1. Introduction

Overview of the System

Durjog Prohori (দুর্যোগ প্রহরী) is a comprehensive disaster management web application developed to support real-time disaster monitoring, coordinated emergency response, and effective preparedness in Bangladesh. Designed to bring together key stakeholders—such as authorities, emergency responders, NGOs, and citizens—onto a single digital platform, the system ensures timely communication, accurate information sharing, and swift decision-making during crises. With features like early warning alerts, live disaster tracking, interactive maps, and citizen reporting tools, Durjog Prohori strengthens the country's disaster response framework and helps minimize the impact of both natural and human-induced disasters.

Purpose and Goals

- **Create a centralized platform for disaster management and emergency response**
  The system aims to bring together all stakeholders—such as government agencies, NGOs, fire station, emergency responders, and citizens—onto a unified digital platform. This centralization reduces fragmented communication and delays, enabling streamlined coordination during crisis events.
- **Enable real-time monitoring and reporting of various types of disasters**
  By incorporating live data feeds, geolocation services, and user-submitted reports, the system ensures that ongoing disasters are tracked in real time. This allows responders to act quickly and with accurate situational awareness, improving the efficiency of emergency operations.
- **Facilitate coordination between different emergency response stakeholders**
  The platform includes role-based dashboards, shared communication channels, and a unified resource management interface. These features promote synchronized actions between organizations, minimize redundant efforts, and improve overall disaster response logistics.
- **Provide citizens with immediate access to emergency services and information**
  Through an intuitive and mobile-responsive interface, users can receive emergency alerts, request help via SOS features, and access verified disaster-

related updates. This ensures that help is just a few clicks away during life-threatening situations.

- **Support multilingual access (English and Bengali) for wider accessibility**
  Recognizing the linguistic diversity of Bangladesh, the system supports both English and Bengali. This makes the platform inclusive, ensuring that users from all educational backgrounds can understand and interact with the system effectively during emergencies.

Intended Users

1. **Citizens/General Public**
   The platform is designed to serve the everyday needs of the general population during disaster events. Citizens can report incidents, send SOS alerts, receive timely notifications, and access critical emergency instructions. The user-friendly interface ensures that even those with limited technical knowledge can use the system effectively.
2. **Emergency Response Teams**
   These are frontline personnel responsible for executing on-ground relief and rescue operations. The system provides them with real-time data, task assignments, resource tracking, and location-based alerts to ensure a quick and efficient response.

   ● **Firefighters**
   Firefighters use the system to respond to fire-related emergencies, track high-risk zones, and receive notifications about active SOS requests that may require immediate intervention.

   ● **NGOs**
   Non-Governmental Organizations contribute by offering humanitarian aid, supplies, and manpower. The platform allows them to receive donation allocations, manage relief distribution, and coordinate with authorities and other NGOs for broader impact.

   ● **Authorities**
   Government and municipal bodies play a vital role in decision-making, resource allocation, and policy enforcement. Through the system, they monitor all incoming disaster data, assign responsibilities, and track the overall disaster response progress.

3. **System Administrators**
   These users are responsible for maintaining the platform's performance, data integrity, and security. They manage user accounts, configure system settings, and ensure the system runs reliably, especially during high-demand periods.
4. **Disaster Management Officials**
   Officials from national or regional disaster management agencies use the system to implement disaster preparedness plans, evaluate threat levels, coordinate with response teams, and oversee post-disaster assessments and recovery efforts.

Scope and Limitations

- **Geographic Focus: Primary focus on Bangladesh**
  The system is tailored specifically for the geographical and environmental conditions of Bangladesh, a country frequently affected by natural disasters. The platform's design considers regional infrastructure, local governance structures, and the specific needs of Bangladeshi communities to ensure maximum effectiveness.
- **Supported Disasters: Earthquakes, Floods, Cyclones, Landslides, Tsunamis, Fires**
  Durjog Prohori covers a wide range of both sudden and slow-onset disasters. By supporting these major disaster types, the system can cater to the most common and destructive events affecting the country, providing relevant tools for early warning, reporting, and response coordination.
- **Platform: Web-based application with mobile responsiveness**
  The system is accessible through any modern web browser and adapts seamlessly to smartphones, tablets, and desktops. This ensures that users from urban and rural areas alike can interact with the platform regardless of their device type.
- **Authentication: Role-based access control**
  Each user type is granted specific access rights depending on their role. For example, citizens can report incidents and request help, while authorities have access to administrative tools and coordination dashboards. This structure enhances security and ensures that users only see features relevant to their responsibilities.
- **Language Support: Bilingual (English and Bengali)**
  The interface and content are available in both English and Bengali, ensuring inclusivity and usability for a broader range of users. This is especially critical in emergencies, where clear and understandable communication can save lives.

2. Problem Statement / Rationale

Bangladesh is highly vulnerable to natural disasters due to its geographical location and climate conditions. It frequently experiences floods, cyclones, earthquakes, and landslides, causing significant damage to lives and infrastructure. Despite the recurring nature of these events, the current disaster management infrastructure still lacks the efficiency and integration needed for swift, coordinated responses.

- **Lack of real-time disaster information dissemination**
  One of the most critical shortcomings is the absence of systems that can relay disaster information to all relevant parties instantly. Without real-time alerts and updates, both citizens and response teams face delays in taking appropriate action, which can cost lives and resources.
- **Poor coordination between different emergency response teams**
  Emergency responders—including firefighters, medical teams, and NGOs—often operate independently, leading to overlapping efforts or critical gaps in action. The lack of a shared platform or unified communication channel makes collaboration difficult during high-stress disaster situations.
- **Difficulty in tracking and managing emergency resources**
  There is no centralized system for monitoring the availability and allocation of essential resources such as rescue equipment, medical supplies, and shelters. This results in inefficient resource utilization and delays in delivering aid to affected areas.
- **Language barriers in emergency communication**
  Many citizens may not understand emergency messages if they are not delivered in their native language. In a country where Bengali is predominantly spoken, reliance on English-only systems can exclude a significant portion of the population during crises.
- **Need for immediate emergency response coordination**
  Disasters require rapid and well-coordinated responses to minimize damage. However, the current system lacks the technological framework to support real-time decision-making and synchronized deployment of emergency services, creating bottlenecks during crucial moments.

3. Information Gathering and System Study

Methods

- Analysis of existing disaster management systems
- Study of emergency response protocols
- Review of stakeholder requirements
- Technical feasibility assessment

Sources

- Government disaster management protocols
- Emergency response best practices
- User feedback and requirements
- Similar disaster management systems

Key Findings

- Need for real-time disaster tracking
- Importance of multilingual support
- Requirement for role-based access control
- Necessity of map-based visualization
- Critical need for mobile responsiveness

Survey Questions & Result Summary:

1. Have you or someone you know ever faced a natural disaster?

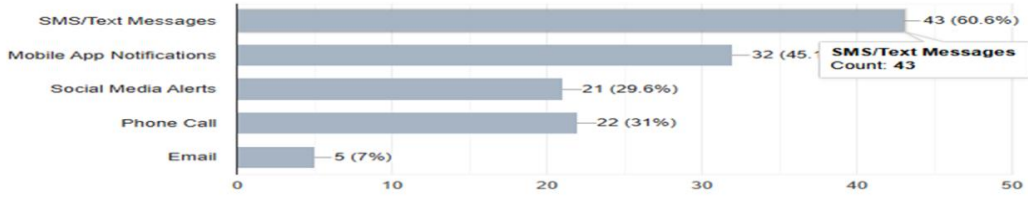
Copy chart

71 responses

2. If you could receive real-time disaster alerts, how would you prefer to get them?

Copy chart

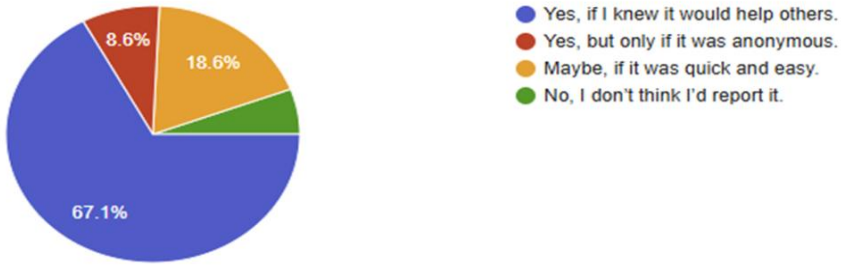
71 responses



5. If you saw a disaster happening (e.g., a flood, fire, or building collapse), would you report it on an app?

Copy chart

70 responses



8. If an app could help you request emergency aid (food, shelter, medical help), what would be most important?

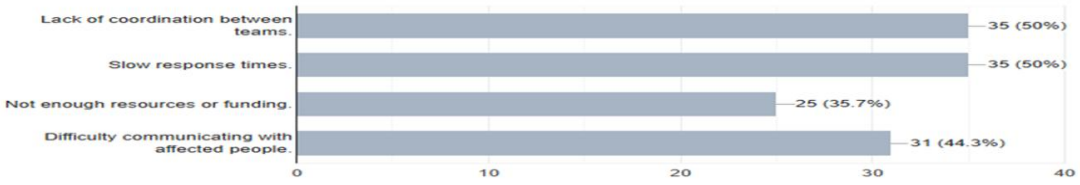
Copy chart

70 responses

9. What's the biggest problem government agencies face during disasters?

Copy chart

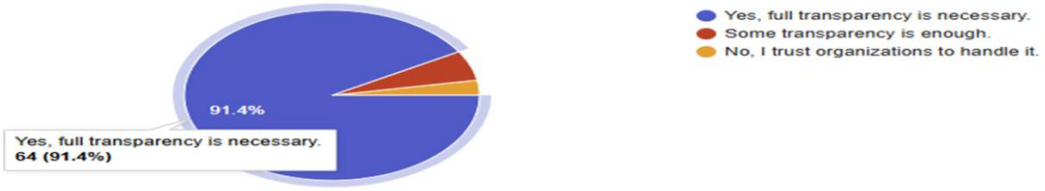
70 responses



11. Should disaster relief funds be more transparent?

Copy chart

70 responses



12. Would you be more likely to volunteer in a disaster if you could do it through an app?

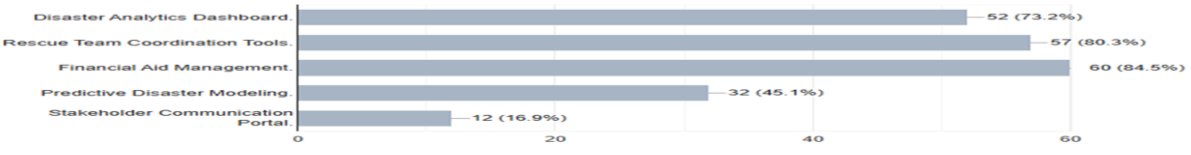
Copy chart

70 responses

14. Which features would you prioritize for government agencies?(Select Top 3)

Copy chart

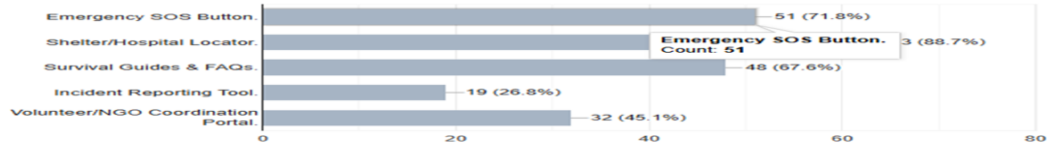
71 responses

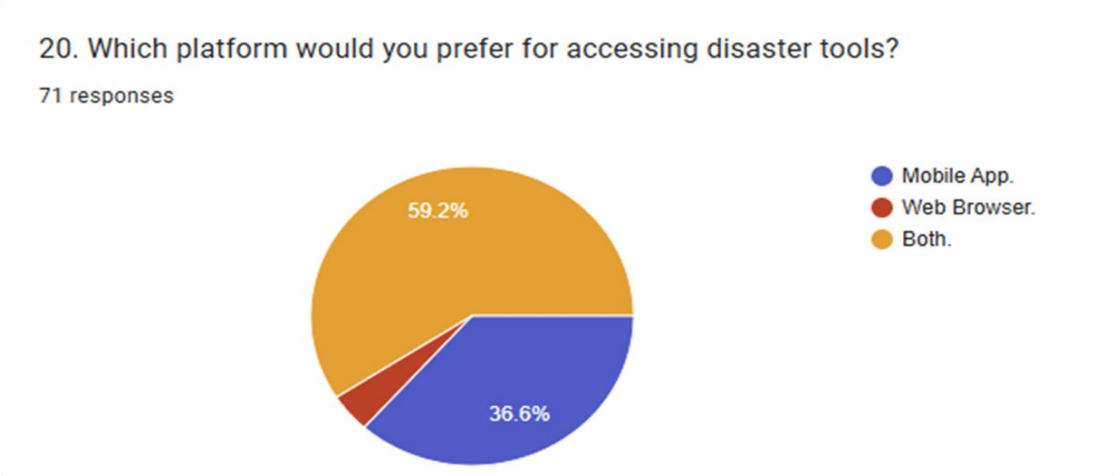


15. For citizens, which tools are most critical during a disaster? (Select Top 3)

Copy chart

71 responses





4. Feasibility Analysis

SWOT Analysis

Strengths:

1. **Real-time disaster tracking and reporting:**
   The system allows immediate tracking and reporting of disasters, ensuring quick information dissemination and timely responses, which is critical in emergency scenarios.
2. **Multi-stakeholder coordination:**
   It facilitates collaboration between various agencies like government bodies, NGOs, rescue teams, and healthcare providers, leading to more organized and efficient disaster response efforts.
3. **Bilingual support:**
   Offering services in multiple languages (e.g., Bengali and English) ensures better accessibility and inclusivity, especially in multilingual communities.
4. **Role-based access control:**
   Secure access to system functionalities based on user roles (e.g., admin, responder, viewer) improves data privacy and system integrity while reducing the chances of unauthorized usage.
5. **Map-based visualization:**
   Geospatial data representation through maps helps stakeholders to visualize the disaster-affected areas, resources, and routes, aiding in better decision-making and resource allocation.

Weaknesses:

1. **Internet dependency:**
   The system relies heavily on internet connectivity, which can be unreliable or unavailable during natural disasters, affecting its usability when it is needed the most.
2. **Limited offline functionality:**
   Without strong offline capabilities, field workers or affected users in remote areas may face difficulty accessing or updating crucial data during emergencies.
3. **Initial user adoption challenges:**
   New systems often face resistance or confusion among users unfamiliar with digital platforms. Training and awareness programs may be necessary to overcome this hurdle.

Opportunities:

1. **Integration with other emergency services:**
   The platform can be connected with healthcare, fire, police, and weather alert systems for a unified and more efficient disaster response ecosystem.
2. **Expansion to other regions:**
   After successful deployment in one region, the system can be scaled and customized to serve other disaster-prone areas both nationally and internationally.
3. **Addition of more disaster types:**
   Initially focusing on common disasters, the system can later include other types (e.g., epidemics, industrial accidents) to increase its relevance and usability.
4. **Mobile app development:**
   Developing a dedicated mobile app would enhance user accessibility, especially in field operations, and encourage wider adoption among the general public.

Threats:

1. **System reliability during disasters:**
   Server downtimes, software bugs, or power outages during disasters could cripple the system at critical times, leading to serious consequences.
2. **Data security concerns:**
   Handling sensitive location and user data makes the system a potential target for cyberattacks. Ensuring robust cybersecurity measures is essential.
3. **Network infrastructure limitations:**
   In areas with weak telecommunications infrastructure, system performance and accessibility may suffer, especially during large-scale disasters.

4. **User resistance to change:**
   Traditional disaster management personnel may be hesitant to adopt new technology, preferring manual methods they are already comfortable with.

5. Requirement Specification

Functional Requirements

1. User Management
    a. User registration and authentication
    b. Role-based access control
    c. Profile management
    d. Password recovery
2. Disaster Management
    a. Real-time disaster reporting
    b. Multiple disaster type support
    c. Disaster status tracking
    d. Location-based disaster mapping
3. Emergency Response
    a. SOS alert system
    b. Resource allocation
    c. Team assignment
    d. Response tracking
4. Communication
    a. Real-time notifications
    b. Multi-language support
    c. Emergency contact management
    d. Status updates
5. Reporting
    a. Incident reports
    b. Response reports
    c. Resource utilization reports
    d. Historical data analysis

Non-Functional Requirements

1. Performance
    a. Response time < 3 seconds
    b. Support for multiple concurrent users
    c. Real-time data updates
    d. Map rendering optimization
2. Security
    a. Encrypted data transmission
    b. Secure authentication
    c. Role-based access control
    d. Data backup and recovery
3. Usability
    a. Intuitive interface
    b. Mobile responsiveness
    c. Bilingual support
    d. Accessibility features
4. Reliability
    a. 99.9% uptime
    b. Data consistency
    c. Error handling
    d. Backup systems

6. Feature List Fixation

Main Features

1. Real-time Disaster Tracking
2. Interactive Map Interface
3. Multi-stakeholder Coordination
4. Emergency Response Management
5. Resource Allocation System
6. SOS Alert System
7. Bilingual Support
8. Reporting and Analytics
9. User Management

10. Notification System

Future Works

1. **Mobile Application Development:**
   Creating a dedicated mobile app will enhance user accessibility and convenience, allowing stakeholders and the public to receive alerts, report incidents, and access disaster-related information on-the-go, especially in field operations.
2. **Offline Mode Support:**
   Implementing offline functionality will ensure that critical features (like data entry and viewing important instructions) remain available even in low or no internet connectivity zones—common during disasters.
3. **AI-powered Disaster Prediction:**
   Integrating artificial intelligence models can enable the system to predict potential disasters based on historical data, sensor inputs, and environmental signals. This will allow authorities to take preventive actions before a disaster occurs.
4. **Integration with Weather Services:**
   Real-time linkage with meteorological departments or weather APIs will allow automatic updates and alerts about severe weather conditions, improving preparedness and early warning mechanisms.
5. **Extended Regional Support:**
   The system will be expanded to support more geographic regions with region-specific disaster profiles, language preferences, and localized resource mapping to ensure broader and more inclusive coverage.
6. **Advanced Analytics Dashboard:**
   A more detailed and interactive dashboard with data visualization, trend analysis, and performance metrics will empower decision-makers to monitor disaster trends, evaluate response effectiveness, and optimize strategies.
7. **Community Forum Integration:**
   A built-in community or public forum will allow users to share local updates, tips, and resources. It will also promote engagement, awareness, and crowd-sourced intelligence during and after disaster events.
8. **Resource Optimization Algorithm:**
   Developing intelligent algorithms to allocate and dispatch resources (like rescue teams, food, and medical aid) based on real-time needs and availability will enhance the efficiency and speed of disaster response operations.

7. System Design

Used Technologies and Description

The disaster management system is built using the **MERN stack**, a robust and scalable JavaScript-based framework comprising **MongoDB**, **Express.js**, **React.js**, and **Node.js**. This full-stack architecture allows for a seamless, real-time, and responsive user experience, ideal for managing critical disaster-related information.

Frontend – React.js

- **Technology:** React.js (JavaScript library)
- **Purpose:** Builds the user interface (UI) of the application.
- **Features:**
    - Responsive and dynamic web interface for users (e.g., admins, responders, general public).
    - Component-based architecture allows reusable UI elements like alert cards, map modules, and report forms.
    - Integrates with external APIs (e.g., weather services, location tracking).
    - Offers bilingual support and map-based visualizations.
    - Utilizes state management (e.g., Redux or Context API) for efficient data handling.
    - Ensures fast updates and smooth navigation using React Router.

Backend – Node.js and Express.js

- **Technology:** Node.js (JavaScript runtime) and Express.js (backend web framework)
- **Purpose:** Handles server-side logic, API endpoints, and business processes.
- **Features:**

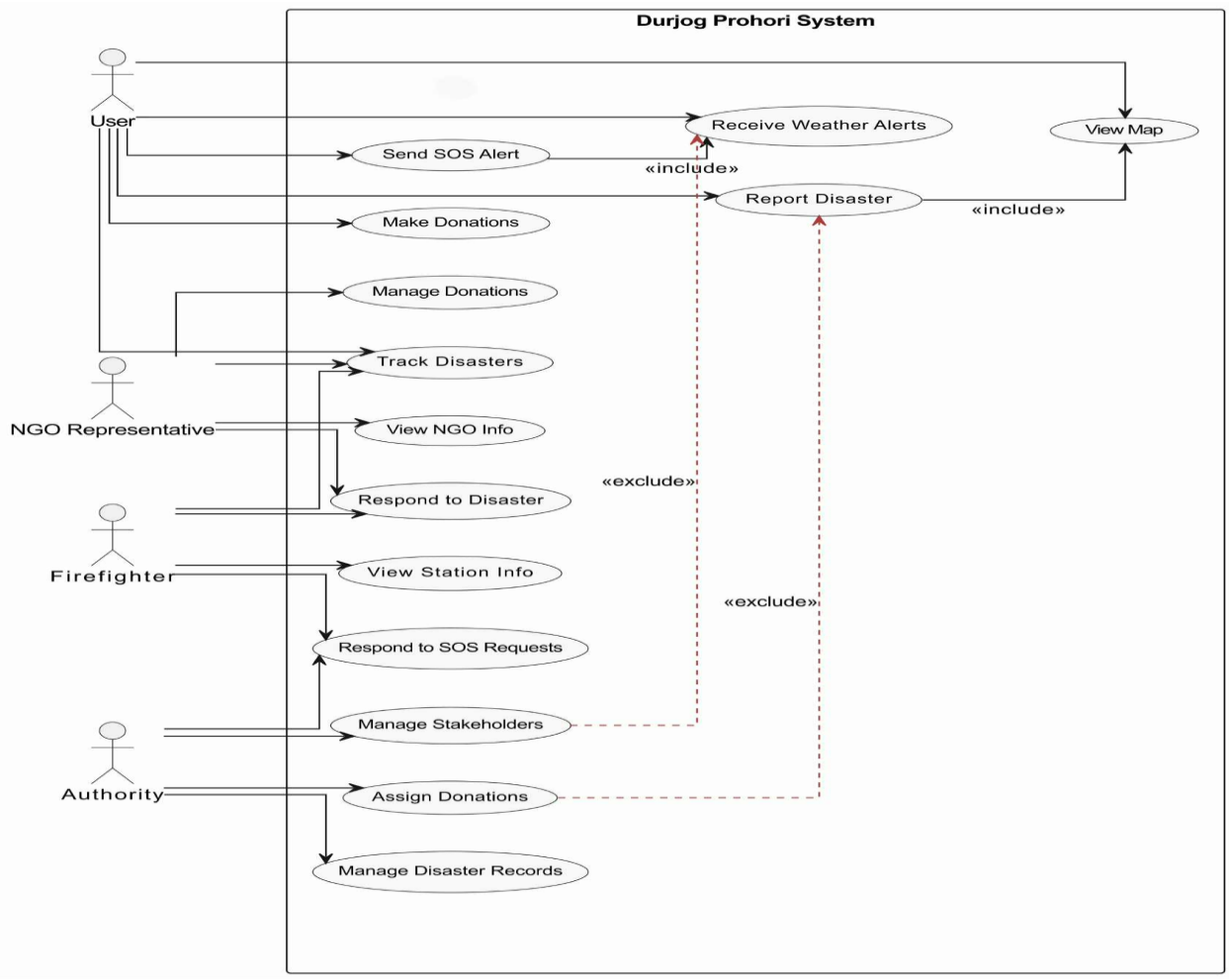
    1. Manages RESTful APIs for communication between frontend and database.
    2. Implements user authentication, role-based access control, and session management.
    3. Handles real-time notifications and data synchronization (e.g., via WebSockets or third-party services).
    4. Integrates with external services such as weather APIs and emergency contacts.
    5. Ensures modular and secure route management for different user roles (e.g., admin, responder, viewer).

Database – MongoDB

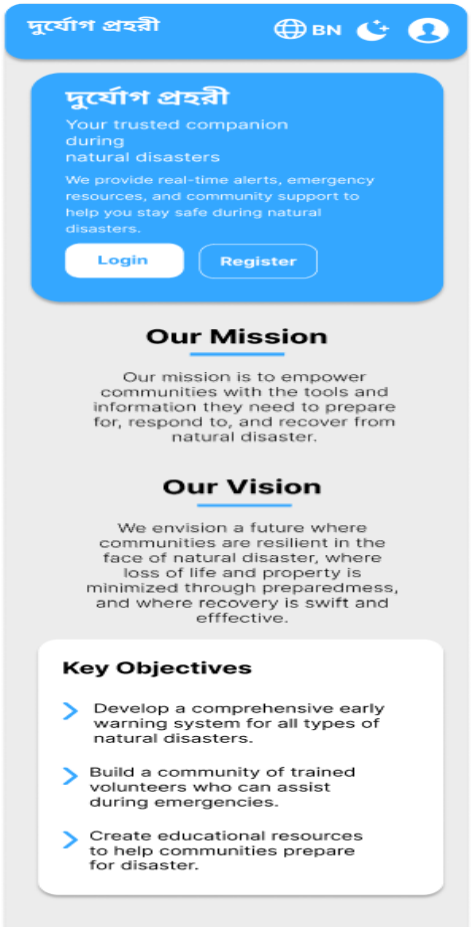
- **Technology:** MongoDB (NoSQL document-oriented database)
- **Purpose:** Stores and manages structured and unstructured data.
- **Features:**

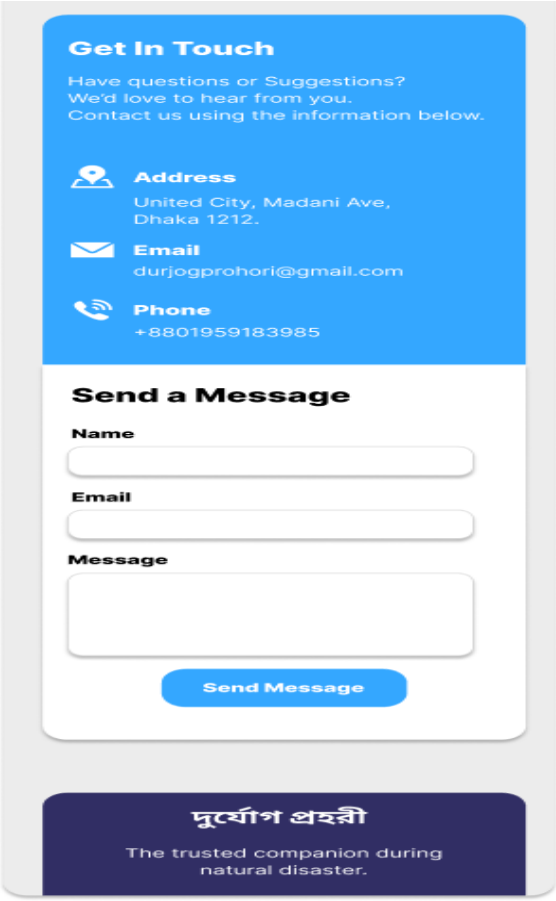
1. Stores disaster reports, user profiles, roles, resource data, and historical records.
2. Flexible schema allows easy addition of new disaster types and user roles.
3. High scalability and performance for handling large volumes of real-time data.
4. Backup and replication features ensure data reliability during system failures.

System Design:

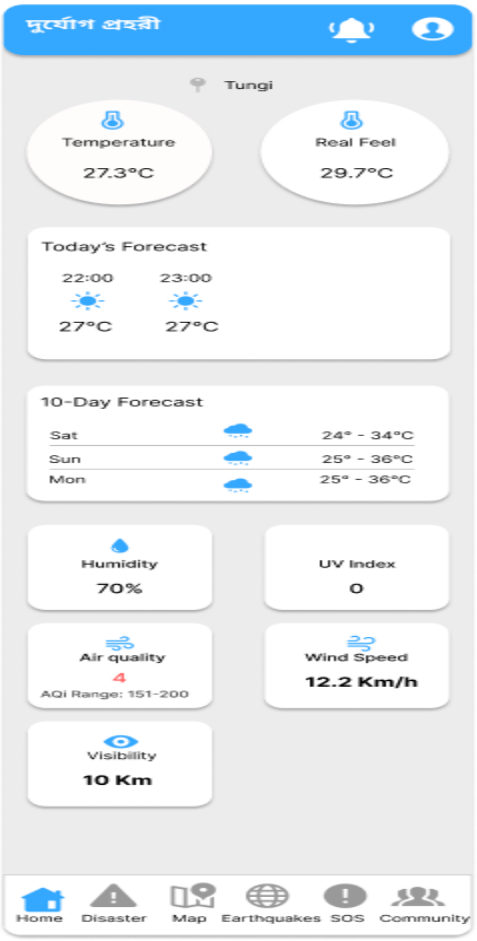


8. User Interface (UI) Design

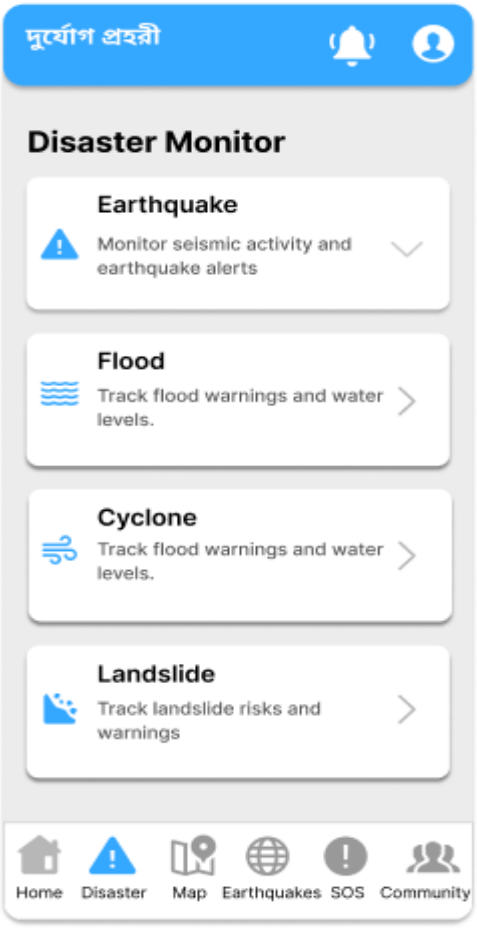




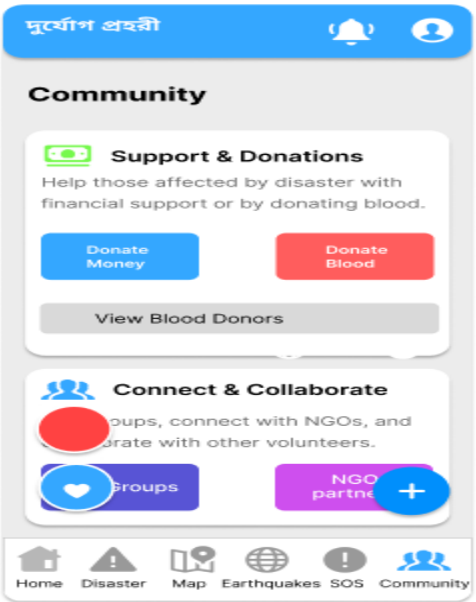
1. **Landing page**



2. Home



3. Disaster Monitor



4. Community

5. Emergency SOS

9. Implementation Plan

Phase 1: Core Development (3 weeks)

- System architecture setup
- Database design
- Basic user management
- Authentication system

Phase 2: Feature Implementation (3 weeks)

- Disaster management module
- Emergency response system
- Resource management
- Map integration

Phase 3: Enhancement (2 weeks)

- Multi-language support
- Reporting system
- Analytics dashboard
- Performance optimization

Phase 4: Testing and Deployment (1 weeks)

- System testing
- User acceptance testing
- Performance testing
- Production deployment

10. Faced Difficulties

1. Leaflet Map API Integration

- **Issue:** Difficulty in rendering real-time location data and managing dynamic updates on the map.
- **Solution:** Optimized state management and used proper lifecycle methods to ensure smooth data syncing and rendering.

2. Weather API Fetching

- **Issue:** API rate limits, data formatting issues, and inconsistent updates caused problems in fetching real-time weather data.
- **Solution:** Implemented error handling, caching, and localized formatting for better reliability.

3. Responsive Design

- **Issue:** UI components broke or didn’t scale properly on mobile and tablet screens.
- **Solution:** Redesigned layouts using mobile-first CSS techniques and media queries to ensure full responsiveness.

4. Payment System

- **Issue:** Third-party payment gateways were hard to integrate due to API restrictions and testing issues.
- **Solution:** Developed a custom in-app payment system with user request forms, admin approval, and transaction tracking.

11. Conclusion

Durjog Prohori represents a comprehensive solution for disaster management in Bangladesh. The system's multi-stakeholder approach, real-time capabilities, and bilingual support make it a valuable tool for emergency response coordination. The modular architecture ensures scalability and future enhancements, while the user-friendly

interface promotes widespread adoption among different user groups.The success of the system will depend on:

- Reliable infrastructure
- User adoption and training
- Regular updates and maintenance
- Stakeholder coordination
- Community engagement

This SRS document serves as a foundation for the development and implementation of the Durjog Prohori system, ensuring all stakeholders have a clear understanding of the system's capabilities, requirements, and implementation plan.