**Two Phases:**

* 🌲 **Travelers/tourists** visiting forest-adjacent areas.
* 👨‍🌾 **Villagers/locals** living near the forest.

**🛠 Project Plan (Hackathon Prototype)**

**1. Core Idea**

A **mobile/web app** that provides:

* **Conflict risk score** (real-time & location-based).
* **Safe route suggestions & alerts** (maps with hotspots).
* **Tour guide features** (nearest forest check-post, wildlife sighting history, do’s & don’ts).
* **Local villager mode** (predicts crop raid risks, nearest forest official contact, fire alerts).

**2. Data Sources**

For hackathon, you can use **open or proxy datasets** (don’t need highly accurate field data):

* **Forest boundaries & protected areas** → [Forest Survey of India shapefiles], [OpenStreetMap].
* **Wildlife distribution & corridors** → [WWF India reports], [Wildlife Institute of India datasets].
* **Human–wildlife incidents** → News datasets (scrapable), [GBIF occurrence data], or sample Kaggle datasets.
* **Weather & vegetation** → IMD Open Data, NASA Earthdata (for rainfall, NDVI).
* **Tourist locations near forests** → Google Maps API / OpenStreetMap.

**3. ML Model**

Yes ✅ — it’s a **classification model** but you can frame it as **Risk Prediction**:

* **Input Features**:
  + GPS coordinates (proximity to forest/core zone)
  + Time of day / season (crop raid/elephant movement patterns)
  + Weather (rainy season → higher elephant/tiger movement)
  + Land use (farmland vs settlement vs tourist spot)
  + Past incidents in that area
* **Output (Target)**:
  + Risk Category → **Low / Medium / High** (multi-class classification)

**Baseline Model (Hackathon Ready)**:

* **Logistic Regression / Random Forest** (easy to implement, explainable).
* Later → **LSTM (temporal) + Geospatial ML** for better prediction.

**4. System Workflow**

1. **User opens app** → Shares location or enters travel destination.
2. **Backend fetches risk score** → Using ML model + incident data.
3. **Map shows overlays** →
   * 🟥 High-risk zones
   * 🟨 Medium risk
   * 🟩 Safe zones
4. **Tourist mode** → Suggests safe trails, do’s & don’ts.
5. **Villager mode** → Predicts possible crop raids, sends SMS alerts in regional language.

**5. Hackathon Deliverables**

* **Frontend**: React Native / Flutter (basic mobile UI).
* **Backend**: Flask / FastAPI + SQLite (to keep it lightweight).
* **Map Integration**: Leaflet.js or Google Maps API with heatmaps.
* **ML Model**: Scikit-learn Random Forest (train with mock + scraped data).
* **Demo Flow**:
  + Tourist chooses “Bandhavgarh Safari” → Sees a **Medium Risk** warning + safety tips.
  + Villager at forest edge → Receives SMS alert of **High Risk: Elephant crossing nearby**.