Overview:

The Flask Emergency Notification System is a comprehensive web application aimed at enhancing communication between individuals facing medical emergencies and healthcare providers. Designed with user-centric functionality, the system allows users to manage their personal medical information, report emergencies with real-time location tracking, and ensures hospitals can respond effectively to these emergencies.

Key Features:

- User Registration and Login:
  - Sign-Up Process:
    - Users can create an account by filling out a registration form that requires their full name, email address, password, weight, height, age, medical history, and allergies.
    - The system includes validations to ensure all required fields are filled and that the password and confirm password fields match.
  - Password Security:
    - User passwords are hashed using SHA-256 encryption before being stored in the database, ensuring enhanced security against unauthorized access.
  - Login Functionality:
    - Registered users can log in using their credentials. The application retrieves user details from the database to establish a session for authenticated users.
- User Dashboard:
  - After logging in, users are redirected to their personalized dashboard.
  - The dashboard serves as a central hub for managing their personal information, including updating medical history and allergies.
  - Users can initiate an emergency notification directly from this dashboard.

- Emergency Notification System:
  - Emergency Button:
    - Users can trigger an emergency alert by providing their current location via latitude and longitude coordinates.
  - Location Geocoding:
    - The application leverages the HERE API to convert the geographical coordinates into a human-readable address, which is then included in the emergency notification.
  - Emergency Record Keeping:
    - All emergency notifications are stored in a dedicated database table, allowing hospitals to access and review them as needed.
- Hospital Registration and Login:
  - Hospital Sign-Up Process:
    - Hospitals can create an account by providing essential details such as the hospital name, address, email, and password.
    - Similar password validation and hashing procedures are followed to maintain security.
  - Hospital Login Functionality:
    - Hospital staff can log in to access their dashboard, which displays all reported emergencies.
- Hospital Dashboard:
  - The dashboard for hospitals provides an overview of all emergencies reported by users, allowing for timely responses.
  - Hospitals can view detailed information about each emergency, including the user’s medical history and allergies.
- Contact Form:
  - Users can reach out to the application administrators through a contact form, submitting their name, email address, and a message.

- Emergency Notification System:
    - Emergency Button:
        - Users can trigger an emergency alert by providing their current location via latitude and longitude coordinates.
    - Location Geocoding:
        - The application leverages the HERE API to convert the geographical coordinates into a human-readable address, which is then included in the emergency notification.
    - Emergency Record Keeping:
        - All emergency notifications are stored in a dedicated database table, allowing hospitals to access and review them as needed.
- Hospital Registration and Login:
    - Hospital Sign-Up Process:
        - Hospitals can create an account by providing essential details such as the hospital name, address, email, and password.
        - Similar password validation and hashing procedures are followed to maintain security.
    - Hospital Login Functionality:
        - Hospital staff can log in to access their dashboard, which displays all reported emergencies.
- Hospital Dashboard:
    - The dashboard for hospitals provides an overview of all emergencies reported by users, allowing for timely responses.
    - Hospitals can view detailed information about each emergency, including the user's medical history and allergies.
- Contact Form:
    - Users can reach out to the application administrators through a contact form, submitting their name, email address, and a message.

- Emergency Notification System:
  - Emergency Button:
    - Users can trigger an emergency alert by providing their current location via latitude and longitude coordinates.
  - Location Geocoding:
    - The application leverages the HERE API to convert the geographical coordinates into a human-readable address, which is then included in the emergency notification.
  - Emergency Record Keeping:
    - All emergency notifications are stored in a dedicated database table, allowing hospitals to access and review them as needed.
- Hospital Registration and Login:
  - Hospital Sign-Up Process:
    - Hospitals can create an account by providing essential details such as the hospital name, address, email, and password.
    - Similar password validation and hashing procedures are followed to maintain security.
  - Hospital Login Functionality:
    - Hospital staff can log in to access their dashboard, which displays all reported emergencies.
- Hospital Dashboard:
  - The dashboard for hospitals provides an overview of all emergencies reported by users, allowing for timely responses.
  - Hospitals can view detailed information about each emergency, including the user’s medical history and allergies.
- Contact Form:
  - Users can reach out to the application administrators through a contact form, submitting their name, email address, and a message.

    - All inquiries are recorded in a separate contact database, facilitating effective communication with users.
- Database Structure:

The application utilizes two main databases to manage its data efficiently:

    - Users Database (users.db):
        - Users Table:
            - Fields: id, full_name, password, email, weight, height, age, medical_history, alergies
            - Stores essential information about users and their medical conditions.
    - Emergencies Table:
        - Fields: id, user_id, username, weight, height, location, email, age, medical_history, alergies
        - Logs all emergency notifications from users along with relevant personal information.
- Hospitals Database (hospitals.db):
    - Hospitals Table:
        - Fields: id, hospital_name, address, email, password
        - Contains hospital details for authentication and emergency response.
- Contact Database (contact.db):
    - Contact Table:
        - Fields: id, name, email, message
        - Stores user inquiries and feedback for administrative purposes.
- Technologies Used:
    - Flask Framework:
        - A lightweight WSGI web application framework for Python that simplifies building web applications.
    - SQLite Database:
        - A serverless, self-contained SQL database engine that provides a reliable storage solution for user and emergency data.

    - HTML/CSS:
        - Used to create the front-end interface of the application, ensuring a user-friendly experience.
    - HERE API:
        - A mapping and location data API that provides geocoding services to translate geographical coordinates into addresses.
- Security Considerations:
    - Data Protection:
        - User passwords are securely hashed, and sensitive data is protected through secure database practices.
    - Session Management:
        - User sessions are managed securely to prevent unauthorized access to user dashboards and emergency information.
- Conclusion:
    - The Flask Emergency Notification System represents a crucial tool in bridging the gap between users experiencing medical emergencies and healthcare providers. By facilitating quick and efficient communication, the system aims to enhance emergency response times and improve overall user safety. Through its robust features and user-centric design, it provides an essential service that prioritizes health and safety in urgent situations.

Authors: Wilayat Ali Kawoosa, Mohammad Zain, Siyan Showkat

==

END