

# Team No-249

**Problem Statement Title:**

**Crowd Safety Intelligence System (Advanced)**

**Team Members:**

1. **Gudimetla Susritha**
2. **Seerla Akhila**
3. **Gaddala Sravani**



# Problem Statement

- Large-scale events face crowd control challenges (e.g., concerts, festivals, rallies).
- Manual monitoring cannot predict critical crowd surges in time.
- Overcrowding can lead to accidents or stampedes.
- Need: An AI-powered real-time monitoring and alert system to predict and prevent risks before escalation.

# Proposed Solution

- Crowd Safety Intelligence System combines AI, IoT, and computer vision for real-time situational awareness.
- Collects crowd density data via:
  - Cameras and GPS sensors
  - Simulated IoT devices (via simulator.py)
- Machine Learning model predicts future crowd density.
- Issues automatic alerts and shows interactive heatmaps on a dashboard for quick decisions.



# { Tech Stack }

Our entire system was built in Python, allowing us to go from idea to a full-stack, AI-powered application at maximum speed.

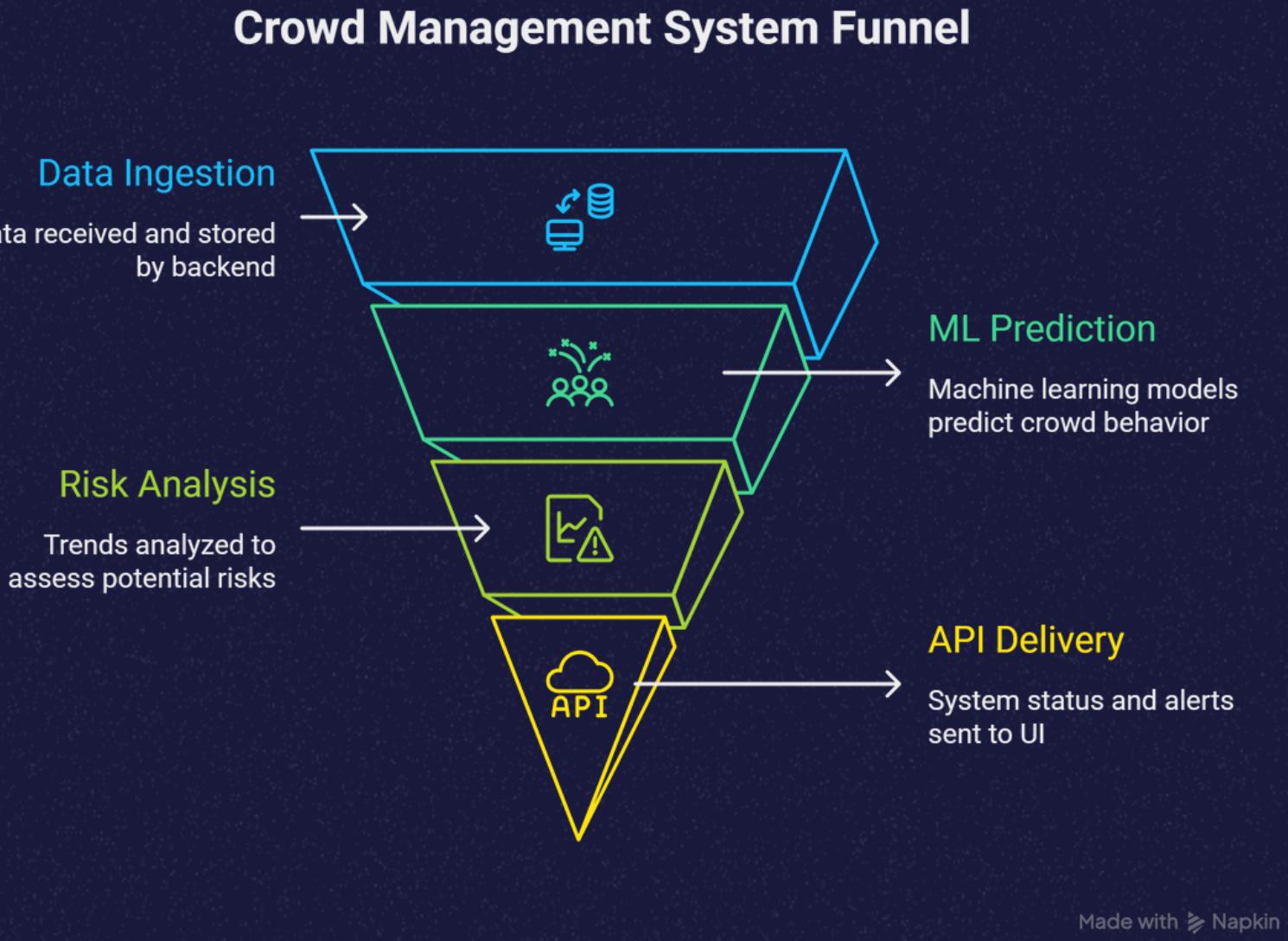
- Backend (The "Brain")
  - FastAPI: A high-performance Python framework to create a "blazing fast" API that ingests live data and serves predictions.
  - Scikit-learn: Used to build and run our predictive machine learning model (.pkl) for forecasting risk.
- Frontend (The "Command Center")
  - Streamlit: The core of our frontend. We chose it to build a beautiful, interactive data dashboard in pure Python, saving us days of complex web development.
  - Pydeck: Powers our primary 3D data map, allowing us to extrude GeoJSON zones and color them based on live risk.
  - Folium: Used for the user-friendly 2D "Reference Map" tab with clear OpenStreetMap labels.



# System Architecture

## Components:

-  Simulator (Person A):
  - Sends crowd count data from virtual cameras.
-  FastAPI Backend (Person B):
  - Processes incoming data, updates crowd trends, predicts risks using ML model (crowd\_predictor.pkl).
-  ML Model:
  - Learns patterns from previous density logs (crowd\_data\_log.csv).
-  Streamlit Dashboard (Person C):
  - Displays live map, density heatmap, alerts, and trend graphs.



# Implementation

Component	Technology	Function
Data Ingestion	FastAPI	Receives live sensor/camera data
ML Prediction	scikit-learn(joblib)	Predicts crowd density trend
Visualizatin	Streamlit + PyDeck	Live dashboard & heatmap
Simulator	Python	Real-time crowd simulation
Storage	CSV / Firebase-ready	Logs data for training

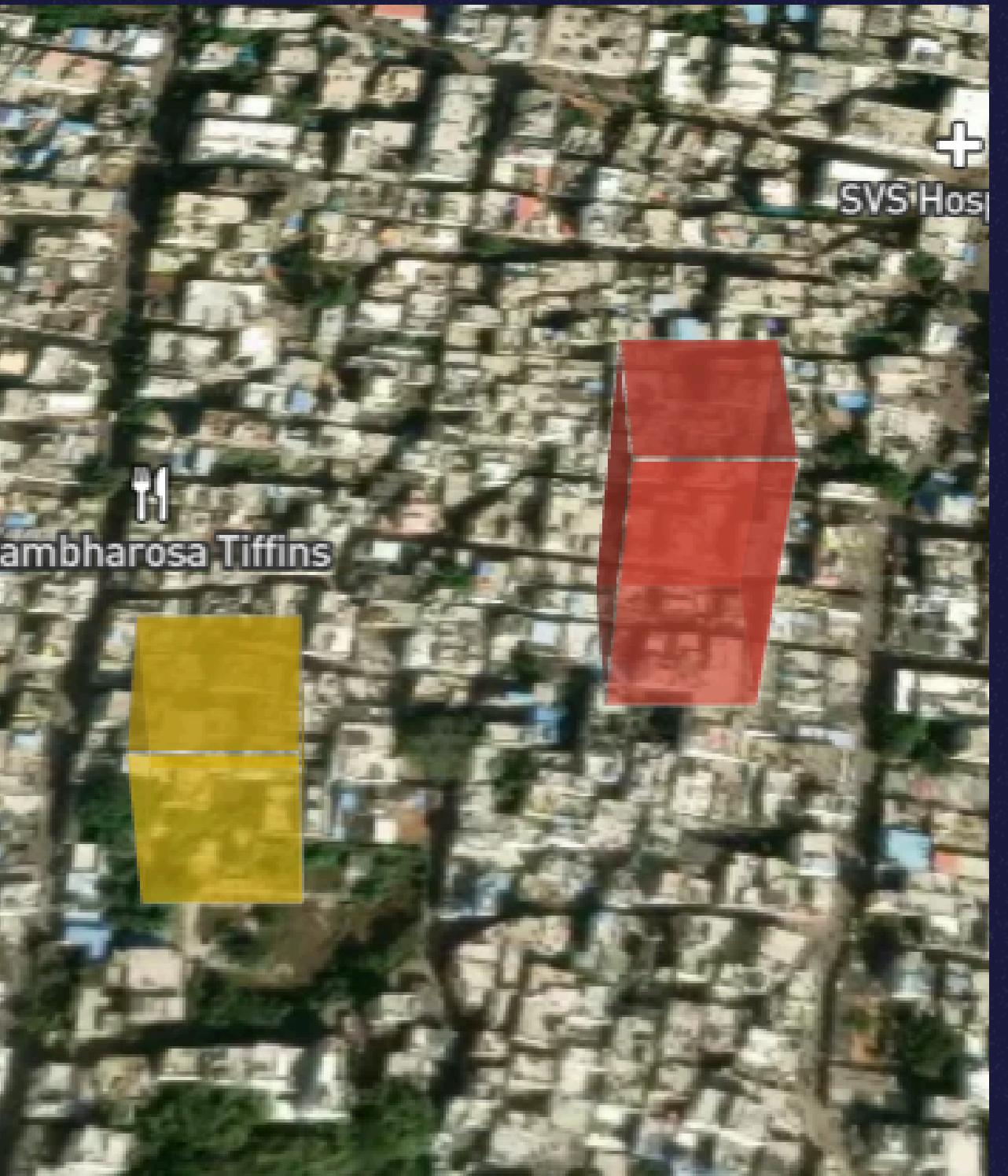
## Key Features:

- Automated risk prediction and trend-based forecasting
- Dynamic alert generation and visualization
- Custom risk threshold control via dashboard
- Modular backend for easy scalability



# Results & Demo

- Real-Time Performance:
  - Dashboard updates every 2 seconds with live simulator data.
- Dynamic Alerts:
  - Automatically generated when crowd density crosses thresholds.
- Predictive Insights:
  - uses ML + trend analysis to forecast high-risk zones.
- Live Demo:
  - The system will be demonstrated in real-time showing:
- Simulator → Backend → Dashboard data flow
- Heatmap visualization and alert generation



# Impact & Future Scope

Impact:

- Enhances public safety at large events.
- Enables proactive crowd management.
- Reduces human response time for emergencies.

Future Enhancements:

- Integration with CCTV video analytics.
- Drone-based aerial crowd tracking.
- Mobile app for on-ground security teams.
- Cloud-deployed version for smart cities.

