Business Requirements Document (BRD)

1. Project Title

Personal Safety and Well-being Cloud-Native Application

2. Project Overview

The goal of this project is to develop a cloud-native application that enhances personal safety and well-being by offering user-centric features like registration, login, emergency alerting, real-time chat, and location tracking. The app will provide an admin dashboard for monitoring and management.

3. Business Objectives

- Provide a secure platform for users to register and access well-being services.
- Allow users to raise alerts in emergencies.
- Enable administrators to monitor users and handle role-based functionalities.
- Offer real-time chat for emergency assistance or user support.
- Track user location for safety verification during emergency alerts.
- Ensure high availability, scalability, and security using cloud-native architecture.

4. Scope

In-Scope:

- User registration and login (with JWT/OAuth2)
- Role-based access (Admin/User)
- Admin dashboard to view users
- React-based responsive frontend
- RESTful APIs using Spring Boot
- MySQL integration
- Dockerized deployment
- CI/CD pipeline
- AWS cloud deployment
- Emergency alert button
- Real-time chat feature
- Location tracking using browser/location API

Out-of-Scope:

- Native mobile app
- Payment gateway integration
- AI/ML-based recommendations

5. Stakeholders

- Developer: Rogan Raj C - Full-stack development
- Trainer: UST Capstone Mentor - Guidance and evaluation
- End Users: General Public / Admins - Use application features

6. Functional Requirements

- User Registration & Login
- JWT-based authentication
- Role-based page access
- Admin dashboard with user list
- Emergency alert button
- Real-time chat module
- Location tracking feature
- REST APIs with validations

7. Non-Functional Requirements

- Responsive UI (React)
- Secure data transmission (HTTPS)
- Scalable microservice architecture
- Cloud deployment (AWS EC2/RDS/S3)
- CI/CD via GitHub Actions or Jenkins
- Real-time features using WebSocket or Firebase

8. Assumptions

- Users have internet access
- AWS account is available for deployment
- Database is hosted on MySQL (local or cloud)

9. Constraints

- Must be completed within 1 week
- Must follow Spring Boot and React-based tech stack
- Cloud-native principles must be applied

10. Acceptance Criteria

- Functional login/register
- Admin can view/manage users
- Role-based access works correctly
- Dockerized backend & frontend
- Successfully deployed on AWS

- Codebase follows best practices
- Emergency alert works and stores log
- Chat is real-time and usable
- Location tracking is enabled during alerts