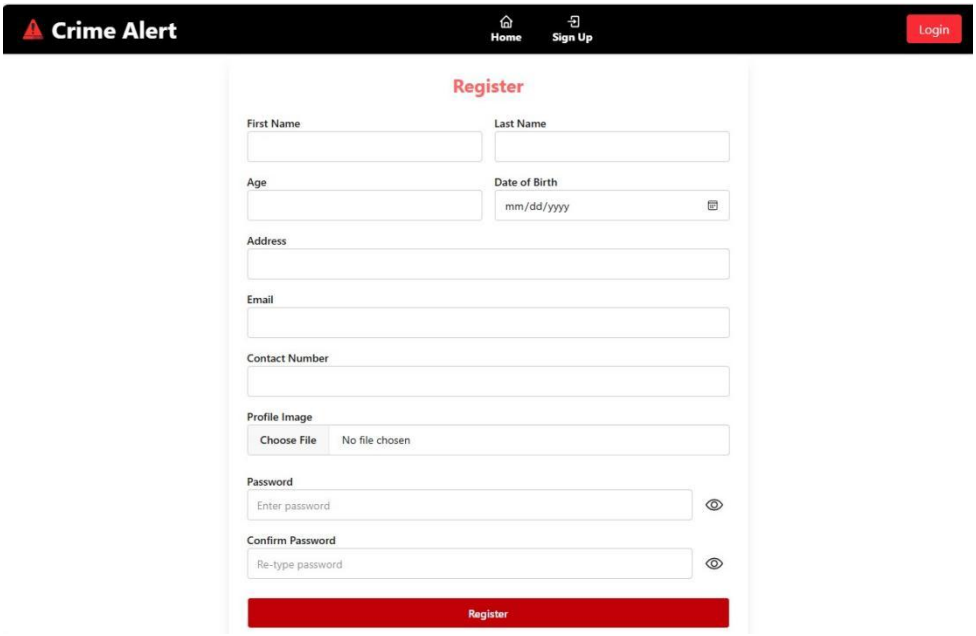


Fig.5.2: Sign Up

5.3 Sign in

The Fig.5.3 below illustrates the login interface of the Crime Alert system, which provides registered users with secure access to their accounts. The form includes fields for entering either an email address or a contact number along with a password, offering flexibility in the authentication process. A password visibility toggle is integrated into the design to minimize input errors, ensuring users can verify their credentials before submission. The login button is prominently displayed in red, emphasizing its importance as the primary action on the page [5].

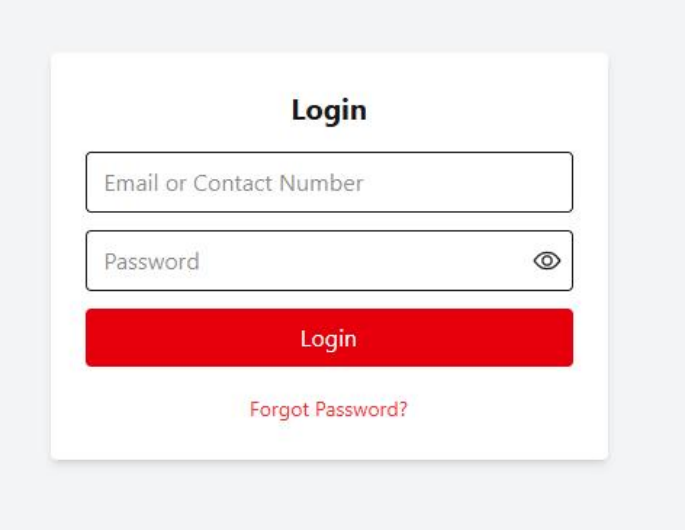


Fig.5.3: Sign in

5.4 Logout

Logging out means to end access to the website. When a user finished his work he/she can logout from the website.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

With our proposed Crime Alert Application, we aim to create a reliable and user-friendly system that enables people to report crimes quickly and receive timely assistance through both web and mobile platforms. The system uses location-based services to identify the exact position of the user and share it instantly with nearby police stations and emergency responders, ensuring fast and accurate action. It not only allows users to report incidents but also provides alerts about crime-prone areas and community safety updates, making it more effective and robust. The application is designed to be cost-effective, easy to use, and efficient in saving valuable time during emergencies, while the real-time flow of information strengthens coordination between citizens and law enforcement agencies. Thus, this Crime Alert Application stands as a practical and scalable solution for enhancing public safety through emergency location-based crime reporting and alert services.

6.2 Future Works

With our proposed Crime Alert Application, we aim to create a reliable and user-friendly system that enables people to report crimes quickly and receive timely assistance through both web and mobile platforms. The system uses location-based services to identify the exact position of the user and share it instantly with nearby police stations and emergency responders, ensuring fast and accurate action. It not only allows users to report incidents but also provides alerts about crime-prone areas and community safety updates, making it more effective and robust. The application is designed to be cost-effective, easy to use, and efficient in saving valuable time during emergencies, while the real-time flow of information strengthens coordination between citizens and law enforcement agencies. Thus, this Crime Alert Application stands as a practical and scalable solution for enhancing public safety through emergency location-based crime reporting and alert services.

As we can’t fulfill the modern technique, those are interested to work like our project are suggested to make mobile applications and future work real-time monitoring system.

REFERENCES

[1] M. Fowler, “UML Distilled: A Brief Guide to the Standard Object Modeling Language”, 3rd ed., Boston, MA, USA: Addison-Wesley, 2003, [Book].

[2] J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters”, Communications of the ACM, vol. 51, no. 1, pp. 107–113, Jan. 2008, [Journal].

[3] M. Mwiya and J. Phiri, “Utilisation of ICTs in Crime Reporting and Monitoring: A Case of Zambia”, International Journal of Computer Applications, vol. 4, no. 3, pp. 1–6, 2007, [Journal].

[4] A. K. Tripathi and R. Pandey, “Enhancing public safety through crime alert mobile applications”, in Proc. IEEE Int. Conf. Advances in Computing, Communications and Informatics (ICACCI), Mysore, India, 2013, pp. 2042–2047, [Conference].

[5] A. Sharma and P. Kumar, “A real-time crime reporting and alert system using IoT and cloud”, in Proc. IEEE Int. Conf. Computing, Communication and Automation (ICCCA), Greater Noida, India, 2015, pp. 1184–1189, [Conference].

[6] A. Al-Hasan and H. Al-Khateeb, “Secure mobile crime reporting application using cloud computing”, in Proc. IEEE Int. Conf. Cyber Security and Cloud Computing (CSCloud), New York, NY, USA, 2015, pp. 45–50, [Conference].

[7] OWASP Foundation, “Mobile Security Testing Guide”, September 7, 2015, [Online]. Available: https://owasp.org/.

[8] Node.js Foundation, “Node.js Documentation”, September 7, 2015, [Online]. Available: https://nodejs.org/en/docs/.

[9] MongoDB Inc., “MongoDB Atlas Security Features”, September 7, 2015, [Online]. Available: https://www.mongodb.com/.

[10] React Native, “React Native Documentation”, September 7, 2015, [Online]. Available: https://reactnative.dev/.