🚨 Aapda Setu — Technical Documentation

AI-Powered Emergency & Disaster Response Platform

> Judge-ready documentation covering APIs, alert systems, geofencing, and implementation status.

📋 Table of Contents

Section 1 — APIs Used

🗺️ Mapping & Location APIs

MapLibre + OpenStreetMap Tiles

- **Why not Google Maps?** — Google Maps charges per API call; OSM is free and open-source
- **Implementation:**
  - MapLibre GL renders interactive disaster maps
  - Markers show verified incidents with severity colors
  - Geofencing circles drawn for alert radius visualization

Geolocation APIs

- **Device GPS** — Native Flutter location services
- **Reverse Geocoding** — Convert coordinates to human-readable addresses
- **Endpoint:** `GET /api/v1/location/reverse?lat=28.6&lng=77.2`

🌧️ Disaster & Weather APIs

IMD Weather Alerts API

- **Source:** India Meteorological Department
- **Data:** Heavy rain, cyclone, heatwave warnings
- **Polling:** Every 10 minutes via scheduled job

NDMA Disaster Feed APIs

- **Source:** National Disaster Management Authority
- **Data:** Official disaster declarations, evacuation orders
- **Format:** JSON/RSS feed parsing

Earthquake Alert Feeds (USGS)

- **Source:** United States Geological Survey
- **Data:** Real-time seismic activity
- **Threshold:** Magnitude ≥ 4.0 triggers alert

Flow:

```
IMD/NDMA API → Scheduler fetches every 10 min → AI categorizes severity →
Broadcast to affected region
```

🔥 Fire Incident Alert System APIs

Endpoint	Method	Description
`/api/v1/reports`	POST	Submit fire report with GPS + photo
`/api/v1/reports/:id/verify`	POST	Admin verification of report
`/api/v1/alerts`	POST	Create geofenced emergency alert
`/api/v1/alerts/nearby`	GET	Get alerts within radius

Fire Department Notification (Future)

- Webhook integration with local fire stations
- Auto-dispatch based on proximity

🔔 Notification APIs

Firebase Cloud Messaging (FCM)

- **Use:** Real-time push notifications to mobile app
- **Targeting:** Topic-based (by city/district) or device-specific
- **Endpoint:** Internal `services/fcm.service.ts`

Twilio SMS (Offline Backup)

- **Use:** Reaches users with no internet
- **Trigger:** High-severity alerts only
- **Endpoint:** Internal via `.env` credentials

🤖 AI/ML Service APIs

Service	Port	Purpose
Text Classification	8000	DistilBERT disaster type detection
Image Forensics	8001	EfficientNet scene verification
Fusion Model	8000	LightGBM combining text + image scores
Gemini Chatbot	5000	Emergency guidance via Google AI

Security

- All internal ML calls use **HMAC signatures**
- Callback URLs verified before processing

Section 2 — Fire Alert System (5 km Radius)

🔥 Step-by-Step Fire Alert Pipeline

```
1. User submits fire report with GPS + photo
         ↓
2. Report stored in MongoDB with GeoJSON location
         ↓
3. AI validates report (text + image analysis)
         ↓
4. If confidence ≥ 0.75 → trigger geofence alert
         ↓
5. Notify:
   • All users within 5 km radius
   • Nearest fire response unit (future)
```

📐 How 5 km Radius is Calculated

Haversine Distance Formula

```
a = sin²(Δlat/2) + cos(lat1) × cos(lat2) × sin²(Δlng/2)
c = 2 × atan2(√a, √(1−a))
distance = R × c   (where R = 6371 km)
```

MongoDB 2dsphere Index

- **Why?** — Enables efficient geospatial queries without brute-force calculation
- **Performance:** O(log n) vs O(n) for full table scan

Query Example

```
// Find all users within 5 km of fire incident
db.users.find({
  location: {
    $near: {
      $geometry: {
        type: "Point",
        coordinates: [77.2090, 28.6139]  // [lng, lat]
      },
      $maxDistance: 5000  // meters
    }
  }
})
```

Scalability

- Indexed queries handle **millions of users** efficiently
- No distance calculation in application code
- MongoDB handles spherical geometry natively

🎯 Notification Targeting Logic

1. **Radius Check** — Only users inside 5 km receive alert
2. **Priority Delivery** — Critical alerts bypass queue
3. **Rate Limiting** — Max 1 alert per incident per user (no spam)
4. **Fallback** — SMS for users with disabled push notifications

---

Section 3 — Heavy Rain / Weather Alerts

⛈️ Weather Alert Flow

```
1. Scheduler fetches IMD alert feed every 10 minutes
         ↓
2. AI categorizes severity (Low / Medium / High / Critical)
         ↓
3. Match affected coordinates to user locations
         ↓
4. Push notification to all users in impacted region
```

Severity Classification

Level	Trigger	Action
Low	Light rain forecast	No alert

Level	Trigger	Action
Medium	Heavy rain warning	In-app notification
High	Flood risk	Push + SMS
Critical	Evacuation order	Emergency broadcast

Example Alert

> 🌧 **Heavy Rainfall Warning** Mumbai — Avoid low-lying areas. Expected: 150mm in next 6 hours. Emergency: 108 | NDMA: 1078

Section 4 — Accident Emergency Alert System

🚗 Accident Alert Pipeline

```
1. Accident report submitted (marked HIGH URGENCY)
         ↓
2. Immediate priority bypass (no ML delay)
         ↓
3. System identifies within 2 km:
   • Nearest hospitals
   • Nearby responders
   • Police stations
         ↓
4. Notify responders with GPS coordinates + victim count
```

Responder Dashboard Features

- Real-time incident feed
- Accept/Reject assignment
- Navigation to scene
- Status updates (En route → On scene → Resolved)

Emergency Escalation Rules

- If no response in **3 minutes** → expand search to 5 km
- If no response in **10 minutes** → escalate to district emergency control

Section 5 — Implementation Status

☑ Implemented Now (Hackathon MVP)

Feature	Status	Location

Feature	Status	Location
Report submission API	✅ Done	`routes/reports.routes.ts`
MongoDB GeoJSON storage	✅ Done	`models/Report.model.ts`
MapLibre live map UI	✅ Done	`admin-dashboard/`
Redis queue pipeline	✅ Done	`queues/ml.queue.ts`
ML text classification	✅ Stub	`ml-service-cpu/`
FCM push notifications	✅ Demo	`services/fcm.service.ts`
5 km geofence query	✅ Done	MongoDB 2dsphere index
Gemini AI Chatbot	✅ Done	`routes/chatbot.routes.ts`
JWT + Phone Auth	✅ Done	`routes/auth.routes.ts`

🔮 Planned Next (Startup Scale)

Feature	Priority	Timeline
Full IMD/NDMA live feeds	High	Week 2
Fire department API automation	High	Week 3
Image forensics GPU deployment	Medium	Week 4
Multi-city SaaS onboarding	Medium	Month 2
IoT + drone sensor integration	Low	Month 3
Blockchain report authenticity	Low	Month 4

Section 6 — Unique Innovation Points

🏆 Why Aapda Setu Stands Out

1. AI-Based Misinformation Control

- NLP detects generic/bot-like text
- Image forensics flags manipulated photos
- Duplicate detection using perceptual hashing

2. Verified Reporting + Trust Scoring

- Multi-layer validation: User → Moderator → NGO → Authority
- User reputation score affects report priority

3. 5 km Instant Fire Alerts

- MongoDB geospatial queries (O(log n) performance)

- Sub-second notification delivery

4. Offline Support

- SQLite local storage for report queue
- SMS fallback for critical alerts
- Sync when connectivity restored

5. Multi-Tenant SaaS Architecture

- Isolated data per organization
- Customizable alert rules per city
- NGO and government dashboard access

6. Hybrid AI Pipeline

- **CPU Service:** Text + Fusion (low cost)
- **GPU Service:** Image analysis (high accuracy)
- **LLM:** Emergency chatbot guidance

Section 7 — Sample Alert Messages

🔥 Fire Alert Push

```
🔥 FIRE ALERT — Sector 12, Delhi

Active fire reported 1.2 km from your location.
Evacuate immediately. Avoid Main Road.

Fire Dept: 101 | Emergency: 112
📍 Tap for safe route
```

⛈️ Heavy Rain Warning

```
⛈️ HEAVY RAIN WARNING — Mumbai

IMD forecasts 150mm rainfall in next 6 hours.
• Avoid low-lying areas
• Stock emergency supplies
• Monitor for flood alerts

NDMA Helpline: 1078
```

🚗 Accident Responder Message

```
🚨 URGENT: Traffic Accident

Location: NH-48, Km 23 (2.1 km from you)
Victims: 3 reported
Time: 2 minutes ago

ACCEPT to navigate to scene →
```

💬 Chatbot Safety Guidance

```
User: What should I do during an earthquake?

AI: 🌍 **Earthquake Safety:**

1. Drop, Cover, and Hold On
2. Stay away from windows and heavy objects
3. If outdoors, move to an open area
4. After shaking stops, check for injuries
5. Be prepared for aftershocks

Emergency: 112 | Disaster: 108
```

🏗️ System Architecture

```
┌─────────────────────────────────────────────────────────────────┐
│                       CLIENT APPLICATIONS                         │
├───────────────────────┬─────────────────────────────────────────┤
│  Flutter Mobile App   │      Next.js Admin Dashboard             │
└───────────────────────┴─────────────────────────────────────────┘
           │                          │
           ▼                          ▼
┌──────────────────────────────────────────────────────────────────┐
│              NODE.JS API GATEWAY (Port 5000)                      │
│  ┌─────────┬───────────┬──────────┬───────────┬──────────────────┐│
│  │ Auth    │ Reports   │ Alerts   │ Chatbot   │ Analytics/Export ││
│  └─────────┴───────────┴──────────┴───────────┴──────────────────┘│
└─────────────────────────────────────────────────────────────────┘
        │            │              │              │
        ▼            ▼              ▼              ▼
  MongoDB      Redis Queue      AWS S3        Firebase FCM
  (GeoJSON)    (BullMQ)         (Media)       (Notifications)
        │
        ▼
  ML Services (FastAPI)
  • Text Classification (CPU)
```

```
      • Image Forensics (GPU)
      • Gemini Chatbot (API)
```

📞 Judge Q&A Quick Answers

Q: How do you ensure only nearby users get alerts?

> We use MongoDB 2dsphere indexing with geospatial `$near` queries and 5000m maxDistance. This scales efficiently to millions of users without brute-force distance calculations.

Q: How do you prevent fake reports?

> AI validates text + images, we have user reputation scoring, multi-layer human verification, and image forensics to detect manipulation.

Q: What happens if there's no internet?

> Reports queue locally in SQLite, SMS backup for critical alerts, and auto-sync when connectivity returns.

Generated for Aapda Setu Hackathon — January 2026