**Why Each File is Necessary:**

**1. Notebook (notebook.ipynb)**

* **Purpose**: Trains the machine learning model using Titanic data.
* **Why Needed**:
  + Prepares the titanic\_model.pkl file, which the Flask app uses for predictions.
  + Ensures the machine learning model is trained and saved for deployment.

**2. Flask Application (app1.py)**

* **Purpose**: Runs the web application to handle user input and predictions.
* **Why Needed**:
  + Loads the trained model (titanic\_model.pkl) to make survival predictions.
  + Provides routes:
    - /: Displays the prediction form.
    - /predict\_form: Handles form submissions and displays the prediction.
    - /predict: API endpoint for JSON-based predictions.
    - /records: Lists all stored predictions from the database.
  + Creates a database (predictions.db) to store user inputs and prediction results.

**3. HTML Form (form.html)**

* **Purpose**: Provides the user interface to input passenger details.
* **Why Needed**:
  + Collects passenger data (Pclass, Sex, Age) for predictions.
  + Dynamically displays the prediction result to the user after submission.
  + Ensures user-friendly interaction with the application.

**4. CSS Stylesheet (style.css)**

* **Purpose**: Styles the HTML form and result page.
* **Why Needed**:
  + Enhances the user experience by making the form visually appealing.
  + Provides layout, fonts, and colors for the web application.

**How They Work Together:**

1. **Notebook**: Trains the model and creates titanic\_model.pkl.
2. **Flask App**: Uses the model and sets up routes to interact with the user via a web interface or API.
3. **HTML Form**: Collects user input and displays predictions.
4. **CSS**: Makes the web interface attractive and user-friendly.

# Titanic Prediction Web Application

## \*\*1. Machine Learning Notebook (`notebook.ipynb`)\*\*

- Trains a machine learning model using Titanic survival data.

- Exports the trained model as `titanic\_model.pkl`.

## \*\*2. Flask Application (`app1.py`)\*\*

- A Python script to:

- Load the trained model.

- Create a web server for predictions.

- Store user inputs and predictions in a database.

- Main Functions:

- Display the prediction form.

- Handle user submissions and provide predictions.

- Save predictions and inputs to the database.

## \*\*3. HTML Form (`form.html`)\*\*

- Provides a simple web interface to:

- Enter passenger details (class, gender, and age).

- Display the prediction result.

## \*\*4. CSS Stylesheet (`style.css`)\*\*

- Adds styling to the form and result page:

- Makes the interface visually appealing.

- Improves usability for users.

## \*\*How They Work Together\*\*

1. \*\*The Notebook\*\* trains the model.

2. \*\*The Flask app\*\* hosts the web application and uses the trained model for predictions.

3. \*\*The HTML form\*\* collects input from users.

4. \*\*The CSS stylesheet\*\* makes the interface attractive.

**Prompt**

**Suggested Prompt for Rendering Assistance:**

**"Help me combine my Titanic prediction Flask app (app1.py), HTML form, and CSS to render predictions dynamically, ensuring everything works correctly. How do I integrate these files and deploy them?"**