ResQSync : Transforming Crisis Response

During crises such as pandemics, natural disasters, or large-scale accidents, managing critical resources efficiently poses significant challenges. Hospitals, ambulances, and government agencies often face difficulties .

D by Deepak Jeena

Execution Plan for ResQSync

1. Backend Development

Develop robust Spring Boot APIs to manage critical resources securely, with JWT-based authentication ensuring system integrity.

2. Frontend Development

Create a comprehensive hospital resource dashboard using Next.js and Tailwind CSS, featuring WebSockets for live updates, volunteer management, and real-time mapping powered by OpenStreetMap.

3. Real-Time Notifications

Implement Firebase Cloud Messaging (FCM) to deliver instantaneous emergency alerts across the platform.

4. Crisis Visualization

Utilize Leaflet.js to generate dynamic crisis heatmaps, providing immediate visual insights into emergency situations.

5. UI/UX Finalization

Design an intuitive, responsive interface with minimal styling to enable rapid, efficient navigation during high-stress scenarios.

Practicality & Social Impact

Practicality

Our approach prioritizes rapid development, leveraging open-source tools to create an essential that can be prototyped quickly.

Social Impact

- **Real-Time Crisis Response:** Accelerate emergency decision-making processes
- **Resource Optimization:** Enhance access to critical resources during emergencies
- **Community Collaboration:** Facilitate seamless communication among first responders
- **Scalability:** Design capable of managing large-scale emergency scenarios

ResQSync

Innovative Crisis Management Platform

Team Name: Code Core

Hackathon Name: InFroniX

Date: 16th February 2025

Team Members:

- Deepak Singh Jeena - (Team Leader) , Backend Specialist
- Rohan Moura - Full Stack Developer

The Problem: Inefficient Crisis Management

Lack of Real-Time Data

Inadequate visibility into hospital bed availability and resource tracking.

Delayed Ambulance Dispatch

Suboptimal hospital allocation and route management.

Inefficient Coordination

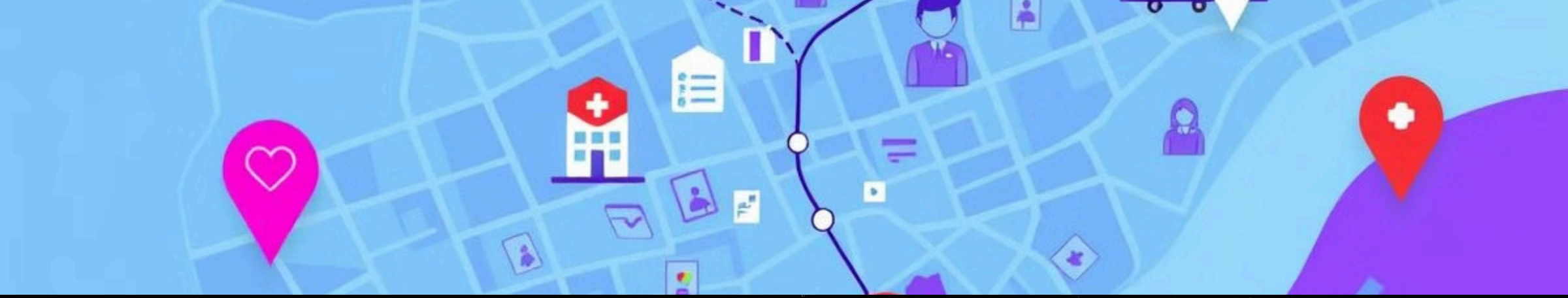
Poor communication between volunteers, responders, and authorities.

Manual Response Systems

Time-consuming decision-making due to manual communication.

Introducing ResQSync: A Smart Solution

ResQSync is a smart crisis management system built to handle large-scale emergencies efficiently. It combines real-time monitoring, optimized resource allocation, predictive analytics, and automated alerts to improve decision-making and crisis response outcomes.



Why Choose ResQSync?

1 **Efficiency in Crisis Management**

Addresses critical time delays and coordination issues during emergencies.

2 **Real-Time Updates**

Offers live monitoring of resources, ensuring up-to-date information for responders and decision-makers.

3 **Centralized Communication**

Unites hospitals, ambulances, volunteers, and authorities under one platform for seamless communication.

4 **Data-Driven Insights**

Historical data and predictive analytics help in future-proofing resource allocation during similar crises.

Key Features & Functionalities

Smart Hospital & Resource Management

Real-time monitoring of hospital resources, manual updates, and live inventory dashboards.

Nearest Hospital

Can see nearest hospital of your area via map inside it .

Emergency Response & Communication System

Automated emergency alerts, volunteer coordination, and communication channels.

Historical Data & Predictive Insights

Data dashboards, interactive crisis heatmaps, and data-driven decision support.

Multi-Language Support

Provides multi-language support (e.g., English, Hindi) using i18n localization.

Tech Stack for Implementation

Component	Tech Stack
Frontend	Next.js (UI/UX), TailwindCSS, Leaflet.js, Framer Motion
Backend	Spring Boot (API & Business Logic), WebSockets
Database	MySQL (Main Storage), Redis (Caching for Fast Lookups)
Real-Time Data	Firebase/WebSockets for live updates
Communication	SMS/WhatsApp Alerts

Frontend Flow (Next.js)

Implements JWT-based authentication using NextAuth.js or a custom solution.Stores tokens for authorized API requests after successful login.

Displays real-time hospital data using WebSockets.Includes forms for manual resource updates by hospital staff.

Visualizes ambulance routes and locations using *OpenStreetMap + OSRM.* Shows ETA and hospital recommendations.

Allows volunteers to sign up and view task assignments via a table.

Displays real-time crisis zones with color-coded markers using *Leaflet.js*. Enables crisis filtering by type.

Backend Flow (Spring Boot)

Authentication & Authorization

Implement JWT authentication and configure Spring Security for endpoint access control.

Hospital Resource Management APIs

Allow updates to hospital resource availability.

Caching Data

Use of reddis for caching the data and send it to the frontend.

Emergency Alert Notifications

SMS/WhatsApp alerts to volunteers and responders.

Historical Data Management

Save historical crisis data and provide endpoints for data visualization.