## Disaster Management System Documentation

**Project Title: *Disaster Management System***

**Objective :*The objective of this project is to create a comprehensive disaster management system using IBM Cloud Foundry and related technologies.***

**1)Technology Stack**

IBM Cloud Foundry: We have chosen IBM Cloud Foundry as the primary platform for deploying our disaster management application due to its scalability and ease of use.

**2)IBM Cloud Services:**

**IBM Watson:** We will leverage IBM Watson for natural language processing to enhance our incident reporting system.

**IBM Cloudant**: Cloudant will be our database solution, allowing us to store and retrieve incident data efficiently.

**IBM Cloud Functions:** IBM Cloud Functions will be used for serverless computing to handle specific real-time data processing tasks.

**Database:** We will use IBM Db2 as our relational database to store critical data.

Disaster Management Functions

1. **Data Collection**: Our system will collect real-time data from various sources, including weather APIs, sensors, and social media feeds, providing real-time information on disasters and incidents.

**2. Incident Reporting**: Users will be able to report incidents through our user-friendly interface, providing information such as incident location, type, and severity.

**3. Resource Allocation** :We have developed resource allocation algorithms that consider incident severity and real-time data to efficiently deploy emergency responders and resources.

**4. Communication Tools**: Our system includes notification and chat features for effective communication and coordination among stakeholders during disaster events.

**5. Geospatial Data Analysis:** We use geospatial data and mapping services to analyze the impact of disasters, monitor affected areas, and track resource deployment.

**IBM Cloud Foundry Setup**

**Steps Taken:** To set up our application on IBM Cloud Foundry, we followed the platform's documentation, created an account, and deployed our application following best practices. Challenges During setup, we faced challenges related to configuring our environment variables and service bindings correctly, which we resolved with the help of IBM Cloud support.

**Implementation Details**

**Data Collection**: We implemented data collection through REST APIs and webhooks, ensuring a constant inflow of real-time data.

**Resource Allocation**: Our resource allocation logic considers incident priority, resource availability, and distance to maximize the efficiency of resource deployment.

**Communication Tools**: We integrated a chat application and implemented notification services using IBM Cloud Functions.

**Documentation Process**

Documentation Tools: For documentation, we used Markdown and a version control system to manage changes.

Challenges : Documenting our progress and decisions was challenging due to time constraints, but we prioritized essential aspects of the project.

**Assessment**

**Assessment Process** : We plan to share this documentation with project assessors for evaluation and feedback

**Security**

**Security Measures** :Security measures include data encryption, user access controls, and intrusion detection systems to protect sensitive data and system integrity.

**Data Protection:** We use data encryption both in transit and at rest, ensuring the confidentiality and integrity of data.

**User Training**

User Training Materials User training materials and user guides have been created to help end-users and administrators understand and use the system effectively.

**Maintenance Plan:** We have established a routine maintenance plan that includes regular updates, patches, and system monitoring.

**Future Features**: Future features planned include advanced machine learning for predicting disaster impact and enhanced reporting capabilities.

