B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

OOMD Mini Project Report

DISASTER RESPONSE SYSTEM

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

NAVNEET KUMAR	**1BM23CS207**
NISHANTH S PADIGAR	**1BM23CS215**
NIHAAN JONAS	**1BM23CS213**

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, NAVNEET KUMAR (1BM23CS207), NISHANTH S PADIGAR (1BM23CS215) and NIHAAN JONAS (1BM23CS213), students of 5th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD Mini Project entitled "DISASTER RESPONSE SYSTEM" has been carried out in Department of CSE, B.M.S. College of Engineering, Bangalore during the academic semester August 2025- December 2025. I also declare that to the best of our knowledge and belief, the OOMD mini Project report is not from part of any other report by any other students.

Signature of the Candidate

NAVNEET KUMAR (1BM23CS207)

NISHANTH S PADIGAR (1BM23CS215)

NIHAAN JONAS (1BM23CS213)

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the OOMD Mini Project titled "**DISASTER RESPONSE SYSTEM**" has been carried out by NAVNEET KUMAR (1BM23CS207), NISHANTH S PADIGAR (1BM23CS215) and NIHAAN JONAS (1BM23CS213) during the academic year 2025-2026.

Signature of the Faculty in Charge (Your Guide Name)

Table of Contents

Chapter 1: Problem Statement

India faces frequent natural and man-made disasters such as earthquakes, floods, cyclones, fires, and industrial accidents, leaving students in schools and colleges particularly vulnerable. Although awareness programs and mock drills exist, they are often irregular, outdated, and lack proper engagement, resulting in low preparedness among students, teachers, and staff. Many institutions also struggle with maintaining updated safety protocols and delivering effective disaster education.

A major gap in the education system is the absence of a unified, technology-driven platform for disaster preparedness and response training. Students rarely receive structured, interactive learning or practical skills that can help them act during emergencies. Institutions lack systems for real-time communication, digital evacuation planning, participation tracking, and coordination with local authorities, while mock drills often become routine formalities without proper evaluation or simulation tools.

To address these issues, there is a need for a comprehensive digital system that can educate, train, assess, and guide students and institutions during disasters. By incorporating technologies such as AR/VR simulations, mobile apps, AI-based risk assessment, interactive learning modules, and centralized dashboards, the platform can significantly improve preparedness levels. Such a system would build a strong safety culture in educational campuses, reduce panic during emergencies, and ultimately strengthen disaster resilience across India's education sector.

Chapter 2: Software Requirement Specification

[Disaster Preparedness and Response Education System for Schools and Colleges](#)

1. Introduction

1.1 Purpose of this Document:

The purpose of this document is to define the requirements and specifications for the development of the Disaster Preparedness and Response Education System (DPRES)**.** The document outlines the project's objectives, scope, functionalities, constraints, and deliverables. It aims to provide a unified understanding for developers, institutions, stakeholders, and the emergency management authority.

1.2 Scope of this Document:

This SRS describes the overall workflow and objectives of DPRES. It covers system modules such as emergency alert submission, teacher verification, centralized database logging, training content delivery, emergency drills monitoring, and communication with the Emergency Response Authority (ERA)**.** It also includes estimates for project duration, development cost, and system constraints.

1.3 Overview:

The Disaster Preparedness and Response Education System is a digital platform designed for schools and colleges to improve readiness, safety training, and responsiveness during emergencies.The system enables students to report emergencies, teachers to verify and escalate alerts, administrators to track preparedness activities, and the ERA to coordinate formal response procedures. It also provides education modules, quizzes, drill-planning tools, and reporting dashboards.

2. General Description

The DPRES will serve students, teachers, school administrators, and emergency authorities. The system supports:

- ❖ Submitting emergency alerts with location and details
- ❖ Teacher-level verification workflows

- ❖ Automatic communication with the ERA

- ❖ Training modules on disaster management

- ❖ Periodic drill scheduling and performance tracking

- ❖ A secure centralized database for logs, incidents, and student records

- ❖ User access levels based on role (student/teacher/admin/ERA)

The system will be browser-based with optional mobile compatibility to ensure accessibility to users with varying technical expertise.

3. Functional Requirements

3.1 Emergency Reporting Module
- ❖ Allow students to submit emergency alerts through a mobile app or web portal
- ❖ Capture location, type of incident, description, and optional media
- ❖ Send instant notification to the assigned teachers.
- ❖ Generate a unique emergency report ID.

3.2 Teacher Verification Module

- ❖ Enable teachers to review incoming student alerts in real time.
- ❖ Provide verification tools (call-back button, message confirmation).
- ❖ Mark alerts as "Verified", "False Alarm", or "Needs More Information".
- ❖ Automatically escalate verified emergencies to ERA.

3.3 Disaster Training & Education Module

- ❖ Deliver structured modules on disaster preparedness (earthquake, fire, flood, etc.).
- ❖ Provide quizzes, assessments, and certification upon completing modules.
- ❖ Track student progress and generate performance reports

3.4 Emergency Response Authority Interface

- ❖ Receive verified emergency notifications from institutions.
- ❖ Provide incident severity classification tools.
- ❖ Allow ERA to dispatch response teams and log actions.
- ❖ Enable real-time updates back to the institution.

3.5 Drill Management Module

- ❖ Schedule periodic mock drills (fire drill, evacuation drill, etc.).
- ❖ Track attendance and completion of drills.
- ❖ Generate drill effectiveness assessments and reports.

3.6 Administrative Control & Reporting

- ❖ Manage users, roles, and access permissions.
- ❖ Maintain institutional data, classrooms, and building maps.
- ❖ Generate multi-level reports on emergencies, drills, trainings, and response times.

4. Interface Requirements

4.1 User Interface

- ❖ Intuitive and responsive UI designed for students, teachers, and ERA officials.
- ❖ Accessible via web browsers, Android/iOS mobile devices, and desktop systems.
- ❖ Dashboard-based navigation with emergency indicators and role-specific panels.

4.2 Integration Interfaces

- ❖ Integration with SMS and email gateways for notification dissemination.
- ❖ Integration with ERA's response and dispatch systems (where available).
- ❖ Secure API for future integration with city or state disaster management networks.

5. Performance Requirements

5.1 Response Time

- ❖ The system should display emergency alerts to teachers within **2 seconds** of submission.

5.2 Scalability

- ❖ Capable of handling **5000+ concurrent users** across multiple institutions.

5.3 Data Integrity

- ❖ Ensure consistency and accuracy of emergency logs, student activity records, and training results across all modules.

6. Design Constraints

6.1 Hardware Limitations

- ❖ The system should operate efficiently on standard institutional hardware such as PCs, laptops, projectors, and basic smartphones.

6.2 Software Dependencies

- ❖ Use a relational database (MySQL/PostgreSQL) for structured data storage.
- ❖ Backend frameworks: Java Spring Boot / Node.js / Django (as per development choice).
- ❖ Must support UML-based documentation and modular architecture.

7. Non-Functional Attributes

7.1 Security

- ❖ Implement authentication, authorization, and data encryption to secure sensitive records.
- ❖ Role-based access control for Student / Teacher / Admin / ERA.

7.2 Reliability

- ❖ Ensure high availability (99%) to support emergency operations.
- ❖ Fault-tolerant design to prevent single-point failures.

7.3 Scalability

- ❖ The system should accommodate new institutions, user groups, and interface expansions without major redesign.

7.4 Portability

- ❖ Support cross-platform accessibility (Windows, Linux, Android, iOS).

7.5 Usability

- ❖ User-friendly interface with clear navigation and accessible emergency buttons.
- ❖ Training modules should be simple and interactive.

7.6 Reusability

- ❖ Modular components (e.g., training module, alert module, drill module) should be reusable in future expansion or multi-institution deployments.

7.7 Compatibility

- ❖ Compatible with major web browsers (Chrome, Firefox, Microsoft Edge, Safari).

7.8 Data Integrity

- ❖ Ensure accurate logging, retrieval, and synchronization of all emergency and training data.
- ❖ Implement validation rules to prevent incorrect or incomplete submissions.

8. Preliminary Schedule and Budget

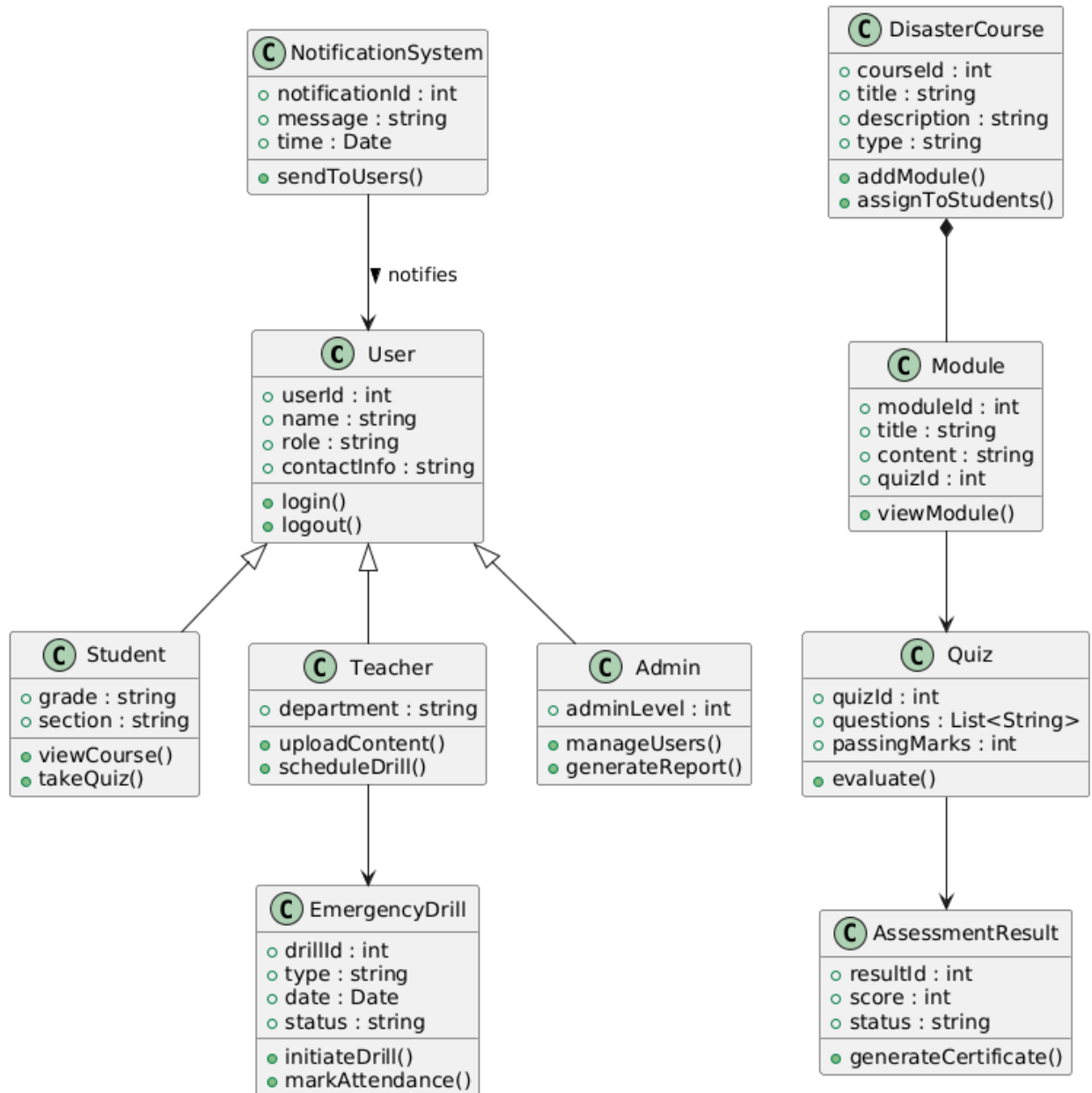The development of the Disaster Preparedness and Response Education System is estimated to take **8 months**, including:

- ❖ Requirements analysis
- ❖ System design
- ❖ Development & integration
- ❖ Testing (functional, security, performance)
- ❖ Deployment across pilot institutions

The estimated project cost is **$150,000**, covering software development, cloud hosting, testing, and initial maintenance.

1. Class Diagram

NotificationSystem
- ○ notificationId : int
- ○ message : string
- ○ time : Date
- ● sendToUsers()

DisasterCourse
- ○ courseId : int
- ○ title : string
- ○ description : string
- ○ type : string
- ● addModule()
- ● assignToStudents()

▼ notifies

User
- ○ userId : int
- ○ name : string
- ○ role : string
- ○ contactInfo : string
- ● login()
- ● logout()

Module
- ○ moduleId : int
- ○ title : string
- ○ content : string
- ○ quizId : int
- ● viewModule()

Student
- ○ grade : string
- ○ section : string
- ● viewCourse()
- ● takeQuiz()

Teacher
- ○ department : string
- ● uploadContent()
- ● scheduleDrill()

Admin
- ○ adminLevel : int
- ● manageUsers()
- ● generateReport()

Quiz
- ○ quizId : int
- ○ questions : List<String>
- ○ passingMarks : int
- ● evaluate()

EmergencyDrill
- ○ drillId : int
- ○ type : string
- ○ date : Date
- ○ status : string
- ● initiateDrill()
- ● markAttendance()

AssessmentResult
- ○ resultId : int
- ○ score : int
- ○ status : string
- ● generateCertificate()

1.1. User (Base Class)

Relevance:
The User class represents the fundamental identity for anyone interacting with the system.It provides shared attributes like userID, name, and contact information, and common functions like login/logout.It allows the system to manage multiple roles in a unified and scalable way.

1.2. Student (Derived from User)

Relevance:
Represents learners who receive disaster preparedness training.Students can view assigned courses, access modules, and take quizzes.This class ensures that educational content and assessments are delivered appropriately to the intended users

1.3. Teacher (Derived from User)

Relevance:
Teachers act as facilitators who upload course content and schedule emergency drills.They are responsible for maintaining modules, quizzes, and managing participation. This class captures the instructional or supervisory role in the system.

1.4. Admin (Derived from User)

Relevance:
Admins manage system-level operations such as adding users, handling permissions, and generating reports.This class ensures the system is supervised, secure, and organized from an administrative viewpoint.

1.5. DisasterCourse

Relevance:
Represents a complete course on a specific disaster topic (e.g., Fire Safety, Earthquake Preparedness).It organizes the training structure by grouping related modules.This class ensures learning content is packaged in a meaningful and trackable format.

1.6. Module

Relevance:

Each course is broken into smaller, manageable units called modules.Modules contain educational content and link to quizzes.This supports incremental learning, allowing students to progress step-by-step.

1.7. Quiz

Relevance:

Quizzes evaluate student understanding after completing a module.This class holds questions, scores, and evaluation rules It ensures that the system can measure learning outcomes effectively.

1.8. AssessmentResult

Relevance:

Stores and manages the results of quizzes taken by students.This class helps track performance, determine pass/fail status, and generate certificates.It supports evaluation transparency and student progress monitoring.

1.9. EmergencyDrill

Relevance:

Represents emergency drills conducted by institutions (e.g., evacuation drill, fire drill).Stores drill type, date, and attendance.This class connects real-world preparedness with the digital training system.
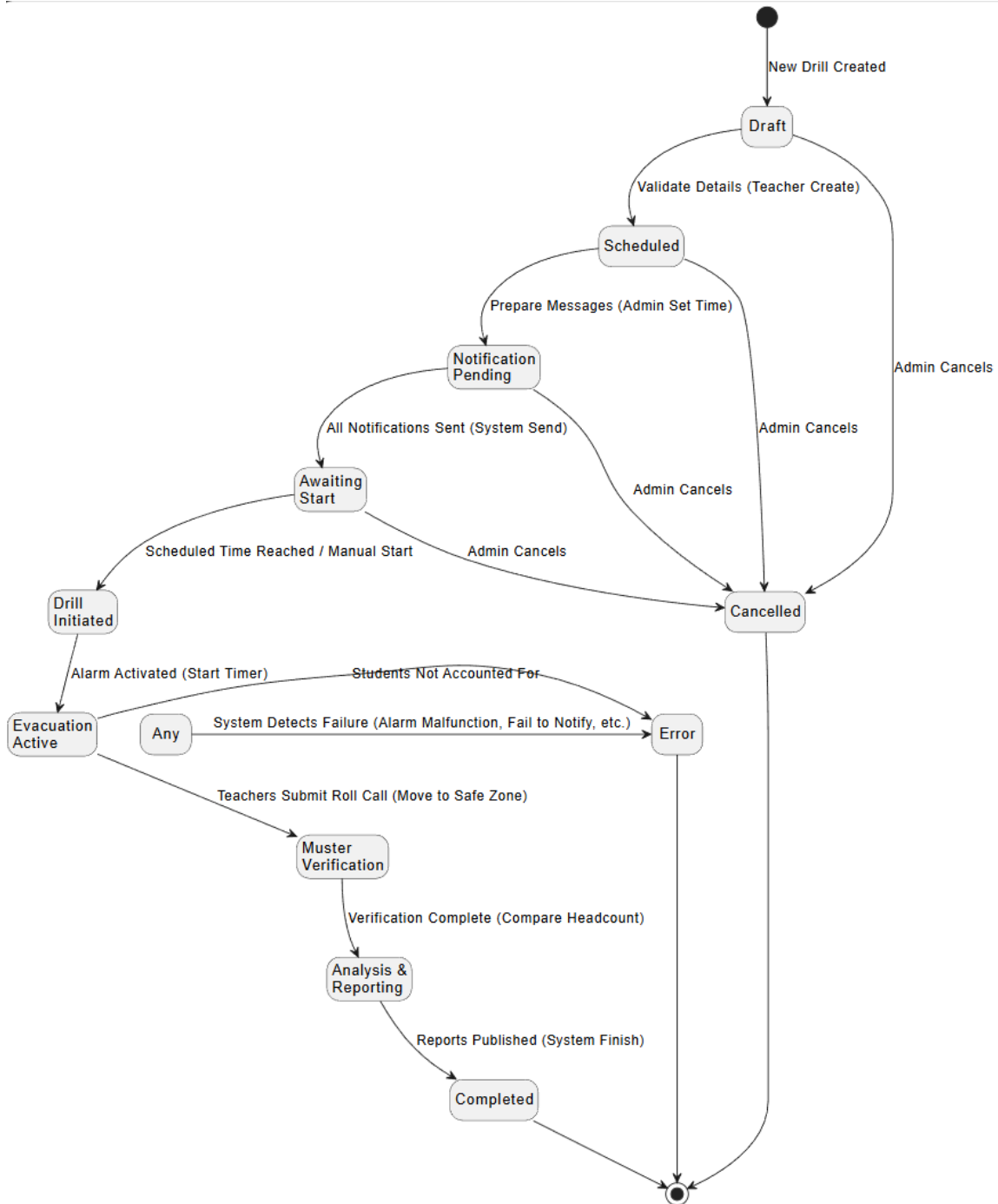
1.10. NotificationSystem

Relevance:

Responsible for sending announcements and alerts to users (students, teachers, admins).This class ensures effective communication during drills, course updates, and emergencies. It improves system responsiveness and user awareness.

2. State Diagram

2.1 Explanation of Each State

2.1.1. Draft

- ❖ This is the initial state of a newly created drill.

- ❖ Basic drill details (date, time, type) are entered.

- ❖ The drill is not active and can be modified or cancelled.

2.1.2. Scheduled

- ❖ The drill has been validated and approved.

- ❖ The date and time are finalized.

- ❖ The system prepares for notification activities.

- ❖ Admin/teacher can still cancel here.

2.1.3. Notification Pending

- ❖ System is preparing and queuing alerts for students, teachers, and staff.

- ❖ Messages might include drill instructions, timings, routes, etc.

2.1.4. Awaiting Start

- ❖ All notifications have been sent.

- ❖ The system is now "waiting" for the scheduled time OR admin-triggered manual start.

- ❖ Drill is ready but not yet active.

2.1.5. Drill Initiated

- ❖ Drill officially begins.
- ❖ Alarm may be activated.
- ❖ Timer starts to track evacuation performance.

2.1.6. Evacuation Active

- ❖ Students and staff begin evacuation.
- ❖ Movement towards safe zones starts.
- ❖ System monitors progress—students' status, sensor alerts, etc.

2.1.7. Any (Decision Point)

- ❖ A branching state where one of two things can happen:
- ❖ Normal evacuation proceeds → Muster Verification
- ❖ A fault/error occurs → Error State

2.1.8. Error

- ❖ Triggered when:
- ❖ Alarm malfunction
- ❖ Notification didn't reach users
- ❖ Students missing
- ❖ System detects abnormal behavior
- ❖ Drill may need corrective action or cancellation.

2.1.9. Muster Verification

- ❖ Teachers verify attendance in safe zones.

- ❖ Roll call is submitted.

- ❖ System compares expected vs. actual headcount.

2.1.10. Analysis & Reporting

- ❖ System analyzes drill performance:

- ❖ Time taken

- ❖ Missing students

- ❖ Route congestion

- ❖ Teacher response

- ❖ Admins generate the final report.

2.1.11. Completed

- ❖ Drill successfully concludes.

- ❖ Reports are published.

- ❖ System resets for next drill.

2.1.12. Cancelled

- ❖ Drill may be cancelled at any point before "Evacuation Active".

- ❖ No further events are processed.
- ❖ Record stored as cancelled in the system.

2.2 Explanation of Each Event / Transition

New Drill Created

- ❖ A teacher/admin initiates a new drill definition.

- ❖ Moves system → Draft.

Validate Details (Teacher Create)

- ❖ Teacher validates drill data.

- ❖ If valid → moves to Scheduled state.

Prepare Messages (Admin Set Time)

- ❖ Admin configures notification content and timing.

- ❖ System moves → Notification Pending.

All Notifications Sent (System Send)

- ❖ System completes sending:

- ❖ Emails

- ❖ SMS alerts

- ❖ App notifications

- ❖ Moves → Awaiting Start.

Scheduled Time Reached / Manual Start

- ❖ Timer indicates the drill start time OR admin clicks "Start Drill".

- ❖ Moves → Drill Initiated.

Alarm Activated (Start Timer)

- ❖ Evacuation alarm rings throughout campus.

- ❖ Evacuation countdown initiated.

- ❖ Moves → Evacuation Active.

Students Not Accounted For

- ❖ System detects missing students via:

- ❖ RFID ID card readers

- ❖ Classroom sensors

- ❖ Attendance mismatch

- ❖ Moves → Error.

System Detects Failure

- ❖ Triggered by:

- ❖ Sensor malfunction

- ❖ Failed alarms

- ❖ System not able to notify users

- ❖ Moves → Error.

Teachers Submit Roll Call

- ❖ After reaching safe zones, teachers submit attendance.

- ❖ Moves → Muster Verification.

Verification Complete (Compare Headcount)

- ❖ System matches expected vs actual roll call.

- ❖ If correct → moves to Analysis & Reporting.

- ❖ If mismatched → could re-trigger error or remain in verification.

Reports   Published   (System   Finish)
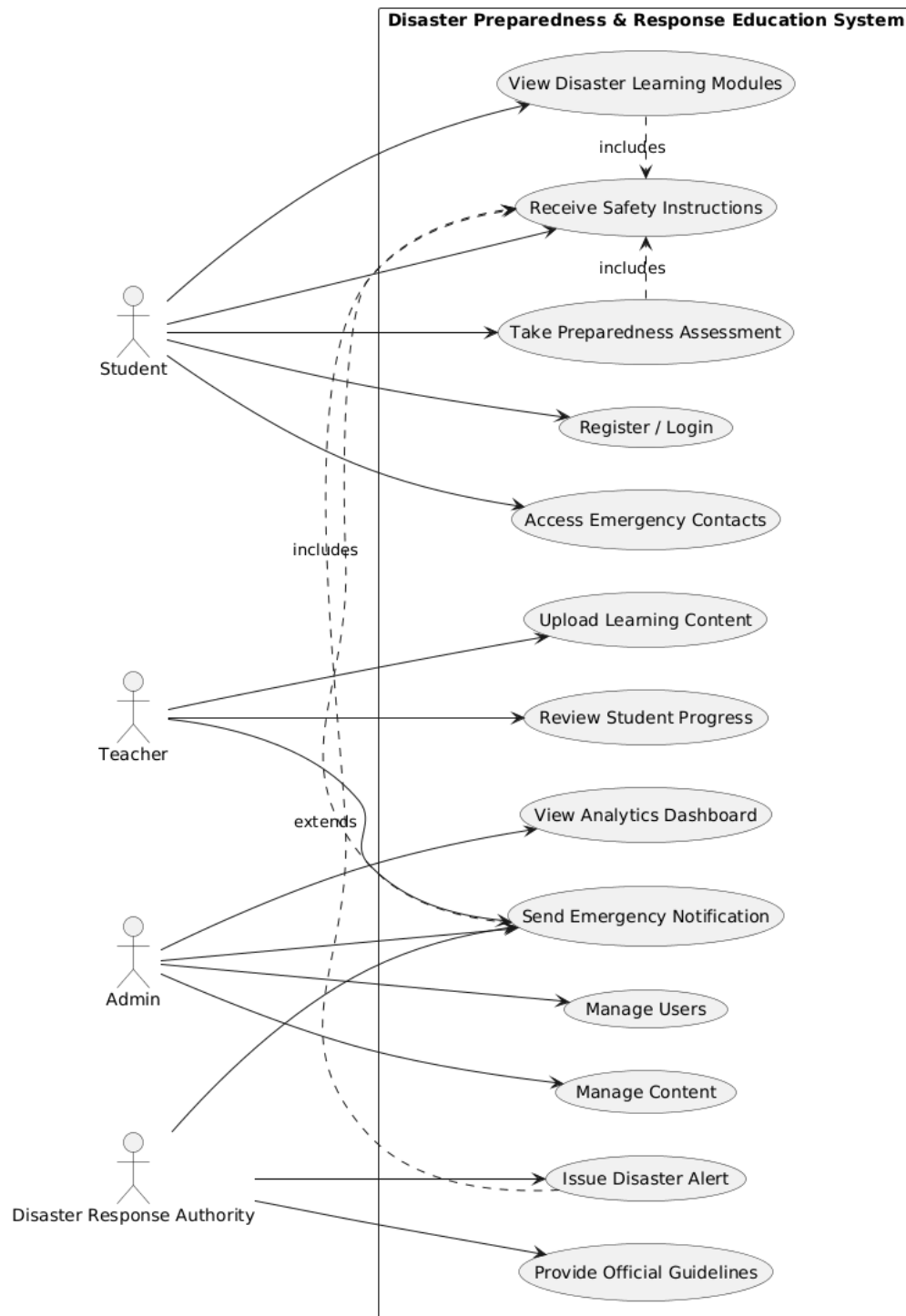
- ❖ Performance reports

- ❖ Evacuation timing

- ❖ Teacher performance

- ❖ Compliance report

- ❖ Moves → Completed.

Admin Cancels

- ❖ Admin may cancel at any point before evacuation.

- ❖ Moves → Cancelled.

Chapter 5: Interaction Modeling

5.1 Use Case Diagram

Disaster Preparedness & Response Education System

Actors: Student, Teacher, Admin, Disaster Response Authority

Use Cases:
- View Disaster Learning Modules
- Receive Safety Instructions
- Take Preparedness Assessment
- Register / Login
- Access Emergency Contacts
- Upload Learning Content
- Review Student Progress
- View Analytics Dashboard
- Send Emergency Notification
- Manage Users
- Manage Content
- Issue Disaster Alert
- Provide Official Guidelines

Relationships: includes, extends

Relevance of Each Actor

Student

Students are the primary learners who need disaster awareness, training, and real-time safety instructions. Their participation directly determines the overall preparedness level of the institution.

Teacher

Teachers provide learning material and guide students in understanding disaster response practices. They also monitor student progress to ensure learning outcomes are achieved.

Admin

Admins manage users, content, and system operations to keep the platform functional and updated. They also coordinate alerts and analytics to improve institutional preparedness.

Disaster Response Authority

Authorities provide verified disaster guidelines and official alerts to ensure accurate information reaches users. Their involvement aligns the system with national safety protocols.

Relevance of Each Use Case

Register / Login

Allows users to securely access the system with role-based permissions.

View Disaster Learning Modules

Provides students with educational content on disaster types and safety practices.

Receive Safety Instructions

Delivers real-time emergency actions and safety alerts during crises.

Take Preparedness Assessment

Evaluates how well students understand disaster response procedures.

Access Emergency Contacts

Gives users quick access to important emergency phone numbers and services.

Upload Learning Content

Enables teachers to add disaster-related educational materials to the system.

Review Student Progress

Allows teachers to track student learning, assessment scores, and preparedness.

View Analytics Dashboard

Shows admins insights about usage, preparedness levels, and system activity.

Send Emergency Notification

Lets admins or teachers broadcast urgent alerts to all users instantly.

Manage Users

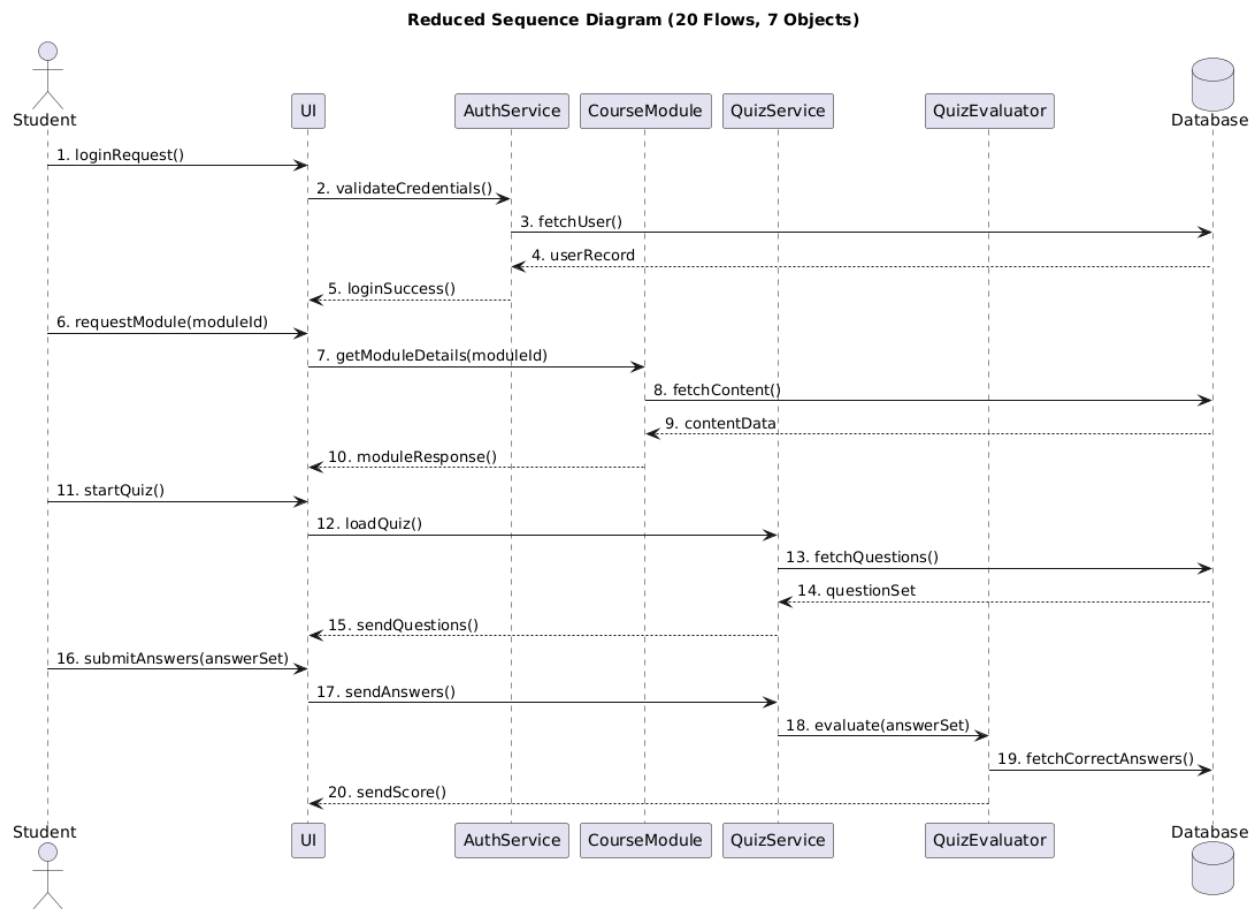Allows admins to add, edit, or remove system users and control access.

Manage Content

Enables admins to update and organize learning modules and safety resources.

Issue Disaster Alert

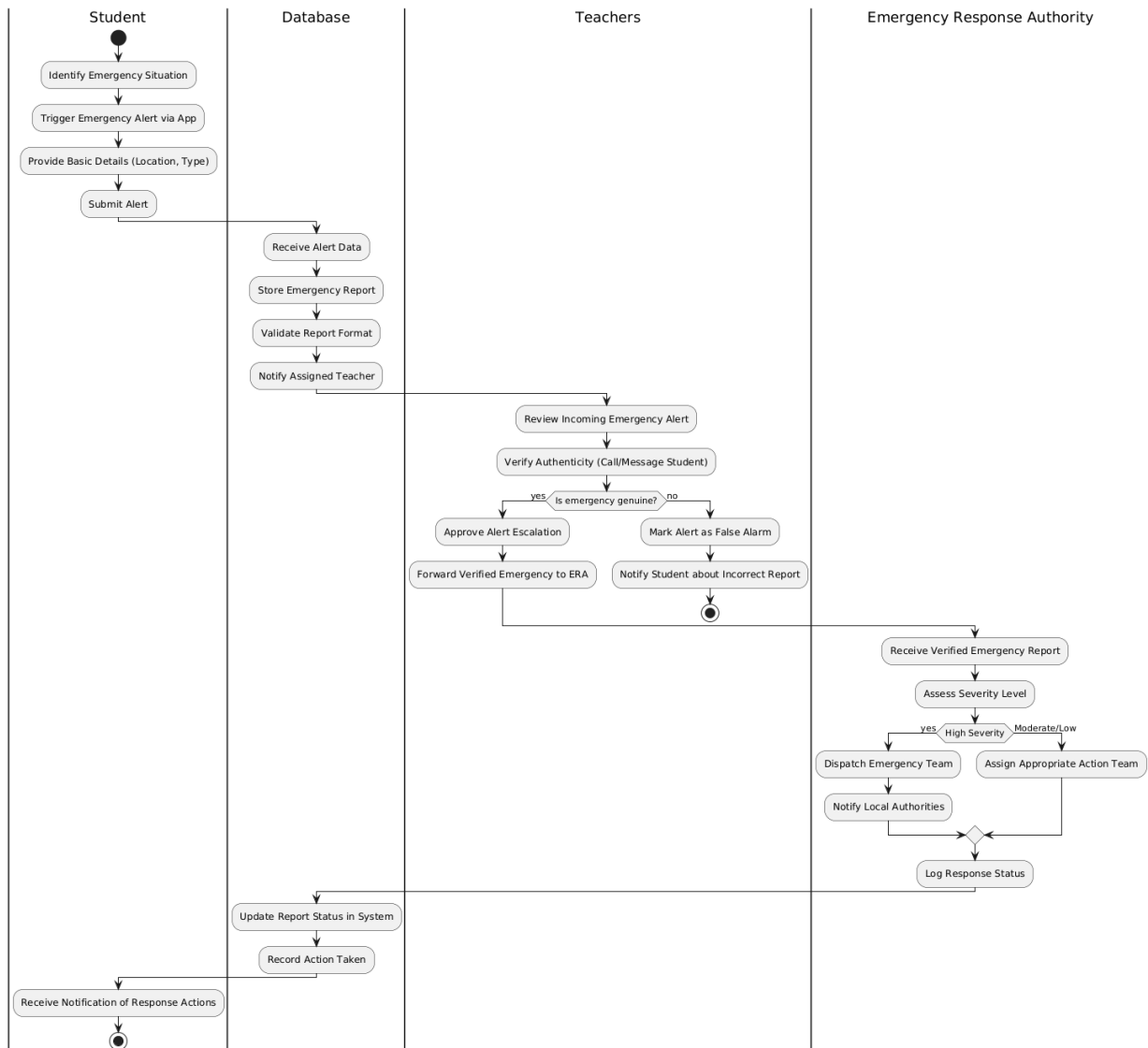Allows authorities to send official warnings or disaster announcements.

5.2 Sequence Diagram

Reduced Sequence Diagram (20 Flows, 7 Objects)

Objects: Student, UI, AuthService, CourseModule, QuizService, QuizEvaluator, Database

1. loginRequest()
2. validateCredentials()
3. fetchUser()
4. userRecord
5. loginSuccess()
6. requestModule(moduleId)
7. getModuleDetails(moduleId)
8. fetchContent()
9. contentData
10. moduleResponse()
11. startQuiz()
12. loadQuiz()
13. fetchQuestions()
14. questionSet
15. sendQuestions()
16. submitAnswers(answerSet)
17. sendAnswers()
18. evaluate(answerSet)
19. fetchCorrectAnswers()
20. sendScore()

Explanation:

The sequence diagram shows how the system handles a student's learning flow, starting from login to accessing a disaster preparedness module. When the student sends a login request, the UI forwards the credentials to the AuthService, which retrieves the user's record from the database and confirms the login. After logging in, the student requests a learning module, prompting the UI to contact the CourseModule service, which fetches the relevant module content from the database and returns it for display.

The second part of the diagram focuses on the quiz interaction. When the student chooses to start the quiz, the UI loads the quiz by requesting questions from the QuizService, which pulls them from the database. After the student submits answers, the UI sends them to the QuizEvaluator, which retrieves the correct answers, evaluates the submission, and sends the final score back to the UI.

5.3 Activity Diagram

Student

- Identify Emergency Situation
- Trigger Emergency Alert via App
- Provide Basic Details (Location, Type)
- Submit Alert
- Receive Notification of Response Actions

Database

- Receive Alert Data
- Store Emergency Report
- Validate Report Format
- Notify Assigned Teacher
- Update Report Status in System
- Record Action Taken

Teachers

- Review Incoming Emergency Alert
- Verify Authenticity (Call/Message Student)
- Is emergency genuine?
  - yes → Approve Alert Escalation → Forward Verified Emergency to ERA
  - no → Mark Alert as False Alarm → Notify Student about Incorrect Report

Emergency Response Authority

- Receive Verified Emergency Report
- Assess Severity Level
- High Severity
  - yes → Dispatch Emergency Team → Notify Local Authorities
  - Moderate/Low → Assign Appropriate Action Team
- Log Response Status

The activity diagram outlines a school emergency alert system with four roles: Student, Database (system), Teachers, and Emergency Response Authority (ERA). It starts when a student witnesses an emergency (e.g., fire, injury, or intruder) and immediately uses a mobile app to trigger an alert, entering critical details like location and type of incident before submitting it.

The system instantly receives and stores the report, validates its format, and notifies the teacher assigned to that area or class. The teacher promptly reviews the alert, contacts the student (via call/message) to confirm authenticity, and decides: if it's a false alarm, they mark it as such and inform the student; if genuine, they approve escalation and forward the verified report to the ERA.

Upon receiving the confirmed report, the ERA begins coordinated response: two actions start simultaneously — assessing the severity level to decide whether to dispatch a full emergency team or take lighter action, and logging the response status for records. Only after both parallel tasks are completed does the flow merge again, updating the system with final actions taken. This design ensures quick reporting, teacher-level verification to reduce false alarms, and efficient, documented handling by authorities.
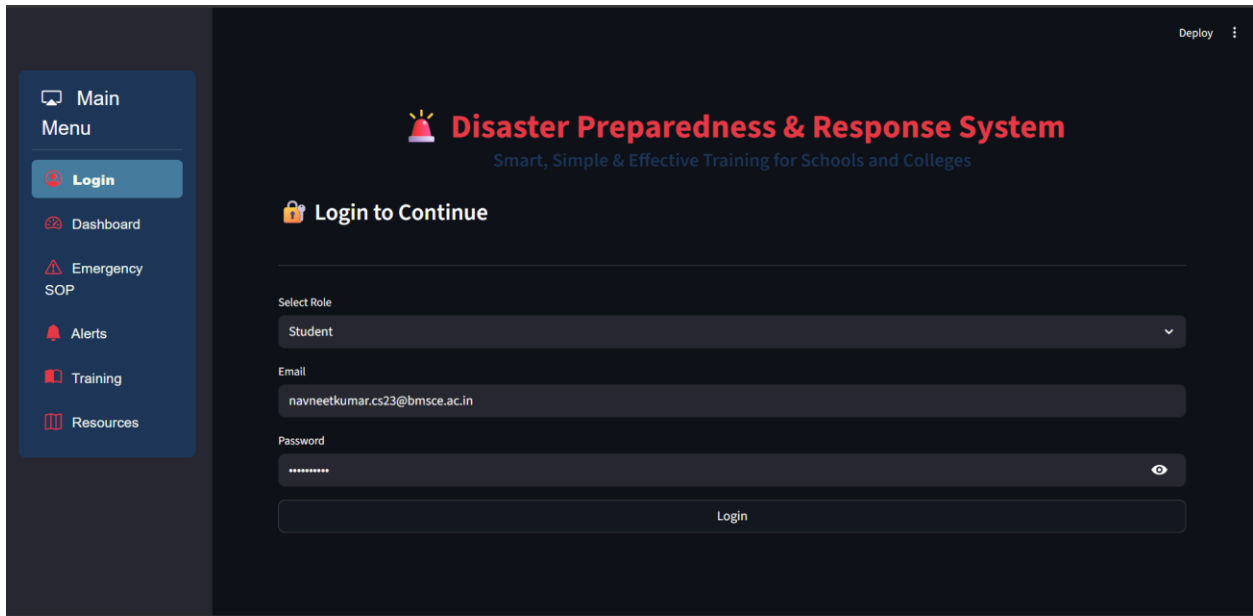
Chapter 6: UI Design with Screenshots

Fig 6.1 Login page

- ❖ A secure login interface for a school-focused disaster preparedness system, offering role-based access and training tools. [Fig 6.1]
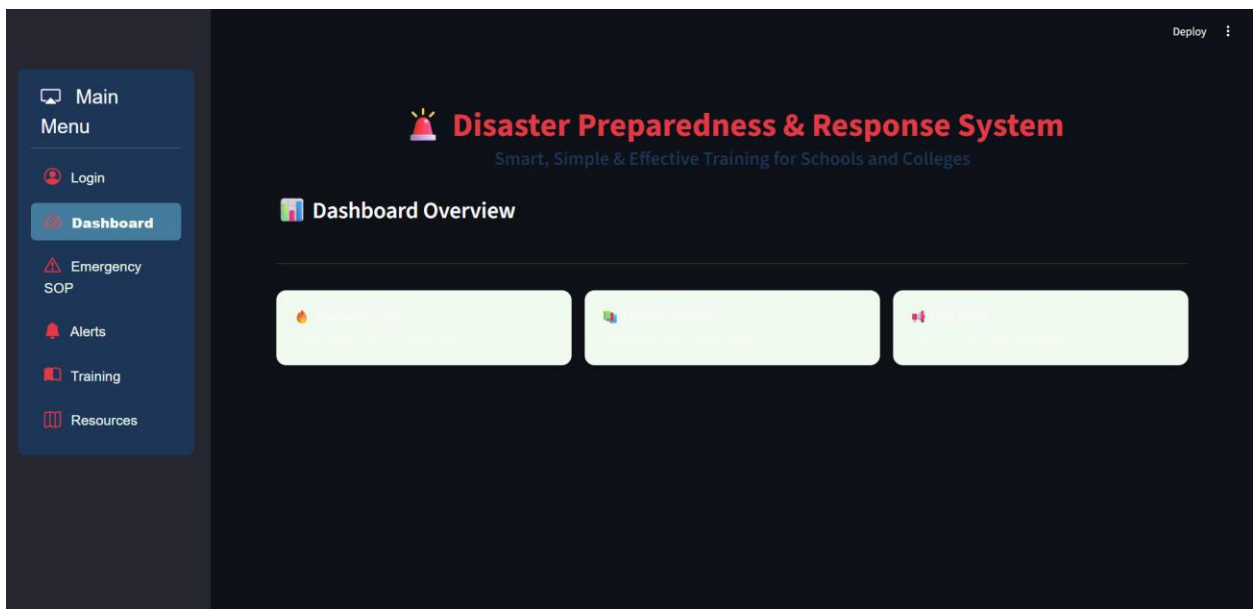

Fig 6.2 Dashboard

- ❖ A sleek dashboard interface for a school disaster preparedness system, offering quick access to SOPs, alerts, training, and resources. [Fig 6.2]
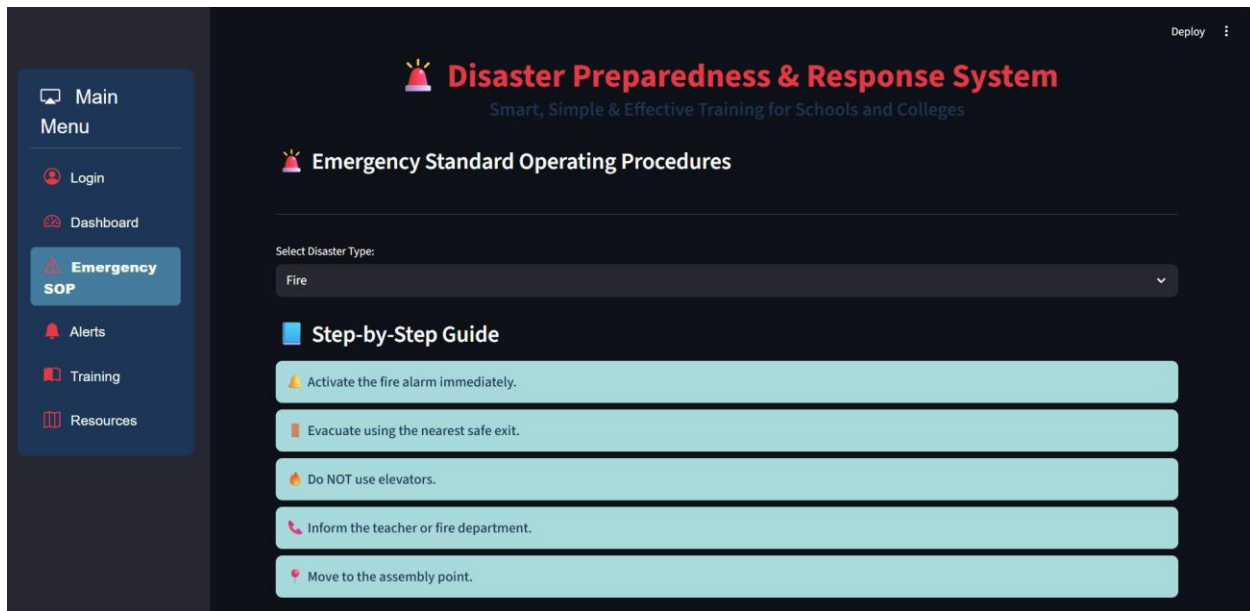
Fig 6.3 Emergency SOPs

- ❖ A clear SOP interface for school disaster response, guiding users through step-by-step emergency actions based on disaster type. [Fig 6.3]
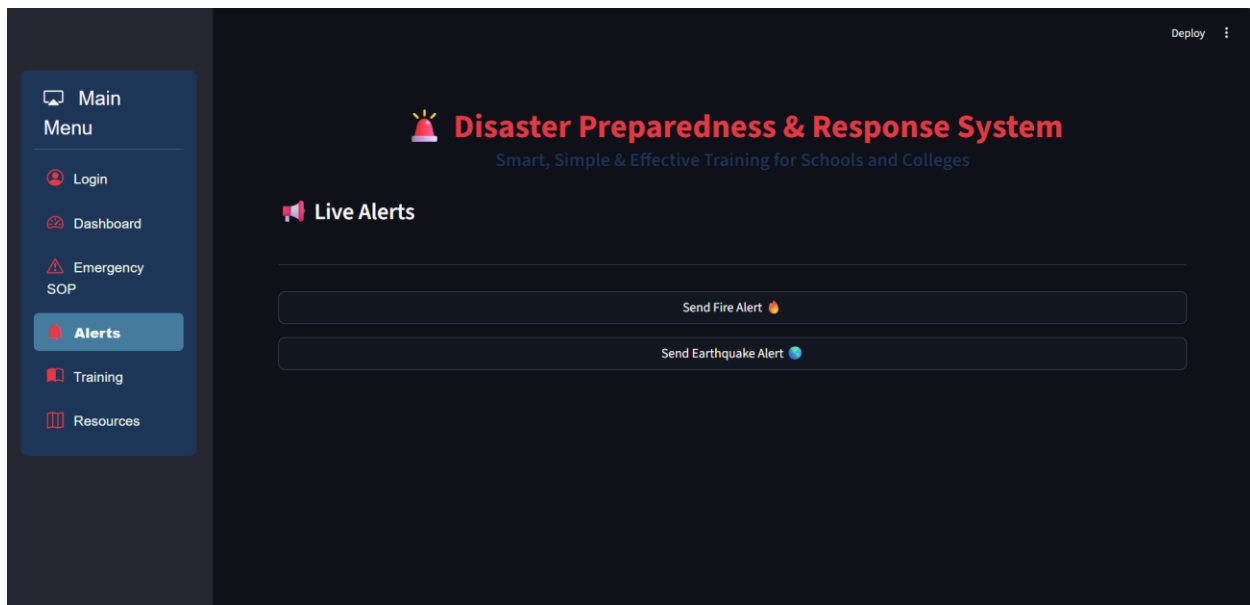
Fig 6.4 Alert Page

- ❖ A fast-response alert interface for school disaster management, enabling instant fire or earthquake notifications. [Fig 6.4]
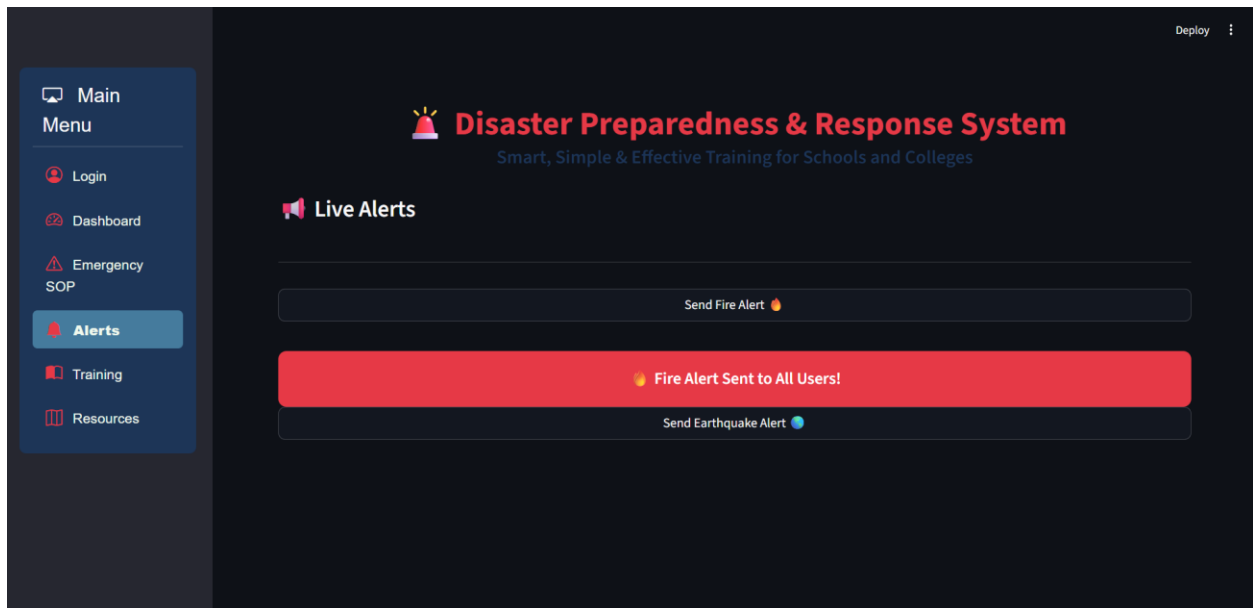
Fig 6.5 Fire Alert Sent to All Users

❖ A real-time alert system for schools, confirming fire notifications have been sent while offering fire alert options. [Fig 6.5]
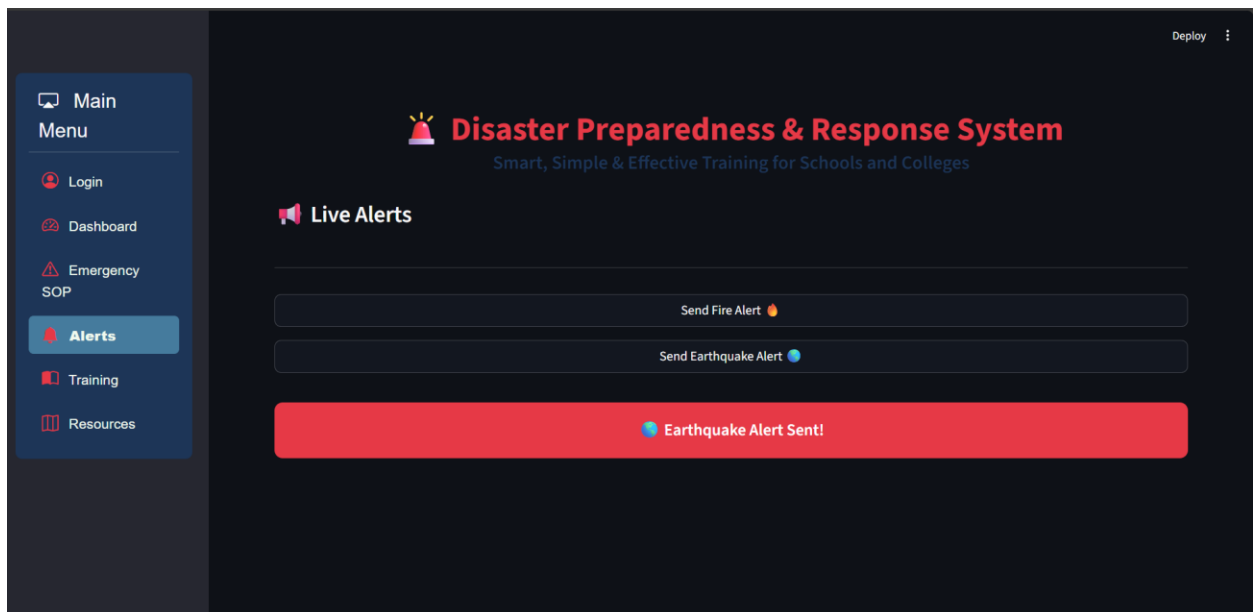

Fig 6.6 Earthquake Alert Sent to All Users

❖ A responsive alert interface for school emergencies, confirming an earthquake alert has been sent to all users. [Fig 6.6]
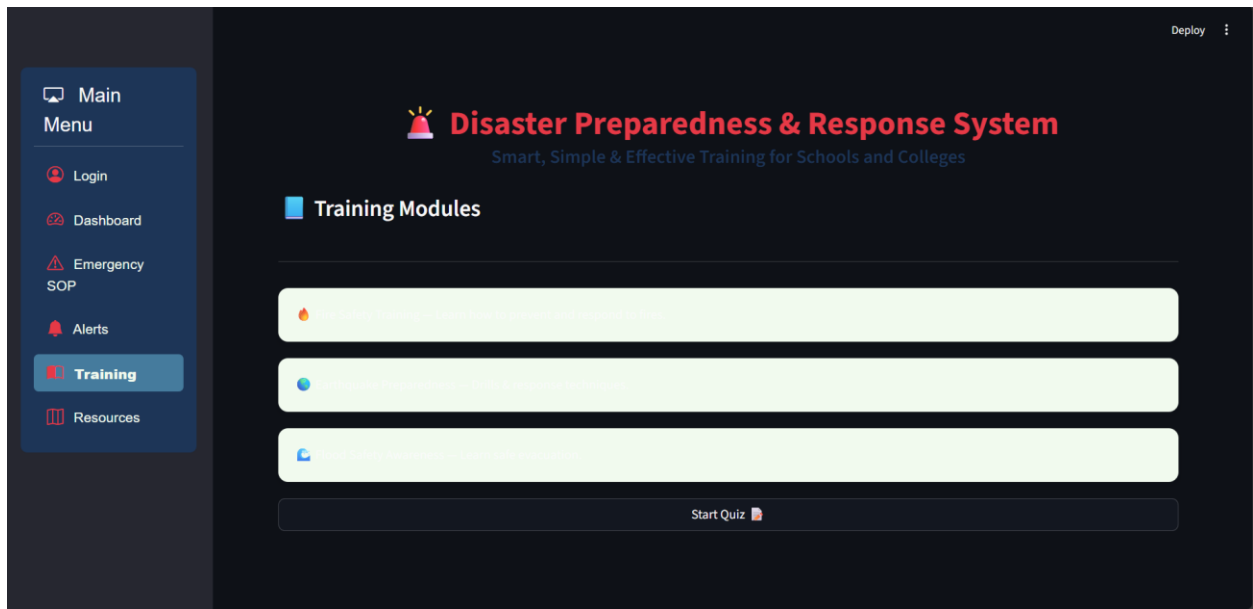
Fig 6.7 Training Modules Page

❖ An interactive training interface for school disaster readiness, offering fire, flood, and earthquake safety modules with a quiz. [Fig 6.7]

Fig 6.8   Resources & Evacuation Maps

❖ A resource interface for school disaster planning, offering evacuation maps for building and assembly areas. [Fig 6.8]