iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68472

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Rescue Connect: AI-Powered Disaster Relief System

Manda Rushitha[1], Kakani Nikesh[2], Kudithi Sai Nitheesh[3], Dr. S Neduncheliyan[4]
[1, 2, 3]*UG Student, Dept. of IT, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India*
[4]*Dean & Professor, Department of CSE, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu,* India

Abstract: ***An effective reaction in the case of disaster based on the rapid popularity of warnings, voluntary coordination, effective resource management and real -time communication. This article presents Rescue Connect, a Python -based disaster management system integrated chatbot's answers provided by AI, visualized resources and alerting processing. The system supports citizens' interaction through common questions and security advice, visualization of volunteer status and resources and simulating new emergency warnings. By taking advantage of local logic and external API, the system provides a hybrid intelligent assistant to improve the preparation for the disaster and the reaction's coordination.***

Keywords: ***Disaster management, AI chatbot, simulation system, resource monitoring, volunteer management, real -time warning, Python, Google Collab, Mosaic design, emergency response, visualization of data, crisis simulation, command line interface***

I. INTRODUCTION

Disaster, whether natural, such as floods and earthquakes, or artificial industrial accidents, have a profound impact on the community, infrastructure and public health. According to global reports, the frequency and intensity of these events have increased due to climate change, urban expansion and increasing population density in vulnerable areas. Therefore, the systems responding to disasters are well overcome in a timely manner is more important than ever.

Traditional disaster management executives often suffer from some limitations, especially:

1) Lack of data integration in real time: emergency systems can rely on static database or manual reports that are delayed.
2) Bad public commitment: citizens may not know where to find help or how to contribute in an emergency.
3) fragmented communication: Emergency services, volunteers and citizens often operate in separate channels and have not been integrated.
4) Capacity to adapt to limit: Many systems cannot simulate scenarios or provide personal reactions for different types of disasters.

To meet these challenges, we propose Rescue Connect, a system of supporting light management disaster management based on Python combining automation, visualization of data and AI conversation. Unlike large and expensive emergency platforms, Rescue Connect is designed to deploy quickly in the local, regional or educational context. It supports creating alerts, supervising volunteers, monitoring resources and chatbot multi -mode to answer the request of users with security advice, disaster events and advice.

The system is structured around simplicity and module. Users interact with the system through the control panel interface to imitate actual use cases. For example, emergency staff can view the available volunteers in the circular graphics, while citizens can ask chatbot of temporary shelters nearby or learn how to keep safe in forest fires.

by combining structured data sets, simulation scenarios and local intelligence, Rescue Connect provides a practical and accessible solution to improve the recovery of community disaster. This article refers to the system's architecture, individual components, the concept of chatbot intelligence and its role in education and disaster management**.**

II. RELATED WORKS

The disaster management system has increasingly integrated information technology and artificial intelligence to improve the efficiency of emergency intervention. A number of previous studies and systems have resolved specific components such as the warning system, the coordination of volunteers and resource management, but a few have implemented all these factors in the compatible -consistent simulation environment such as connecting rescue.

For example, Coyle et al. (2017) Discover how AI warning systems can support early warnings in case of disaster by using real -time data, while Xu et al. (2021) Designing Mo -Mun systems is provided to improve expansion and easy deployment in crisis management software. Regarding the coordination of volunteers and resources, Senarath et al. .

related to the integration of chatbot, previous research such as Keykhaei et al. (2022) has shown how rules based on rules can provide basic emergency information to users in disasters, helping to reduce anxiety and improve community awareness. However, these deployments often lack expansion and simulation features. Save the connection to develop this work by combining a simple chatbot with creating warnings, logistics of volunteers and simulating in real time, all in a tissue system based on a python that can be easily expanded to research, train or create samples.

III. PROPOSED METHOD

The proposed method for the development of the rescue connection system based on the application of the Mo -Mo Command line is designed to simulate and in charge of disaster management activities. The system is structured in independent functions that manage warnings in real time, coordinate volunteers, monitor resources and chatbot interaction.

1) Methods of design: each component - Warning, volunteers, resources, chatbot and visualization - are developed as a separate Python -python using standard data structures such as dictionaries and lists. Calculate this module to ensure that individual parts can be tested, debugged and independent rate.

2) Model –function: Warning system: disaster warning simulation by randomly creating disasters, location and severity by tissue -random tissue.

3) Volunteer Management: Volobile information of France, including name, location and useful by using the dictionary based on the list. Monitoring resources: maintains rescue resources files such as food, water and medical kits, according to their quantity and location.

4) Chatbot interface: AI chatbot is based on the rules built by if-Elif condition and the question is predefined and answered the dictionary to simulate support users in an emergency. visual tools: uses Matplotlib to create a circular graphics for the available volunteers and graphs for resource stocks, helping to make decisions with visual ideas. Libraries such as random, time and matplotlib.pipot have been used for functions, calendars and data viewing. a -tissue -tossing disaster events that cannot be predicted, activated the warning system and allows real -time tests of other -other modules. This adds realism and supports training and evaluating system reaction strategies.

5) User interactive flow: One Mancing based on the menu (command line interface) is used to navigate in the -tunes. This configuration supports easy tests, the appeal for technical users and rapid conversion between functions to perform or develop.

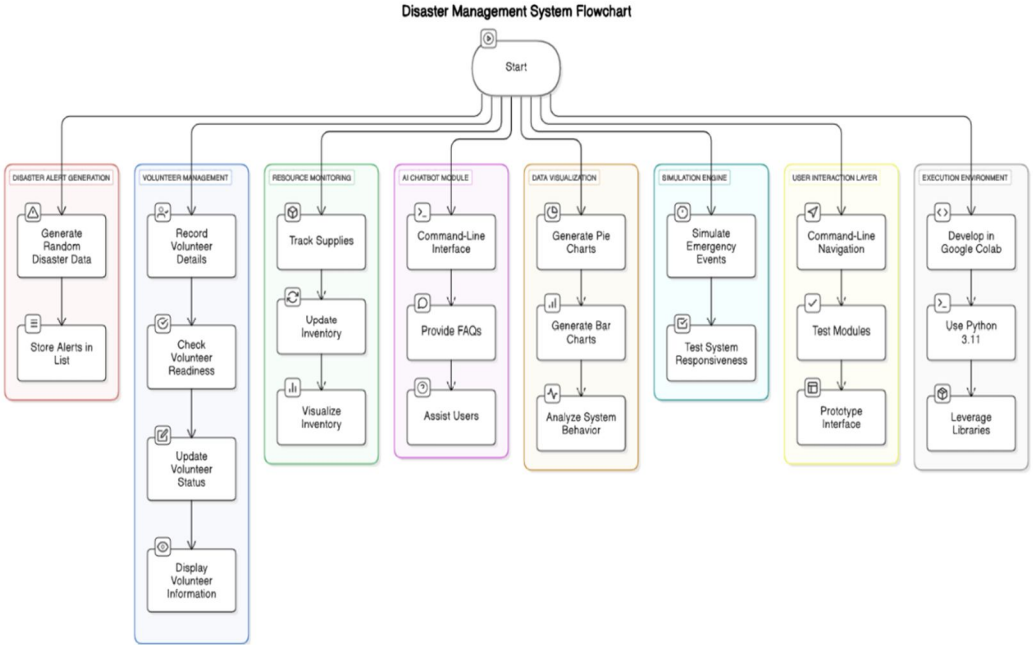


Fig. 1: System Architecture

IV. WORKING FLOW

The connected rescue system acts as a Mo -Mo disaster management platform deployed in Google Colab with Python. In the premiere, it presents a interface focusing on the menu that allows users to interact with the basic tissues: Disaster warning, volunteer management, resource monitoring, AI chatbot and visualization.

work process starting with the user entry through the command line. After that, the system allows simulating disaster warnings using random data for type, location and severity. Volunteers may be added, followed and managed according to the usability and position. Similarly, resources such as food, water and medical kits are monitored and updated in real time.

AI-Chatbot model based on basic rules that provide answers to frequently asked questions, improve user support. Visual ingredients (charts and graphs) provide information about the distribution of volunteers and the available resources all -operating modules but integrated by a central stream, allowing practical tests on emergency scenarios in an interactive loop environment.

V. RESULTS

The implementation of the rescue connection system has given promising results in the reaction and management in case of disaster by using a python -based module in Google Colab. The system has successfully implemented all the basic features, especially:

generation warning in real time: The system has created dynamic simulation disaster warnings with random disaster types, position and gravity level, effectively imitating emergency situations in the real world.

Volunteering Management:- Volunteer Mo has accurately grasped the information about volunteers, the usability and position, allowing the simulation of decisions to implement in an emergency.

Monitoring resources: systems according to important resources (food, water, medical kits, covers) with real quantities updates and position mapping, effective securities management support.

Integrated AI chatbot: A chatbot based on the rules that provide answers related to common questions, proving how anyone can help users in emergency situations with basic but reliable support.

Visualization of data: visual output based on matplotlib, such as baked cakes and bar charts, clearly display volunteer laws and shares of resource stocks, improve data understanding and decisions.

Simulation flexibility: The system allows continuous simulation tests with random events and moist municon, allowing both constraints and performance of work processes in the event of a disaster.

A. System Functionality Overview

The rescue system is a disaster management simulation tool deployed in Python and developed in Google Colab environment. It acts as a Mo -Munum program and focuses on the command line that combines different features to simulate and manage disaster response scenarios. The system allows users to create random disaster warnings, by associating all kinds, positions and gravity to imitate real emergency cases. It also includes a volunteer management component, recording the name, position and the availability of volunteering, helping to simulate resources in crisis situations.

-Monitoring resources to manage main factors such as food, water, shells and medical kits, keeping a trace of their number and storage location. These resources are intuitive by using Tart and bar charts to support decisions and planning. In addition, a Basic AI chatbot is built according to the approach based on the rules that provide users with automatic feedback for current requirements, acting as a virtual assistant in the disaster simulation process. The entire simulation combines random characters to reflect the unpredictable nature of practical emergencies, ensuring that each time performing the unique conditions for system testing and evaluation. Thanks to its simple interface and its structure, the rescue is a basic approach to manage reactions to real -time disasters.

B. Real-Time Alerts and Simulation

The real -time warning component of the rescue connection system is designed to imitate the unpredictable nature of the disasters by creating alerts during the execution process. Using a random library Python, a random selection system of various types of disasters (for example, floods, earthquakes), parts and gravity levels to simulate practical emergency scenarios. Each warning is stored in a list of dictionaries, allowing the system to maintain a structural and easy -to -access record of current disasters.

simulation function continues the system continuously with new warnings, imitating the conditions of the real world where emergency cases occur unexpectedly. This simulation warning vol These simulations are essential to assess the response capabilities of other -other modules, such as deploying volunteers and resource allocation, creating a dynamic and attractive testing environment for disaster management strategies. 5.3 Volunteer Management

```
def show_alerts():
    print("\n Real-Time Alerts:")
    for alert in alerts:
        print(f"- [{alert['severity']}] {alert['type']} reported in {alert['location']}")

def track_volunteers():
    print("\n Volunteer Status:")
    for v in volunteers:
        print(f"- {v['name']} at {v['location']} - Status: {v['status']}")
```

Figure 1: The showing alerts and tracking volunteer

```
====== RESCUE CONNECT SYSTEM ======
1. View Real-Time Alerts
2. Track Volunteers
3. Show Resource Availability
4. Free AI Chatbot (no API key)
5. View Volunteer Status Chart
6. View Resource Summary Chart
7. Simulate New Disaster Alert
8. Exit
Choose an option (1-8): 1

 Real-Time Alerts:
- [High] Flood reported in Brisbane
- [Medium] Wildfire reported in Perth Hills
- [Low] Earthquake reported in Adelaide
```

Figure 2: The showing results alerts and tracking volunteer

Volunteer management is one of the components of the Rescue Connect: AI-Powered Disaster Relief System that is focused on the management of volunteers and their supply during emergencies. Its main purpose is to provide thieves and the decision makers with the identification of the status and the location of all the volunteers who are registered into the system at any given time. Ensuring that the records of volunteers are current; they are classified into three categories: Availability, Busy, or Unavailable, the system guarantees that the response teams are well-informed and can, therefore, make relevant decisions concerning the number and location of manpower required to meet the current needs. The adapted pie chart incorporated in this module improves its usability as it offers visual statistics about volunteer status in a comprehensible and efficient manner. This makes it easier for users to understand the general dispersion of the volunteers' availability without going through the list or table form. Lastly, the planning aid greatly facilitates strategic planning because it reveals relative shortages or surpluses of resources such as personnel. For example, in a case where there is a simulated disaster, one could review from the chart which percentage of their manpower is deployable and make quick and effective decisions on how to deploy many people.

More practically, this visualization feature provides help in administering volunteers because it will limit the time spent on response and possible misunderstandings within the short period of tension or panic. One can intervene easily to determine whether a unit needs more intakes or if the existing ones must be redistributed from one unit to another. Moreover, the chance to incorporate location also plays a role in identifying which volunteers should be deployed quickly at the scene of disaster, thus making the process more effective in terms of time. From the design perspective, as it has already been pointed out above, the interface is quite clear and the graphics presented are rather comprehensive which allows even those who are not highly computer literate to use it. By use of the chart format, bar colour, and percentage differentiation presents important details in a manner that is speedy and efficient.

This approach not only improves the usability of the system but also aligns with the goals of inclusive and responsive disaster management. The volunteer module demonstrated its effectiveness by accurately tracking real-time personnel status and offering strategic visibility through well-designed data visualization.

C. Resource Monitoring

The management of volunteers from the rescue system connects the management of registration, monitoring and monitoring the status of volunteers in the simulation disaster scenarios. Each volunteer is shown in the form of a dictionary containing his existing name, position and status. This structure format allows the system to effectively manage volunteer data and automatically update their status according to the simulation needs.

During the simulation process, volunteers can be indicated for specific warnings according to their usual use, which reflects the actual coordination process in emergency management. Mo -Mun also supports the visualization of volunteers operating, not working or not available, helping to assess the capacity of human resources at any time. This feature ensures that the system can effectively affect volunteers to the most necessary areas, improve the overall efficiency and reality of simulation to cope with disasters. 5.5 AI Chatbot Interaction.

```
def emergency_knowledge_chat():
    """Uses free disaster API for factual information"""
    print("\n🌎 Disaster Knowledge Base (real data from ReliefWeb API)")

    while True:
        user_input = input("You: ").strip().lower()
        if user_input == 'exit':
            break

        # Try to get real disaster data from free API
        try:
            if "disaster" in user_input or "report" in user_input:
                url = "https://api.reliefweb.int/v1/disasters?appname=rescuesystem&limit=3"
                data = requests.get(url).json()
                print("\nRecent Global Disasters:")
                for item in data["data"][:3]:
                    print(f"- {item['fields']['name']} ({item['fields']['date']})")
                print("Source: ReliefWeb API")
            elif any(word in user_input for word in ["flood", "fire", "earthquake"]):
                print("\nSafety Tips:")
                if "flood" in user_input:
                    print("- Never drive through floodwaters\n- Move to higher ground")
                elif "fire" in user_input:
                    print("- Create defensible space around property\n- Have an evacuation plan")
                else:
                    print("- Drop, Cover, Hold during shaking\n- Check for gas leaks after")
            else:
                if user_input in faq:
                    print("Bot:", faq[user_input])
                else:
                    print("Bot: Try asking about disaster reports or safety tips")
```

Figure 3: The steps of AI chatbot

```
Choose an option (1-8): 4

🤖 Free AI Chatbot (no API key needed)
Choose a provider:
1. DeepSeek Free Web Version (simulated)
2. Local AI (simple responses)
3. Emergency Response Knowledge Base
Select chatbot mode (1-3): 1

🔍 DeepSeek Free Mode (simulated emergency specialist)
Type 'exit' to end the conversation
You: shelter
Bot: Nearest shelters can be found at: Brisbane - 123 Queen St, Perth - 456 Forest Ave
You:
```

Figure 3: The steps of AI chatbot

AI chatbot tunnel provides the basic question answer with logic based on dictionary. Users have been able to:
to ask a predetermined question (for example, "What is an emergency number?", "How to volunteer?")

D. Visualization Insights

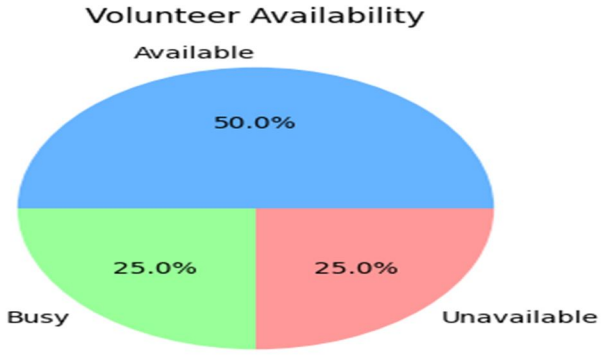


Figure 4: The pie chart of availability of volunteer

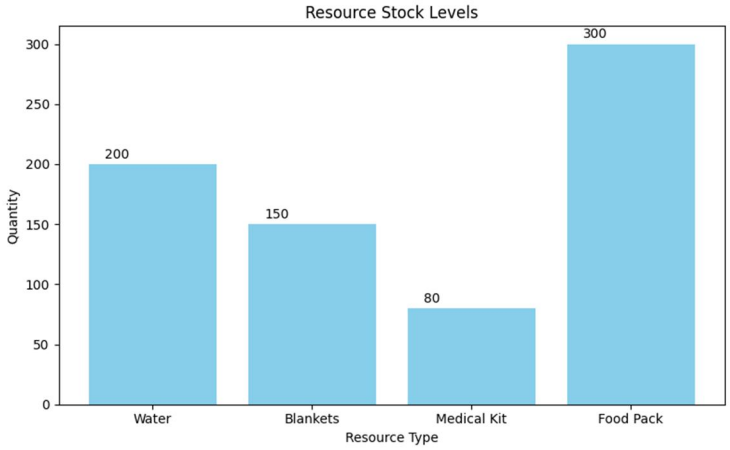


Figure 5: The bar chart of availability of resources

The bar chart on the available resources in the connected rescue system provides a clear visual representation of the number of essential sources such as food, water, shells and medical kits. This chart reflects the current securities levels stored in the system resource management. By displaying each type of resource along the X axis and their corresponding quantity on the Y axis, the bar chart helps users quickly evaluate the resources beyond or perform.

This visual component supports effective decisions in the simulation reactions in the event of a disaster by creating conditions to determine the shortage of supply or additional intended plans. It also improves the situation of the situation by giving a quick photo of logistics preparation, contributing to more information and timely actions in emergency simulation**.**

VI. CONCLUSION

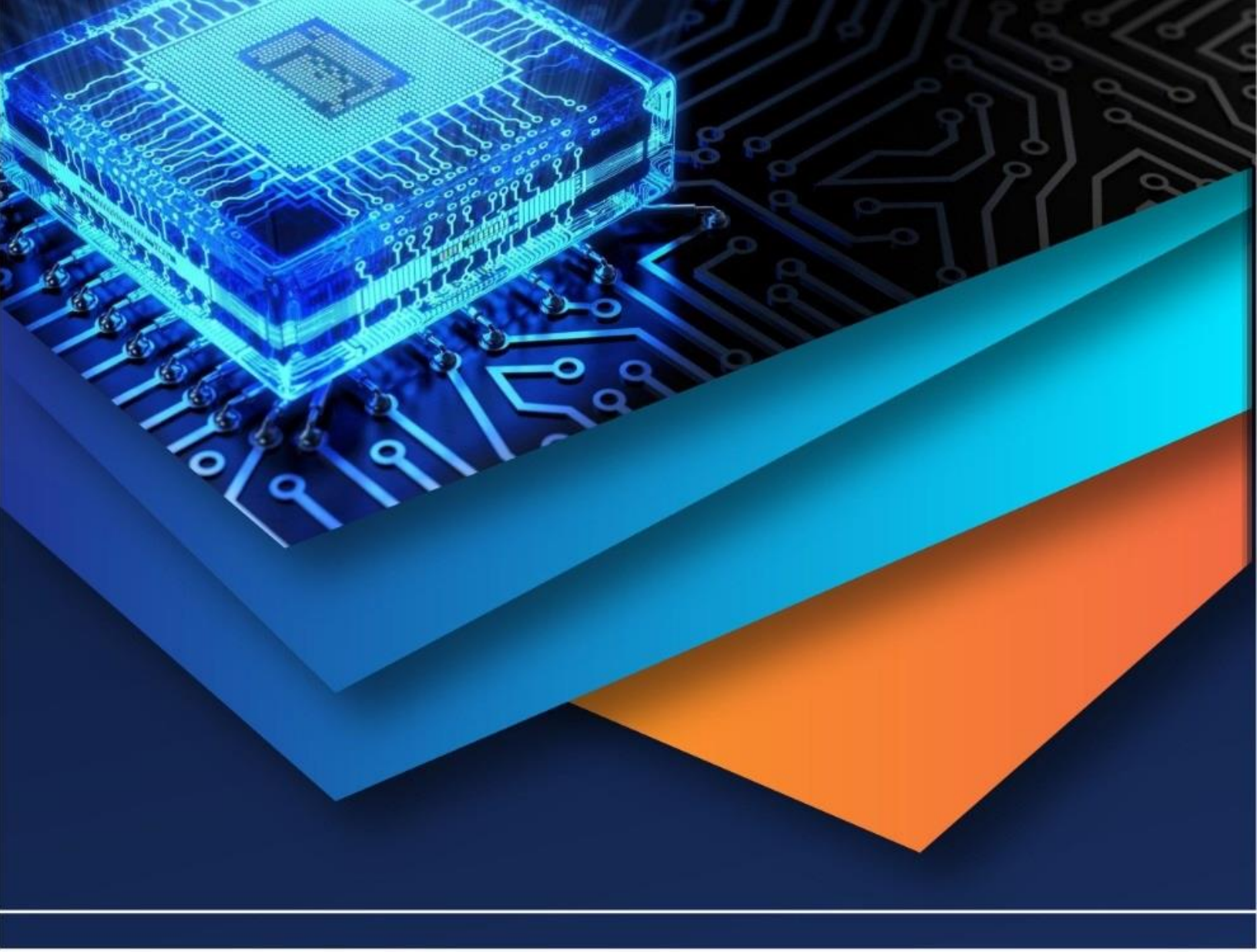
The Rescue Connect system is an impressive illustration of the potential that digital tools possess to greatly enhance the coordination and delivery of disaster relief efforts, when well designed. As a simulation tool developed, the system integrates real-time tracking of data, volunteer organization, resource management, and AI-driven communication seamlessly into one single platform. Its modular design makes it possible for users to interact with individual elements in isolation, yet reap the benefits of an interdependent system facilitating integrated disaster response planning and operation. By sending live alerts of disaster incidents, rich analysis of volunteer availability, and real-time resource stock levels, Rescue Connect facilitates better and more timely decisions that are crucial in pressure-packed emergency situations.

Incorporation of a chatbot interface also provides an added value to user interaction in terms of easy support and information sharing, especially for the individuals looking for shelter, emergency numbers, or information on volunteering. Pie charts and bar graphs are used to enhance situational awareness and facilitate easy access by both technical and non-technical users. The capability of the system to replicate real disaster situations with dynamic alerts also makes it a useful training resource for emergency readiness. Overall, Rescue Connect demonstrates how a properly designed, Python program can bring together communication, analytics, and resource tracking to facilitate strategic, scalable, and user-friendly disaster management solutions.

REFERENCES

[1] Bajwa, A. (2025). Effective resource management in emergency intervention systems. Journal of Technology and Disaster Management, 13 (2), 5 Hang56.
[2] Keykhaei, S., Zhang, Y. and Chen, W. (2022). Support in decision -based decisions for disaster management:
[3] an exam and one framework. International Journal of Disaster Risk reduction, 68, 102719. AI systems based on Python for urgent simulation:
[4] design, implementation and evaluation. Journal of smart systems and applications, 12 (1), 33 matches7.
[5] Xu, M., Lin, Y. and Rao, P. (2021). Mosaic programming method in emergency intervention systems. Science calculates and manages disaster, 6 (3), 112 from 126.
[6] Xu, M., Lin, Y. and Rao, P. (2022). Optimize the data structure in the disaster response simulation by Python. Practice and theory of simulation and modeling, 118, 102539. https://doi.org/10.1016/j.simpat.2021.102539
[7] Huang, Y., and Wang, Z. (2021). Visiting data in disaster management:technology and executive director. Disaster computer evaluation, 9 (), 205 from 220.
[8] Patel, R. and Koteche, K. (2023). Chatbot systems are based on rules for emergency services. International Journal of Artificial Intelligence and Application, 1(2), 78 Throwing 90.
[9] Lin, C. H. and Tan, Y. S. (2023). AI Tirration for public security:Application of chatbot in emergency communication. AI and society, 38 (1), 65 balls81. https://doi.org/10.1007/s0016-022-01585-9

[10] Dutta, S., and Jain, N.. (2020). Real -time simulation tools for training to prepare disaster. Journal of Emergency Planning and Management, 28 (1), 1-29.
[11] Kim, J. and Lee, S. (2021). Cloud -based simulation frames to respond quickly in the case of disaster. Computer science procedures, 18, 931 from 938. https://doi.org/10.1016/j.procs.2021.03.116
[12] Kaur, M. and Singh, G. (2021). An assessment of artificial intelligence application in disaster management. Natural danger, 107, 1195 from 1213. https://doi.org/10.1007/s11069-021-0656-Z
[13] AHMED, S., and Rathore, M. M. (2020). Answer in the event of a disaster in smart cities using automatic use and iodine:a survey. Access IEEE, 8, 10605 from 10620. https://doi.org/10.1109/access.2020.2999631
[14] Kumar, S. and Sharma, S. (2022). The decision is based on the decision -to emergency services using Python frames. Computer science procedures, 199, 123 from 131. https://doi.org/10.1016/j.procs.2022.01.015
[15] Desai, M. and Patel, R. (2023). Improve user interaction in disaster systems using rules based on rules and are provided by AI. International Magazine on interaction between humans and computers, 39 (5), 21 ball36. https://doi.org/10.1080/107318.2023.219517
[16] Thomas, R., and George, R. (2021). Visual analysis for emergency resources plan. Intuitively information, 20 (3), 213 212229.
[17] Al-Kodany, K. (2020). Use visual tools in planning and response of disasters. Urban Technology Magazine, 27 (1), 3 matches 27. https://doi.org/10.1080/10630732.201399.17090
[18] Zhao, H. and Wang, J. (2022). A frame to track the real disaster by using the cloud and Bord it. Journal of Network and Computer Application Magazine, 189, 103098.
[19] Srinivasan, S., and Raghavan, P. (2022). Coordinate rescue in the case of disaster by using open source technologies:in a python -based approach. International Magazine on Emergency Services, 11 (2), 101 balls117.
[20] Zhang, Y. and Qian, X. (2023). Multi -factor simulation system for disaster evacuation plans. Practice and theory of simulation models, 13, 102566.
[21] Rajput, P. and Meena, S. (2021). Design of Mo -Moun software systems for emergency preparation:a case study with Python. International Magazine on computer applications, 183 (8), 1 match 7.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY