9/13/2025



CSE471: System Analysis and Design Project Report

Project Title: Enhancing Women’s Safety through Real-Time Volunteer Response (SentryAid)

|  |  |
| --- | --- |
| Group No: 02, CSE471 Lab Section: 03,  Summer 2025 | |
| Student ID | Name |
| Farah Tabussum | 22101185 |
| Erfan Khan | 22101590 |
| Abrar Ibne Haque | 21201734 |
| Rubaiyat Haque | 22101358 |

Submission Date: 13.09.2025

Contents

[1. Business Need: 2](#_Toc208697011)

[1.1 Business Requirements: 2](#_Toc208697012)

[1.2 Business Value: 2](#_Toc208697013)

[1.3 Special Issues or Constraints: 2](#_Toc208697014)

[2. Functional Requirements 3](#_Toc208697015)

[3. Technology (Framework, Languages) 3](#_Toc208697016)

[4. Backend Development 4](#_Toc208697017)

[5. User Interface Design 7](#_Toc208697018)

[6. Frontend Development 7](#_Toc208697019)

[7. SentryAid User Manual 12](#_Toc208697020)

[8. Performance and Network Analysis 18](#_Toc208697021)

[9. GitHub Repo [Public] Link 19](#_Toc208697022)

[10. Link of Deployed Project 19](#_Toc208697023)

[11. Individual Contribution 20](#_Toc208697024)

[12. References 20](#_Toc208697025)

# Business Need:

Many women feel unsafe walking alone at night due to harassment, being followed, or assault, and traditional emergency services are often too slow to respond. SentryAid offers a quick-response, community-driven safety solution by connecting women with verified local volunteers for immediate help. The system supports discreet SOS alerts, ensuring reliable assistance even in unfamiliar areas. By empowering women through technology-enabled safety, SentryAid builds confidence, strengthens community trust, and fosters collective responsibility. It maps high-risk zones using SOS data to guide preventive actions and provides administrators and law enforcement with actionable insights. Expected outcomes include reduced emergency response times (40–60%), prevention of potential assaults, increased reporting of harassment incidents (3–5x), improved volunteer engagement (70% active participation), and a 60% improvement in women’s perceived safety.

## Business Requirements:

* Secure registration with identity verification to ensure trust and prevent misuse.
* SOS alert system allowing women to send instant distress signals with location sharing.
* Real-time notifications to volunteers for quick response coordination.
* Direct in-app chat system between users and volunteers for immediate assistance.
* Admin tools for approval, emergency broadcasts, blacklisting abusers, and abuse reporting.
* Heatmap risk visualization to highlight unsafe areas at night for female users.
* Community bulletin board for sharing safety tips, warnings, and missing person alerts.

## Business Value:

* Enhances women’s safety by providing a real-time, community-driven help system.
* Builds confidence and trust through strict user verification and admin oversight.
* Creates safer communities by mapping high-risk zones and enabling preventive action.
* Provides administrators and law enforcement with actionable, data-driven insights.
* Expected outcomes include faster emergency response, prevention of assaults, increased reporting of harassment, improved volunteer engagement, and a 60% improvement in women’s perceived safety.

## Special Issues or Constraints:

* Must ensure strict data privacy and security, particularly for women’s personal information and location data.
* Requires high availability during late-night hours when risks are elevated.
* Strong misuse prevention mechanisms, such as blacklisting and reporting, are essential.
* Reliant on GPS accuracy and internet connectivity, which may affect performance in low-network areas.

# Functional Requirements

### Module 1: User Interaction & Communication

* Secure User Registration and Authentication (name, contact info, NID, encrypted login, optional 2FA)
* Password Recovery via email or phone verification
* Approve/Reject Users & Volunteers by Admins
* Emergency Broadcasts by Admins
* Profile Update for Users

### Module 2: Volunteer Management & SOS Response

* Volunteer SOS Response (notification and accept/decline functionality)
* Volunteer Availability and Suggestions (Active/Inactive status, mass alerts, estimated response times)
* SOS Alert System for Users (instant alert with location, message, silent/discreet mode, cancel alert)
* Emergency Contacts Integration
* In-App Chat System between Users and Volunteers

### Module 3: Analytics, Safety & Administration

* Heatmap Risk Visualization (high-alert zones in red, safe zones in green)
* Blacklist Suspicious Accounts by Admins
* Read Receipts for SOS Alerts
* Abuse Reporting by Users (tracking resolved or dismissed reports)
* Volunteer Ranking System (based on successful SOS responses or user ratings)
* Community Bulletin Board (blog-style feature for safety tips, alerts, missing persons)

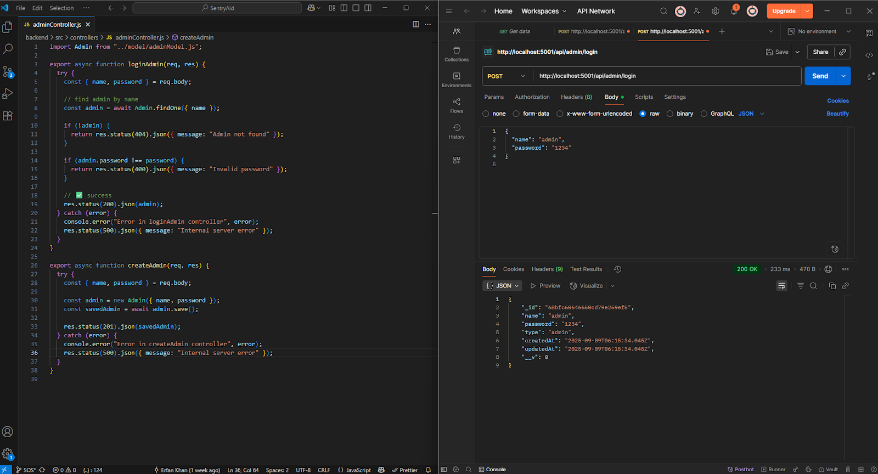
# Technology (Framework, Languages)

* **Frontend:** React (JSX) with Vite, Tailwind CSS, DaisyUI, Framer Motion, React Router, React Leaflet, Leaflet Heatmap, React Icons, Axios, EmailJS, Socket.IO Client, React Toastify
* **Backend:** Node.js + Express with Socket.IO, CORS
* **Database:** MongoDB Atlas (with Mongoose ORM)
* **Styling:** Tailwind CSS + DaisyUI (utility-first CSS framework)
* **Deployment:** Render (frontend), Render (backend), MongoDB Atlas (database)
* **Integrations:** EmailJS, Socket.IO real-time communication, Leaflet/Leaflet.heat (maps and heatmaps)

# Backend Development

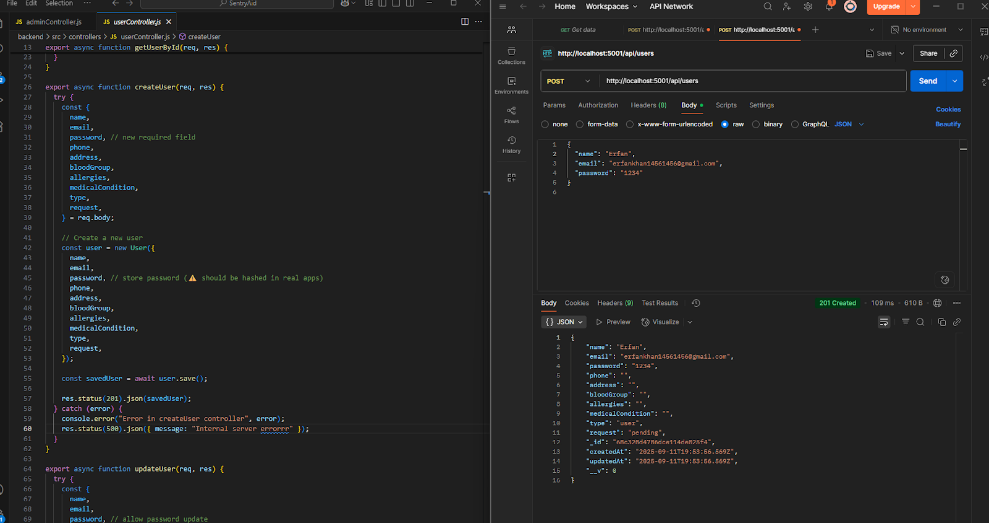
### Admin Controller – Login Function

The Admin Controller manages administrative operations in the backend. The loginAdmin function allows admins to log in by verifying their name and password, returning admin details on success, handling invalid credentials, and properly managing server errors. It ensures secure access for administrative users.



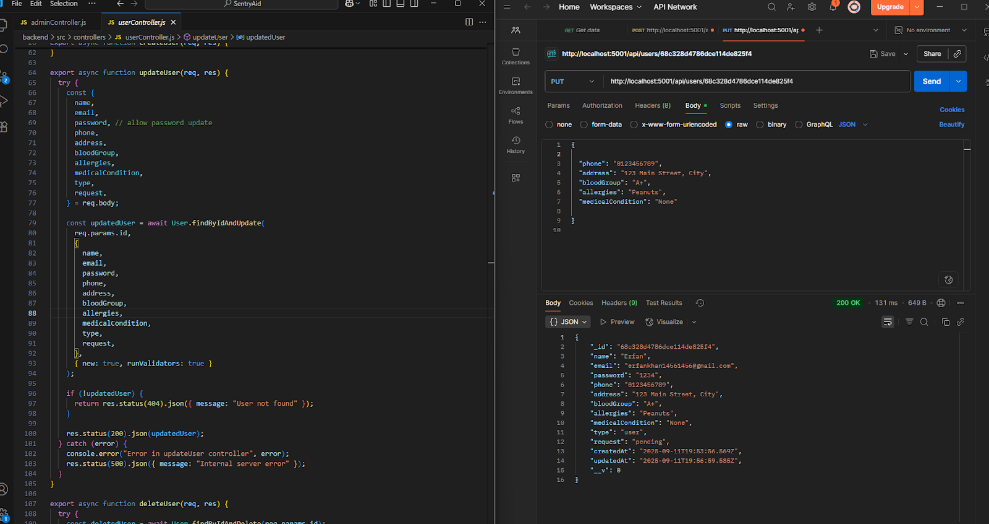
### User Controller – Create User Function

The User Controller manages user-related operations in the backend. The createUser function allows creating a new user by accepting details such as name, email, password, phone, address, blood group, allergies, medical conditions, user type, and request status. It saves the user to the database and returns the saved record, while handling any server errors.



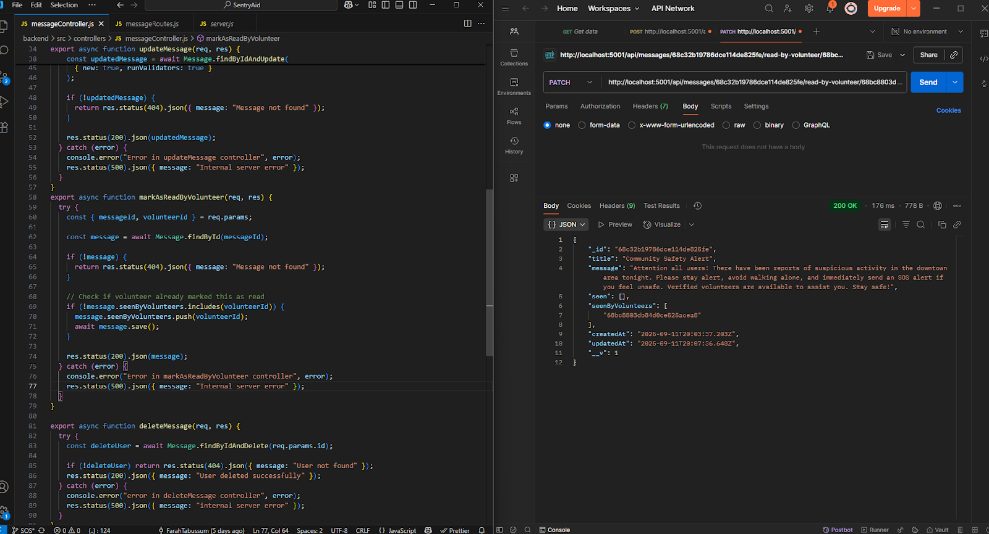
### User Controller – Update User Function

The User Controller handles user management in the backend. The updateUser function allows updating an existing user’s details, including name, email, password, phone, address, blood group, allergies, medical conditions, user type, and request status. It searches for the user by ID, applies the updates, returns the updated record, and handles errors if the user is not found or if a server error occurs.



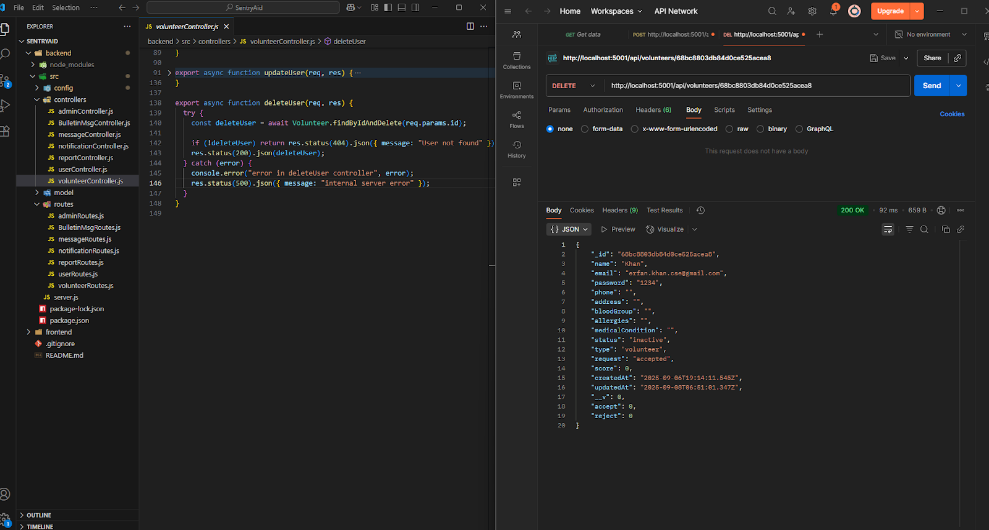
### Message Controller – Mark as Read by Volunteer

The Message Controller manages message interactions between users and volunteers. The markAsReadByVolunteer function allows a volunteer to mark a specific message as read. It checks if the message exists and whether the volunteer has already marked it as read. If not, the volunteer ID is added to the seenByVolunteers array, and the updated message is saved. Proper error handling ensures appropriate responses for missing messages or server issues.



### Volunteer Controller – Delete User Function

The Volunteer Controller manages volunteer-related operations in the backend. The deleteUser function allows deletion of a volunteer by their ID. It checks if the volunteer exists, removes the record from the database, and returns the deleted data. Proper error handling ensures responses for cases where the volunteer is not found or if a server error occurs.



### Socket.IO Integration in SentryAid Backend

SentryAid uses Socket.IO for real-time communication between users and volunteers, enabling instant SOS alerts, live chat, and location sharing. Users can join rooms, send SOS requests, and volunteers can accept or reject them, while messages and location updates are delivered only to relevant participants. The system also handles disconnections and cleans up active connections, ensuring reliable, immediate coordination during emergencies.

|  |  |
| --- | --- |
|  |  |

# User Interface Design

The design prototypes were developed before the actual implementation of the project. However, these initial designs are not fully consistent with the final system. In total, four separate Figma designs were created, along with one combined design that consolidated all of them.

|  |  |
| --- | --- |
| Created By | Link to Figma Design |
| Abrar Ibne Haque (21201734) | [Design 1](https://www.figma.com/design/7qra82Dj16P9yIZ0YEjwrE/Abrar?node-id=1-2) |
| Farah Tabussum (22101185) | [Design 2](https://www.figma.com/proto/HSXBdXXK1ffJ2NoPQdO0hb/Assignment2?node-id=1-9&starting-point-node-id=1%3A9) |
| Rubaiyat Haque (22101358) | [Design 3](https://www.figma.com/design/jIgdqjJlyNCSNOKUWzBAKt/Trial?node-id=0-1&p=f) |
| Erfan Khan (22101590) | [Design 4](https://www.figma.com/design/Gf9OpfqaM7wpA5rVU8uWdW/22101590_Erfan?node-id=0-1&p=f&t=gdwc9cDIp93dZDNu-0) |
| Team Collaboration | [Combined](https://www.figma.com/design/EMpVXpYlSutcJPStOI71C7/Assignment-2--Group-2-?node-id=0-1&p=f&t=xvpYjTrR0281kmHG-0) |

# Frontend Development

### React Router Configuration for SentryAid

This setup uses createBrowserRouter to manage navigation for users, volunteers, and admins.

* User Routes (users/:id): Profile, volunteer ranking, edit profile, heatmap, SOS, bulletin board.
* Volunteer Routes (volunteers/:id): Profile, ranking, edit profile, heatmap, SOS alerts, bulletin board.
* Admin Routes (admin): Dashboard, alert messages, heatmap, bulletin board.
* Standalone Pages: Login pages, bulletin creation, report volunteer.
* Supports nested routes, error handling, and role-based access for organized navigation.

|  |  |
| --- | --- |
|  |  |

### SOS Sender Component

The SOS component handles real-time emergency communication for the sender in the SentryAid system. It integrates with Socket.IO to:

* Connect to receivers in a designated room.
* Share live location continuously or stop sharing.
* Send and receive chat messages with connected receivers.
* Track connection status, chat history, and connected participants.
* Provide UI components for connection buttons, location sharing, message input, and instructions.
* It ensures real-time SOS coordination, location updates, and instant messaging for emergencies.

|  |  |
| --- | --- |
|  |  |
|  |  |

### SOS Alert Receiver Component

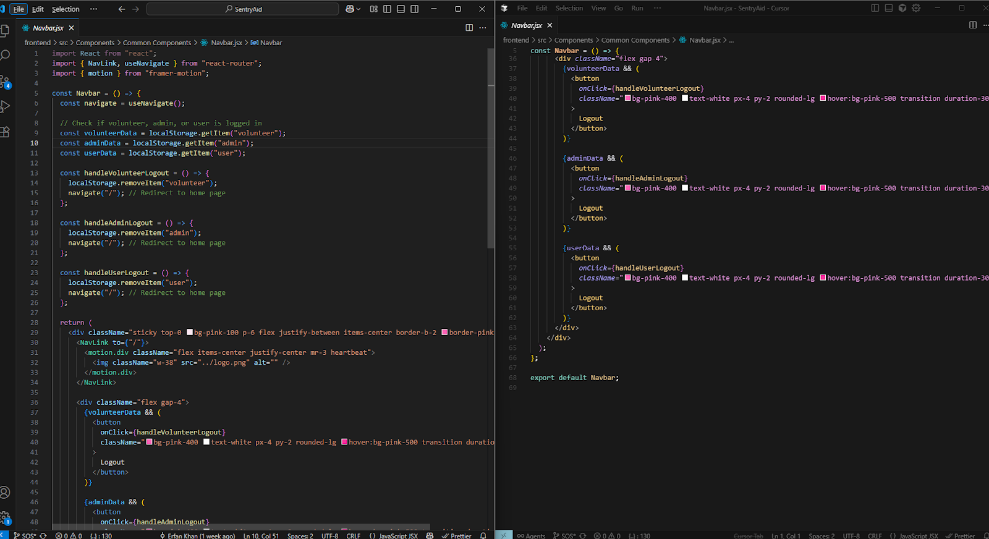
The SosAlert component allows volunteers/receivers to monitor and respond to SOS requests in real-time:

* Connection Requests: Accept or reject incoming sender connections.
* Location Tracking: View live sender locations on a Leaflet map, with color-coded markers and connecting paths.
* Map Management: Auto-zooms to fit all senders, tracks the receiver’s own location, and updates dynamically.
* Chat System: Real-time messaging with connected senders, including chat history and message input.
* User & Volunteer Management: Tracks volunteer status and updates scores upon accepting requests.
* It provides a full dashboard for monitoring multiple senders, handling emergency coordination, and chatting simultaneously.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

### Navbar Component in SentryAid Frontend

The Navbar in SentryAid provides a sticky, top-level navigation bar visible across all pages. It displays the website logo and dynamically renders a logout button based on the current logged-in role (user, volunteer, or admin). When a logout button is clicked, the corresponding role is removed from local storage, and the user is redirected to the home page. This ensures role-based session management and quick access to logout functionality while maintaining a clean and responsive UI.



### AlertMessage Component in SentryAid Admin Panel

The AlertMessage component allows administrators to create and manage emergency broadcasts for all users. Admins can compose a broadcast by entering a title and message, which is then sent to the backend via Axios. On success, the broadcast is confirmed with a toast notification, and the UI updates to show the latest messages. The component also displays a log of past broadcasts with timestamps, ensuring admins can track all emergency alerts. It handles form validation, loading states, and error handling, providing a smooth and reliable interface for urgent communications.

|  |  |
| --- | --- |
|  |  |

### Accident Heatmap in SentryAid

The Heat Map component provides a visual representation of accident-prone areas using a heatmap overlay on a Leaflet map. It simulates different regions with varying accident intensity, where green indicates low risk, yellow indicates moderate risk, and red indicates high-risk zones. The component generates random points within each region to simulate density and uses leaflet.heat to render the heatmap with color gradients. Users can pan and zoom the map to explore accident hotspots in real time, helping volunteers and authorities identify areas requiring urgent attention.



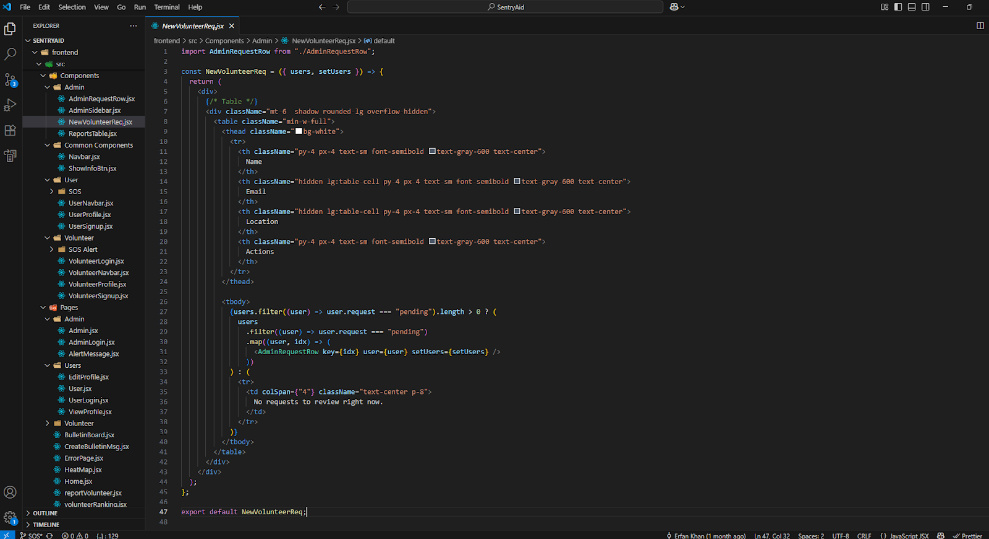
### Admin Request Management in SentryAid

The AdminRequestRow component allows administrators to review, approve, or reject user or volunteer requests. Each row displays the requester’s name, email, and location. Admins can accept requests with a single click, which updates the database and sends an acceptance email automatically. For rejections, a modal prompts the admin to provide a reason, which is then sent to the requester via email. After any action, the component refreshes the list of users or volunteers to reflect the latest status. This ensures smooth and transparent management of incoming requests while keeping requesters informed immediately.

|  |  |
| --- | --- |
|  |  |

### New Volunteer Request Management in SentryAid

The NewVolunteerReq component displays all pending user or volunteer requests in a responsive table with name, email, and location. Each row uses AdminRequestRow to handle approval or rejection, updating the backend and sending email notifications. If no requests are pending, a message is shown. This ensures efficient and transparent request management while keeping users informed.



# SentryAid User Manual

Welcome to SentryAid, a community-driven emergency response platform designed to connect people in distress with volunteers and provide administrators with oversight and control. This manual explains how the system works for all account types: Admin, User, and Volunteer.

|  |  |
| --- | --- |
|  |  |

### User Account

* Sign up and log in to the platform.
* Access is restricted until approved by the Admin.
* After approval, users can:
  + Edit their profile and view personal information.
  + Receive emergency broadcast messages from Admin.
  + Send and manage SOS alerts during emergencies.
  + Access the bulletin board to post and read blogs.
  + View the risk zone heatmap.
  + Reports & Moderation (User side):
  + Users can file reports against volunteers under categories such as:
    - Unprofessional Behavior
    - No Show to SOS (Failure to Respond)
    - Misconduct
  + Users can track the status of their reports (e.g., Resolved, Under Review, or Dismissed).

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

### Volunteer Account

* Sign up and log in to the platform.
* Access is restricted until approved by the Admin.
* After approval, volunteers can:
  + Toggle Active/Inactive status (inactive volunteers will not receive SOS alerts).
  + Receive SOS alerts from nearby users.
  + Accept or reject SOS requests.
  + Chat with users once a request is accepted.
  + Track live location shared by users during emergencies.
  + Access broadcasts from Admin.
  + View personal profile, bulletin board, risk zone heatmap, and ranking system.

|  |  |
| --- | --- |
|  |  |
|  |  |

### Admin Account

* Admin accounts are selective and created manually.
* Default Admin: Username: admin | Password: 1234
* Admin privileges include:
  + Reviewing and approving/rejecting new user and volunteer requests.
  + Sending email notifications upon acceptance or rejection.
  + Managing reports submitted against volunteers (resolve, dismiss, or delete volunteer accounts).
  + Broadcasting emergency alerts to all users and volunteers.
  + Viewing and monitoring the accident heatmap.
  + Managing the bulletin board (reviewing, approving, and adding blogs).

|  |  |
| --- | --- |
|  |  |
|  |  |

### SOS Alerts

The SOS Alert system is the core of SentryAid, enabling real-time emergency assistance between users and volunteers.

For Users

1. Navigate to the Send SOS page.
2. Press the Send SOS Alert button during emergencies.
3. Volunteers are notified instantly.
4. Once a volunteer accepts:

You can chat directly with them.

A new option appears to share your live location.

You can see which volunteers have accepted your request.

1. When the situation is resolved, press Cancel SOS to end the alert.

For Volunteers

1. Go to the Get SOS Alert page.
2. Incoming SOS requests appear in real time.
3. Volunteers may accept or reject requests.

Accepting notifies the user that help is on the way.

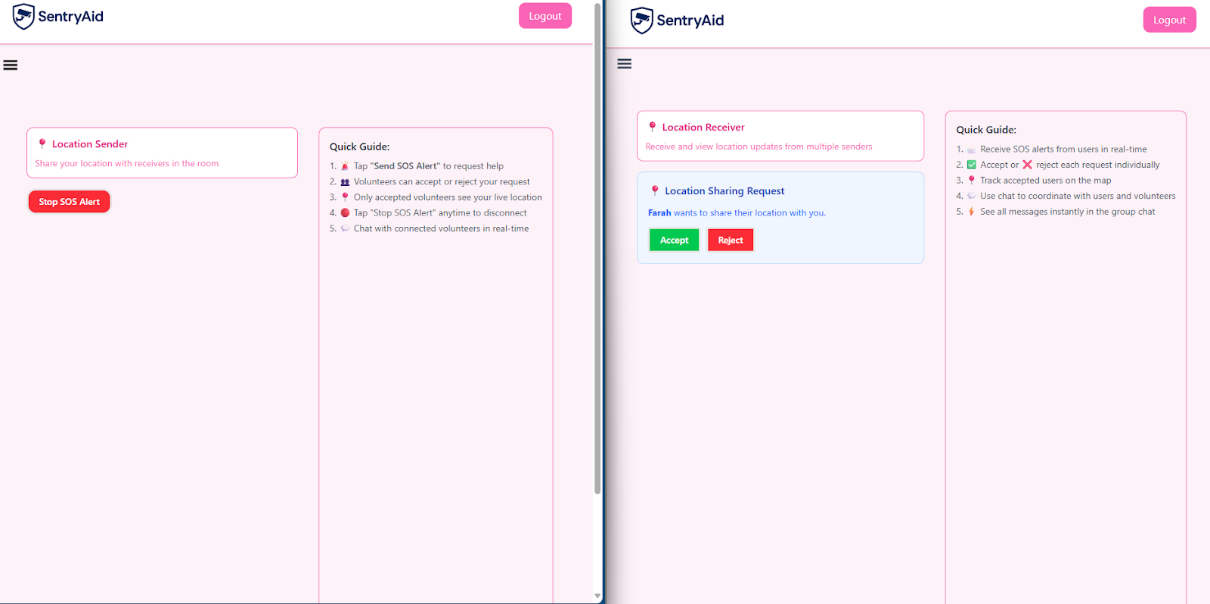
Rejecting passes the request to other available volunteers.

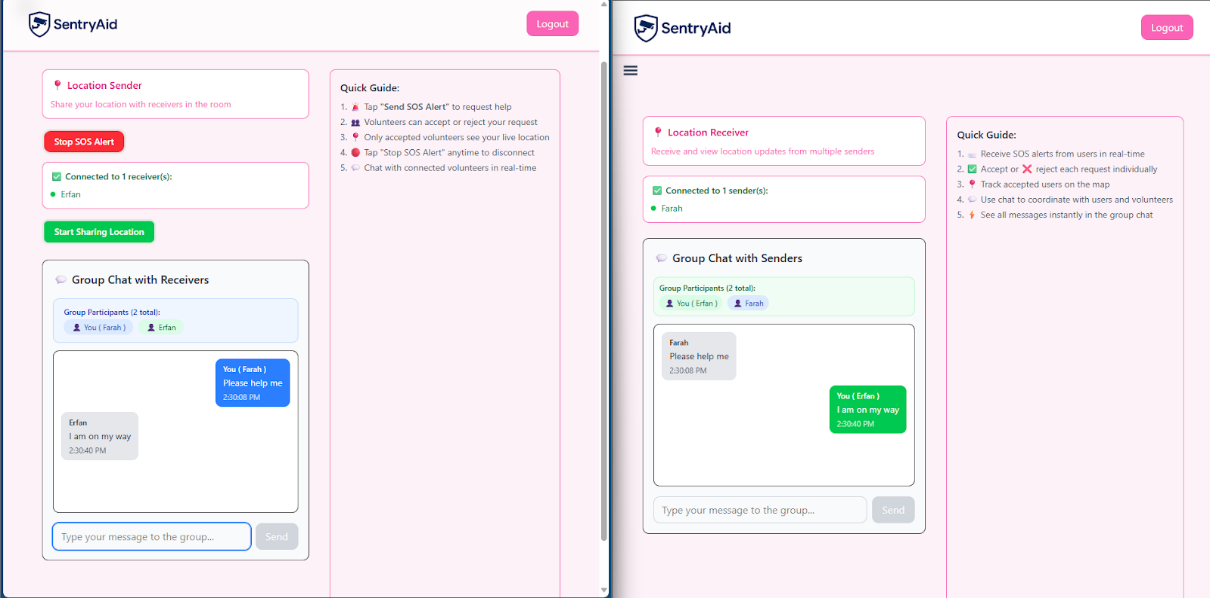
1. Once accepted:

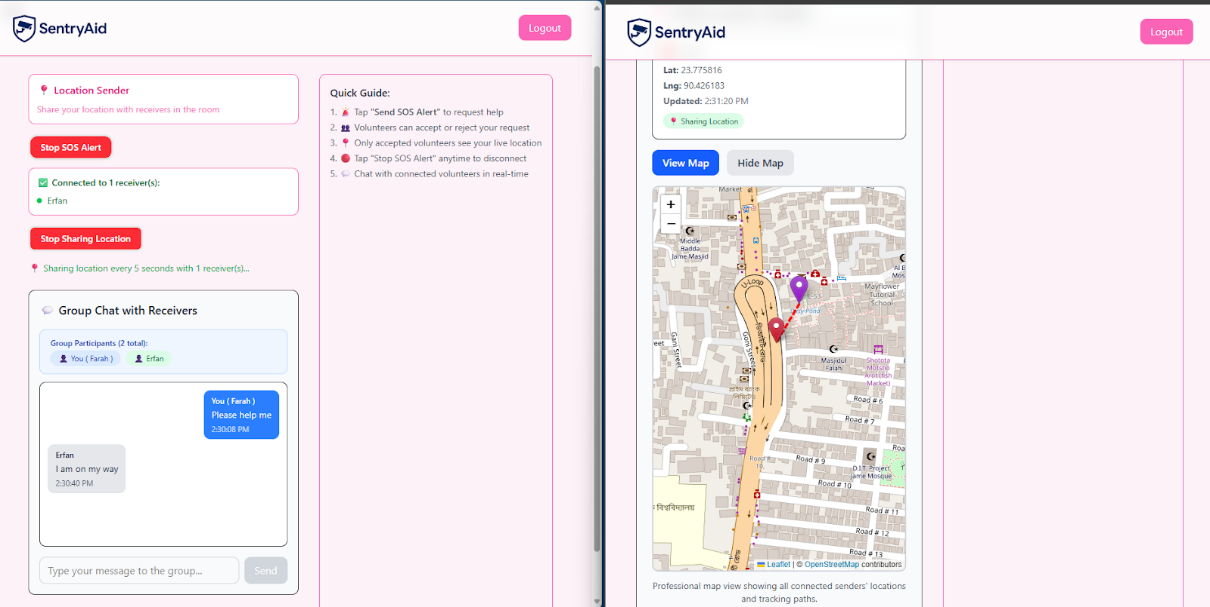
The chat system activates.

If the user shares their location, you can track their live movement on the map.

1. SOS details disappear when the user cancels the request.







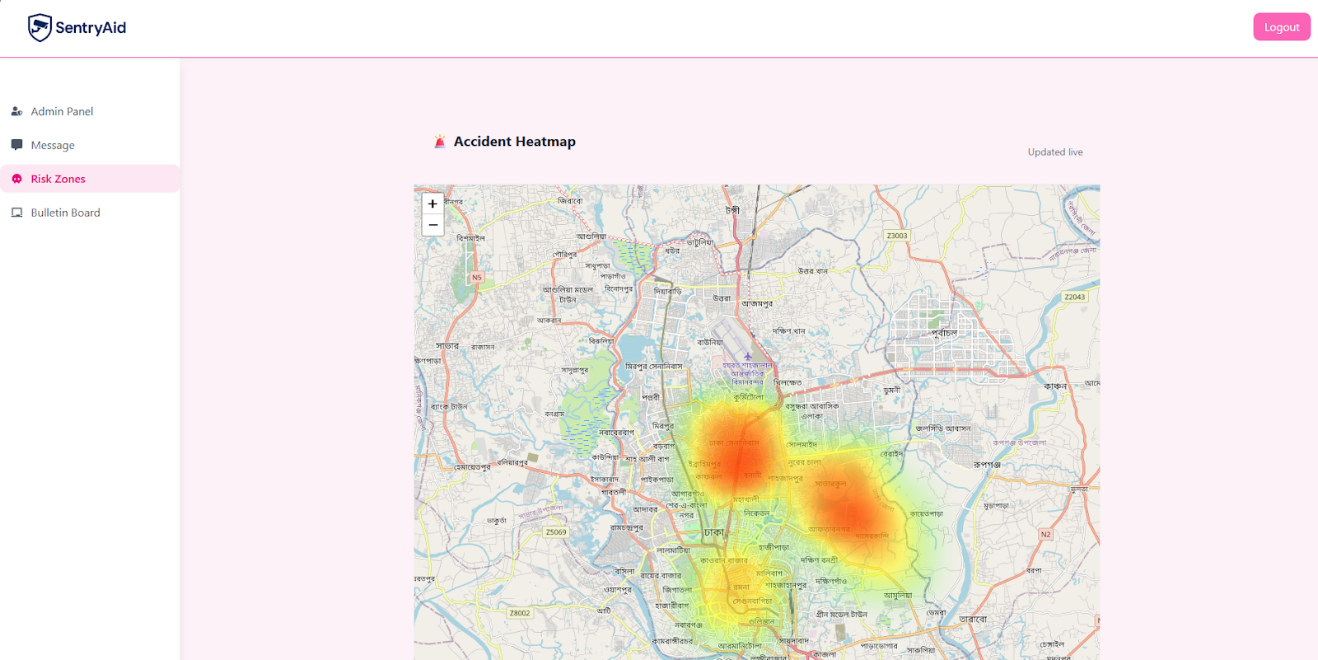
### Emergency Broadcasts

* Admins can send emergency alerts to all users and volunteers simultaneously.
* Alerts include a title and detailed message.
* Users and volunteers see these broadcasts on their dashboard.

|  |  |
| --- | --- |
|  |  |

### Risk Zone Heatmap

* Admins, Users and Volunteers can view the heatmap of emergency incidents reported by users.
* Heat levels are shown in green, yellow, and red zones to indicate severity and frequency.
* Helps identify accident-prone areas for faster and more prepared responses.



# Performance and Network Analysis

### Lighthouse Performance Metrics (Desktop)

* Performance: 96% – The application loads quickly and responds efficiently.
* Accessibility: 90% – Users can navigate the app easily, with minor improvements possible.
* Best Practices: 70% – Some updates are needed for optimal security and modern coding standards.
* SEO: 91% – Content is well-structured for search engine indexing.

The app demonstrates excellent performance and accessibility. Best practices could be improved slightly, and SEO is strong.

### Network Highlights (Desktop)

* Total requests: 15, with 7 critical requests completed during page load
* Page size: 1,163 kB of resources, 3.9 kB transferred
* Critical resources: 2.5 kB
* DOMContentLoaded: 508 MS
* Full page load: 510 MS
* Finish time: 26.04 s
* Status: No failing requests; critical resources load efficiently

The network performance is efficient, with fast rendering and minimal data transfer, ensuring a responsive user experience.

|  |  |
| --- | --- |
|  |  |

### Lighthouse Performance Metrics (Mobile)

* Performance: 80% – The application loads efficiently, with room for optimization.
* Accessibility: 83% – Users can navigate the app comfortably; some improvements recommended.
* Best Practices: 79% – Minor updates needed for security and coding standards compliance.
* SEO: 91% – Content is well-structured for search engine indexing.

The app performs well overall, with solid accessibility and SEO scores. Best practices and performance could be slightly improved.

### Network Highlights (Mobile)

* Total requests: 13, with 6 critical requests completed during page load
* Page size: 1,162 kB of resources, 2.2 kB transferred
* Critical resources: 0.9 kB
* DOMContentLoaded: 115 MS
* Full page load: 116 MS
* Finish time: 1.71 s
* Status: All requests completed successfully; resources load efficiently

The network performance is fast and lightweight, with minimal data transfer and quick page rendering.

|  |  |
| --- | --- |
|  |  |

# GitHub Repo [Public] Link

[SentryAid GitHub Link](https://github.com/Erfan-Khan-Dhrubo/SentryAid)

# Link of Deployed Project

[SentryAid Deployed Link](https://sentryaid-frontend.onrender.com/)

# Individual Contribution

|  |  |
| --- | --- |
| Group member - 01 | |
| Name: Rubaiyat Haque | Student ID: 22101358 |
| Functional Requirements which are developed by this member: | |
| 1. Secure User Registration and Authentication (Module 1) | |
| 2. Password Recovery (Module 1) | |
| 3. Blacklist Suspicious Accounts (Module 3) | |
| 4. Community Bulletin Board (Blog-Style Feature) (Module 3) | |

|  |  |
| --- | --- |
| Group member - 02 | |
| Name: Erfan Khan | Student ID: 22101590 |
| Functional Requirements which are developed by this member: | |
| 1. Approve/Reject Users & Volunteers (Module 1) | |
| 2. Volunteer SOS Response (Module 2) | |
| 3. SOS Alert System (Module 2) | |
| 4. In-App Chat System (Module 2) | |

|  |  |
| --- | --- |
| Group member - 03 | |
| Name: Farah Tabussum | Student ID: 22101185 |
| Functional Requirements which are developed by this member: | |
| 1. Emergency Broadcasts (Module 1) | |
| 2. Volunteer Availability and Suggestions (Module 2) | |
| 3. Abuse Reporting (Module 3) | |
| 4. Volunteer Ranking System (Module 3) | |

|  |  |
| --- | --- |
| Group member - 04 | |
| Name: Abrar Ibne Haque | Student ID: 21201734 |
| Functional Requirements which are developed by this member: | |
| 1. Profile Update (Module 1) | |
| 2. Emergency Contacts Integration (Module 2) | |
| 3. Heatmap Risk Visualization (Module 3) | |
| 4. Read Receipts for SOS Alerts (Module 3) | |

# References

N/A