International Journal of Computer Science and Engineering Development
Volume 1, Issue 4 | September - October 2026 | www.ijcsed.com
ISSN: 3139-0862

A Real-Time Disaster Intelligence and Evacuation Support Platform *Using Geospatial Visualization, Live Event Synchronization, and Climate Risk Inference*

Maseeha Banu[1](Assistant Professor), Gagan D S[2], Likitha C D[3], Lokesh M M[4], Bhoomika R Kandgal[5]

Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning),
Malnad College of Engineering, Hassan, Karnataka, India
*Meseeha Banu(*Assistant professor),all are other authors are students
Corresponding author: likithadevraj006@gmail.com
likithadevraj006@gmail.com, lokeshmmankith@gmail.com, mb@mcehassan.ac.in

Abstract:

Natural disasters create rapidly changing information environments in which citizens, volunteers, and administrators require a common operational picture. This paper presents a full-stack prototype for real-time disaster tracking and evacuation support. The platform integrates a React-based geospatial interface, a Python Flask and Flask-SocketIO backend, JSON persistence, role-aware authentication, browser-based live location, OpenWeather current-weather data, road-network routing, multilingual emergency guidance, SOS signaling, and PDF situation-report generation. Administrators can publish incident and response-resource information, while citizens can view hazards, locate resources, request evacuation routes, and transmit distress information. A transparent climate-risk heuristic combines rainfall, wind speed, and humidity into a bounded preliminary screening score. Functional evaluation was performed across authentication, incident and shelter workflows, climate analysis, live-location display, weather-data integration, administrator-to-user synchronization, routing, SOS, and reporting. The observed tests demonstrate integrated operation of these components; however, no numerical accuracy claim is made for geolocation, route estimation, or the heuristic because dedicated ground-truth and historical validation datasets were not part of the supplied evaluation. The principal contribution is an integrated, explainable emergency-coordination workflow rather than a new prediction algorithm.

Keywords-disaster management, emergency response, geospatial information system, real-time synchronization, evacuation routing, climate risk, OpenWeather, Socket.IO.

INTRODUCTION

Disaster response depends on timely answers to three operational questions: where an incident is occurring, what response resources are available, and how affected people can reach safer locations. The Sendai Framework for Disaster Risk Reduction identifies understanding disaster risk, strengthening risk governance, investing in resilience, and enhancing preparedness as core priorities for disaster risk reduction [1]. Digital coordination systems can support these priorities by combining geographic information, environmental observations, communication channels, and operational records in a shared interface.

In local and community settings, emergency information can be fragmented across telephone calls, messaging groups, static notices, and separate mapping applications. Geospatial emergency-management research has repeatedly identified the value of integrating GIS, positioning, field information, and information dissemination for response coordination [2], [3]. Recent evacuation systems also demonstrate the importance of real-time routing and operational decision support [4], [5].

The proposed system addresses this coordination problem through a browser-based disaster intelligence platform integrating incidents, shelters, hospitals, volunteers, SOS signals, live weather information, user location, evacuation routing, and administrative updates. Unlike a study that introduces a new optimization or prediction algorithm, this work focuses on the engineering integration of multiple response functions into a single operational workflow.

The objectives are to: 1) provide a shared geospatial view of incidents and response resources; 2) synchronize administrator updates with connected users; 3) support authenticated citizen and administrator workflows; 4) provide route-based evacuation assistance; 5) integrate current weather variables into an interpretable preliminary risk score; and 6) support emergency interaction in English, Kannada, and Hindi.

RELATED WORK

GIS-based emergency systems have long been used to support situational awareness, evacuation planning, and resource coordination. Zerger and Smith identified challenges in using GIS for real-time disaster decision support, including the need for appropriate spatial and temporal information for emergency managers [2]. A later geospatial emergency-management framework emphasized integrating GIS, GPS, remote sensing, field information, and decision-support functions [3].Utami and Ramdani developed GEMAR, a web-based GIS for emergency reporting and ambulance routing, combining geolocation, geocoding, and routing and evaluating technology acceptance and usability with expert users [6]. González-Villa et al. proposed a real-time

ISSN: 3139-0862

evacuation management system that combines GIS-based traffic, pedestrian, and routing models and adapts evacuation strategies to changing conditions [4]. Suwanno et al. demonstrated GIS-based evacuation-area and route analysis for flood-prone areas, including road-network analysis under flood scenarios [7].

More recent research continues to move toward real-time and AI-assisted evacuation decision support. A 2025 IEEE conference paper, SafeRoute, describes real-time geospatial evacuation support with route optimization and dynamic information updates [8]. A 2026 study in Environmental Modelling & Software describes human-AI convergence for flood evacuation decision-making by combining AI, transportation geospatial data, hydraulic modeling, and human-generated information [9]. These studies show that the research direction is moving toward dynamic, data-rich emergency decision support.

The gap addressed here is narrower: the proposed platform emphasizes an accessible, low-cost full-stack implementation that connects live user location, current weather, administrator-generated incident information, real-time client synchronization, resource mapping, routing, SOS, multilingual interaction, and reporting in one prototype. The contribution is therefore architectural and integrative rather than a claim of superiority over specialized evacuation algorithms.

TABLE I — RELATED WORK AND RELEVANCE

- Zerger & Smith [2] — GIS challenges for real-time disaster decision support; motivates temporal and spatial coordination.
- Bhanumurthy et al. [3] — GIS, GPS and near-real-time field information for emergency management.
- Utami & Ramdani [6] — Web GIS for emergency reporting and ambulance routing.
- González-Villa et al. [4] — Real-time evacuation management with GIS-based routing.
- Suwanno et al. [7] — GIS-based flood evacuation areas and road-network route analysis.
- SafeRoute [8] — Real-time geospatial evacuation and dynamic route updates.
- HAC flood evacuation [9] — AI, GIS and human information for advanced decision support.

SYSTEM OBJECTIVES AND REQUIREMENTS

The system is designed around two principal user roles. Citizens require access to current hazards, their location, nearby response resources, evacuation directions, weather context, and SOS communication. Administrators require authenticated access to incident and resource management, monitoring, and reporting functions.

TABLE II — USER AND ADMINISTRATOR REQUIREMENTS

Function	Requirement
Live location	Citizen: view position; Administrator: location-aware monitoring.
Incidents	Citizen: view hazards; Administrator: create, update and remove.
Shelters/hospitals	Citizen: find and route; Administrator: manage records.
Weather	Citizen: view current context; Administrator: monitor.
SOS	Citizen: submit distress location; Administrator: view and coordinate.
Reporting	Citizen: view information; Administrator: generate report.
Authentication	Citizen: login/session; Administrator: privileged session.

Data provenance and implementation scope: The prototype uses live browser location, OpenWeather current-weather data, map/geocoding services, OSRM routing, and internally managed incident/resource records. No synthetic accuracy values or fabricated benchmark results are introduced.

SYSTEM ARCHITECTURE

The platform follows a client-server architecture with presentation, application, persistence, and external-service layers. React and React Leaflet form the client interface. Flask provides REST endpoints and Flask-SocketIO provides event-based synchronization. JSON is used for prototype persistence. OpenWeather supplies current weather information; OpenStreetMap-based services support mapping/geocoding; and OSRM supplies road-network routing [10]–[13].

Figure 1. Proposed system architecture

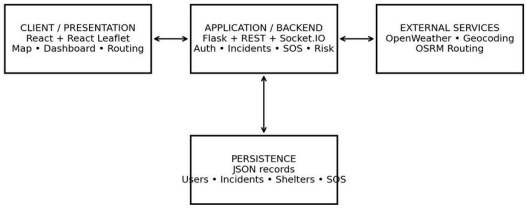


Fig. 1. Conceptual architecture of the proposed disaster intelligence platform.

Data Sources and Data Flow

The principal inputs are shown in Fig. 6. Browser geolocation provides latitude and longitude when permission is granted. OpenWeather returns current weather observations for the requested coordinates. OpenStreetMap/Nominatim can provide place-search and reverse-geocoding functionality subject to its usage policy. OSRM accepts coordinate pairs and returns road-route information. Internal JSON records maintain users and operational resources.

Figure 6. Data-source and processing pipeline

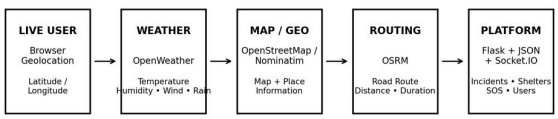


Fig. 6. Data-source and processing pipeline.

Data Entities

Incident records contain an identifier, disaster type, latitude-longitude pair, location name, description, severity, and timestamp. Shelter and hospital records contain an identifier, name, coordinates, capacity, type, operational status, and distance information. Volunteer records contain identity and communication fields together with region and coordinates. SOS records contain an

ISSN: 3139-0862

identifier, coordinates, status, and timestamp. User records contain email, password hash, name, role, and creation time.

Real-Time Interaction

After authentication, the client retrieves operational collections and establishes a Socket.IO connection. When an administrator creates or updates operational information, the backend emits an event to connected clients. Flask-SocketIO supports server-side event emission and broadcasting to connected clients [10], which is the mechanism used by the prototype to propagate operational changes without requiring a full-page reload.

Figure 4. Administrator-to-user real-time synchronization



Fig. 4. Administrator-to-user real-time synchronization path.

Figure 5. Role-based interaction model



Fig. 5. Role-based interaction model.

METHODOLOGY AND IMPLEMENTATION

Live Location Acquisition

The browser geolocation mechanism is used to obtain the user's current position when location permission is available. Latitude and longitude are displayed on the map and can be passed to weather and routing services. The platform therefore provides location-aware visualization. Because browser geolocation accuracy depends on device, browser, permission state, satellite/network availability, and environmental conditions, the paper does not assign a numerical location-error value without a ground-truth experiment.

OpenWeather Integration

The weather component requests current weather data using geographic coordinates. OpenWeather documents a current-weather API using latitude, longitude, and an API key and returns structured JSON including fields such as temperature, pressure, humidity, wind, and precipitation when available [11]. The platform uses the returned environmental variables for dashboard display and as inputs to the preliminary risk heuristic. Weather values should be accompanied by retrieval time and source status in a production deployment.

Figure 3. OpenWeather-to-climate-risk processing flow

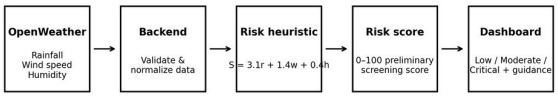


Fig. 3. OpenWeather-to-climate-risk processing flow.

Authentication and Authorization

Registration validates email and password input, and passwords are stored using a password-hashing mechanism. Login verifies the stored hash and issues a JSON Web Token containing identity and role information. The client uses the token for authenticated requests. The prototype separates citizen and administrator workflows.

A security review is required before production deployment. Every state-changing endpoint must enforce server-side authentication and authorization. Client-side role visibility must never be treated as an authorization mechanism. Stronger password requirements, token storage controls, input validation, audit logging, rate limiting, and secret management should be applied in production.

Geospatial Visualization and Distance

Leaflet renders hazards, shelters, hospitals, volunteers, SOS records, and the current user as map features. Approximate geographic distance is calculated using the Haversine formula:

$$a = \sin^2(\Delta\varphi/2) + \cos(\varphi_1)\cos(\varphi_2)\sin^2(\Delta\lambda/2)$$
$$d = 2R\,\mathrm{atan2}(\sqrt{a}, \sqrt{(1-a)}),\ R = 6371\ km$$

The Haversine value is used for approximate local distance presentation. Road-following navigation is delegated to OSRM, whose route service accepts coordinate sequences and transportation profiles [12].

Climate-Risk Heuristic

The prototype does not use a trained machine-learning prediction model. Instead, it implements an interpretable heuristic. For rainfall r in millimeters, wind speed w in kilometers per hour, and humidity h in percent:

$$S = 3.1r + 1.4w + 0.4h$$
$$P = \min(\ \mathrm{floor}((S / 250) \times 100), 100\ )$$

Values above 75 are labeled Critical, values above 45 are labeled Moderate, and the remaining values are labeled Low. The coefficients are manually defined and are not calibrated against a labeled historical disaster dataset. Therefore, P is a preliminary screening score and not a validated probability of disaster occurrence.

This distinction is important for research integrity: the presence of a formula and live weather inputs does not by itself establish predictive accuracy. A future model should be trained and evaluated using event-labeled historical data, with appropriate calibration and class-imbalance analysis.

ISSN: 3139-0862

Evacuation Routing

When a user selects a destination, the platform requests a road-following route using current and destination coordinates. The returned geometry is rendered as a polyline, and route distance and duration are displayed. The current prototype may approximate walking time from route distance; a production version should request an explicit pedestrian profile and accessibility-aware alternatives.

Communication and Reporting

Socket.IO events distribute operational changes to connected clients. Flask-SocketIO supports broadcasting and server-generated events [10]. The frontend can generate a PDF situation report containing incident, shelter, weather, and coordination information. Multilingual labels are provided in English, Kannada, and Hindi.

EXPERIMENTAL EVALUATION

The evaluation in this study is functional and observational. It documents implemented behavior that was exercised in the prototype rather than inventing numerical performance results. The supplied project verification script covers backend health, administrator login, invalid credentials, citizen registration, current-user verification, incident creation, shelter creation, climate analysis, and cleanup. Additional functional checks cover live location, OpenWeather integration, administrator updates, real-time propagation, routing, SOS, and reporting.

TABLE III — FUNCTIONAL EVALUATION SUMMARY

Component	Observed functional check
Authentication	Citizen/admin authentication and role workflows exercised.
Live location	Browser-provided position displayed when permission was available.
OpenWeather	Current weather information retrieved for configured coordinates.
Administrator update	Incident/resource updates propagated to connected clients.
Shelter/hospital	Resources displayed and available as route destinations.
Socket.IO	Connected clients received operational events without full-page reload.
Routing	Road route rendered when the routing service was available.
SOS	SOS information transmitted with location for response visibility.
Risk heuristic	Weather variables converted into a preliminary risk score/category.
PDF report	Situation report generated from platform information.

Quantitative Evaluation Required for Publication-Grade Validation

The current evidence establishes functional integration but not numerical accuracy. A stronger experimental study should repeat each measurement under controlled conditions. Recommended metrics are: (1) median and 95th-percentile REST latency; (2) publisher-to-receiver Socket.IO propagation delay; (3) geolocation error against a reference coordinate; (4) route-distance and ETA error against a trusted navigation reference; (5) climate-risk precision, recall, F1-score and calibration against labeled historical events; (6) task completion time and error rate for representative users; and (7) rejection rate for unauthorized state-changing requests.

Results should report sample size, test environment, software versions, date/time, external-service status, and measurement procedure. If these measurements are not performed, the paper should retain the current qualitative wording and should not introduce invented percentages.

DISCUSSION

The primary contribution is integration. A user can move from location awareness to hazard visualization, weather context, response-resource discovery, route guidance, SOS communication, and reporting within one platform. An administrator can publish operational information and distribute it to connected users through the real-time communication layer.

The architecture is suitable for educational, campus, community, and early-stage local emergency-coordination scenarios. It uses widely available web technologies and replaceable service adapters.

However, the system should not be presented as a validated national emergency-management platform or as an accurate disaster-prediction model. The climate heuristic is intentionally simple and explainable but omits topography, exposure, vulnerability, historical patterns, disaster-specific mechanisms, and spatial hazard layers. External weather, geocoding, and routing services can also fail or impose usage restrictions. JSON persistence is suitable for prototyping but not for high-concurrency transactional workloads. These constraints motivate the future-work plan.

LIMITATIONS

First, the study does not include a labeled historical dataset for validating the climate-risk heuristic. Second, the current evaluation does not provide a controlled benchmark of geolocation error, route accuracy, real-time latency, or usability. Third, the persistence architecture is not designed for multi-process production concurrency. Fourth, all state-changing routes require a complete server-side authorization audit before operational deployment. Fifth, external-service availability and policy restrictions may affect continuity.

SECURITY, PRIVACY, AND SAFETY

Location and SOS records can expose sensitive information about people in danger. Production deployment should minimize data collection, restrict visibility according to operational need, encrypt communications and stored records, define retention periods, and maintain access logs. Exact SOS coordinates should not be exposed publicly unless operationally justified.

The risk indicator must not be treated as an official warning or used as the sole basis for evacuation. Official meteorological and disaster-management authorities remain the appropriate sources for authoritative warnings. The interface should expose timestamps, data provenance, uncertainty, and external-service failure states.

The public Nominatim service has explicit usage limits and requires appropriate identification and attribution; heavy or systematic use is restricted [13]. A production

ISSN: 3139-0862

application should therefore use caching, a suitable service arrangement, or a self-hosted/contracted geocoding service when scale requires it.

FUTURE WORK

Future work should prioritize centralized JWT and role middleware, schema validation, audit trails, transactional database storage with spatial indexing, and conflict handling. The risk component should be calibrated against local historical data while retaining interpretable explanations. Routing should support pedestrian and accessibility-aware profiles, route alternatives, offline caching, and hazard-aware route avoidance.

A human-centered study should involve representative citizens, volunteers, and emergency coordinators. Additional engineering work should include service-health monitoring, message acknowledgements, duplicate-event suppression, SOS rate limiting, stronger password policies, secret rotation, automated accessibility testing, and separation of public map information from protected operational records.

CONCLUSION

This paper presented a real-time disaster intelligence and evacuation-support prototype integrating browser-based live location, OpenWeather current-weather information, geospatial visualization, administrator-to-user synchronization, incident and response-resource management, evacuation routing, SOS communication, multilingual interaction, and PDF reporting. Functional testing demonstrates that these components can operate together as an integrated emergency-coordination workflow. The study deliberately does not claim numerical prediction accuracy because the climate-risk heuristic has not been validated against labeled historical disaster outcomes. The main research value is the integration of heterogeneous data and response functions into an explainable, low-cost web architecture. Future controlled evaluation of latency, location accuracy, route accuracy, usability, security, and risk-model validity is required before production deployment or claims of predictive effectiveness.

ACKNOWLEDGMENT

The authors may add institutional, laboratory, mentor, funding, or infrastructure acknowledgments here if applicable.

AI-ASSISTED DEVELOPMENT AND RESEARCH INTEGRITY STATEMENT

AI-assisted development tools, including conversational coding and writing assistants, were used during software development and manuscript preparation. The authors remain responsible for the architecture, implementation, verification, interpretation of results, factual accuracy, citations, and final manuscript. The use of AI assistance does not replace experimental validation. Any additional disclosure required by the target venue should be added before submission.

REFERENCES

[1]United Nations Office for Disaster Risk Reduction, “Sendai Framework for Disaster Risk Reduction 2015–2030,” United Nations, 2015.

[1] A. Zerger and D. I. Smith, “Impediments to using GIS for real-time disaster decision support,” Computers, Environment and Urban Systems, vol. 27, no. 2, pp. 123–141, 2003, doi: 10.1016/S0198-9715(01)00021-7.

[2] V. Bhanumurthy, G. Jai Shankar, K. Ram Mohan Rao, and P. V. Nagamani, “Defining a framework for integration of geospatial technologies for emergency management,” Geocarto International, vol. 30, no. 9, pp. 963–983, 2015, doi: 10.1080/10106049.2015.1004132.

[3] J. González-Villa, A. Cuesta, D. Alvear, and A. Balboa, “Evacuation Management System for Major Disasters,” Applied Sciences, vol. 12, no. 15, 7876, 2022, doi: 10.3390/app12157876.

[4] P. Suwanno et al., “GIS-based identification and analysis of suitable evacuation areas and routes in flood-prone zones of Nakhon Si Thammarat municipality,” IATSS Research, vol. 47, no. 3, pp. 416–431, 2023, doi: 10.1016/j.iatssr.2023.08.004.

[5] I. Q. Utami and F. Ramdani, “GEMAR: web-based GIS for emergency management and ambulance routing,” Informatics for Health and Social Care, vol. 47, no. 2, pp. 123–131, 2022, doi: 10.1080/17538157.2021.1948856.

[6] C. Oxendine, M. Sonwalkar, and N. Waters, “A Multi-Objective, Multi-Criteria Approach to Improve Situational Awareness in Emergency Evacuation Routing Using Mobile Phone Data,” Transactions in GIS, vol. 16, no. 3, pp. 375–396, 2012, doi: 10.1111/j.1467-9671.2012.01341.x.

[7] S. Ponugupati, T. S. Yallamelli, P. Chilukuri, K. S, and K. S. Pande, “Saferoute: Multi-Modal Optimization for Crisis Management and Evacuation,” in Proc. 3rd Int. Conf. Smart Systems for Applications in Electrical Sciences (ICSSES), 2025, doi: 10.1109/ICSSES64899.2025.11009902.

[8] “Converging human intelligence with AI systems to advance flood evacuation decision,” Environmental Modelling & Software, vol. 201, 106958, 2026, doi: 10.1016/j.envsoft.2026.106958.

[9] M. Grinberg, “Flask-SocketIO Documentation,” Flask-SocketIO.

[10] OpenWeather, “Current Weather Data,” OpenWeather API Documentation.

[11] Project OSRM, “OSRM HTTP API Documentation.”

[12] OpenStreetMap Foundation, “Nominatim Usage Policy (Geocoding Policy).”

[13] I. Fette and A. Melnikov, “The WebSocket Protocol,” RFC 6455, Internet Engineering Task Force, 2011.

[14] National Institute of Standards and Technology, “Digital Identity Guidelines: Authentication and Lifecycle Management,” NIST Special Publication 800-63B.

AUTHOR INFORMATION

Before submission, replace the placeholder corresponding-author email, add ORCID identifiers if required, confirm author order and affiliations, and provide funding, conflict-of-interest, data/code availability, and ethics statements according to the selected venue.