**1. System Design Overview**

The system will consist of the following components:

1. **User Interface (UI):**
   * Input fields for starting location, destination (GPS coordinates), and preferences (e.g., avoid specific areas, prioritize well-lit streets).
   * A map visualization to display the recommended route.
2. **Backend:**
   * Route calculation using map APIs. (google maps API, OSRM (problem🡪 needs frequent updating)
   * Integration of historical crime data to adjust route recommendations.
3. **Data Processing:**
   * Analyze crime data for patterns, locations, and risk scores.
   * Aggregate crime incidents by time of day, type, and severity.
4. **Recommendation Engine:**
   * Combine user preferences and crime data to compute and rank safe routes.

**2. Input Requirements**

**User Input:**

* **Starting Location**: GPS coordinates (latitude, longitude).
* **Destination**: GPS coordinates (latitude, longitude).
* **Preferences**: Options like:
  + Avoid areas with high crime rates.
  + Prefer routes with public surveillance cameras.
  + Prefer routes with street lamps.
  + Time of travel (to adjust based on daytime/nighttime safety).

**Historical Data Input:**

* **Crime Data**:
  + **Location**: Latitude and longitude of incidents.
  + **Type of Crime**: E.g., robbery, assault, vandalism.
  + **Date and Time**: For time-specific risk adjustments.
  + **Severity**: Assign weight based on crime type.

**3. Technical Workflow**

1. **Data Collection:**
   * Use public crime datasets (e.g., police records, open crime APIs) or proprietary datasets.
   * Regularly update the database to maintain relevance.
2. **Data Preprocessing:**
   * Clean and normalize crime data.
   * Categorize crimes by type and severity.
   * Group crime incidents spatially using geohashing or clustering algorithms.
3. **Risk Mapping:**
   * Create a risk map by assigning risk scores to map tiles based on crime density and severity.
   * Use heatmaps to visualize high-risk areas.
4. **Route Generation:**
   * Use routing APIs like Google Maps, OpenStreetMap (OSRM), or Mapbox to generate multiple possible routes.
   * Overlay risk data onto these routes to calculate risk-adjusted scores.
5. **Preference Filtering:**
   * Apply user-defined filters (e.g., avoid certain areas, prioritize certain features).
6. **Recommendation Engine:**
   * Rank routes based on risk-adjusted scores and user preferences.
   * Suggest the safest route along with alternatives.

**4. Implementation**

**Tools and Technologies:**

1. **Frontend**: React, Mapbox/Leaflet.js for maps.
2. **Backend**: Node.js, Python (Flask/Django) for APIs.
3. **Database**: PostgreSQL with PostGIS extension for geospatial data.
4. **Crime Data Processing**: Pandas, Geopandas for analysis; scikit-learn for clustering.
5. **Routing**: Google Maps API, OSRM, or GraphHopper.

**Integration Flow:**

1. Accept user input and fetch crime data for the relevant region.
2. Calculate multiple potential routes using the map API.
3. Score each route based on historical crime data and user preferences.
4. Display the safest route and alternatives on the UI.

**6. Enhancements**

* **Real-Time Data**: Incorporate live reports of incidents.
* **Community Feedback**: Allow users to report unsafe areas.

**Crime history datasets:**

**1. Open Government Data (OGD) Platform India: This platform offers datasets related to crime statistics across various regions in India. While specific APIs for crime data may not be directly available, datasets can be accessed and integrated into your system.**

[**Data.gov.in**](https://www.data.gov.in/dataset-group-name/Crime%20Statistics?utm_source=chatgpt.com)

**2. CrimeCheck API: CrimeCheck provides access to a vast database of court records across India, including pending and disposed cases. Their API allows for detailed searches and can be integrated to fetch relevant crime data.**

[**CrimeCheck**](https://apidoc.crimecheck.in/?utm_source=chatgpt.com)

**3. CrimeoMeter: CrimeoMeter offers a worldwide crime data API that integrates various sources of crime data. While its primary focus is global, it may provide valuable insights for specific locations within India.**

[**Crimeometer**](https://www.crimeometer.com/?utm_source=chatgpt.com)

**4. Kaggle Datasets: Kaggle hosts datasets such as "Crime in India," which contains comprehensive information about various crimes in India from 2001 onwards. These datasets can be downloaded and used to analyze crime patterns and trends.**

[**Kaggle**](https://www.kaggle.com/datasets/rajanand/crime-in-india?utm_source=chatgpt.com)

**5.National Crime Records Bureau (NCRB): The NCRB publishes annual reports like "Crime in India," which serve as principal references for crime statistics. While direct APIs may not be available, their reports can be a valuable source of data.**

[**Wikipedia**](https://en.wikipedia.org/wiki/National_Crime_Records_Bureau?utm_source=chatgpt.com)

**CRIME HISTORY DATA-BASE**

**Categories:**

(a) Crimes against people (robbery, theft, assault, kidnapping, abduction, sexual assault/harassment)

(b ) Property related crimes (burglary, vehicle related theft)

(c ) Crimes in public places

(d ) Traffic and road safety violation (drunk driving, hit and run, traffic rule violations)

(e ) Area specific crimes (drug related, gang-related violence, police response time)

(f )