AI-Driven Road Safety Project

# Project Structure:

ai\_road\_safety\_project/  
├── data/  
│ └── accidents.csv  
├── notebooks/  
│ ├── data\_exploration.ipynb  
│ └── model\_training.ipynb  
├── src/  
│ ├── data\_preprocessing.py  
│ ├── model.py  
│ ├── visualization.py  
│ └── utils.py  
├── dashboard/  
│ ├── app.py  
│ └── templates/  
├── reports/  
│ └── final\_report.pdf  
├── requirements.txt  
└── README.md

# 1. Data Preprocessing (`src/data\_preprocessing.py`)

import pandas as pd  
  
def load\_data(filepath):  
 return pd.read\_csv(filepath).dropna() # minimal cleaning

# 2. Model Training (`src/model.py`)

from sklearn.ensemble import RandomForestClassifier  
from sklearn.model\_selection import train\_test\_split  
from sklearn.metrics import accuracy\_score  
  
def train\_model(df, target):  
 X = df.drop(columns=[target])  
 y = df[target]  
 X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)  
 model = RandomForestClassifier()  
 model.fit(X\_train, y\_train)  
 predictions = model.predict(X\_test)  
 return model, accuracy\_score(y\_test, predictions)

# 3. Visualization (`src/visualization.py`)

import folium  
  
def plot\_map(data, lat\_col='latitude', lon\_col='longitude'):  
 m = folium.Map(location=[data[lat\_col].mean(), data[lon\_col].mean()], zoom\_start=10)  
 for \_, row in data.iterrows():  
 folium.CircleMarker([row[lat\_col], row[lon\_col]], radius=5, color='red').add\_to(m)  
 return m

# 4. Flask Web App (`dashboard/app.py`)

from flask import Flask, render\_template  
from src.data\_preprocessing import load\_data  
from src.visualization import plot\_map  
  
app = Flask(\_\_name\_\_)  
  
@app.route('/')  
def index():  
 df = load\_data('data/accidents.csv') # Load and preprocess data  
 map\_html = plot\_map(df).get\_root().render() # Generate the map  
 return render\_template('index.html', map\_html=map\_html)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

# 5. Frontend Template (`dashboard/templates/index.html`)

<!DOCTYPE html>  
<html>  
<head>  
 <title>AI Road Safety Dashboard</title>  
</head>  
<body>  
 <h1>Accident Hotspot Map</h1>  
 {{ map\_html|safe }} <!-- Embed the generated map -->  
</body>  
</html>

# 6. Requirements (`requirements.txt`)

flask  
pandas  
folium  
scikit-learn

# 7. README File (`README.md`)

# AI-Driven Road Safety Project  
  
## Overview  
This project uses AI to analyze and predict traffic accident risks, providing insights into accident hotspots and contributing factors.  
  
## How to Run  
1. Clone the repository:  
 ```bash  
 git clone https://github.com/yourusername/ai\_road\_safety\_project.git  
 cd ai\_road\_safety\_project  
 ```  
  
2. Install the dependencies:  
 ```bash  
 pip install -r requirements.txt  
 ```  
  
3. Run the Flask app:  
 ```bash  
 python dashboard/app.py  
 ```  
  
4. Open a web browser and visit `http://localhost:5000` to view the accident hotspot map.

# Expected Output

1. \*\*Model Accuracy Output\*\*:  
When you run the model (`train\_model()`), you will get the accuracy printed in the console.  
  
Example output:  
```python  
Model Accuracy: 0.85 # or a value depending on your dataset  
```  
  
2. \*\*Accident Hotspot Map\*\*:  
- When you run the Flask app (`python dashboard/app.py`), it will start a server at `http://localhost:5000`.  
- Open this URL in a web browser, and you'll see a map with \*\*red circle markers\*\* indicating accident hotspots. The map is interactive and based on the \*\*latitude\*\* and \*\*longitude\*\* data from your CSV file.  
  
\*\*Expected Web Output\*\*:  
- A web page titled \*\*"AI Road Safety Dashboard"\*\* with a title and the map rendered below it. The map shows the locations of accidents based on the data provided.

# How to Run

1. Clone the repository:  
 ```bash  
 git clone https://github.com/yourusername/ai\_road\_safety\_project.git  
 cd ai\_road\_safety\_project  
 ```  
  
2. Install dependencies:  
 ```bash  
 pip install -r requirements.txt  
 ```  
  
3. Run the Flask app:  
 ```bash  
 python dashboard/app.py  
 ```  
  
4. Access the dashboard at `http://localhost:5000`.