**INFOSYS SPRINGBOARD INTERNSHIP 5.0**

**EMERGENCY AND MENTAL WELLBEING BATCH-7, TEAM-1**

**STORY POINTS**

**TEAM MEMBERS:**

1.Harinee S

2. Sailaja

3. Yash Raj

4. Swetha

**CONTENT:**

1.Description

2. Story points

3. Functional requirements

4. Non-functional requirements

5. DB Design

**DESCRIPTION**

**In the Emergency and Mental Wellbeing application, several key features—Account Setup, Journal, SOS Button, and Deployment—are designed to support user needs. Here’s how each feature is implemented and benefits the user:**

**1. User Authentication**

* **Microservice Involvement:**
  + **Authentication Microservice:** Handles user login, password storage and validation, session management, and basic error handling.
  + **Main Application Microservice:** Coordinates the data flow, sending the verification and authentication status to the frontend.
* **User Experience:**
  + Users can log in with their email and password on a secure, user-friendly login page. Error messages for invalid credentials or connection issues are handled gracefully.

**2. Secure Journal Setup**

* **Microservice Involvement:**
  + **Journal Microservice:** Stores and manages journal entries with encryption for privacy, allowing text, image, and document entries.
  + **Main Application Microservice:** Aggregates and delivers journal data securely to the frontend, ensuring privacy and encryption.
* **User Experience:**
  + Users can create, view, and edit journal entries securely. They’ll have options to add text, images, and documents to their journal entries, knowing their data is securely encrypted.

**3. SOS Button Implementation**

* **Microservice Involvement:**
  + **Emergency Microservice:** Manages SOS alerts, geolocation sharing, and emergency contact details.
  + **Main Application Microservice:** Coordinates between the emergency and location microservices, sending data to the frontend.
* **User Experience:**
  + Users have quick access to an SOS button on the interface, which, when activated, shares their geolocation and sends an alert to emergency contacts in real time.

**4. Positive Affirmations**

* **Microservice Involvement:**
  + **Affirmations Microservice:** Manages a collection of positive affirmations and provides random selections for display.
  + **Main Application Microservice:** Coordinates with the affirmations microservice and sends the data to the frontend.
* **User Experience:**
  + Users are presented with a random positive affirmation on each visit, or they can refresh for a new one, enhancing the sense of positivity and motivation.

**Overall Architecture**

**The Emergency and Mental Wellbeing system is designed with user privacy and security as top priorities. By focusing on essential features, such as private journaling, a responsive emergency alert system, and reliable access, users are supported in their personal wellbeing journey. The app is organized to scale as more features are added, while maintaining a simple and intuitive experience**

**USER STORIES**

1. **User Registration and Login**
   * **Title:** User Authentication
   * **As a:** User
   * **I want to:** Log in securely with my email and password
   * **So that I can:** Access my account and personalized features safely
   * **Acceptance Criteria:**
     + Users can log in with email and password on a secure login page.
     + Error messages appear for invalid credentials or connection issues.
2. **Journal Creation and Management**
   * **Title:** Secure Journal Setup
   * **As a:** User
   * **I want to:** Create and manage private journal entries
   * **So that I can:** Document my thoughts securely with privacy assured
   * **Acceptance Criteria:**
     + Users can add, view, and edit journal entries with text, images, and documents.
     + All journal data is encrypted to ensure privacy.
3. **Emergency Assistance**
   * **Title:** SOS Button Implementation
   * **As a:** User
   * **I want to:** Send an emergency alert quickly
   * **So that I can:** Notify my contacts and share my location in real-time during emergencies
   * **Acceptance Criteria:**
     + Users have access to an SOS button that sends alerts and shares geolocation with emergency contacts.
4. **Positive Motivation**
   * **Title:** Positive Affirmations
   * **As a:** User
   * **I want to:** Receive uplifting affirmations
   * **So that I can:** Feel motivated and encouraged regularly
   * **Acceptance Criteria:**
     + A random positive affirmation is displayed on each visit, with an option to refresh for a new one.

**FUNCTIONAL REQUIREMENTS**

**User Functional Requirements**

* **Registration and Login**: Users must be able to create an account, log in, and manage their profiles securely.
* **Journal Management**: Users should be able to create, view, edit, and securely store private journal entries.
* **SOS Button**: Users should have access to an SOS button to quickly send alerts and share their geolocation with emergency contacts.
* **Positive Affirmations**: Users should receive random positive affirmations to boost motivation on each visit.

**NON-FUNCTIONAL REQUIREMENTS**

**1. Performance**

* The application should handle a growing number of users and requests with minimal latency.
* Ensure that login, journal management, and SOS alerts are processed quickly and efficiently.

**2. Scalability**

* The system should be designed to scale horizontally, accommodating increasing users and data securely and efficiently.

**3. Security**

* Implement role-based authentication to restrict access to different parts of the system.
* Ensure data encryption and privacy measures for user information, especially for sensitive journal entries and emergency contacts.

**4. Reliability**

* The system should be reliable with minimal downtime and include robust error handling mechanisms.

**5. Usability**

* The application should provide a user-friendly interface that enables easy navigation for journaling, SOS alerts, and viewing positive affirmations.
* Ensure intuitive access to core features to enhance the user experience.

**6. Maintainability**

* The system should be modular and maintainable, with a clear separation of concerns across microservices for authentication, journal, emergency, and affirmations functions.

**DB DESIGN**

Table users {

  \_id ObjectId [pk]

  name String

  email String

  passwordHash String

}

Table emergencyContacts {

  \_id ObjectId [pk]

  userId ObjectId [ref: > users.\_id]

  contactName String

  contactPhone String

  contactEmail String

}

Table sosAlerts {

  \_id ObjectId [pk]

  userId ObjectId [ref: > users.\_id]

  alertTime Date

}

Table sosAlertContacts {

  \_id ObjectId [pk]

  sosAlertId ObjectId [ref: > sosAlerts.\_id]

  contactId ObjectId [ref: > emergencyContacts.\_id]

  contactName String

  contactPhone String

  contactEmail String

}

Table journalEntries {

  \_id ObjectId [pk]

  userId ObjectId [ref: > users.\_id]

  entryContent String

}

Table affirmations {

  \_id ObjectId [pk]

  userId ObjectId [ref: > users.\_id]

  affirmationText String

}

