Buildathon — Team Roles & Responsibilities

Congratulations Team! 🎉

Congratulations team for successfully completing the first step! Now let's focus on giving our absolute best. Here below are the expectations from each one of you to ensure we deliver an outstanding solution that stands out in the competition.

Mahesh — Frontend Lead (Face of the Operation)

Mission

Design and build the interface judges interact with, turning system intelligence into a clear, trustworthy experience.

Core Responsibilities

- **Live Incident Dashboard**: Show real-time list of incidents with columns for Incident ID, Source, Status, and Severity
- **Incident Detail View**: Click-through view rendering a neat, readable incident_summary for any case
- **Chat Interface**: A clean, WhatsApp-style chat window for the Safety Assistant demo
- **Source Visibility**: Clearly display input source (WhatsApp, SMS, Call, Video, App) for each incident

Collaboration

- Consumes all read/write endpoints exposed by the backend
- Co-owns real-time updates layer with the backend (e.g., push/live updates) for incident and dispatch status
- Integrates the chat UI with the Safety Assistant service in coordination with Deva

Definition of Done

- Dashboard updates without manual refresh
- Incident detail view readable in under 3 clicks
- Chat interface live for the demo flow

Naseer — Backend Lead (Central Hub)

Mission

Own the data layer and the API surface that connects AI Core, ingestion, chatbot, and frontend.

Core Responsibilities

- **Database Architecture**: Design and manage the schema; all other modules consume via APIs only
- **API Platform**: CRUD for incidents and teams; endpoints required by frontend and AI Core; strict ownership of data contracts
- **WebSocket Implementation**: Together with Mahesh, own the WebSocket layer to provide real-time updates for the dashboard
- **Real-Time Updates**: Co-own the live update channel with the frontend to push incident status and dispatch changes

R&D Stretch (Time-Permitting)

- Explore live location tracking exposure for rescue teams and provide an easy interface for the frontend to render movement

Collaboration

- Single source of truth for data; no direct DB access by other modules
- Works with Deva to ensure chatbot and inbound channels can store and fetch needed data cleanly
- Aligns with AI Core on contracts for nearest team lookup and incident updates

Definition of Done

- Stable API contracts shared early
- Live updates working for incident status changes
- Logs/health checks available for quick debugging

Deva — Communication & RAG Lead (The Bridge)

Mission

Connect outside world to our system and return life-saving guidance back to victims over the same channel.

Core Responsibilities

- **Channel Automation**: Set up webhooks for WhatsApp and SMS to capture text/media/sender and forward to AI Core's intake
- **Channel Fidelity**: Ensure responses go back via the originating channel (WhatsApp/SMS) once an incident is confirmed
- **Safety Assistant (RAG)**: Build a retrieval pipeline over NDMA SOPs and generate short, authoritative, hazard-aware guidance; run proactive guidance until responders arrive

Flow Alignment

- After AI Core verifies and allocates, begin the Safety Assistant session and maintain the conversation thread
- Coordinate with frontend to render chat sessions cleanly and with backend for any needed persistence

Stretch Collaboration (Time-Permitting)

- Team up with backend to surface live location tracking to the frontend

Definition of Done

- Incoming WhatsApp/SMS routed to intake reliably
- Safety Assistant can send first guidance message automatically on verified incidents
- RAG answers are grounded in SOP content, short, and safe

Ramesh — AI Core Architect (The Brain)

Mission

Implement the core intelligence in three cooperating agents to turn messy inputs into verified, structured, and actionable incidents.

Core Responsibilities

- **Agent 1 (Scribe)**: Normalize inputs from text/images/audio/video into structured evidence using top-tier ASR and vision models
- **Agent 2 (Analyst)**: Derive hazard type, damages, and initial severity with clear reasoning and a consistent JSON structure
- **Agent 3 (Orchestrator)**: Verify with real-world context, finalize severity, select nearest responders via backend API, produce incident_summary, and trigger notifications and chatbot

Interfaces

- Provide a single intake endpoint for ingestion to call
- Consume backend contracts to fetch nearest teams and persist updates
- Hand off verified incident details to Deva for the Safety Assistant flow

Definition of Done

- End-to-end: intake → analyze → verify → allocate → notify → chatbot kick-off
- incident_summary is complete, audit-ready, and consistent across cases
- Clear logs and deterministic demo path for judges

Working Agreements

Single Owners, Clear Handoffs

- **Data ownership and APIs**: Backend
- **UI experience and rendering**: Frontend
- **Channel automation and Safety Assistant**: Communication & RAG
- **AI reasoning and orchestration**: AI Core

Shared Ownership

- **Real-time updates**: Co-owned by Frontend + Backend
- **Channel fidelity and guidance continuity**: Owned by Communication & RAG

Database Access Protocol

- **Important**: No one will communicate with database operations directly
- All database operations will be done through Naseer's APIs only
- Everyone will be calling Naseer's API wrappers for any data needs

---

Risk and Escalation

If any scope feels risky or unclear, raise it immediately in the team channel for a quick alignment/backup plan.