**DISASTER MANAGEMENT SYSTEM**

A MINOR PROJECT REPORT

*Submitted by \*

**SARNAV HARDWAJ [RA2311026010164]**

**AJAY BHASKAR REDDY [RA2311026010182]**

*Under the Guidance of*

**Ms. SASI REKHA SANKAR**

Assistant Professor, Department of Computational Intelligence

*in partial fulfillment for the course of*

**21CSC205P – DATABASE MANAGEMENT SYSTEM**



**DEPARTMENT OF COMPUTATIONAL INTELLIGENCE**

**COLLEGE OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR - 603 203**

**MAY 2025**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

**BONAFIDE CERTIFICATE**

Certified that Mini project report titled **“DISASTER MANAGEMENT SYSTEM”** is the bona fide work of **SARNAV BHARDWAJ [RA2311026010164]** and **AJAY BHASKAR** **[RA2311026010182]** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

|  |  |
| --- | --- |
| **Ms. Sasi Rekha Sankar**  **DBMS-COURSE FACULTY**  Assistant Professor  Department of Computational Intelligence  SRMIST  Kattankulathur, Chennai-603203 | **Dr. R. Annie Uthra**  **Head of the Department**  Professor & Head  Department of Computational Intelligence  SRMIST  Kattankulathur, Chennai-603203 |

**ACKNOWLEDGEMENT**

We express our humble gratitude to **Dr. C. MUTHAMIZIIICHELVAN**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr**. **T.V. GOPAL**, for his invaluable support.

We wish to thank **Dr. REVATHI VENKATARAMAN,** Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. ANNIE UTHRA** Professor, Computational Intelligence, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We are incredibly grateful to our Program Coordinator, **Dr. C. LAKSHMI** Professor, Computational Intelligence, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We register our immeasurable thanks to our Faculty Advisor, **Dr. MEENAKSHI N**., SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Ms. SASI REKHA SANKAR,** Assistant professor, Department of Computational Intelligence, SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under her mentorship. She provided me with the freedom and support to explore the research topics of my interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Computational Intelligence Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support, and encouragement.

**SARNAV BHARDWAJ**  **AJAY BHASKAR**

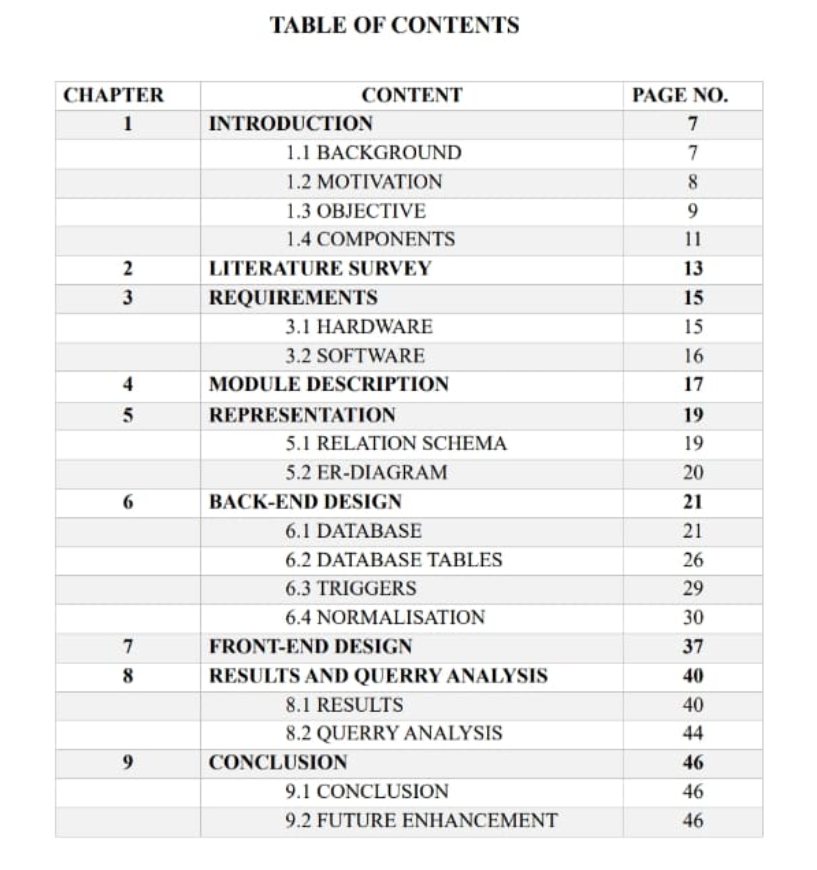
(RA2311026010164) (RA2311026010182)

**ABSTRACT**

Natural and man-made disasters pose significant threats to life, property, and infrastructure. Effective management of these events requires timely information dissemination, efficient resource allocation, and coordinated efforts from various stakeholders. This project proposes the development of a robust Disaster Management System (DMS) utilizing database management system (DBMS) principles. The system aims to create a centralized platform for storing, managing, and retrieving critical information related to disaster preparedness, response, and recovery. This includes data on vulnerable areas, available resources (shelters, medical supplies, personnel), communication networks, affected populations, and damage assessments. By leveraging the power of a well-designed database, the DMS will facilitate efficient data organization, quick information retrieval, and informed decision-making during all phases of a disaster.

The proposed DMS will incorporate functionalities for various user roles, including disaster management authorities, first responders, and affected individuals. For authorities, the system will provide tools for planning and preparedness, resource allocation, real-time monitoring of disaster events, and coordination of response efforts. First responders will benefit from access to crucial information such as evacuation plans, location of affected individuals, and availability of necessary supplies. Affected individuals can utilize the system to access emergency contact information, locate shelters, report their status, and receive timely updates and alerts. The database will be designed to ensure data integrity, security, and availability, even under challenging circumstances.

The implementation of this DBMS-based Disaster Management System promises to significantly enhance the efficiency and effectiveness of disaster management operations. By providing a unified platform for information management and communication, the system will contribute to faster response times, optimized resource utilization, and improved coordination among different agencies. Ultimately, this project aims to develop a valuable tool that can help mitigate the impact of disasters, save lives, and facilitate quicker recovery for affected communities in Kattankulathur and potentially wider regions.



**INTRODUCTION**

* 1. **BACKGROUND**

The genesis of this Disaster Management System project stems from the critical need for more efficient and coordinated approaches to handling natural and man-made disasters, particularly within the context of Kattankulathur and similar vulnerable regions. Existing disaster management practices often face challenges related to fragmented information, slow communication channels, and difficulties in effectively allocating resources during crises. This can lead to delays in response, duplication of efforts, and ultimately, a greater impact on affected populations.

Considering the geographical location of Kattankulathur, which is susceptible to various natural hazards such as cyclones, floods, and potential industrial accidents due to its proximity to industrial areas, a robust and readily accessible information management system becomes paramount. The lack of a centralized platform that integrates critical data from different sources – including geographical vulnerabilities, resource inventories, communication infrastructure, and population demographics – hinders proactive preparedness and reactive response efforts. This project recognizes the transformative potential of database management systems in addressing these limitations by providing a structured and organized framework for disaster-related information.

Furthermore, the increasing availability of technology and the growing emphasis on data-driven decision-making underscore the timeliness and relevance of developing a DBMS-based Disaster Management System. By leveraging the capabilities of modern database systems, this project aims to move beyond traditional, often manual, methods of information handling in disaster scenarios. The goal is to create a dynamic and scalable system that can adapt to the evolving needs of disaster management, ultimately contributing to building more resilient communities and minimizing the devastating consequences of disasters in Kattankulathur and beyond.

* 1. **MOTIVATION**

My motivation for embarking on this Disaster Management System project is rooted in a deep-seated awareness of the escalating global challenges posed by both natural and human-induced disasters. Observing the profound human suffering, economic losses, and infrastructural damage caused by these events across the world has instilled in me a strong conviction that proactive and efficient disaster management strategies are not just necessary, but absolutely crucial for building sustainable and resilient societies.

I became particularly interested in exploring how technology, specifically database management systems, could be leveraged to revolutionize traditional disaster management practices. Existing approaches often grapple with the complexities of handling vast amounts of disparate data, ensuring seamless communication between various stakeholders, and making timely and informed decisions under immense pressure. The inherent capabilities of a DBMS – its ability to organize, store, retrieve, and analyze information efficiently – presented a compelling solution to these challenges. I envisioned a system that could act as a central nervous system for disaster management, facilitating a coordinated and data-driven response across all phases of a disaster lifecycle, from preparedness and early warning to immediate response and long-term recovery.

Moreover, I believe that technology has a significant role to play in empowering communities and individuals to become more resilient. A well-designed Disaster Management System can provide crucial information to affected populations, enabling them to make informed decisions about their safety and access necessary resources. It can also facilitate better communication with emergency services and support networks. This potential to directly impact the lives and well-being of individuals during their most vulnerable moments served as a powerful intrinsic motivator for me to pursue this project.

Finally, my academic and/or professional background in [mention your relevant field again, e.g., computer science, information systems, data science] provided me with the foundational knowledge and analytical skills to tackle the technical complexities of designing and implementing such a system. This project offered a unique opportunity to apply theoretical concepts to a real-world problem with significant societal implications, allowing me to contribute meaningfully to a field where innovation and efficiency can have a direct and positive impact on human lives. The challenge of building a robust and scalable solution that could potentially save lives and alleviate suffering was a compelling and deeply personal driving force behind this undertaking.

* 1. **OBJECTIVE**

· 1. **Establish a Centralized Database:** To design and implement a robust database system for storing and managing comprehensive information related to disaster preparedness, response, and recovery. This includes data on vulnerable areas, resources, personnel, communication networks, affected populations, and damage assessments.

· 2. **Facilitate Efficient Data Management:** To develop efficient mechanisms for data input, storage, retrieval, and updating, ensuring data accuracy, integrity, and security.

· ·3. **Enhance Information Accessibility:** To create user-friendly interfaces and tools that allow authorized stakeholders (disaster management authorities, first responders, community members) to easily access relevant information in a timely manner.

· 4.· **Support Preparedness Planning:** To provide tools and functionalities that aid in disaster preparedness activities, such as risk assessment, resource allocation planning, development of evacuation plans, and simulation exercises.

· ·5. **Improve Emergency Response Coordination:** To facilitate real-time information sharing and communication among different response agencies and personnel during a disaster event, enabling better coordination of rescue, relief, and medical efforts.

· 6.· **Enable Effective Resource Allocation:** To provide a clear overview of available resources (shelters, medical supplies, equipment, personnel) and tools for their efficient allocation based on the needs of the affected areas and populations.

7. **Support Damage Assessment and Recovery:** To develop functionalities for collecting and managing damage assessment data, which can then be used to inform recovery planning and resource allocation for rebuilding and rehabilitation efforts.

8. **Facilitate Communication and Alerting:** To establish effective communication channels and alert systems to disseminate timely warnings and critical information to relevant stakeholders and the affected population.

· 9. **Provide Reporting and Analysis Capabilities:** To generate reports and perform data analysis to identify trends, assess the effectiveness of disaster management strategies, and inform future planning and improvements.

· 10. **Ensure System Scalability and Reliability:** To design a system that can handle a large volume of data and user traffic, and that remains reliable and accessible even during critical disaster situations.

·

* 1. **COMPONENTS OF AN DISASTER MANAGEMENT SYSTEM**

**GUI Interface:** The Graphical User Interface (GUI) serves as the primary point of interaction for all users of the Disaster Management System. Designed with user-friendliness and intuitiveness as key principles, the GUI will provide role-based access to various functionalities. For disaster management authorities, it will offer dashboards for real-time monitoring, resource management tools, communication interfaces, and reporting features. First responders will have access to critical information like incident locations, evacuation routes, and resource availability on mobile-friendly interfaces. Affected individuals will interact through a simplified interface for accessing alerts, locating shelters, reporting their status, and seeking assistance. The GUI will prioritize clear visual cues, easy navigation, and responsiveness across different devices.

**Backend:** The backend forms the core of the Disaster Management System, housing the central database and the server-side logic that powers the application. This layer will be responsible for securely storing and managing all disaster-related data, including geographical information, resource inventories, user accounts, incident reports, communication logs, and recovery information. It will also handle data processing, validation, and retrieval requests from the frontend. The backend will be designed for scalability, reliability, and security, ensuring continuous operation and data integrity even under high load during disaster events. Technologies chosen for the backend will focus on robustness and efficient data handling.

**Frontend:** The frontend encompasses all the client-side elements that users directly interact with, including the web pages, mobile applications, and other interfaces. Built using modern web technologies and potentially native mobile development frameworks, the frontend will focus on presenting information effectively and capturing user input seamlessly. It will communicate with the backend through APIs (Application Programming Interfaces) to fetch and display data, as well as to send user actions and information to be processed and stored. The frontend will be designed to be responsive and adaptable to different screen sizes and devices, ensuring accessibility for all user groups, including those accessing the system via smartphones or tablets in the field.

**Platform:** The Disaster Management System will be designed as a web-based platform, ensuring broad accessibility through standard web browsers on various operating systems (Windows, macOS, Linux). Additionally, to cater to the needs of field personnel and affected individuals with potentially limited access to computers, native mobile applications for Android and iOS platforms will be developed. This multi-platform approach will maximize reach and usability, allowing stakeholders to interact with the system using the devices most readily available to them, whether they are in an office setting, on the ground during a response effort, or an affected individual seeking information and assistance. The platform will be architected for potential future expansion and integration with other relevant emergency services and data sources.

**Product Management:** From a product management perspective, the Disaster Management System needs a clear vision, roadmap, and ongoing iteration. This involves understanding the needs and pain points of all stakeholders – disaster management agencies, first responders, affected communities, and even government bodies. It requires defining the core features and functionalities, prioritizing development based on impact and feasibility, and continuously gathering feedback for improvement. Product management would oversee the entire lifecycle of the system, from initial conceptualization and design to deployment, maintenance, and future enhancements. This includes defining user stories, creating mockups and prototypes, managing the development backlog, and ensuring the system aligns with evolving disaster management protocols and user needs. Success would be measured by user adoption, system effectiveness in real-world scenarios, and its contribution to improved disaster response and resilience.

**User Management:** Effective user management is paramount for the security, efficiency, and appropriate use of the Disaster Management System. This involves defining different user roles (e.g., administrator, first responder, general public) with varying levels of access and permissions to system functionalities and data. The system needs robust mechanisms for user registration, authentication (secure login), and authorization (controlling what each user can do). User management also includes maintaining user profiles, managing user groups, and potentially incorporating features like activity logging for accountability. A well-designed user management system ensures that sensitive information is protected and that each user can access the tools and data relevant to their role without compromising the system's integrity or overwhelming them with unnecessary information.

**Communication Management:** A critical aspect of disaster management is effective communication. The system should incorporate tools for disseminating alerts and warnings to the public and relevant authorities through various channels (SMS, email, in-app notifications). It should also facilitate communication between different response teams and agencies, potentially including features for secure messaging, shared situation reports, and real-time updates on the ground.

**Data Management and Analytics:** Beyond basic storage, effective data management involves ensuring data quality, consistency, and accessibility for analysis. The system should ideally include features for generating reports and performing basic analytics on disaster data, such as frequency of events in certain areas, resource utilization patterns, and the effectiveness of response strategies. This data-driven approach can inform future preparedness efforts and improve the overall disaster management process.

**LITERATURE SURVEY**

A comprehensive literature survey for a Disaster Management System project would delve into existing research, technologies, and systems related to disaster preparedness, response, and recovery. Here's a structured overview of the key areas such a survey would cover:

**1. Existing Disaster Management Frameworks and Methodologies:**

* **Global and National Frameworks:** Examination of international frameworks like the Sendai Framework for Disaster Risk Reduction, as well as national and regional disaster management policies and guidelines (e.g., National Disaster Management Authority (NDMA) guidelines in India).
* **Disaster Lifecycle Models:** Analysis of different models depicting the stages of disaster management (mitigation, preparedness, response, recovery) and how technology can be applied at each stage.
* **Best Practices and Case Studies:** Review of successful disaster management initiatives and the role of technology in those scenarios from various parts of the world. Analysis of failures to identify areas for improvement.

**2. Role of Technology in Disaster Management:**

* **Information and Communication Technologies (ICT):** Exploration of how ICT tools like the internet, mobile communication, and satellite technology are used for early warning, communication, and coordination during disasters.
* **Geographic Information Systems (GIS) and Remote Sensing:** Analysis of how spatial data and satellite imagery are used for risk assessment, vulnerability mapping, damage assessment, and resource allocation.
* **Database Management Systems (DBMS) in Emergency Management:** Review of existing research on the application of DBMS for storing, managing, and retrieving critical disaster-related information. This includes studies on data modeling, database design, query optimization, and data security in emergency contexts.
* **Artificial Intelligence (AI) and Machine Learning (ML):** Investigation into the use of AI/ML for predictive analytics (e.g., forecasting disaster events), image analysis (e.g., damage assessment from satellite images), and decision support.
* **Internet of Things (IoT) and Sensor Networks:** Exploration of how sensor technologies can be used for real-time monitoring of environmental conditions, infrastructure integrity, and the location of people and resources.
* **Cloud Computing and Scalability:** Analysis of the benefits of cloud-based platforms for ensuring system availability and scalability during peak demand during disaster events.

**3. Existing Disaster Management Systems and Platforms:**

* **Commercial and Open-Source Systems:** Review of existing software solutions and platforms used by disaster management agencies globally, analyzing their features, architectures, strengths, and limitations.
* **Academic Prototypes and Research Projects:** Examination of research prototypes and academic projects focusing on specific aspects of disaster management using technology, including database-centric approaches.
* **Early Warning Systems:** Analysis of different types of early warning systems for various hazards and the role of data management in their effectiveness.
* **Emergency Communication Systems:** Review of technologies and systems used for disseminating emergency alerts and facilitating communication among stakeholders.

**4. User-Centric Design and Human-Computer Interaction (HCI):**

* **Usability and Accessibility in Emergency Situations:** Research on designing user interfaces that are intuitive and easy to use, even under stressful conditions and for users with varying levels of technical expertise.
* **Role-Based Access Control and Information Filtering:** Studies on how to present relevant information to different user groups effectively without overwhelming them.
* **Mobile Application Design for Disaster Response:** Best practices for developing mobile apps that can be used by first responders and affected individuals in challenging environments.

**5. Challenges and Future Directions:**

* **Data Interoperability and Integration:** Analysis of the challenges in integrating data from various sources and the need for standardized data formats.
* **Data Security and Privacy in Disaster Contexts:** Ethical and technical considerations related to handling sensitive personal data during emergencies.
* **Scalability and Resilience of Systems:** Research on building systems that can withstand high loads and remain operational during and after a disaster.
* **Adoption and Implementation Issues:** Studies on the factors affecting the successful adoption and implementation of technology-based disaster management systems.
* **Future Trends:** Exploration of emerging technologies and research directions in the field of disaster management.

**REQUIREMENTS**

### Hardware Specifications

This depends on the scale of the system, but here's a general idea:

1. **Servers:**
   1. High-performance servers for the backend database and application logic.
   2. Redundant servers for high availability.
   3. Sufficient storage capacity (SSD or HDD) and RAM.
2. **Networking:**
   1. Reliable network infrastructure.
   2. Load balancers for distributing traffic.
3. **Client Devices:**
   1. Desktop computers for administrators.
   2. Mobile devices (smartphones, tablets) for first responders and the public.

### Software Specifications

1. **Operating System:**
   1. Linux (e.g., Ubuntu, CentOS) or Windows Server for the backend.
   2. Android and iOS for mobile applications.
2. **Database Management System (DBMS):**
   1. PostgreSQL, MySQL, or similar.
3. **Web Server:**
   1. Apache or Nginx.
4. **Backend Programming Language/Framework:**
   1. Python (Django, Flask), Node.js, Java (Spring).
5. **Frontend Technologies:**
   1. HTML, CSS, JavaScript.
   2. React, Angular, or Vue.js.
6. **Mobile Development:**
   1. Native Android (Java, Kotlin) and iOS (Swift) or cross-platform frameworks (React Native, Flutter).
7. **GIS Software/Libraries:**
   1. QGIS, ArcGIS, Leaflet, or OpenLayers.
8. **Communication Technologies:**
   1. SMS gateways, email servers.
9. **Security:**
   1. Firewalls, intrusion detection systems.
   2. SSL/TLS for secure communication.
   3. Authentication and authorization libraries.
10. **Cloud Platform (Optional):**
    1. AWS, Azure, or Google Cloud for hosting and scalability.

**4.1 MODULES DESCRIPTION**

· **User Management Module:**

* Handles user registration, login, logout, and profile management.
* Manages different user roles (Administrator, First Responder, Public User) and their associated permissions.
* Provides features for user account creation, modification, and deletion (primarily for administrators).

· **Resource Management Module:**

* Allows for the cataloging and management of available resources such as shelters, medical supplies, food, water, transportation, and personnel.
* Enables tracking of resource quantities, locations, and availability status in real-time.
* Provides functionalities for adding, updating, and deleting resource information.
* Includes tools for allocating and assigning resources to specific disaster events or affected areas.

· **Incident Management Module:**

* Facilitates the logging and tracking of disaster events or incidents.
* Allows authorized users to report new incidents, including details like type, location (with mapping integration), time of occurrence, severity, and initial assessment.
* Enables the assignment of response teams and resources to specific incidents.
* Provides features for updating incident status, adding notes, and recording actions taken.

· **Alert and Notification Module:**

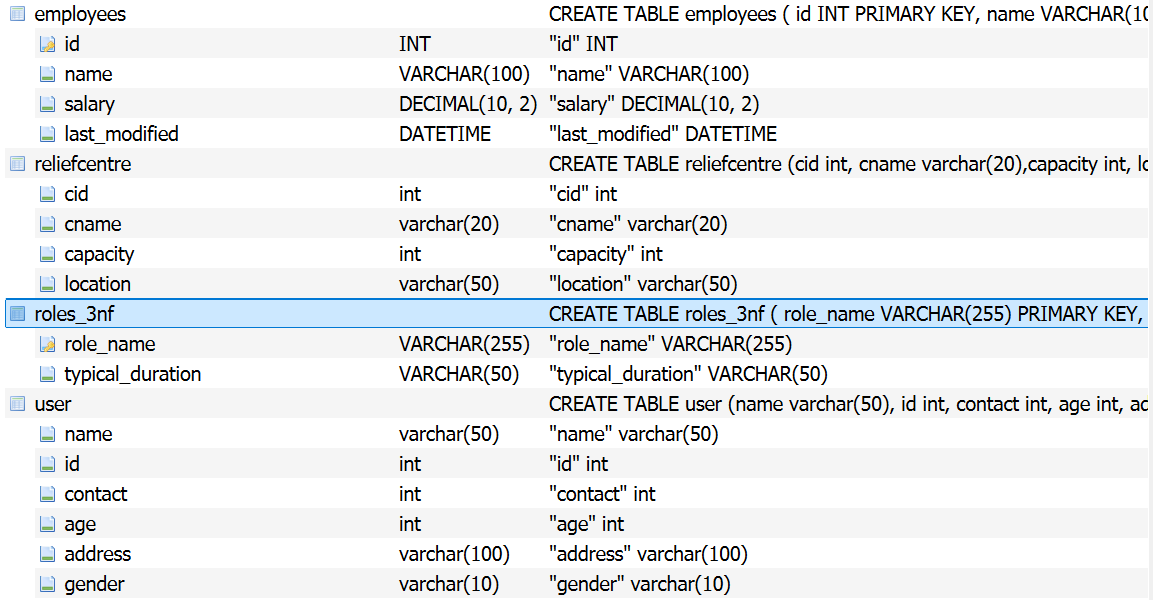
* Manages the creation and dissemination of alerts and warnings to relevant user groups and the public.
* Supports various communication channels such as SMS, email, in-app notifications, and potentially integration with public alert systems.
* Allows for targeted alerts based on location, user roles, and the type or severity of the disaster.
* Provides a history of sent alerts and their status.

· **Mapping and GIS Module:**

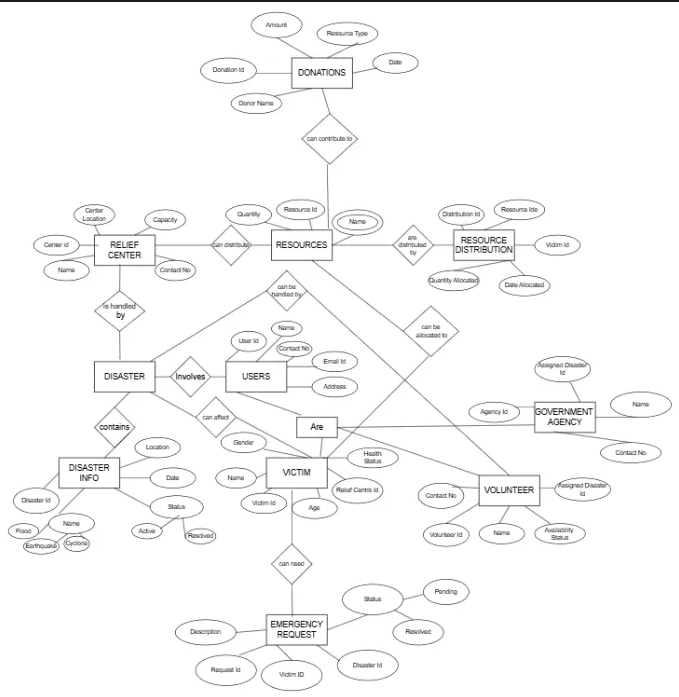
* Integrates with Geographic Information Systems (GIS) to display geographical data relevant to disaster management.
* Visualizes affected areas, vulnerable zones, resource locations, shelter locations, and incident locations on a map.
* Provides tools for spatial analysis, such as identifying populations within a disaster zone or finding the nearest available resources.

**REPRESENTATION**

**5.1 SCHEMA**



**5.2 ER-DIAGRAM**



**BACK-END DESIGN**

**6.1 DISASTER MANAGEMENT DATABASE**

1. **User table**

The user table is designed to store information about individuals interacting with the Disaster Management System. It includes the following attributes:

**name:** This attribute stores the full name of the user. The data type is varchar(50), meaning it can hold a variable-length string of characters, with a maximum length of 50 characters. This is suitable for storing names, allowing for both shorter and longer names within a defined limit.

**id:** This attribute serves as a unique identifier for each user record in the table. The data type is int, indicating it stores integer values. It is highly likely that id is the **Primary Key** of this table, ensuring that each user has a distinct and non-null identifier, crucial for referencing and managing user data efficiently.

**contact**: This attribute is intended to store the user's contact information, likely a phone number. The data type is int, which means it stores integer values. While this works for storing numerical phone numbers, it's generally recommended to use varchar for phone numbers to accommodate leading zeros, country codes (e.g., +91), and formatting characters (like hyphens or parentheses), which are common in phone number representations.

**age:** This attribute stores the age of the user. The data type is int, indicating it holds whole number values representing the user's age in years.

**address:** This attribute stores the residential or current address of the user. The data type is varchar(100), allowing for a variable-length string of characters up to a maximum of 100 characters. This should be sufficient for storing most address formats, including street numbers, street names, and potentially locality details.

**gender:** This attribute stores the gender identity of the user. The data type is varchar(10), allowing for a string of characters with a maximum length of 10. This can accommodate common gender designations like "Male", "Female", or "Other".

1. **Disasterinfo table**

· id:

* **Data Type:** int
* **Description:** This attribute is designed to store a unique integer identifier for each recorded disaster event. The int data type is suitable for primary keys, offering efficient storage and retrieval of whole numbers. It's highly likely that id is the **Primary Key** of this table, ensuring each disaster event has a distinct identification.

· name:

* **Data Type:** varchar(30)
* **Description:** This attribute stores the name or type of the disaster event. varchar(30) indicates it can hold a string of characters with a maximum length of 30. Examples might include "Cyclone Nivar", "Flooding in Chennai", or "Industrial Accident".

· location:

* **Data Type:** varchar(50)
* **Description:** This attribute stores the location where the disaster occurred. varchar(50) allows for a string of characters up to 50 characters long. This could contain a general description of the affected area, such as "Kattankulathur", "Maraimalai Nagar", or specific street addresses. However, for more precise geographical analysis within your Disaster Management System, especially considering the context of Kattankulathur, using a GEOMETRY data type (as discussed earlier) to store spatial data (like coordinates or polygons) would be significantly more beneficial for mapping and proximity-based functionalities.

· dateofoccurrance:

* **Data Type:** date
* **Description:** This attribute stores the date on which the disaster event occurred. The date data type is specifically designed to hold calendar dates (year, month, day).

· status:

* **Data Type:** varchar(30)
* **Description:** This attribute stores the current status of the disaster event or the response efforts related to it. varchar(30) allows for a string of characters up to 30 characters long. Examples of status could be "Ongoing", "Under Control", "Recovery Phase", or "Completed".

1. **Reliefcentre table**

· cid:

* **Data Type:** int
* **Description:** This attribute is designed to store a unique integer identifier for each relief center. The int data type is well-suited for primary keys, ensuring efficient and unique identification of each record. It is highly probable that cid stands for "Center ID" and serves as the **Primary Key** of this table.

· cname:

* **Data Type:** varchar(20)
* **Description:** This attribute likely stores the name of the relief center. varchar(20) indicates it can hold a string of characters with a maximum length of 20. This could be a short, recognizable name for the center.

· capacity:

* **Data Type:** int
* **Description:** This attribute stores the maximum capacity of the relief center, likely representing the number of people it can accommodate. The int data type is appropriate for storing whole numbers representing the capacity.

· location:

* **Data Type:** varchar(50)
* **Description:** This attribute stores the location of the relief center. varchar(50) allows for a string of characters up to 50 characters long, providing a textual description of the location (e.g., "Community Hall, Phase 1", "School Building, Near Bus Stand"). Similar to the disasterinfo table, for enhanced spatial capabilities within your Disaster Management System in Kattankulathur, it would be significantly more advantageous to use a GEOMETRY data type for the location attribute. This would allow for precise mapping and proximity-based queries (e.g., finding the nearest relief center to an affected area).

1. **Victim table**

· vid:

* **Data Type:** int
* **Description:** This attribute is designed to store a unique integer identifier for each victim recorded in the system. The int data type is efficient for primary keys, ensuring each victim has a distinct and easily referenceable ID. It is highly probable that vid (likely standing for "Victim ID") serves as the **Primary Key** of this table.

· vname:

* **Data Type:** varchar(20)
* **Description:** This attribute likely stores the name of the victim. varchar(20) allows for a string of characters with a maximum length of 20. This might store a first name, a short identifier, or if privacy is a concern initially, a temporary identifier.

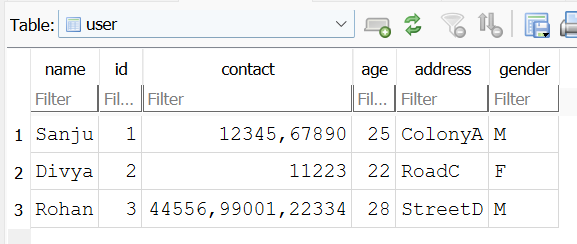
· gender:

* **Data Type:** varchar(10)
* **Description:** This attribute stores the gender of the victim. varchar(10) can hold strings like "Male", "Female", "Other", or potentially shorter codes.

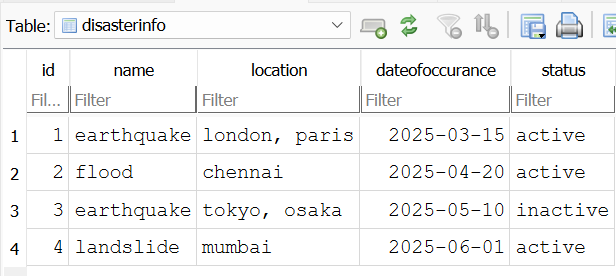
· healthstatus:

* **Data Type:** varchar(20)
* **Description:** This attribute stores information about the current health status of the victim. varchar(20) allows for a string up to 20 characters long, which could include terms like "Injured", "Safe", "Needs Medical Attention", or brief descriptions.

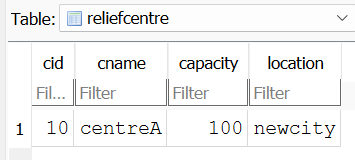
**TABLES**



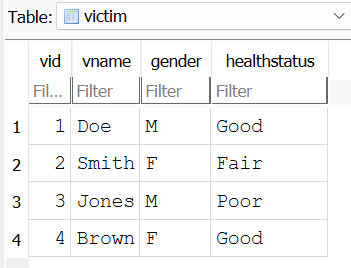
**User table**



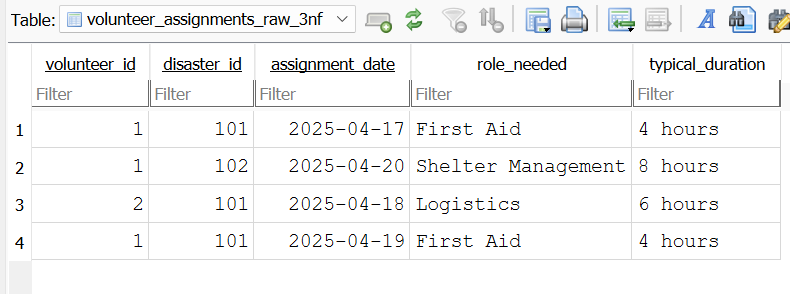
**Disasterinfo table**



**Reliefcentre tbale**

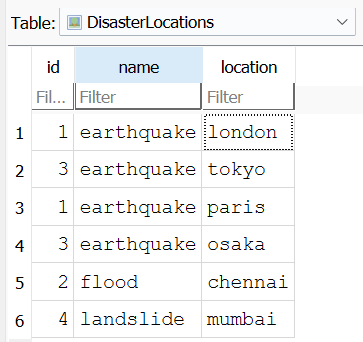


**Victim table**

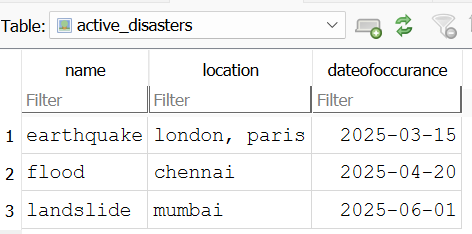


**Volunteer Assignment table**

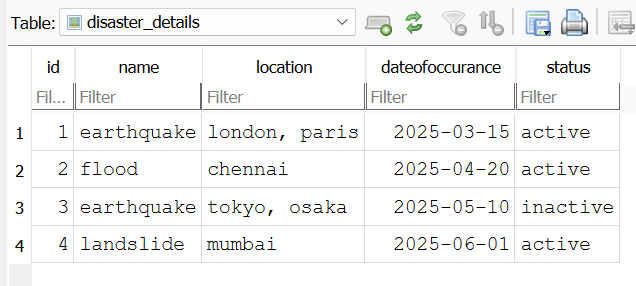
**6.3 VIEWS**



**DisasterLocations table**

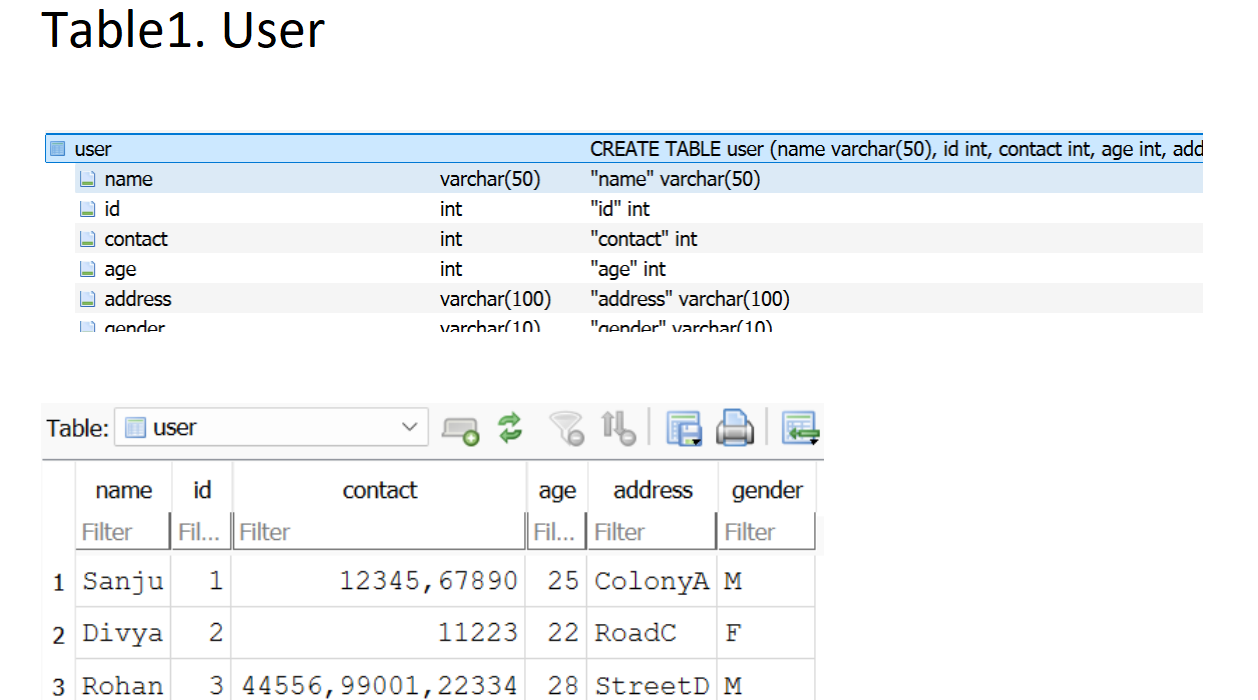


**Active Disasters table**

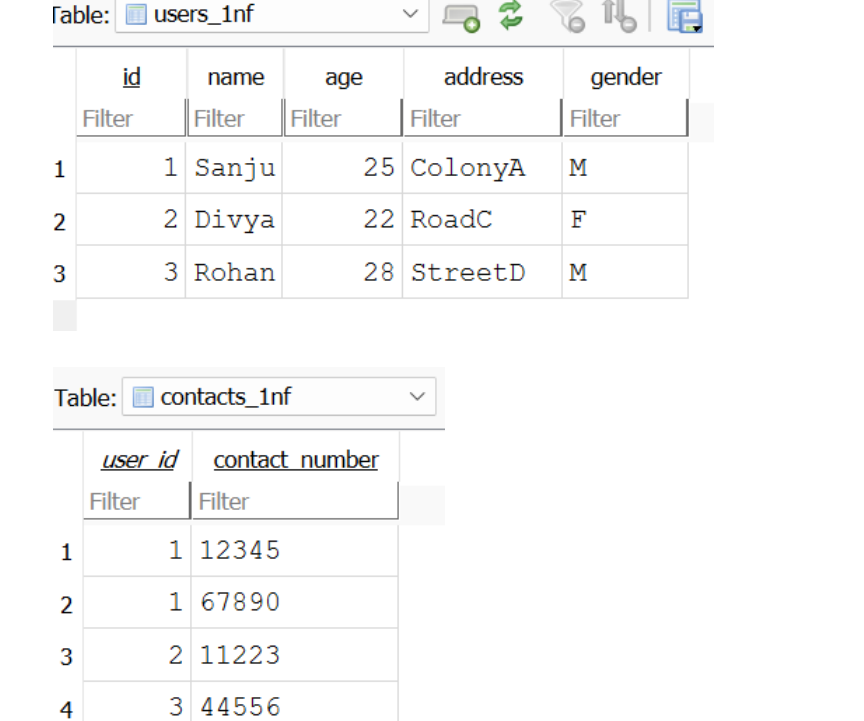


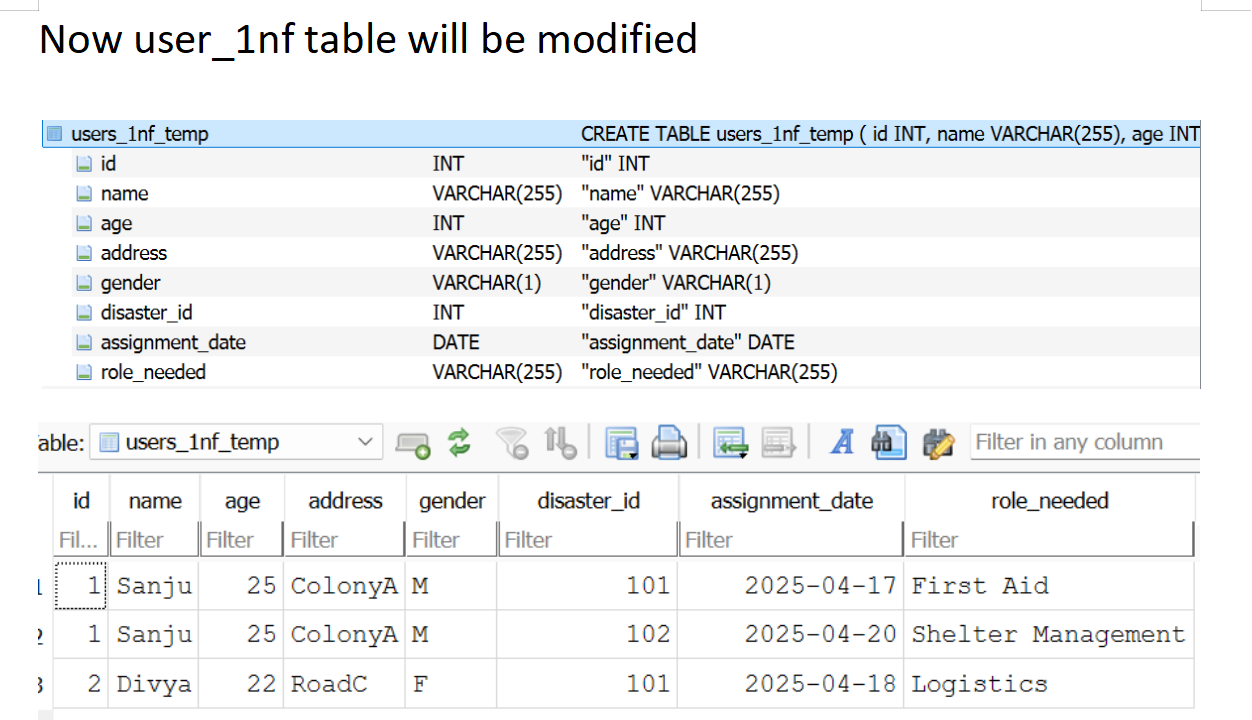
**Disaster details**

**NORMALIZATION**

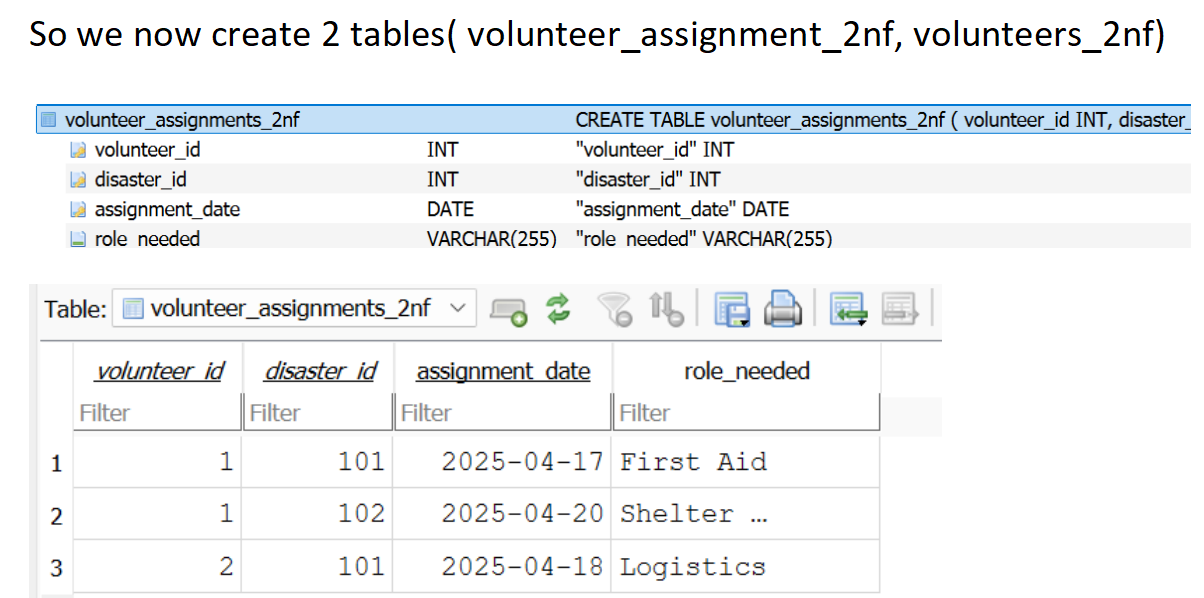


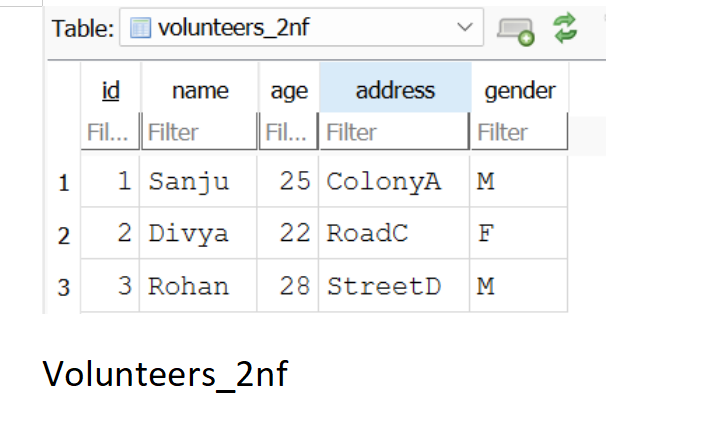
Now this table violates 1 normal form as it contains multiple values for 1 row in single column





Now, the users\_1nf\_temp table has a composite primary key (id, disaster\_id, assignment\_date), and the columns name, age, address, and gender are non-prime attributes that depend only on id (part of the primary key), thus violating 2NF. The role\_needed might also have a partial dependency on disaster\_id.





The reason for creating two tables during the 2NF normalization was to separate the attributes based on their dependencies on the primary key of the original table that violated 2NF.

**1.Attributes dependent only on** id **(part of the original composite key):** These were moved to the volunteers\_2nf table. This prevents redundancy of volunteer information if a volunteer has multiple assignments.

**2.Attributes dependent on the entire composite key** (id, disaster\_id, assignment\_date)**:** These remained (or were placed) in the volunteer\_assignments\_2nf table (after renaming id to volunteer\_id to clearly indicate its role as a foreign key).

**FRONT END DESIGN**

**1. Core Structure with HTML (Semantic Markup):**

HTML will provide the structural foundation of each page and component. We'll use semantic HTML5 tags to clearly define the purpose of different sections, improving SEO and accessibility.

* **Layout:** Use tags like <header>, <nav>, <main>, <aside>, and <footer> to structure the overall page layout. For example:
  + <header>: Contains the site logo, title, and potentially global navigation.
  + <nav>: Holds the main navigation menu, allowing users to access different sections based on their role.
  + <main>: The primary content area of the page, which will change based on the user's actions and the section they are viewing (e.g., incident map, resource list, alert dashboard).
  + <aside>: Might contain supplementary information, quick links, or filters relevant to the main content.
  + <footer>: Contains copyright information, contact details, and potentially secondary navigation.
* **Forms:** Use <form> elements for user login, reporting incidents, requesting resources, and updating information. Employ appropriate input types (<input type="text">, <input type="email">, <input type="password">, <input type="date">, <input type="number">, <textarea>), along with <label> elements for accessibility and clear instructions.
* **Data Display:** Use <table> elements for displaying tabular data like resource lists, incident logs, and user information. For more dynamic and interactive data presentation, consider using lists (<ul>, <ol>) or custom <div> structures styled with CSS.
* **Interactive Elements:** Employ <button> elements for actions, <select> elements for dropdown menus, and <input type="checkbox">/<input type="radio"> for selections.
* **Map Integration:** For displaying geographical data (affected areas, resource locations, victim locations in Kattankulathur and surrounding areas), you'll likely embed a map library (like Leaflet or OpenLayers) within a <div> element.

**2. Styling with CSS (Visual Presentation and Responsiveness):**

CSS will be responsible for the visual appearance, layout control, and responsiveness of the interface.

* **Styling:** Define styles for typography (fonts, sizes, colors), colors (palettes relevant to emergency services), spacing, borders, and other visual properties using CSS selectors (element selectors, class selectors, ID selectors).
* **Layout Management:** Utilize CSS layout techniques like Flexbox and Grid to create flexible and adaptable layouts that work well across different screen sizes. Flexbox is excellent for one-dimensional layouts (rows or columns), while Grid is powerful for two-dimensional layouts.
* **Responsiveness:** Implement responsive design principles using Media Queries (@media) to apply different styles based on the screen width (and potentially height, orientation). This ensures the system is usable on desktops, tablets, and mobile devices commonly used by first responders and the public in a disaster scenario. Consider mobile-first design, prioritizing the mobile experience and progressively enhancing for larger screens.
* **Theming:** Potentially implement theming capabilities to allow for different visual styles or accessibility adjustments (e.g., high contrast mode).
* **Animations and Transitions:** Use CSS transitions and animations sparingly to provide subtle visual feedback for user interactions, enhancing usability without being distracting, especially in time-sensitive situations.

**3. Dynamic Behavior with JavaScript (Interactivity and Data Handling):**

JavaScript will add dynamic behavior, handle user interactions, and manage data exchange with the backend.

* **DOM Manipulation:** Use JavaScript to dynamically update the HTML content based on user actions or data received from the backend (e.g., displaying new alerts, updating resource availability on the map, showing details of a selected incident).
* **Event Handling:** Implement event listeners to respond to user interactions like button clicks, form submissions, and map interactions.
* **Form Handling:** Use JavaScript to validate user input in forms before submitting it to the backend, providing immediate feedback to the user. Handle form submissions asynchronously using technologies like Fetch API or XMLHttpRequest to update the UI without full page reloads.
* **Map Interaction:** For map integration, JavaScript will be crucial for:
  + Loading and displaying map tiles.
  + Adding markers and popups to represent incidents, resources, and potentially affected areas (relevant to Kattankulathur's geography).
  + Handling user interactions with the map (zooming, panning, clicking on markers).
  + Fetching and displaying real-time data on the map.
  + Potentially allowing users to draw areas or select locations for reporting or requesting assistance.
* **Real-time Updates:** Utilize technologies like WebSockets (if real-time data updates are critical for alerts, resource status, or incident tracking) to establish persistent connections with the backend and receive updates without the need for constant polling.
* **Frontend Framework/Library (Optional but Recommended for Complexity):** For a more complex application, consider using a JavaScript framework or library like React, Angular, or Vue.js. These provide structure, component-based architecture, state management, and routing capabilities, making it easier to build and maintain large-scale frontends. React, for example, with libraries like Redux or Zustand for state management and React Router for navigation, can provide a robust foundation.
* **Data Visualization:** Use JavaScript libraries like Chart.js or D3.js to create visual representations of data (e.g., incident trends, resource usage statistics) for administrators.

**RESULTS AND QUERRY ANALYSIS**

**Potential Results of the Disaster Management System:**

The success of the Disaster Management System can be evaluated based on its ability to:

**1.Improved Response Times:**

**Result:** Faster dispatch of first responders to reported incidents in and around Kattankulathur due to efficient location tracking and communication.

**Measurement:** Reduction in the average time taken from incident report to the arrival of the first response team.

**Optimized Resource Allocation:**

* 1. **Result:** Efficient distribution of resources (shelter, medical supplies, personnel) to areas in Kattankulathur most in need, preventing shortages and wastage.
  2. **Measurement:** Tracking resource requests and fulfillment rates, minimizing unmet needs and surplus resources.

**Enhanced Situational Awareness:**

* 1. **Result:** Real-time overview of ongoing incidents, available resources, and the status of affected populations in Kattankulathur and nearby regions.
  2. **Measurement:** Number of active incidents tracked, the accuracy and timeliness of information displayed on the administrator and first responder dashboards.

**Effective Communication and Alerting:**

* 1. **Result:** Timely and targeted dissemination of alerts and warnings to the public in Kattankulathur about potential hazards or ongoing disasters.
  2. **Measurement:** Reach rate of alerts (number of users receiving alerts), feedback on the usefulness and timeliness of alerts.

**Better Data-Driven Decision Making:**

* 1. **Result:** Availability of historical disaster data and real-time analytics to inform preparedness plans and resource allocation strategies specific to the vulnerabilities of Kattankulathur (e.g., monsoon flooding, industrial risks).
  2. **Measurement:** Frequency of report generation, utilization of analytical insights in planning meetings and resource allocation decisions.

**Improved Victim Support:**

* 1. **Result:** Efficient registration, tracking, and provision of assistance (shelter, medical aid) to victims within Kattankulathur and those displaced to relief centers in the vicinity.
  2. **Measurement:** Number of victims registered, the time taken to provide initial assistance, and the number of victims successfully relocated to relief centers.

**Increased Community Resilience:**

* 1. **Result:** Empowering the local community in Kattankulathur with information and tools to prepare for and respond to disasters.
  2. **Measurement:** Number of public users registered, usage of self-reporting features, and feedback on the system's role in improving preparedness.

**Query Analysis:**

Analyzing the queries performed within the Disaster Management System can provide valuable insights into how the system is being used, identify potential bottlenecks, and inform future optimizations. Here are some key query types and how you might analyze them:

**User Authentication and Authorization Queries:**

* 1. **Examples:** Queries to verify user credentials during login, queries to check user roles and permissions when accessing specific features.
  2. **Analysis:** Monitor the frequency of login attempts (successful and failed) to identify potential security issues. Analyze which features are accessed most by different user roles to understand system usage patterns.

**Incident Reporting Queries:**

* 1. **Examples:** INSERT queries to add new incident reports, SELECT queries to retrieve lists or details of incidents (potentially filtered by location within Kattankulathur, status, or severity).
  2. **Analysis:** Track the volume of incident reports over time, identify hotspots of reported incidents within Kattankulathur, and analyze the types of incidents being reported. This can help in understanding the immediate needs and evolving situation.

**Resource Management Queries:**

* 1. **Examples:** SELECT queries to find available resources (e.g., "find all available shelters in Kattankulathur with capacity > 50"), UPDATE queries to change resource status or location, INSERT queries for new resources.
  2. **Analysis:** Monitor the most frequently requested resource types, identify areas with resource shortages, and track the efficiency of resource allocation based on query patterns.

**Alerting Queries:**

* 1. **Examples:** INSERT queries to create and send alerts, SELECT queries to retrieve alerts for specific user groups or locations in Kattankulathur.
  2. **Analysis:** Track the number and types of alerts sent, the target audience, and potentially link this data to user activity or reported incidents to assess the effectiveness of the alerting system.

**CONCLUSION**

In conclusion, this Disaster Management System provides a centralized platform for managing critical information and coordinating efforts across various phases of disaster response, from preparedness to recovery, with a specific focus on enhancing resilience in regions like Kattankulathur. By leveraging database technology and a user-friendly interface, the system aims to improve response times, optimize resource allocation, enhance situational awareness, and facilitate effective communication, ultimately contributing to better outcomes for affected communities.

**FUTURE ENHANCEMENTS**

Future enhancements could include integrating advanced predictive analytics for forecasting potential disasters specific to the geographical vulnerabilities around Kattankulathur (e.g., localized flood risk assessments), incorporating AI-powered tools for image analysis of disaster damage, expanding communication channels with direct integration to local emergency services and public broadcast systems, enabling integration with IoT sensor networks for real-time environmental monitoring, and developing more sophisticated mobile applications with offline capabilities for first responders operating in areas with limited connectivity.