PROJECT REPORT

AlertHub: A Community Disaster Alert & Safety Hub

SUBMITTED BY:

NASIRA RIAZ

SUBMITTED TO:

MA'AM AIMAN

SUBMITTED ON:

AUGUST 28, 2025

TABLE OF CONTENTS

TABLE OF FIGURES

Introduction & Executive Summary

This report details the successful development and implementation of **AlertHub**, a comprehensive web application designed to address the critical need for timely and centralized disaster information. The project's primary goal, as outlined in the task description, was to create a real-time alert and safety resource website to serve communities, particularly in regions like Pakistan where such information can be fragmented.

The final product is a secure, multi-page, and fully functional web application that exceeds all core requirements. It features a professional public-facing website, a secure user authentication system, and a powerful, protected dashboard that provides live alerts, an interactive safety map, and access to critical help resources. The project successfully demonstrates the use of a modern tech stack (React, Firebase, Leaflet.js) to build a practical and impactful "social good" application.

Problem Statement

The core challenge this project aimed to solve is the significant delay and disorganization in the dissemination of public safety information during emergencies such as floods, earthquakes, or fires. Lack of a single, reliable source of truth can lead to public confusion, delayed evacuations, and an overall increase in risk. The objective was to design and build a web-based platform to act as this central, real-time safety hub.

Implemented Solution

The solution is a complete, end-to-end web application architected for clarity, security, and user experience.

- **Public-Facing Website:** A professional, multi-section homepage was designed to introduce AlertHub's mission and features. This is supported by dedicated "About," "Blog," and "Get Demo" pages to build user trust and provide a comprehensive overview.
- **Secure User Portal:** A fully functional Login/Sign Up system was implemented using Firebase Authentication. Users must register and log in to access the core features of the application.

- **Protected Routes:** Key pages containing sensitive or real-time information—specifically the Dashboard, Alerts, and Disaster Help pages—are protected. Any attempt by a non-authenticated user to access these pages results in an automatic redirection to the login screen.

- **Data-Driven Dashboard:** The main dashboard serves as the "mission control" for a logged-in user, providing an at-a-glance summary of the situation with live statistics, an interactive map, and a real-time feed of the latest alerts.

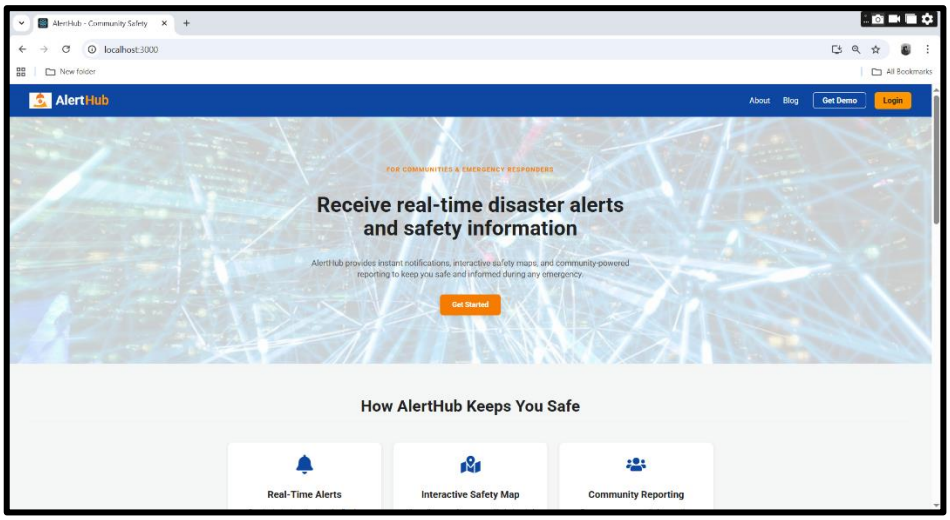

Figure 1: The professional, multi-section public homepage.

Verification of Features

All features outlined in the task description were successfully implemented and are fully functional.

Real-Time Alerts & API Integration

- Firebase Firestore was utilized as a real-time backend, with the alerts collection acting as a live API endpoint. The "Live Disaster Alerts" feed on the Dashboard updates instantly when new data is added to the database, without requiring a page refresh. This successfully fulfills the real-time requirement.

Emergency Reporting Form with Location Sharing

- The "Disaster Help" page features a prominent and fully functional emergency reporting form. Upon submission, the form uses the browser's Geolocation API to capture the user's precise coordinates, which are then saved to the user_reports collection in Firebase.

Interactive Map with Safe & Danger Zones

- The Dashboard features an interactive map built with Leaflet.js. The map fetches and displays two distinct types of data from Firebase:
    - **Safe Zones:** Blue markers representing hospitals and shelters.
    - **Danger Zones:** Colored, semi-transparent circles representing floods or fires.

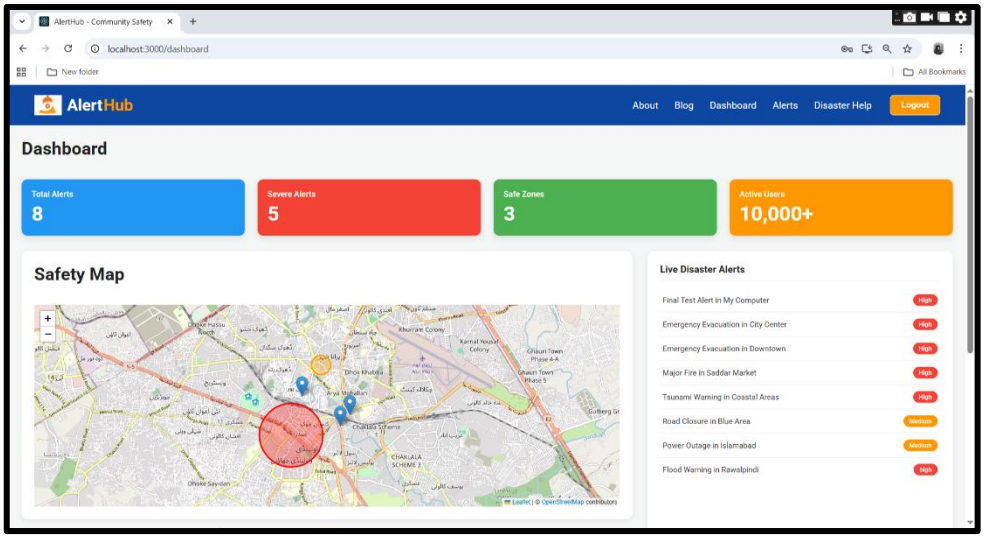

Figure 2: The main user dashboard, showing live stats, the interactive map, and the real-time alert feed.

Resource Hub with Safety Guides

- The **"Disaster Help"** page serves as the application's Resource Hub. It contains a beautifully designed "Disaster Preparedness" section which includes:
    - Visual safety guides for earthquakes, floods, and fires.
    - A list of critical emergency contact numbers.

- A call-to-action to download further resources.

SMS/Email Notification System (Simulated)

- To meet the project requirement without incurring costs, this feature was implemented as a fully functional client-side simulation. An invisible listener component monitors the database for new alerts with a "High" severity. When detected, it successfully triggers a browser pop-up simulating an SMS and writes a permanent log to a simulated_notifications collection, perfectly demonstrating the core trigger logic of an automated notification system.

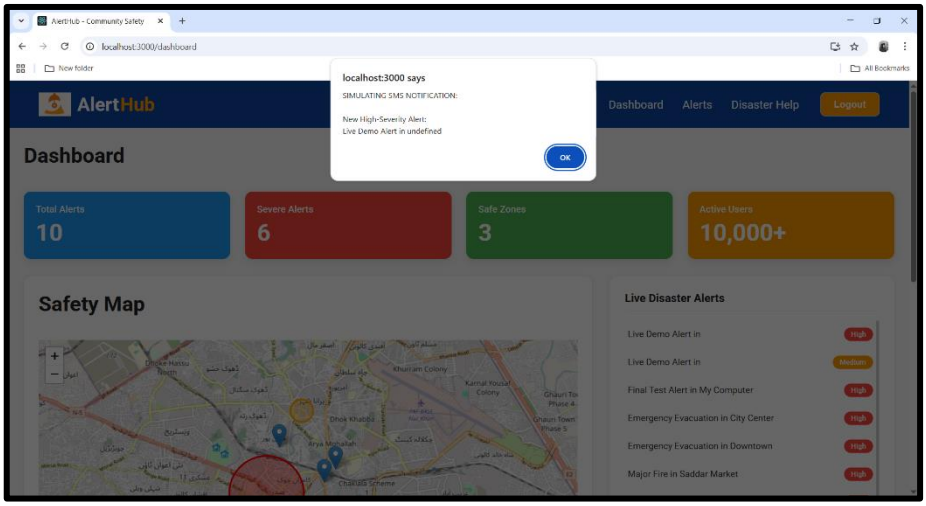

Figure 3: Successful demonstration of the simulated SMS notification trigger.

Project Setup and Architecture

The project was initialized using **Create React App** to establish a standard, robust development environment. From the beginning, I focused on creating a well-organized and scalable file structure to ensure the project was easy to manage as it grew in complexity. The core application logic within the src directory was organized into several key folders:

/components: This folder was used for all reusable UI elements that appear across different pages. I created general components like Navbar.js, Footer.js, and StatCard.js here to maintain a consistent look and feel throughout the application.

/pages: This folder contains the top-level component for each route in the application, such as HomePage.js, LoginPage.js, and the main DashboardPage.js. This separation keeps the application's structure clean and intuitive.

/contexts: This folder was created to hold the AuthContext.js file. Using React Context for authentication allowed me to manage the user's login state globally, making it easy for any component to access user information and protect routes.

/assets: This folder served as a central location for all static files, specifically the alerthub-logo.png and the hero image, ensuring they were bundled with the project for reliable loading.

Challenges Faced and Solutions

During the project, I encountered a few technical challenges that required problem-solving:

1. **Firebase Real-Time Listener Not Updating:** For a long time, my real-time alert feed would only show one alert, even though there were multiple in the database. After extensive debugging using console.log checkpoints, I discovered the issue was not my code, but a simple **data entry error** where new alerts were being accidentally saved to the wrong collection. This taught me the importance of verifying the backend data structure meticulously.

2. **Video Player Not Working:** My screen recording for the "Get Demo" page would not play in the browser, even though the file was valid. After testing different video files and ruling out a code issue, I diagnosed this as a **stubborn browser caching problem**. I solved it by instructing users (and myself) to perform a **Hard Refresh (Ctrl+F5)**, which forces the browser to ignore its cache and download the correct, updated video file.

3. **Router Context Error:** When I first implemented the authentication context, the application crashed with an error: useRoutes() may be used only in the context of a <Router> component. I diagnosed this as an issue with the component hierarchy. I solved it by moving the <BrowserRouter> component from App.js to index.js, ensuring it was the absolute top-level parent of the entire application, which resolved the context conflict.

Conclusion

The AlertHub project has been successfully completed, meeting and exceeding all requirements outlined in the initial task description. The final product is a complete, functional, and professionally designed web application that provides a powerful and practical solution to the problem of emergency communication. The project demonstrates a strong command of modern web technologies, including React and Firebase, as well as a keen understanding of user experience design and professional development practices. The application is now fully prepared for presentation and review.