**Guide for Route Optimisation**

Setting Up the Flight Route Planner:

1. Install Python:

* If Python is not already installed on your system, download and install it from the official Python website: [Python Downloads] (<https://www.python.org/downloads/>).
* During the installation process, make sure to check the option to add Python to your system PATH.

1. Install Flask:

Flask is a lightweight web framework for Python. You can install it via pip, Python's package manager, by running the following command in your terminal or command prompt:

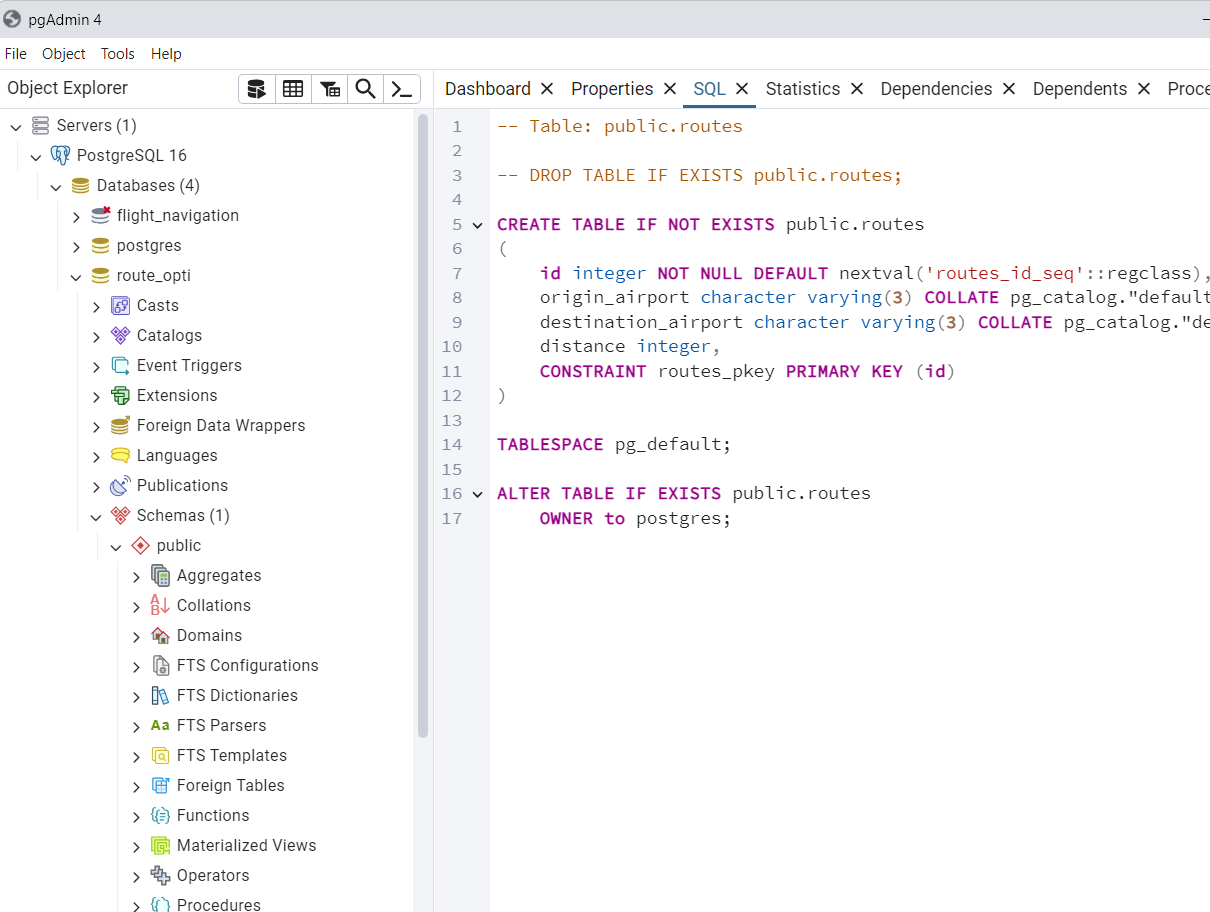
```

pip install flask

```

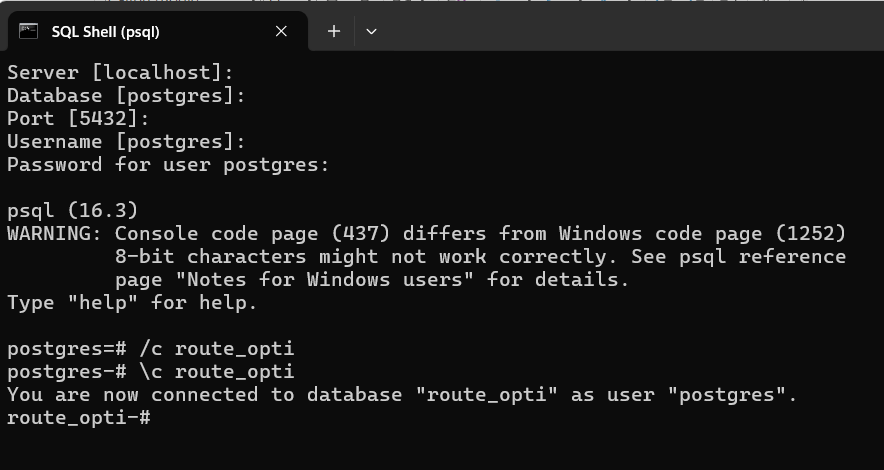
1. Set Up the Database:

* Download and install PostgreSQL from the official website: [PostgreSQL Downloads](<https://www.postgresql.org/download/>).
* During the installation process, make sure to **remember the password** you set for the default PostgreSQL user (`postgres`). [in the code my password is sanj123, it will come error for you so put whatever password you put when you downloaded postgres]
* After installation, open pgAdmin, the PostgreSQL administration tool.



This is pgAdmin

* Create a new database named `route\_opti`.



* Create a new table named `routes` with columns `origin\_airport`, `destination\_airport`, and `distance`.

SQL code:  
CREATE TABLE routes (

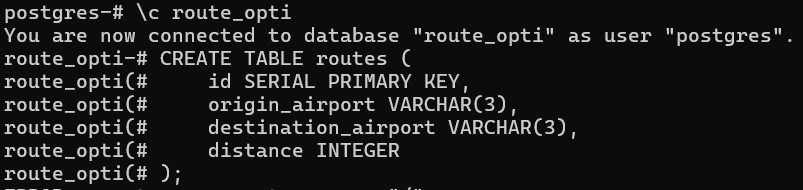
id SERIAL PRIMARY KEY,

origin\_airport VARCHAR(3),

destination\_airport VARCHAR(3),

distance INTEGER

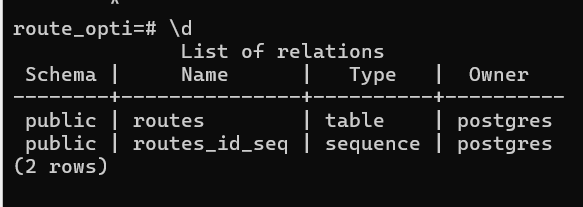
);



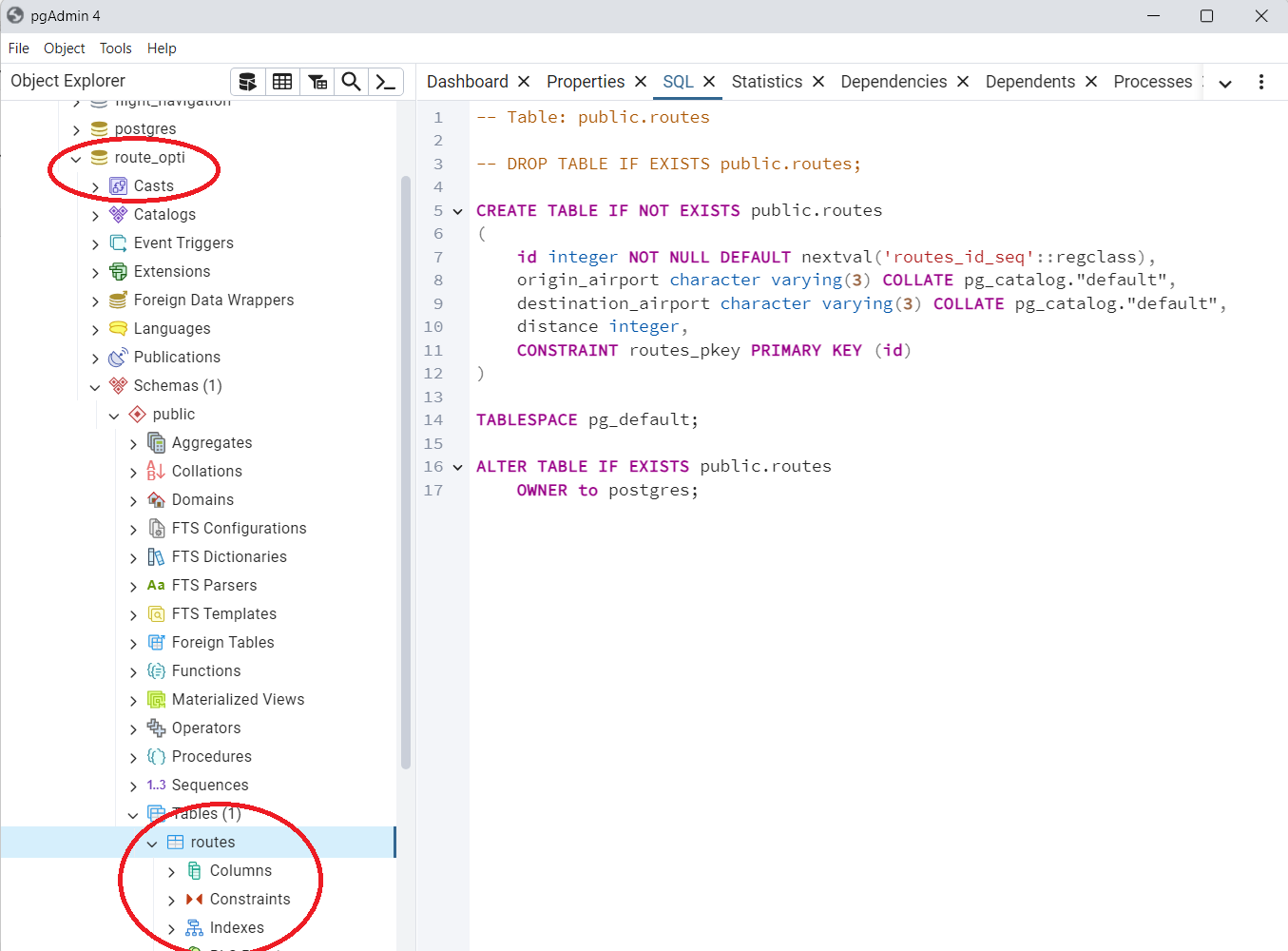
It will come, table created.

To check if the table is there you can enter

\d



* Populate the `routes` table with data representing the distances between airports.



After ruuning the app, it’ll come like this is postgresSQL

4. Install psycopg2:

- Psycopg2 is a PostgreSQL adapter for Python. Install it via pip by running the following command:

```

pip install psycopg2-binary

```

5. Set Up the Flask Application:

- Create a new directory for your Flask application.

- Inside the directory, create a file named `app.py`.

- Copy the code provided in the previous message for `app.py` into this file.

6. Create HTML Template:

- Inside the directory, create a subdirectory named `templates`.

- Inside the `templates` directory, create a file named `index.html`.

- Populate `index.html` with HTML code for the user interface. You can design a simple form for users to input the origin and destination airports.

7. Run the Application:

- Open a terminal or command prompt and navigate to the directory containing `app.py`.

- Run the Flask application by executing the following command:

```

python app.py

```

- Your Flask application should now be running locally. Open a web browser and navigate to `http://127.0.0.1:5000/` to access the flight route planner interface.

8. Test the Application:

- Enter origin and destination airports in the form on the web page.

- Click the "Get Route" button to submit the form.

- The application should display the optimal route between the selected airports along with the total distance.

TO RUN:  
In the terminal, I’d used vs code so view>terminal

Instructions for Running the Project

* Prepare the Data:

Run the data preparation script to clean the data and save it to a new CSV file:

python data\_preparation.py

* Set Up the Database:

Create the routes table and insert data by running the database insertion script:

python insert\_data.py

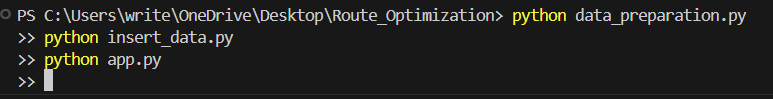
* Run the Flask Application:

Start the Flask app to serve the web interface:

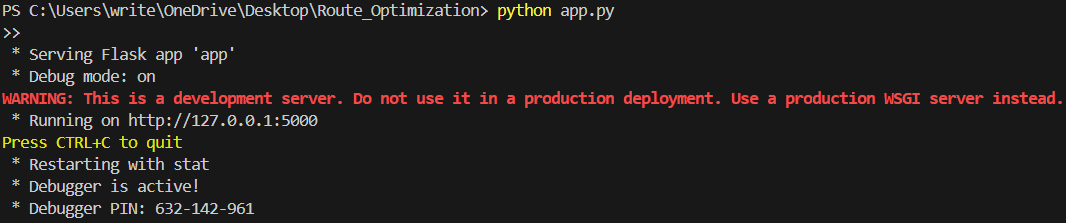
python app.py

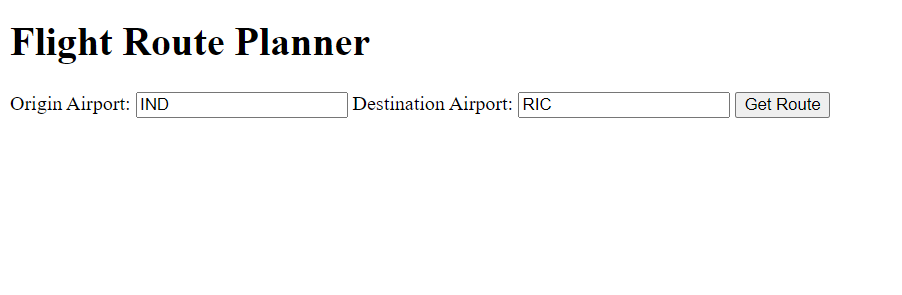
* Access the Web Interface:

Open your web browser and go to http://127.0.0.1:5000 to see the flight route planner interface.



It’ll take some time because all the records from flight\_delay needs to be stored into the database.

  
Once you run it’ll come like this, and go to the localhost link



Then enter and optimal path should be created