Road Report AI

Table of Contents

1. Product Overview

The Road Report AI will be an web application that incorporates an AI model that will be trained on the historic conditions of roads. The application will then be able to take real roads and return a general probability of a crash occurring on a road at any given time. This will be fed with real world data to reflect the real world state of the road.

The AI will be trained on data provided by C.R.I.S. and local to Texas. The user of the application will have access to a map for a general overview of the roads. The user can then query specific roads which will be sent to the AI along with the time, weather which will be provided by the weather.gov api, and interpreted road conditions. The AI will then highlight the road as above, below, or at general risk for crash

1.1 Target Users

- First responders: Those that need to be able to prioritize higher risk areas to respond to
- City planners: Those that are able to improve the quality of roads to reduce risk
- General citizens (later phase): Commuters that might want to be aware of risky roads

1.2 Core Features

- Map display for roads for easy querying
- Automatic updating of local weather and road conditions
- Visual representation of road safety

2. Tech Stack

Layer	Technology	Notes / Constraints
Frontend	Vue.js + Vite + NPM + Typescript	
Backend	Python	REST API
Database	PostgreSQL	
AI/ML	Pytorch + Pandas + NumPy	
API	Google Maps + US Weather.gov	
Version Control	Github	
Auth	Oauth	

3. Project File Structure

4. Data

4.1 Model Training Data

The .csv file pulled from C.R.I.S. that contains the crash information is structured:

```
model Data{
        Id                  String;
        City                String;
        County              String;
        CrashDate           Date;
        CrashMonth          int;
        CrashTime           String;
        DayOfWeek           String;
        RoadClass           Stringl
        StreetName          Stringl
        SurfaceCondition    String;
        WeatherCondition    String;
        Crash               Binary;
}
```

5. Phase 1

Weeks 1 and 2

Phase 1 is complete upon the collection and processing of a .csv file into usable data, a figma is designed, and the boilerplates are completed for the front and backend.

Training Data

Requirement	Acceptance Criteria
A way to visualize the .csv data	A Jupyter Notebook file that has reusable functions to compare frequency of columns based on each other (ex. number of crashes per month) or discover how many entries per column have no data
Cleaning up the .csv file	A Jupyter Notebook file that can delete or modify entries inside the .csv file based on specific conditions. For example, any entry that has multiple columns with 'No Data' should be deleted or entries with a researchable column should be flagged for human editing
Add negative sampling to .csv file	A Jupyter Notebook file is created that will create reasonable non-crash data such that the time slice of every XXXXX is filled. The file needs to either randomize non-critical fields like city and road class while using local entries to extrapolate current environmental conditions like weather.

Figma

Requirement	Acceptance Criteria
Assist in Figma creation	If research capacities allow for it, gather a list of reference applications that display maps for information querying and explain why they are relevant. If image generation capabilities exist, generate color schemes and reference images that humans designs can use to create a modern looking UI for the application.

Boiler Plates

Requirement	Acceptance Criteria
Frontend Boilerplate	Based on the Frontend technologies listed in the tech stack (2. Tech Stack), produce the necessary files to integrate and pull from all needed resources. Ensure that no additional technologies or resources are added without commenting (//) to explain their addition to a human reviewer
Backend Boilerplate	Based on the Backend technologies listed in the tech stack (2. Tech Stack), produce the necessary files to integrate and pull from all needed resources. Ensure that no additional technologies or resources are added without commenting (//) to explain their addition to a human reviewer

6. Phase 2

Weeks 3 and 4