PES UNIVERSITY

CONFLUENCE

24 Hour Hackathon

Team Name – Pack A Byte

Member 1 – Piyush Shiv, SRN 1 – PES1UG24AM191

Member 2 – Harsh Patnaik, SRN 2 – PES1UG24EC085

Member 3 – Harsh Pandya, SRN 3 – PES1UG24CS182

Member 4 – Adyanth Mallur, SRN 4 – PES1UG24CS036

sponsored by ACCURE

IEEE COMPUTER SOCIETY

Problem Statement

Track Selected - AI-Powered Disaster Impact Prediction System

Disaster response teams often face challenges in accurately identifying and prioritizing high-risk areas due to delays in data collection, inefficient resource allocation, and a lack of risk assessment. This leads to ineffective disaster relief efforts, increased casualties, and prolonged recovery times. There is a need for a system that can analyze data and provide immediate, data-driven insights to optimize crisis management and humanitarian aid distribution.

Solution

PES UNIVERSITY

01

Data Collection

Fetch data from APIs, satellite feeds, and user reports.

02

Risk Analysis & Processing

AI filters data and assigns risk scores.

03

Interactive Map Dashboard

Displays disaster zones with risk levels.

04

Automated Alerts

Sends notifications to users and authorities.

05

Response & Resource Allocation

Suggests evacuation routes and aid distribution.

06

Continuous Improvement

Refines AI with feedback and new data.

IEEE COMPUTER SOCIETY

Solution Appendix

PES UNIVERSITY

🔹 Why This Matters:

Traditional disaster response is slow and inefficient. Our AI-powered system predicts impact zones, helping authorities act before situations worsen.

🔹 Key Differentiators:

Uses multi-source data (weather, geospatial, population, social media).

Provides automated, data-driven risk assessments instead of relying on manual evaluations.

Offers mobile-first accessibility, ensuring usability in crisis zones.

🔹 Challenges & Solutions:

Data Reliability: Combining multiple APIs reduces dependence on any single source.

Scalability: Cloud-based architecture ensures performance even during disasters.

User Adoption: A simple UI with push notifications ensures non-tech users can navigate easily.

🔹 Final Vision:

A fully automated, AI-driven disaster response system that minimizes casualties, speeds up relief efforts, and empowers decision-makers with real-time intelligence.

IEEE COMPUTER SOCIETY

Requirements Software stack

- **Frontend (User Interface)**

Next.js (React Framework) – SEO-friendly, fast, and server-rendered.
Tailwind CSS – Lightweight and responsive UI styling.
Mapbox / Leaflet.js – Interactive disaster risk visualization.

- **Backend (API & Server Logic)**

Node.js + Express.js – API handling and business logic.
Firebase Functions – Serverless backend execution.

- **Database & Storage**

Firebase Firestore – NoSQL database for disaster data and reports.
Firebase Storage – Stores media files like satellite images.

- **APIs & Data Sources**

OpenWeather, Google Maps, NASA Copernicus – weather, satellite, and geospatial data.
Twitter API / CrisisNET – Crowdsourced disaster updates.

- **AI & Machine Learning**

TensorFlow / PyTorch – AI models for disaster impact analysis.
Scikit-learn – Statistical analysis and risk classification.

- **Hosting & Deployment**

Firebase Hosting – Deploys frontend and backend with global CDN.
Firebase Authentication – Secure user login and role management.

Deployment

Deployment Targets

- Target 1: Government agencies – Helps in disaster preparedness and resource planning.
- Target 2: NGOs & Relief Organizations – Assists in identifying high-risk areas for aid distribution.
- Target 3: Urban Planners & Researchers – Supports infrastructure planning in disaster-prone zones.

Targeted Users & Use Case

- Targeted Users: Government bodies, disaster relief teams, NGOs, and researchers.
- Use Case: Our app analyzes historical disaster data and key risk factors (sea level, population density, distance from the coast) to predict high-risk zones. This helps organizations prioritize response efforts and optimize resource allocation, ensuring better disaster management and recovery planning.

Deployment

Deployment, Scalability & Distribution

- Easy Deployment – Fully serverless with Firebase Hosting & Firebase Functions, automated CI/CD.
- Scalable – Firestore’s NoSQL database scales dynamically with demand.
- Effortless Distribution – Next.js ensures fast web performance with global CDN support.

Why Our Solution?

- Web-first, AI-driven, and lightweight for seamless disaster risk assessment.
- Cost-efficient & low latency with a serverless backend on Firebase.
- User-friendly UI/UX for quick, actionable insights.

Fast, scalable, and optimized for disaster management on the web.

Bibliography

- IEEE Research Paper – AI-based Disaster Risk Prediction & Management.
  🔗 https://ieeexplore.ieee.org/document/10593506
- Firebase Documentation – Serverless backend, hosting, and database services.
  🔗 https://firebase.google.com/docs
- Next.js Documentation – Optimized framework for web applications.
  🔗 https://nextjs.org/docs
- Mapbox API – Interactive map integration for disaster visualization.
  🔗 https://docs.mapbox.com/
- TensorFlow.js Documentation – AI model deployment for web-based ML.
  🔗 https://www.tensorflow.org/js
- NASA Earth Data – Preloaded datasets for disaster risk analysis.
  🔗 https://earthdata.nasa.gov/