

ReliefMatcher.AI: An AI-Assisted Resource Matching Framework for Disaster Relief Coordination

Sabari S¹, Samuel Dhinakar R², Sanmaya Shree S³, Santhiya T⁴, Santhosh BB⁵

^{1,2,3,4,5}*Department of Computer Science and Engineering, SNS College of Technology, Coimbatore, Tamil Nadu, India*

Abstract—Natural disasters such as floods, cyclones, earthquakes, and landslides create urgent demand for food, drinking water, medicines, shelter, transport, and other essential supplies. Relief operations can be slowed by fragmented communication, limited visibility of available resources, duplicated requests, and difficulty identifying the most urgent needs. This paper presents ReliefMatcher.AI, a web-based disaster relief resource matching platform developed to support coordination between people requesting assistance, donors, volunteers, rescue teams, and relief organizations. The system records available resources and relief requests, then uses a scoring-based matching workflow to compare relevant factors including resource category, quantity, location, availability, and urgency. Requests can be assigned priority levels—Critical, High, Medium, and Low—to help users identify cases that may need earlier attention. The implementation described in the project report uses a React and Vite frontend, a Flask backend, and SQLite for data persistence, with map-based visualization and dashboard-oriented components. The deployed project is available at the project website. Functional checks described in the report cover user access, resource entry, request submission, matching, prioritization, and request tracking. The available documentation does not provide a reproducible benchmark dataset or measured accuracy, latency, or field-outcome results; therefore, the system is presented as a decision-support prototype rather than a validated emergency dispatch service. Future work includes stronger field validation, offline-friendly operation, multilingual access, secure verification, and integration with authorized disaster-management data.

Index Terms—Disaster Relief, Resource Allocation, Humanitarian Logistics, Artificial Intelligence, Decision Support, Priority-Based Matching, Web Application

I. INTRODUCTION

Natural disasters disrupt homes, transport, communication, and access to essential services. During the first stages of an emergency, affected communities may need food, safe drinking water, medicines, temporary shelter, clothing, and transportation. At the same time, donors and relief organizations may have supplies available but lack a shared and current view of where those supplies are needed. When requests and offers are managed through separate phone calls, spreadsheets, messaging groups, or local records, it can be difficult to compare needs, avoid duplicate allocation, and identify critical shortages.

ReliefMatcher.AI was developed to address this coordination problem through a centralized web platform. It is intended to help users register available resources, submit requests for assistance, and review potential matches using a consistent set of factors. The system emphasizes resource type, requested and available quantity, location, and urgency. A priority label can help relief coordinators focus attention on requests marked Critical or High, while a matching score provides a compact explanation of why a resource and request may be compatible.

The platform is designed as a decision-support tool. A match recommendation does not prove that a request is genuine, that a donor has physically verified stock, or that transport is safe and available. Human confirmation and coordination with relevant authorities remain necessary, especially where a delay or incorrect allocation could affect health or safety.

The contributions of this project are: (1) a centralized workflow for recording resource offers and relief requests;

(2) a scoring-based matching approach that considers

category, quantity, location, availability, and urgency; (3) priority-oriented presentation of requests; and (4) a web interface and backend implementation that can be extended with mapping, analytics, and operational integrations.

II. RELATED WORK

Humanitarian logistics research studies how limited supplies are planned, stored, and delivered when demand is urgent and uncertain. Van Wassenhove describes the importance of supply-chain coordination in humanitarian aid, while Altay and Green review operations-research challenges in disaster operations management [1], [2]. These themes are directly relevant to systems that must compare multiple requests against finite stocks and coordinate delivery across locations.

Emergency information can also be collected from digital communities. Work on matching requests and offers in crisis-related social media demonstrates the value of structuring unorganized information so that people offering assistance can find requests that fit their resources [3]. However, information collected from online channels may be incomplete, outdated, duplicated, or difficult to verify. A practical resource matcher therefore needs clear input fields and a review process rather than relying only on free-text messages. Digital technologies are increasingly used to support humanitarian supply chains, information sharing, and tracking. Location-aware logistics can help identify accessible facilities and connect supply points with areas of need. These ideas inform ReliefMatcher.AI's use of location and resource attributes when ranking possible matches. The present project focuses on an implementable web-based workflow; it does not claim to solve route optimization, forecast disaster demand, or automatically verify the truth of submitted information.

III. OBJECTIVES AND SCOPE

The main objective is to support faster and more organized coordination between relief needs and available supplies. The project scope includes recording resource offers, submitting requests, representing urgency, comparing request and resource details, and presenting the resulting matches through a web interface.

Record resources such as food, water, medicine,

shelter, transport, and other supplies. Capture request details including resource category, quantity, unit, location, and urgency.

Compare requests with suitable available resources using a scoring-based matching workflow.

Display priority levels to support review of urgent requests.

Provide a centralized interface for viewing resource and request information.

Support future extension to mapping, reporting, multilingual access, and authorized data sources.

IV. SYSTEM DESIGN AND METHODS

A. Architecture

The documented implementation follows a client-server structure. The frontend is developed with React and Vite and provides forms and views for entering resource availability and relief requests. The backend is implemented using Flask, which exposes application routes and processes submitted data. SQLite is used for structured persistence in the project implementation. The frontend and backend communicate through HTTP requests, with JSON used for structured data exchange where applicable.

The deployed project is published at <https://ai-disaster-relief-matcher.onrender.com>. Deployment and availability depend on the hosting service and network access. The project website demonstrates the application interface; the manuscript does not assume uninterrupted service during a real disaster or offline operation.

B. Data Collection and Representation

A resource offer represents supplies that a donor or organization reports as available. Relevant fields include the resource name, category, quantity, unit, and location. A relief request represents a need reported by an affected person, rescue team, or coordinating organization. Its fields include the requested resource, category, quantity, urgency, and location. Structured fields make it easier to compare records than unstructured messages alone.

The project report describes categories including Food, Water, Medicine, Shelter, Transport, and Other. Request urgency is represented by Critical, High, Medium, or Low. These values are used as operational labels to help organize the queue; they should not be treated as a clinical or official emergency classification

unless a responsible authority defines and approves the policy.

C. Matching and Prioritization Workflow

The matching workflow begins when resource offers and relief requests have been entered. The system identifies candidate pairs that share a compatible resource category and then considers other attributes such as available quantity, required quantity, geographic proximity or location compatibility, availability, and request urgency. A scoring model ranks candidate matches and displays a match percentage or score to make the recommendation easier to inspect. The interface also supports rerunning the matching process when the underlying records change. A simplified conceptual workflow is: (1) collect and validate resource offers and requests; (2) filter candidate pairs by resource category; (3) compare quantity and location; (4) account for urgency and availability; (5) calculate and rank a match score; and (6) display suggested matches for human review. The project describes this as an explainable scoring model. The exact numeric weights and a formal mathematical specification are not provided in the supplied report, so this paper does not invent a scoring equation or claim a particular optimized weighting scheme.

Where one request requires more units than a single donor can provide, a useful allocation strategy is to consider multiple compatible offers and track the quantity allocated from each. For example, a hypothetical request for 1,000 food packets could be covered by several donors whose available quantities sum to the requested amount. This is an illustrative allocation scenario, not a measured field result. Before actual dispatch, the coordinator must verify stock, transport, destination access, and the recipient's need.

D. User Workflow

The primary workflow includes opening the website, entering or reviewing resource information, submitting a request, assigning urgency, running or reviewing the matching process, and checking the resulting records. Donors and relief organizations can use the resource-entry workflow to publish available supplies. Requesting users or rescue teams can record needs and their locations. The dashboard-oriented design is intended to make information easier to review in one place rather than relying on

disconnected records.

Map-based visualization can help users interpret where requests and offers are located. A map marker should be understood as a representation of the submitted location, not proof that a team or resource is physically present there. Accurate coordinates, current inventory, and reliable status updates are necessary for location-aware decisions.

V. IMPLEMENTATION

The documented stack includes React and Vite for the frontend, Tailwind CSS for styling, Flask for the backend, and SQLite for data storage. Leaflet and React Leaflet are used for map-oriented presentation, while Chart.js supports dashboard charts. Lucide React provides interface icons, and jsPDF is included for report or document generation. The project report identifies Python 3.10 or later and Node.js 18 or later as the development environment requirements.

The backend handles application requests and data operations. The frontend presents forms and results, including fields for resource type, quantity, unit, urgency, and location. The matching interface presents candidate matches and their scores so users can review the rationale. A live deployment is available through the project URL, although production readiness under disaster-scale traffic, poor connectivity, or malicious input has not been established by the supplied evidence.

The project report also describes potential links to disaster information and map-related components. Such integrations should be treated carefully: authoritative feeds need to be identified, their terms and update frequency checked, and their data provenance shown to users. Automated recommendations should remain separate from official warnings and emergency instructions.

VI. RESULTS AND FUNCTIONAL CHECKS

The project report describes functional testing intended to verify that key workflows behave according to their requirements. The documented areas include user registration and login, resource entry, relief request submission, validation of resource information, AI-based matching, urgency prioritization, request tracking, and the presentation of updated information. The testing approach is

described as providing valid and invalid inputs, comparing actual behavior with expected behavior, correcting errors, and retesting affected functions.

At the implementation level, the project provides a working deployed website and a user interface for the intended resource-matching workflow. The report describes test scenarios and interface screens, but it does not provide a reproducible test dataset, sample size, measured match precision or recall, response-time measurements, load-test results, or a controlled field study. Accordingly, quantitative claims such as a percentage reduction in delivery time, resource wastage, or allocation errors cannot be concluded from the available documentation.

A suitable next evaluation should define test cases with known correct matches, include partial and insufficient stock scenarios, and test requests at different urgency levels and locations. Results could be reported using category-match accuracy, quantity coverage, ranking quality, response latency, and the number of invalid or duplicate records detected. A field evaluation should be conducted only with appropriate partner organizations and safety procedures.

VII. DISCUSSION

ReliefMatcher.AI brings resource offers and requests into a shared workflow and makes several matching factors visible. Structured categories can reduce ambiguity, while a priority label helps users notice requests that may require earlier review. A scoring-based approach is understandable and comparatively straightforward to implement, which is useful for a student project and for explaining why a candidate match appears in the interface. There are important limitations. Submitted locations may be imprecise, quantities may become outdated, and a high score cannot establish the reliability of a donor or request. Category matching alone may overlook quality, expiry dates, storage requirements, compatibility, road conditions, transport capacity, or legal restrictions. A resource that appears close on a map may not be reachable. The platform therefore should not be used as an autonomous dispatch authority.

Privacy and security also matter. Locations and contact details associated with disaster-affected people may be sensitive. A production system should collect only necessary data, restrict access by user role, protect credentials and communications, maintain audit

records, and define retention and deletion policies. Request verification, duplicate review, stock confirmation, and clear status ownership are necessary safeguards.

The website relies on internet connectivity and hosting availability. During disasters, power and network services may be disrupted. A resilient future version could support local saving and delayed synchronization, but offline behavior is not claimed as a current capability. Similarly, any integration with public disaster feeds should be validated against the authoritative source and should not replace official emergency guidance.

VIII. CONCLUSION AND FUTURE WORK

This paper presented ReliefMatcher.AI, a web-based AI-assisted disaster relief resource matcher intended to improve coordination between relief requests and available supplies. The system organizes resource offers and requests and ranks candidate matches using factors described in the project report, including resource category, quantity, location, availability, and urgency. Priority labels and a centralized interface are intended to help users review needs and potential supply matches more systematically.

The project demonstrates an implementation-oriented approach to humanitarian resource coordination, but the available documentation does not establish operational effectiveness or measured improvements in disaster outcomes. Human verification and coordination with authorized relief organizations remain essential before supplies are allocated or dispatched.

Future work includes evaluating the matching score on labeled test cases; improving inventory and request verification; adding secure role-based access and audit logs; supporting multilingual and accessible interfaces; developing mobile and offline-friendly workflows; integrating reliable GPS and route information; providing notifications and status updates; and conducting usability, security, and load testing. Further work should also define transparent matching weights and report reproducible results under realistic scenarios.

ACKNOWLEDGMENT

The authors acknowledge SNS College of Technology, Coimbatore, Tamil Nadu, India, for the academic environment and support provided for this student project. Any additional acknowledgments should be confirmed by the project team before submission.

Author Contributions

This is a team project. Individual contributions were not itemized in the supplied report; the authors should confirm and describe each member's actual contribution before final submission.

Conflict Of Interest

The authors should confirm whether any competing interests exist before submission.

Funding

No external funding information was provided in the supplied project report.

Data Availability

The manuscript does not report a newly collected, shareable research dataset. The project website is available at <https://ai-disaster-relief-matcher.onrender.com>. A source-code repository or test dataset should be added if the team intends to make it publicly available.

Ethics And Safety Statement

Samuel Dhinakar R was responsible for the software implementation and initial manuscript preparation. The remaining authors are listed as co-authors of this team project; their specific intellectual contributions must be accurately confirmed by the author group before submission. No individual contribution is attributed here without supporting confirmation. The platform should be treated as decision support and not as a replacement for official emergency services.

Journal of Operational Research, vol. 175, no. 1, pp. 475–493, 2006.

- [3] Purohit *et al.*, “Research on matching resource requests and offers during emergencies through social media,” 2014. [Incomplete reference: Verify the original publication title, author list, and publication details before submission.]
- [4] Ö. Özdamar, E. Ekinici, and B. Küçükyazici, “Emergency logistics planning in natural disasters,” *Annals of Operations Research*, vol. 129, pp. 217–245, 2004.
- [5] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Hoboken, NJ, USA: Pearson, 2021.
- [6] Flask, “Flask documentation.” [Online]. Available: <https://flask.palletsprojects.com/>
- [7] Python Software Foundation, “Python documentation.” [Online]. Available: <https://docs.python.org/3/>
- [8] World Health Organization, “Health emergency and disaster risk management.” [Online]. Available: <https://www.who.int/>
- [9] United Nations Office for Disaster Risk Reduction, “Disaster risk reduction.” [Online]. Available: <https://www.undrr.org/>
- [10] Federal Emergency Management Agency, “Disaster preparedness and response resources.” [Online]. Available: <https://www.fema.gov/>
- [11] United Nations Office for the Coordination of Humanitarian Affairs, “Humanitarian Data Exchange.” [Online]. Available: <https://data.humdata.org/>

REFERENCES

- [1] L. N. Van Wassenhove, “Humanitarian aid logistics: Supply chain management in high gear,” *Journal of the Operational Research Society*, vol. 57, pp. 475–489, 2006.
- [2] N. Altay and W. G. Green, “OR/MS research in disaster operations management,” *European*