DS440 Project Instructions

1) Open up a terminal
2) Go to wherever it is saved on your computer for me it is documents/ds440
3) Cd into the evdashboard folder
   C:\Users\slava\Documents\ds440> **cd evdashboard**
4) Cd into the backend folder
   C:\Users\slava\Documents\ds440\evdashboard>**cd backend**
5) Start the backend code up
   C:\Users\garen\Documents\ds440\evdashboard\backend>**python3 app.py**
6) This will start the backend server
7) Type in cd .. to get back to the evdashboard folder then type npm start to start the dashboard

What the Three Main Files Do

1. app.py (Backend Server This is located in the backend folder)

- **Purpose**: `app.py` is the backend server, handling data requests and responses. It serves as a bridge between the frontend (what you see) and the model (how data is processed).
- **Main Functions**:
  - It loads machine learning models for each country's emissions data (stored in a file called `country_emissions_models.pkl`).
  - It has endpoints like `/predict/<country>` that the frontend calls to get predicted CO_2 emissions for a given country.
  - When the frontend requests data, `app.py` processes the request, retrieves data, and sends it back.

2. train_model.py (Training the Models. This is located in the backend folder)

- **Purpose**: `train_model.py` trains machine learning models for each country's CO_2 emissions data, so that future emissions can be predicted.
- **Main Functions**:
  - It reads historical emissions data for each country, trains a machine learning model for each one, and saves these models into a file (`country_emissions_models.pkl`).
  - This file is later loaded by `app.py` to make predictions.

Note: `train_model.py` only needs to be run once to create the models, unless you want to retrain them with updated data.

DS440 Project Instructions

3. App.js (Frontend Interface. This is located in the folder src)

- **Purpose**: `App.js` is the main file for the frontend, written in React. It provides the dashboard interface that users interact with.
- **Main Functions**:
    - It displays data visualizations, charts, and maps based on the data retrieved from `app.py`.
    - It lets users select options like countries and years, then sends requests to the backend to fetch data.
    - It shows the results (e.g., CO_2 emissions predictions) to the user in an interactive way.