# INFOSYS SPRINGBOARD 5.0 INTERNSHIP

Group-4

Team Members

1. G. Sai Satvik
2. M. Srishanth
3. Shraddha Potdar
4. Amurta Bhosale

# Functional and Non-Functional Requirements

## Functional Requirements

### 1. User Functional Requirements

• Registration and Login: Users will be able to create an account, log in securely, and manage their profile information.

• Emergency Contact Management: Users will be able to add, update, or delete emergency contacts who can be notified in times of crisis.

• Mental Well-being Resource Access: Users will be able to browse and access mental health resources, including articles, videos, and self-help tools.

• Emergency Alert System: Users will be able to activate an emergency alert to notify selected emergency contacts and nearby support services.

• Well-being Check-ins: Users will have access to regular check-ins to track their mental health, including mood, stress, and sleep patterns.

• Professional Support Access: Users will be able to search for and connect with certified mental health professionals for appointments or immediate assistance.

### 2. Support Service Functional Requirements

• Support Service Directory Management: Support services will be able to register their details and maintain an updated directory listing, including contact information and service categories.

• Alert Response: Support services will receive emergency alerts with details and will be able to respond to users in real time.

• User Feedback Collection: Support services will have access to collect user feedback after an interaction to improve service quality.

### 3. Admin Functional Requirements

• User and Service Management: Admins will be able to manage registered users and support services, including deactivating or reactivating accounts as needed.

• Resource Management: Admins will be able to upload, categorize, and update mental health resources accessible to users.

• Analytics and Reporting: Admins will have access to reports on app usage, emergency alerts, and user engagement to monitor and enhance platform performance.

• Feedback Analysis: Admins will have tools to analyze user feedback on both services and resources to continuously improve the platform.

## Non-Functional Requirements

### 1. Performance

• The application will handle a high volume of users and emergency notifications with minimal response time.

• Resource browsing, emergency alerts, and support service communications will be processed quickly and efficiently.

### 2. Scalability

• The system will be designed to scale horizontally to accommodate an increasing number of users and mental health resources.

• The platform will support additional mental health resources, support services, and emergency contacts as user demand grows.

### 3. Security

• Role-based authentication will be implemented to secure user, support service, and admin access to various parts of the system.

• Strict data privacy and protection measures will be enforced, especially for sensitive user information and mental health data.

• Secure data transmission will be ensured, particularly for emergency alerts and user-support interactions.

### 4. Reliability

• The application will maintain high availability, with minimal downtime and resilient error handling, especially during emergency alert transmissions.

• The system will handle sudden peaks in demand, such as an influx of emergency alerts or check-ins.

### 5. Usability

• The platform will offer an intuitive and user-friendly interface for users, support services, and admins, with easy navigation and accessibility.

• Users will be able to seamlessly access mental health resources, initiate emergency alerts, and connect with support services.

### 6. Maintainability

• The system will be modular and maintainable, enabling efficient updates, particularly for resources, emergency alert protocols, and support service details.

• Clear documentation for each module will be provided, making it easy for developers to understand and manage the codebase.

## Proposed Tech Stack

### 1. Frontend

• React JS: To build a dynamic and responsive user interface, providing an engaging and accessible experience for users, support services, and admins.

### 2. Backend

• Spring Boot: For creating scalable and efficient backend services. Spring Boot will be used to develop RESTful API endpoints for user management, emergency alerts, and access to mental health resources.

• Spring Security: To implement robust authentication and authorization mechanisms, ensuring secure access to sensitive user and system data.

### 3. Database

• MongoDB/MySQL: A NoSQL/SQL database will be used to store user profiles, emergency contacts, mental health resources, and alert history, offering schema flexibility and efficient data retrieval.

### 4. Microservices

• User Microservice: Built using Spring Boot, this microservice will handle user registration, authentication, profile management, and mental well-being check-ins.

• Emergency Alert Microservice: Developed with Spring Boot, this microservice will manage real-time emergency alert transmissions to designated contacts and support services.

• Resource Management Microservice: A Spring Boot service that will manage the mental health resource library, enabling resource updates and collecting user feedback.

• Support Service Microservice: Created with Spring Boot, this microservice will manage support service listings, registrations, and user feedback for ongoing service improvement.