Introduction：

Natural disasters, such as earthquakes, floods, hurricanes, and wildfires, cause significant damage to infrastructure and human life. In order to understand the impacts of these disasters and devise effective recovery strategies, it is crucial to collect and analyze relevant data. In this project, we developed a back-end management system that allows users to access and explore global data on natural disasters and their associated recovery efforts..

Development：

During the development process, we discussed project requirements, clarified functionalities, and design styles. We adopted the Scrum agile development method, held regular team meetings, and assigned tasks to ensure smooth project execution. Members supported and collaborated with each other, and a team leader was appointed to maintain the team's direction.In the initial phase, we modularized requirements, filtered CSV data, and designed the database. We imported data into tables for processing, generating searchable, editable, and downloadable lists, as well as visualized charts. We implemented administrator login/logout, user management, and internal messaging by manipulating cookies. We utilized Bootstrap for front-end display and wrote tests to ensure proper site functionality. We used Plotly to create intuitive charts and maps. Additionally, we included detailed comments throughout the code to enhance readability. A detailed breakdown of the group's activities can be found at: GroupProjectContributionRecord.pdf

Design:

During the project design phase, we decided to modularize the application functionalities, making the code more readable and allowing each team member to focus on their responsibilities without being disturbed by other parts of the code, reducing GitHub conflicts. More importantly, this approach made it more convenient to expand, split, and reduce back-end functionalities. The application modules include disaster lists, disaster details, data entry, internal messaging systems, user management systems, and archived data lists. We used Bootstrap as the front-end framework and Django and Plotly for the back-end. The database design consists of summary, detail, user, and message tables.In the list section, we provided user-friendly pagination, filtering, CSV downloads, data editing, and disaster detail viewing. By performing a join query on the summary and detail tables, we displayed intuitive disaster descriptions for users. Notably, we generated disaster information descriptions on the detail page through 30 fields following specific rules, allowing users to gain a more comprehensive understanding of the information. We also designed data archiving and restoration features to ensure data quality.To visualize data more intuitively, we used SQLite3 and Plotly to create multi-dimensional charts and rankings, such as a global disaster count map grouped by country, annual disaster count line charts, disaster count pie charts grouped by disaster type, and Top 10 rankings of countries with the most significant impact on population and assets. In the code, the navigation bar and header serve as public files for easier maintenance. We designed login and logout features, and the user management system supports adding and removing administrators and checking login status. Finally, we implemented an internal messaging feature for administrators to post announcements visible to all site users.

Deployment:

Should use leiwang's Codio box, url: **https://sharonpackage-expandfood-8000.codio-box.uk**
The site has been deployed to Render: **https://disaster-management.onrender.com/**