

## Task 6: Database Design and Implementation

### DATABASE DESIGN:

- **Identify Data Entities:** Determine the entities relevant to the disaster management system, such as users, locations, incidents, resources, and notifications. Define attributes for each entity to capture relevant information.
- **Define Relationships:** Establish relationships between entities to represent how they are connected and how data flows within the system. For example, a user may be associated with multiple incidents, and an incident may require various resources.
- **ERD:** Create a visual representation of the database model to better understand the relationships between entities.
- **Schema Design:** Design the database schema by converting the ERD into a relational schema, specifying data types, constraints, and indexes ensuring data integrity and consistency.

### DATABASE IMPLEMENTATION:

- **Select Database Management System (DBMS):** Choose a DBMS that aligns with the requirements of the disaster management system, considering factors such as data volume, transaction frequency, and scalability. Common choices include relational databases (e.g., MySQL, PostgreSQL) and NoSQL databases (e.g., MongoDB, Firebase).
  - Choose a suitable DBMS based on the app's needs and complexity.
  - For mobile apps, SQLite is a popular choice due to its lightweight nature and offline capabilities.
- **Create Database Tables:** Implement the database schema by creating tables based on the defined entities and relationships. Specify primary keys, foreign keys, and constraints to enforce data integrity and consistency.
- **Normalization:** Apply normalization techniques to eliminate redundancy and ensure data integrity. This involves organizing data into tables and minimizing data duplication to reduce the risk of anomalies.
- **Implement CRUD Operations:** Develop functions or methods to perform Create, Read, Update, and Delete (CRUD) operations on the database. This allows the mobile app to interact with the database to store, retrieve, update, and delete data as needed.

Implement functionalities for data storage (inserting data), retrieval (fetching data), and manipulation (updating and deleting data).

- **Data Security:** Implement security measures such as encryption, access control, and data validation to protect sensitive information stored in the database. Follow best practices to mitigate security risks and ensure compliance with privacy regulations.