Team Name: Database team 1
Team Member: Minglu Sun, Hanhan Guo, Yuchen Huang, Jiaqi Guo, Chichi Zhang

SafePath is a data-driven navigation platform
that provides integrated crime statistics, lighting conditions, and community-reported incidents; real-time safety alerts; and safer walking route planning
for urban pedestrians and city visitors
who are unhappy with generic map apps that lack crime and lighting data, don't reflect community safety concerns, and fail to automatically plan safer walking routes

What features/capabilities do we plan to deliver this semester?

1. Interactive Crime Map: Display clustered pins on a map; users can click to view detailed incident cards (type, time, location, city).
2. Search & Filter Panel: Filter crimes by type (multi-select), city, and time range (24h/7d/30d/custom).
3. ETL Data Pipeline: Clean, normalize, and deduplicate CSV crime data into a unified schema and load into a spatially indexed database.
4. Safety Recommendations*(advanced)*: Use ML models trained on historical data to suggest safer routes and areas to avoid.

What could we add in the future?

In the future, we plan to evolve SafePath into a comprehensive urban safety platform by:

- ✓ Enhance ML-driven routing with real-time crime risk prediction, assigning dynamic risk scores to individual road segments based on time, lighting, and recent local incidents, enabling the system to adjust routes in real time.
- ✓ Integrating with wearables and mobile devices to deliver live safety alerts and enable quick location sharing with emergency contacts.
- ✓ Scaling to multiple cities by ingesting open crime and safety datasets, allowing broader geographic coverage and impact.

Data sources and how to extract the data.

1.Historical crime data (where danger lies): Seattle Police Department Crime Data
https://data.seattle.gov/Public-Safety/SPD-Crime-Data-2008-Present/tazs-3rd5/about_data
Extraction: Use Python (requests / pandas) to call the API daily and fetch new records. Store data in a PostgreSQL + PostGIS database.
Processing: Clean timestamps and coordinates; build spatial indexes to support fast querying and map rendering.

2.Real-time emergency data (where incidents are currently occurring)
https://data.seattle.gov/Public-Safety/Seattle-Real-Time-Fire-911-Calls/kzjm-xkqj/about_data
Extraction: Poll the JSON API every 5 minutes or subscribe to the real-time feed. Parse responses and insert new records into a "live incidents" table (e.g., Redis or PostgreSQL).

3.Street network data (to calculate routes)
https://data-seattlecitygis.opendata.arcgis.com/datasets/SeattleCityGIS::seattle-streets/about
Extraction: Download shapefile or GeoJSON, and import into PostGIS using tools like ogr2ogr or shp2pgsql.
Processing: Build routing topology (nodes, edges) to support shortest-path algorithms.