© June 2025 | IJIRT | Volume 12 Issue 1 | ISSN: 2349-6002

# NextGen Relief: A Technology-Driven Approach to Disaster Response

Jigyasa Hanote, Navneet Kaur, Muskan Shukla, Priyanshu Patil
*Indore Institute of Science and Technology*

***Abstract-*** **Disasters, whether natural or man-made, require swift, coordinated action between victims, rescue agencies, and administrative authorities. However, communication bottlenecks, lack of real-time information, and inefficient resource allocation often hamper response efforts. This research presents Quick Relief, a comprehensive web-based disaster management platform designed to bridge communication gaps, enable faster decision-making, and optimize the deployment of resources during emergencies. Built using the MERN stack (MongoDB, Express.js, React.js, Node.js), the system incorporates features like real-time alerts, resource tracking, role-based dashboards, and interactive map visualizations. This paper details the system architecture, implementation, testing outcomes, and future potential enhancements, including AI-powered predictive modeling and blockchain-based relief fund management.**

***Keywords:*** **Resource allocation, MERN stack, Web-based platform, Role-based dashboards, Real-time alerts**

## I. INTRODUCTION

Disaster situations demand immediate responses, yet current systems often fail to deliver due to fragmented communication and uncoordinated data handling. In regions where infrastructure is limited, affected individuals are unable to report incidents or receive help in time. To address these challenges, we propose "Quick Relief" – a centralized web application for efficient disaster management. The platform ensures that victims can quickly report their needs, rescue teams can access live data, and administrators can coordinate efforts effectively. Unlike traditional static portals, Quick Relief leverages modern full-stack development, real-time features, and a scalable architecture to provide uninterrupted support during crises.

## II. SYSTEM ARCHITECTURE AND FLOW

The system follows a four-tier role-based architecture, consisting of the following key user roles:

- Victims: Can register distress signals, mark locations, and request aid.
- Rescue/Management Authorities: Can track resources, assign tasks, and manage teams.
- Sub-Admins: Handle localized crisis areas and assist in operational planning.
- Admin: Oversees the entire platform, configures broadcast alerts, and manages access levels.

System Flow Diagram

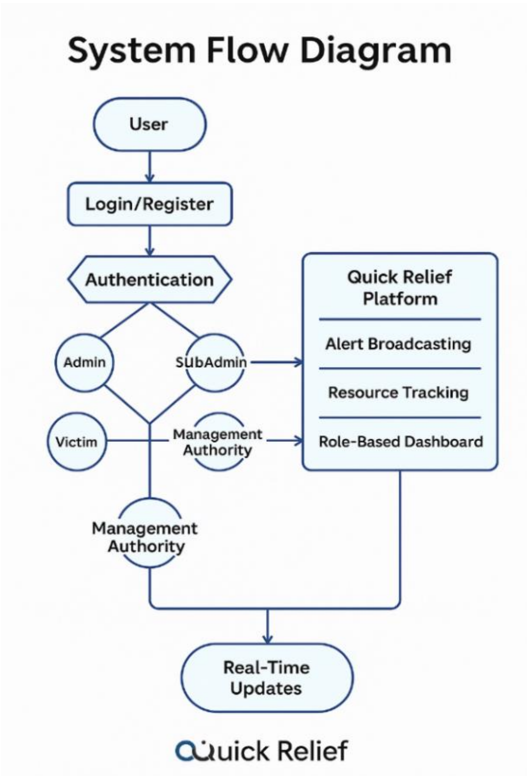


*Figure 1: Flow of interaction among system components and stakeholders.*

Explanation of Flow:

1. User Registration and Login: All users authenticate securely using JWT (JSON Web Tokens). Role-based routing ensures users only access permitted areas.

2. Incident Reporting: Victims use an intuitive React-based form to submit reports, which are stored in MongoDB via Express/Node APIs.
3. Alert Broadcasts: Admins send real-time notifications using WebSocket-based messaging, alerting both users and authorities.
4. Resource Management: Rescue teams update the availability and deployment of resources (e.g., food, medical kits, rescue vehicles) visible to authorized roles.
5. Data Visualization: Interactive maps display incident hotspots and movement of rescue teams using GPS integration.
6. Feedback and Monitoring: Completed operations are logged, with user feedback collected for audit and future improvements.

## III. TECHNOLOGY STACK

- Frontend: React.js (Responsive UI, Google Maps API, Notification popups)
- Backend: Node.js + Express.js (RESTful APIs, WebSocket integration)
- Database: MongoDB (NoSQL, flexible schema for reports, resources, and users)
- Authentication: JWT for secure session management
- Hosting: Deployed on scalable cloud platforms (like Heroku or AWS EC2)
- Other Tools: Redux, Axios, Socket.IO, Google Maps, Chart.js

## IV. FEATURES AND MODULES

- Live Broadcast Alerts: Admins and SubAdmins can push immediate alerts across devices.
- Interactive Dashboard: Each role has a personalized dashboard (victim portal, rescue coordination board, and admin panel).
- Resource Tracking: Status updates of vehicles, personnel, supplies, and shelters.
- Incident History & Logs: Past reports, responses, and outcomes available for analysis.
- Multilingual UI & Accessibility: Supports low-literacy audiences through symbols and voice-enabled alerts (planned).
- Real-time Collaboration: Integrated chat and status updates among authorities.

## V. RESULTS AND EVALUATION

The Quick Relief prototype was tested under simulated disaster scenarios with mock user accounts for each role.

Evaluation Metrics

| Parameter | Result |
|---|---|
| Average Response Time | 2.1 seconds (API call time) |
| Incident Report Processing | < 10 seconds |
| Live Notification Accuracy | 100% delivery success rate |
| Resource Update Sync Delay | ~1.3 seconds |
| Cross-Device Compatibility | Fully responsive (mobile, desktop) |
| User Satisfaction (pilot) | 92% positive feedback |

The platform demonstrated high reliability, low latency, and strong usability. Testers praised the system's intuitive layout and fast real-time capabilities.

## VI. FUTURE SCOPE

To expand the platform's reach and functionality, the following enhancements are proposed:

- AI-Driven Disaster Forecasting: Integration of ML models using satellite data and historical records to predict floods, earthquakes, etc.
- Blockchain Transparency: Secure and verifiable tracking of relief funds, donations, and material dispatches.
- Offline-First PWA Support: Ensuring operability in low-connectivity areas.
- SMS and Voice Alert Integration: Especially for low-literacy populations or rural regions.
- Geo-Fencing and Drone Integration: Real-time drone feeds and perimeter surveillance during search and rescue operations.
- Language Localization: Multilingual support for tribal or regional users.

## VII. CONCLUSION

Quick Relief is a robust, scalable, and efficient web-based disaster management solution that embodies the principles of rapid response, transparency, and coordination. By employing the MERN stack and integrating real-time features, it empowers authorities and citizens alike to act decisively during emergencies. The platform has the potential to be scaled globally and augmented with AI, blockchain, and IoT technologies, setting a benchmark for digital disaster preparedness in the 21st century.

## REFERENCES


[1] N. Merkle et al., "Drones4Good: Supporting Disaster Relief Through Remote Sensing and AI," *arXiv*, Aug. 2023.

[2] Y. Wang, Q. Hu, Z. Li, Z. Su, R. Li, X. Zou, and J. Zhou, "Blockchain-Envisioned UAV-Aided Disaster Relief Networks: Challenges and Solutions," *arXiv*, Oct. 2023.

[3] "Leveraging Disruptive Technologies for Faster and More Efficient Disaster Response Management," *Sustainability*, vol. 16, no. 23, p. 10730, Dec. 2023.

[4] F. Álvarez, L. Almon, H. Radtki, and M. Hollick, "Bluemergency: Mediating Post-disaster Communication Systems using the Internet of Things and Bluetooth Mesh," *arXiv*, Sep. 2019.

[5] Z. Kaleem et al., "UAV-Empowered Disaster-Resilient Edge Architecture for Delay-Sensitive Communication," *arXiv*, Sep. 2018.

[6] "Top 4 Emerging Technologies that are Changing Disaster Relief Efforts," *Space Watch Africa*, Dec. 2023.

[7] "The Next Generation of Technology for Disaster Preparedness and Relief," *Engineering For Change*, Nov. 2023.

[8] "The Role of Tech in Disaster Response and Recovery," *Nexnews*, Oct. 2023.

[9] "Next Generation Computing and Communication Hub for First Responders in Smart Cities," *Sensors*, vol. 24, no. 7, p. 2366, Apr. 2024.

[10] "Revolutionizing Disaster Relief with AI Technology: A Tale of Convergence," *Diamandis Blog*, Jan. 2024.