# INTRODUCTION

In the previous section, we gathered requirements for our disaster management system through a multiple methods.

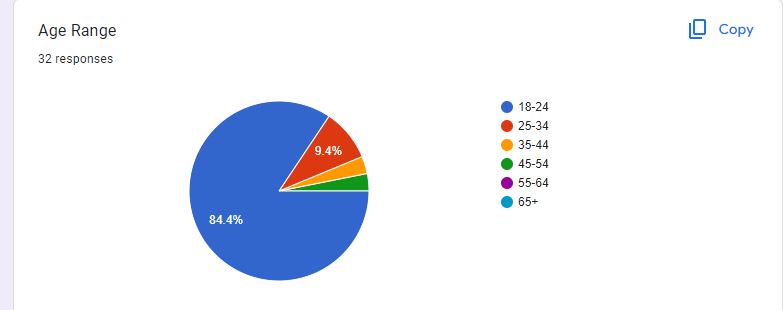
However, gathering requirements alone are just the first step. To translate these needs into a tangible and effective mobile application, we embarked on the critical stage of requirements analysis.

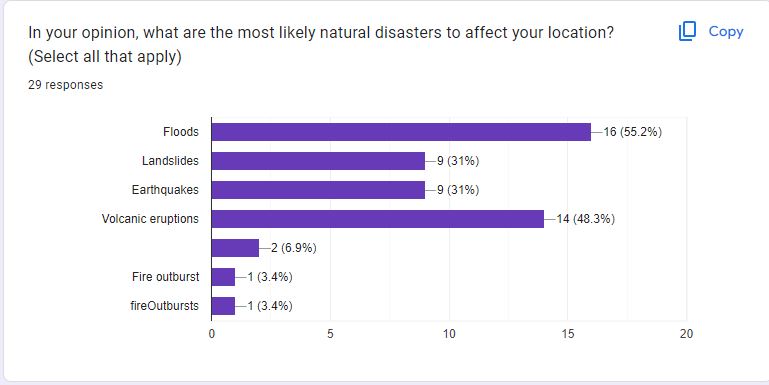
# TOOLS AND METHODS FOR REQUIREMENTS ANALYSIS

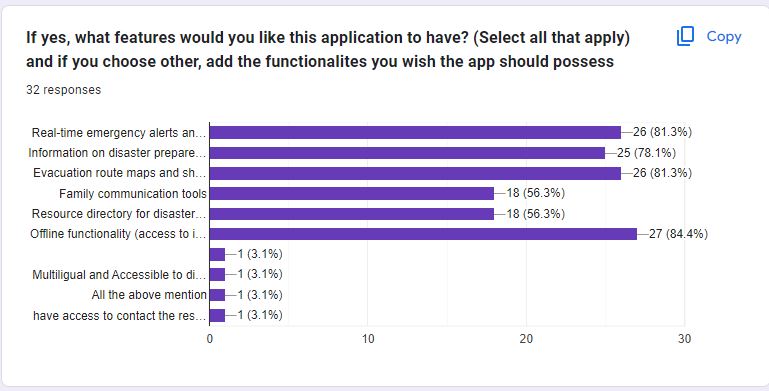
## GOOGLE FORMS

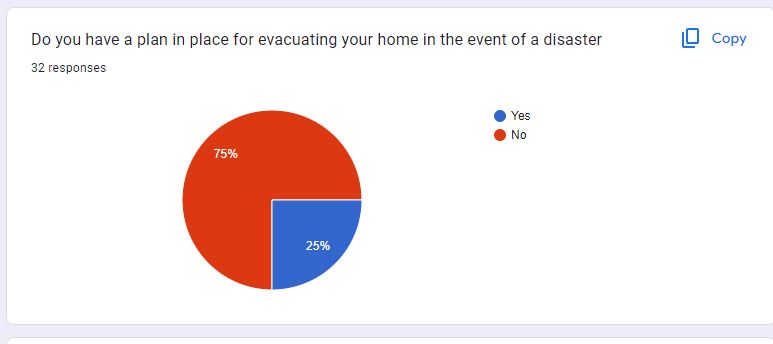
This platform offers a robust suite of built-in analytic tool that proved instrumental in analyzing the survey data. This tool allowed us to:

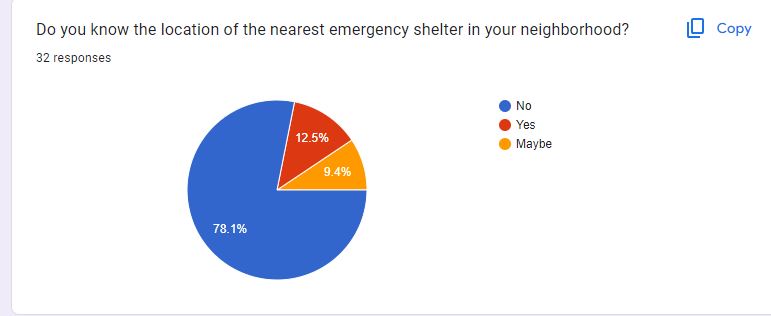
* **Identify trends,** analyze the answer choices and open-ended responses providing insights into user anxieties and concerns, and p**rioritize requirements b**y analyzing the frequency of specific feature requests and the strength of user sentiment surrounding them.
* **Below are some of the images to show the analysis**











## REQUIREMENTS EVALUATION BY BRAINSTORMING

After gathering all our requirements, we evaluated them all by brainstorming using the SMART analysis to ensure the requirements feasibility and alignment with project objectives. Evaluation was also done to see whether or not to consider these requirements during our project development.

**SMART Analysis**

1. Specific (S): Ensure that the requirement is clear, concise, and well-defined.
2. Measurable (M): Make sure the requirement can be quantified or measured.
3. Attainable (A): Evaluate whether the requirement is feasible within the project's constraints (time, budget, resources).
4. Relevant (R): Assess whether the requirement directly contributes to project objectives.
5. Time-Bound (T): Set a timeframe for achieving the requirement and ensure it can be achieved within that time.

Now, in order to analyze these requirements, we need to know what we have gathered so far. Below is a list of all the requirements we got from our requirements gathering process:

1. **User Feedback and Support**
2. **Offline Functionality**
3. **Multi-language Support**
4. **Report Disaster**
5. **Evacuation Routes**
6. **Request Help**
7. Real-time Alerts and Notifications.
8. **Receive Alerts Offline (SMS)**
9. **Usability**
10. Resource Management:
11. Communication:
12. Geospatial Data and Mapping:
13. Incident Reporting:
14. Damage Assessment:
15. Disaster Detection:

We later performed a SMART analysis on all the requirements to evaluate whether or not they are feasible, attainable and non-redundant as shown below:

1. **User Feedback and Support:**
   * **Specific:** Enable users to provide feedback and access support channels.
   * **Measurable:** Track the number of user feedback submissions and support requests.
   * **Achievable:** Feasible to implement feedback forms and support channels.
   * **Relevant:** Relevant for improving system usability and addressing user concerns.
   * **Time-Bound:** Implement within the next development cycle.
2. **Offline Functionality:**
   * **Specific:** Provide offline access to maps, emergency contacts, and saved data.
   * **Measurable:** Test offline functionality in various scenarios.
   * **Achievable:** Feasible with proper design and data synchronization.
   * **Relevant:** Critical for users during emergencies.
   * **Time-Bound:** Ensure offline features are available in the next release.
3. **Multi-language Support:**
   * **Specific:** Support multiple languages for user interfaces and alerts.
   * **Measurable:** Count the number of supported languages.
   * **Achievable:** Possible through localization efforts.
   * **Relevant:** Addresses diverse user needs.
   * **Time-Bound:** Implement language support by the end of the quarter.
4. **Report Disaster:**
   * **Specific:** Enable users to report disaster incidents.
   * **Measurable:** Track the number of reported incidents.
   * **Achievable:** Feasible with incident reporting forms.
   * **Relevant:** Vital for situational awareness and response coordination.
   * **Time-Bound:** Available in the next system update.
5. **Evacuation Routes:**
   * **Specific:** Provide up-to-date evacuation routes.
   * **Measurable:** Validate accuracy of route information.
   * **Achievable:** Possible through geospatial data integration.
   * **Relevant:** Critical for user safety during disasters.
   * **Time-Bound:** Ensure routes are available before the next disaster season.
6. **Request Help:**
   * **Specific:** Allow users to request assistance.
   * **Measurable:** Track help requests.
   * **Achievable:** Implement a help button or chat feature.
   * **Relevant:** Essential for user safety and resource allocation.
   * **Time-Bound:** Available in the next system update.
7. **Real-time Alerts and Notifications:**
   * **Specific:** Provide timely alerts during disasters.
   * **Measurable:** Measure alert delivery time.
   * **Achievable:** Feasible with real-time communication channels.
   * **Relevant:** Critical for situational awareness.
   * **Time-Bound:** Ensure real-time alerts are operational within a month.
8. **Receive Alerts Offline (SMS):**
   * **Specific:** Enable SMS alerts.
   * **Measurable:** Confirm successful SMS delivery.
   * **Achievable:** Possible through SMS gateways.
   * **Relevant:** Addresses connectivity limitations.
   * **Time-Bound:** Implement SMS alerts within two weeks.
9. **Usability:**
   * **Specific:** Ensure an intuitive user interface.
   * **Measurable:** Conduct usability testing and gather feedback.
   * **Achievable:** Achievable through user-centered design.
   * **Relevant:** Enhances user adoption and effectiveness.
   * **Time-Bound:** Continuously improve usability throughout development.
10. **Resource Management:**
    * **Specific:** Efficiently manage resources for each phase.
    * **Measurable:** Monitor resource allocation and utilization.
    * **Achievable:** Requires resource planning and coordination.
    * **Relevant:** Critical for disaster response effectiveness.
    * **Time-Bound:** Implement resource management features by the end of the year.
11. **Communication:**
    * **Specific:** Establish communication channels for each phase.
    * **Measurable:** Evaluate communication effectiveness.
    * **Achievable:** Requires protocols and tools.
    * **Relevant:** Essential for coordination and information sharing.
    * **Time-Bound:** Ensure communication channels are operational before the next disaster event.
12. **Geospatial Data and Mapping:**

* **Specific:** Implement geospatial data integration and mapping technologies.
* **Measurable:** Evaluate accuracy and coverage of mapped data.
* **Achievable:** Feasible through GIS tools and APIs.
* **Relevant:** Critical for situational awareness during disaster response.
* **Time-Bound:** Ensure geospatial features are operational within six months.

1. **Incident Reporting:**

* **Specific:** Provide a mechanism for users to report emergencies.
* **Measurable:** Track the number of incident reports submitted.
* **Achievable:** Implement an incident reporting form or button.
* **Relevant:** Facilitates rapid assessment and resource deployment.
* **Time-Bound:** Available in the next system update.

1. **Damage Assessment:**

* **Specific:** Develop tools for assessing infrastructure damage.
* **Measurable:** Evaluate accuracy and efficiency of damage assessment.
* **Achievable:** Requires collaboration with experts and field data collection.
* **Relevant:** Guides resource allocation and recovery efforts.
* **Time-Bound:** Operational within three months.

1. **Disaster Detection:**

* **Specific:** Implement early detection mechanisms (e.g., sensors, data analytics).
* **Measurable:** Monitor detection accuracy and response time.
* **Achievable:** Requires data integration and predictive models.
* **Relevant:** Mitigates impact by allowing timely intervention.
* **Time-Bound:** Deploy detection systems within a year.

Based on the SMART ANALYSIS, the following requirements were included in our project development:

1. User Feedback and Support
2. User Registration and Authentication
3. Offline Functionality
4. Multi-language Support
5. Evacuation Routes
6. Request Help
7. Real-time Alerts and Notifications
8. Resource Management
9. Communication
10. Geospatial Data and Mapping
11. Incident Reporting
12. Usability
13. Performance
14. Security
15. Reliability
16. Scalability
17. Maintainability
18. Accessibility
19. Interoperability

# CATEGORISING OUR REQUIREMENTS

## FUNCTIONAL REQUIREMENTS

Functional requirements of a system, in simple terms, are the specific tasks and features that the system needs to perform to meet the needs of its users. They define what the system is supposed to do and how it should behave under certain conditions. The functional requirements of our disaster management system are listed below:

* 1. User Registration and Authentication
  2. Real-Time Alerts and Notifications
  3. Incident Reporting
  4. Emergency Resource Access
  5. Communication with Authorities
  6. Geospatial Data Integration
  7. Community Engagement Features
  8. Data Privacy and Security
  9. **Evacuation Routes**
  10. Request Help
  11. Offline Functionality
  12. Multilingual Support
  13. User Feedback and Support.

## NON-FUNCTIONAL REQUIREMENTS

To ensure the effectiveness of our disaster management system, several non-functional requirements are crucial. These non-functional requirements address how the system should perform or behave rather than what specific features it offers.

1. Usability
2. Performance
3. Security
4. Reliability
5. Scalability
6. Maintainability
7. Accessibility
8. Interoperability

# PRIORITISING REQUIREMENTS

Prioritizing the requirements is crucial to ensure the most critical needs are addressed first and determining which requirements should be implemented first and which can be addressed later. By analyzing the urgency, impact, and complexity of the identified requirements, the requirements were prioritized as follows:

1. **High Priority:**
2. Real-time Alerting and Notification System
3. Ability to Report Incidents and Request Assistance
4. Integration with Geospatial Data and Mapping Services
5. Communication with Authorities:
6. Offline Functionality
7. **Medium Priority:**
8. Multifunctional Mobile Application
9. Community Engagement and Collaboration Features
10. User Feedback and Support
11. **Low Priority:**
12. Multi-language Support
13. Receive Alerts Offline (SMS)