# Disaster Relief Supply System - Documentation

## Overview

This system is designed to manage disaster relief operations efficiently. It enables users to request resources, assign volunteers, and allocate inventory based on priority and proximity. The system uses CustomTkinter for UI, MySQL for data storage, and geopy for location-based computations. With this system we can request for the resources from the inventory of major relief centers and we can also request for certain number of the volunteers for the disaster, volunteers are assigned efficiently by how close they are to the corresponding relief centers. When the volunteers are assigned a mail is sent to alert them. The system also provides a way for common people to volunteer. The Government can also restock the items in specific relief centers. We have compared the three methods first-come-first-serve, priority-based, and location-based distribution by giving option to view and allocate the resources based on the three methods and which one they prefer. When the stock in any relief centre becomes less or empty an alert mail is sent to the authorized person to refill.

## Key Features

- Database Management: Handles inventory, requests, volunteers, and locations.  
- CustomTkinter GUI: Provides an interactive interface for users.  
- Automated Resource Allocation: Assigns inventory and volunteers based on demand.  
- Geographical Optimization: Finds the closest available resources.

## System Workflow

1. User Reports a Disaster -> Enters disaster details via UI.  
2. Requests Processed Periodically -> Every minute, the system evaluates pending requests.  
3. Priority-Based Sorting -> Requests are sorted by urgency, severity, and affected population.  
4. Nearest Resources Located -> System finds the closest inventory and volunteers.  
5. Resource Allocation -> Items & volunteers assigned and updated in the database.  
6. Stock Alerts -> If inventory is insufficient, an email notification is sent.

## Database Management

1. setup\_database() : Initializes MySQL database and tables.  
2. stock\_renewal(item\_name, quantity, item\_type, inv\_name) : Updates or inserts inventory items.

## GUI Components

1. disaster\_info() : Collects disaster-related details from the user.  
2. need\_resource() : Displays available inventory and resource request options.  
3. volunteer\_adding() : Allows users to register as volunteers.  
4. add\_resource\_to\_stock() : Adds resources to the inventory via the UI.

## Resource & Volunteer Allocation

1. find\_closest\_city(user\_cities, related\_list) : Sorts cities based on proximity.  
2. priority(disaster\_lst, related\_lst) : Sorts requests based on urgency.  
3. fcfsAlloc(location, resource\_lst) : Assigns inventory and volunteers to requests.  
4. closestVolunteer(aLoc, volunteers) : Identifies the nearest volunteer.  
5. find\_closest\_inventory(user\_city) : Finds the nearest inventory storage.

## Technologies Used

- Python (CustomTkinter, geopy, smtplib)  
- MySQL (mysql-connector-python)

## Future Enhancements

- Improved Concurrency: Optimize database queries for handling large-scale disasters.  
- Real-time Monitoring Dashboard: Display stock and volunteer status dynamically.  
- Mobile App Integration: Extend functionality to mobile platforms.