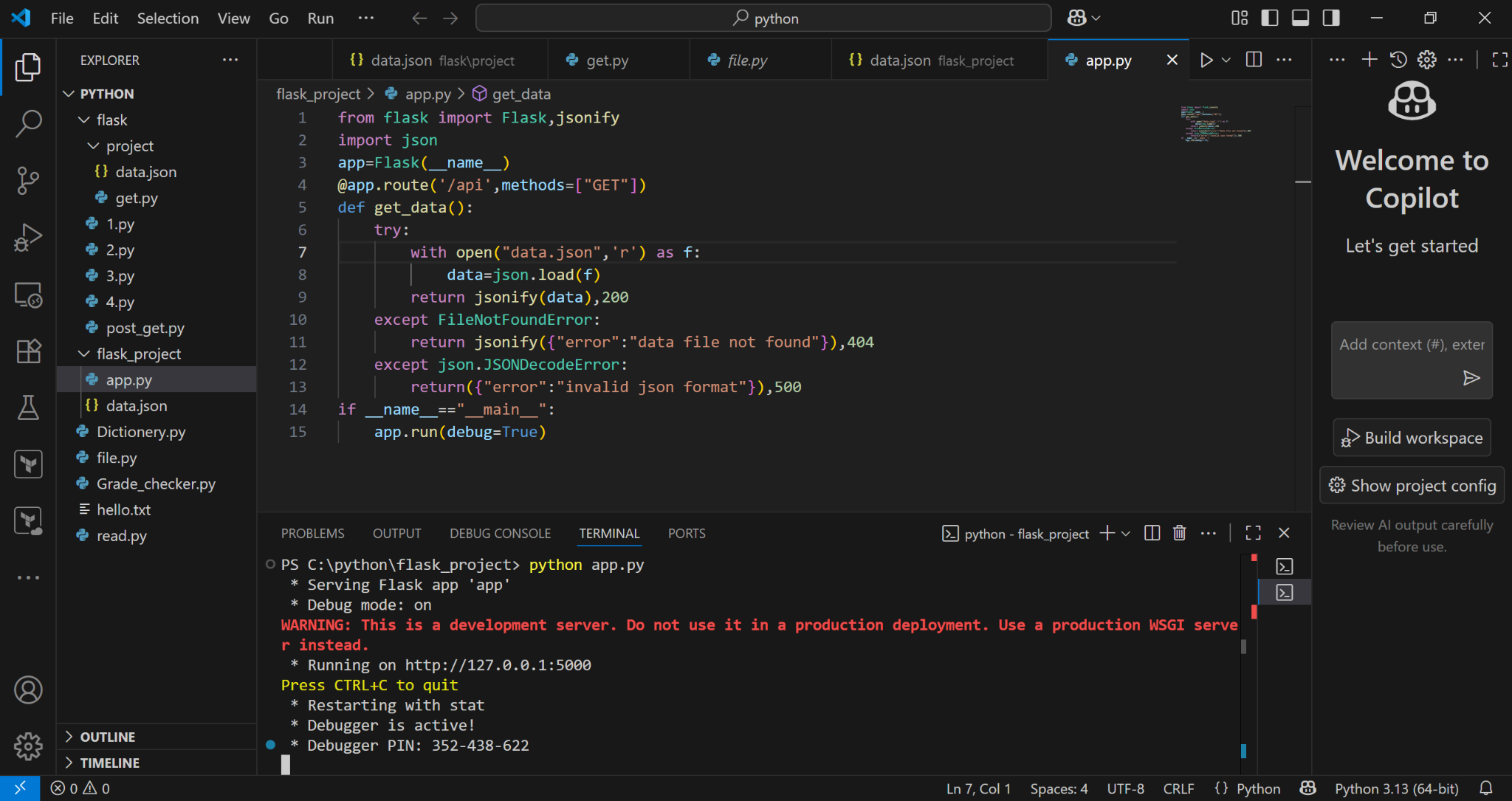
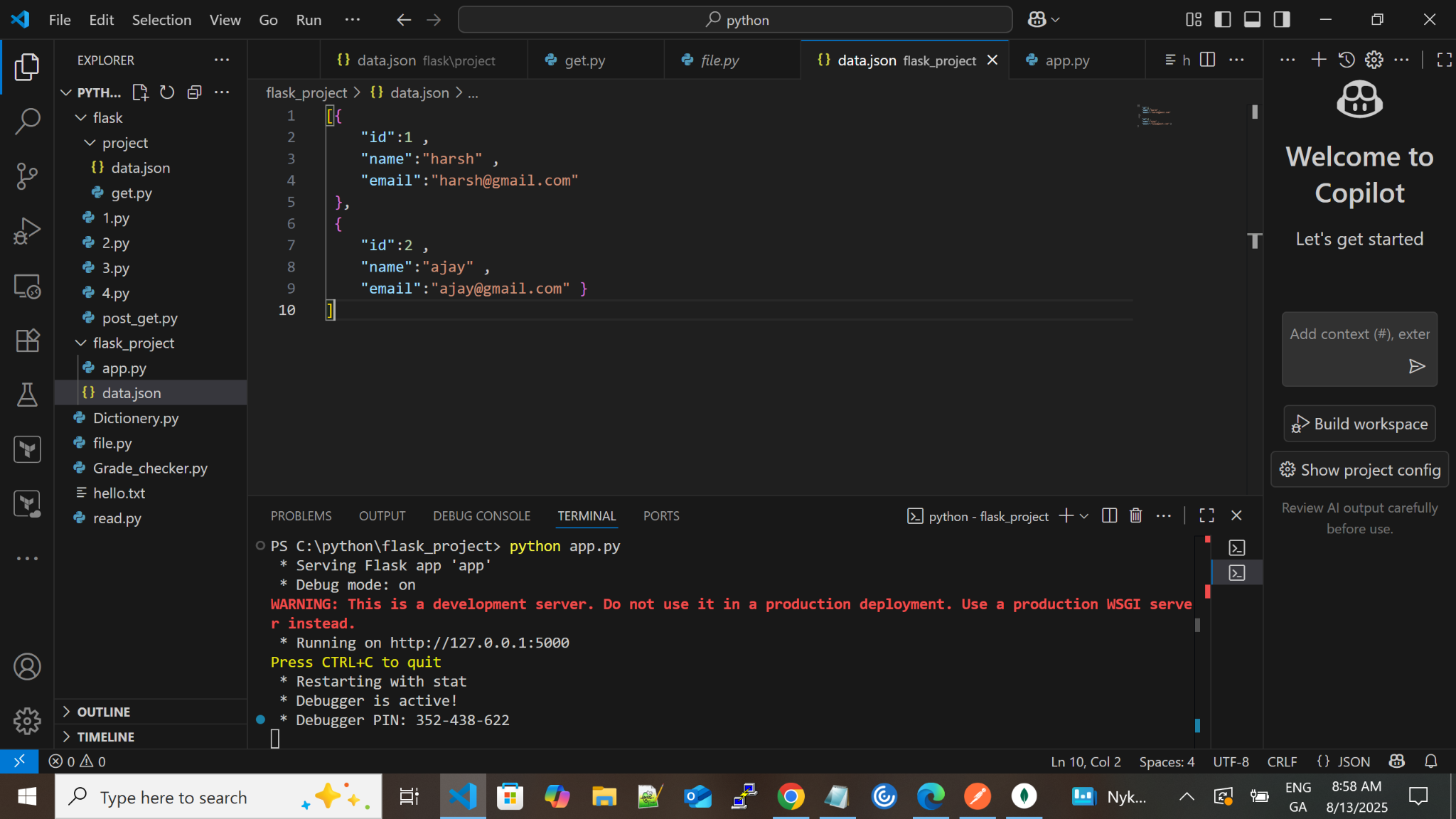
1. Create a Flask application with an /api route. When this route is accessed, it should return a JSON list. The data should be stored in a backend file, read from it, and sent as a response.

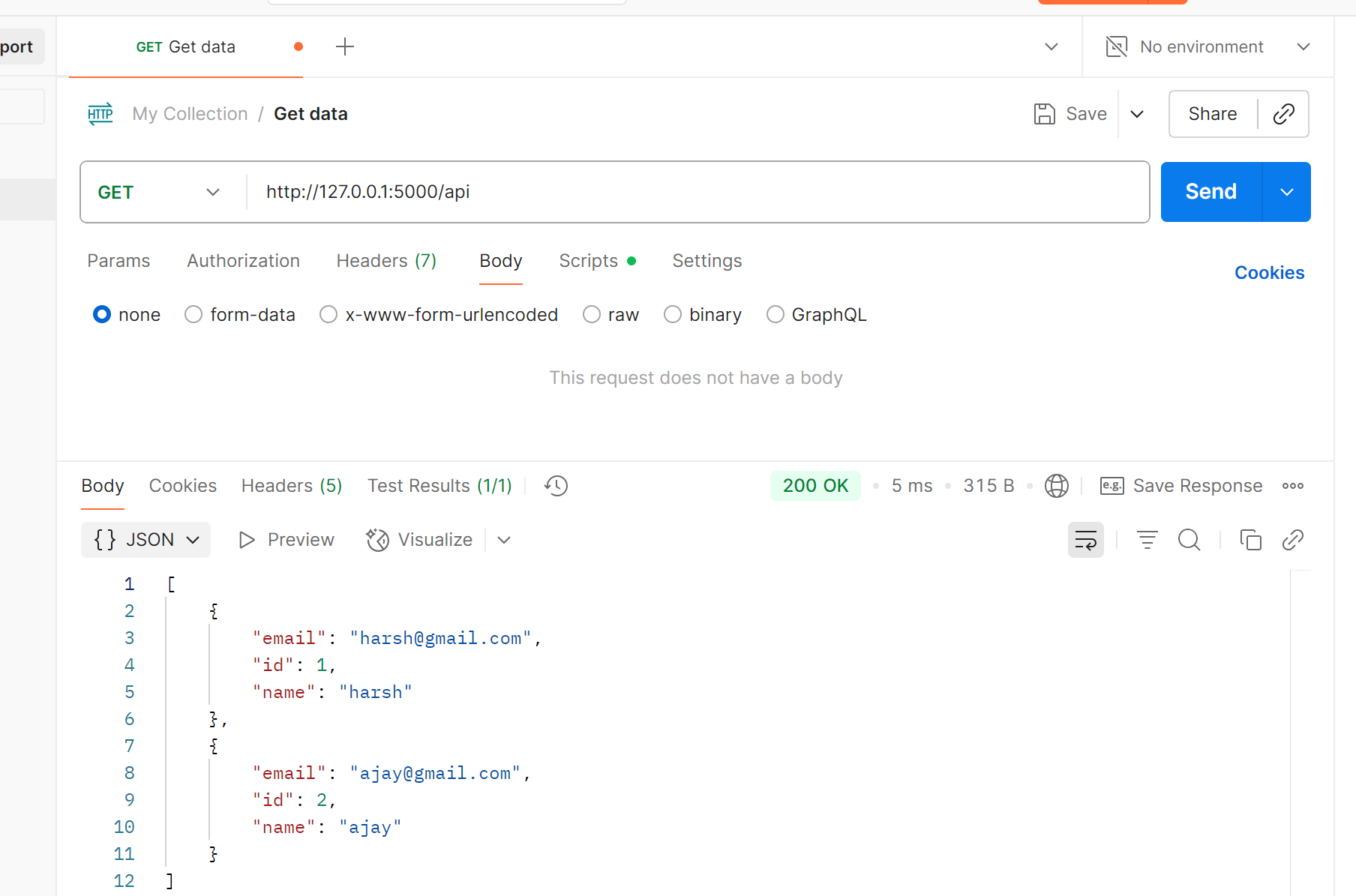
CODE FOR THE PROGRAM:

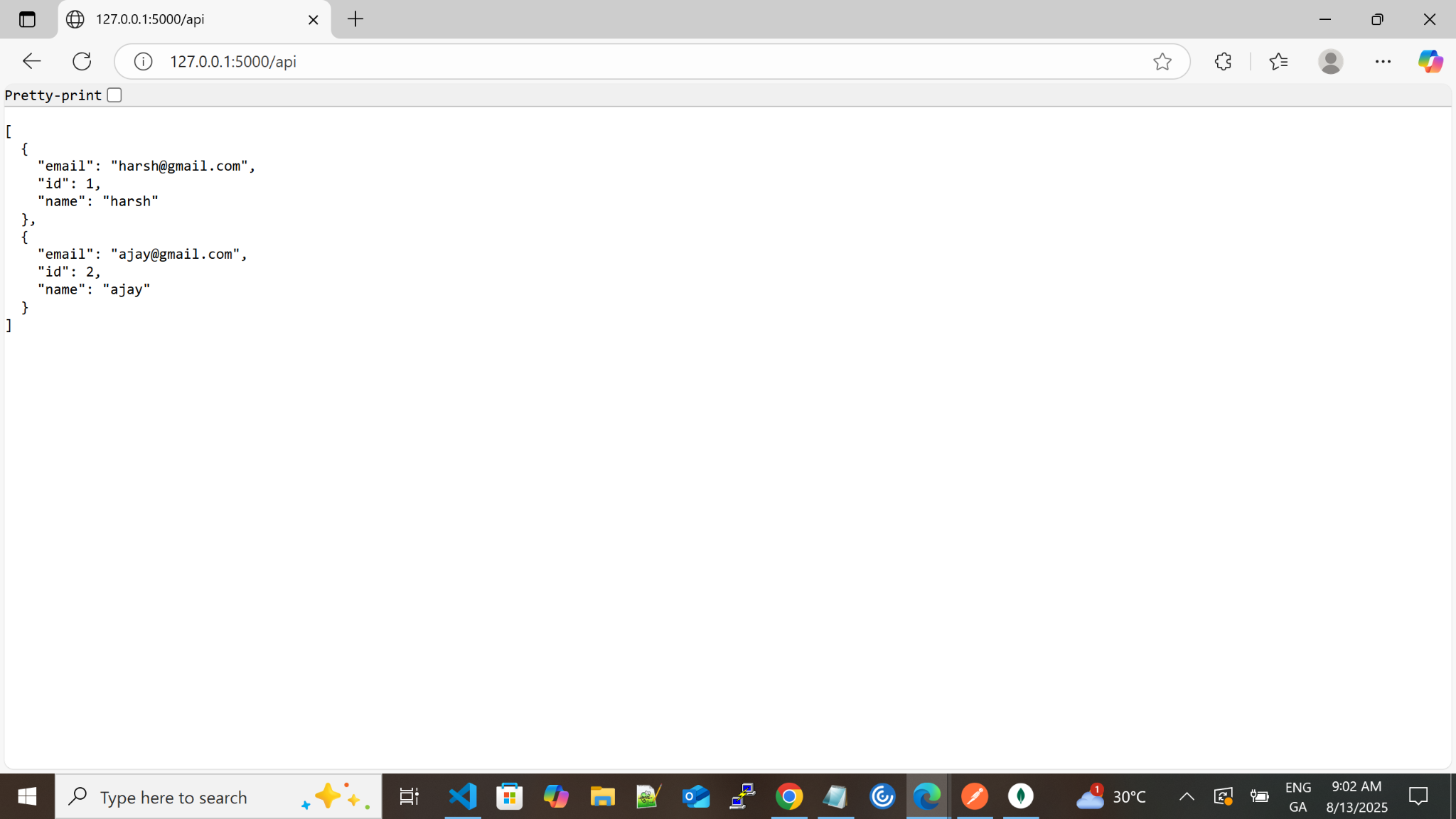


DATA FOR THE PROGRAM SO THE PROGRAM FETCH IT :-



OUTPUT FOR THE PROGRAM BY POSTMAN:



OUTPUT FOR THE PROGRAM BY THE URL:

EXPLANATION OF THE PROGRAM:

**•** from flask import Flask, jsonify

Flask aur jsonify import kar rahe ho.

Flask → Flask app banane ke liye

jsonify → Python ka data (like list or dict) ko JSON response banane ke liye

• import json

Python ka built-in json module import kiya — jisse file se JSON data read kar sakein.

• app = Flask(\_\_name\_\_)

Initializes your Flask application. \_\_name\_\_ tells Flask where to look for resources.

• @app.route('/api', methods=["GET"])

def get\_data():

This sets up a route /api that only responds to GET requests.

When someone visits /api, the get\_data() function runs.

• **Inside the get() function:**

try:

with open("data.json", 'r') as f:

data = json.load(f)

Tries to open a file named data.json in read mode.

If successful, it loads and parses the JSON data from that file into a Python variable called data.

• return jsonify(data), 200

If everything works, it returns the data as a JSON response with HTTP status code 200 OK.

•  **Handling Errors:**

except FileNotFoundError:

return jsonify({"error": "data file not found"}), 404

If data.json doesn’t exist, it returns an error message with status 404 Not Found.

except json.JSONDecodeError:

return({"error": "invalid json format"}), 500

If data.json exists but has invalid JSON syntax, this returns a 500 Internal Server Error with an appropriate message.

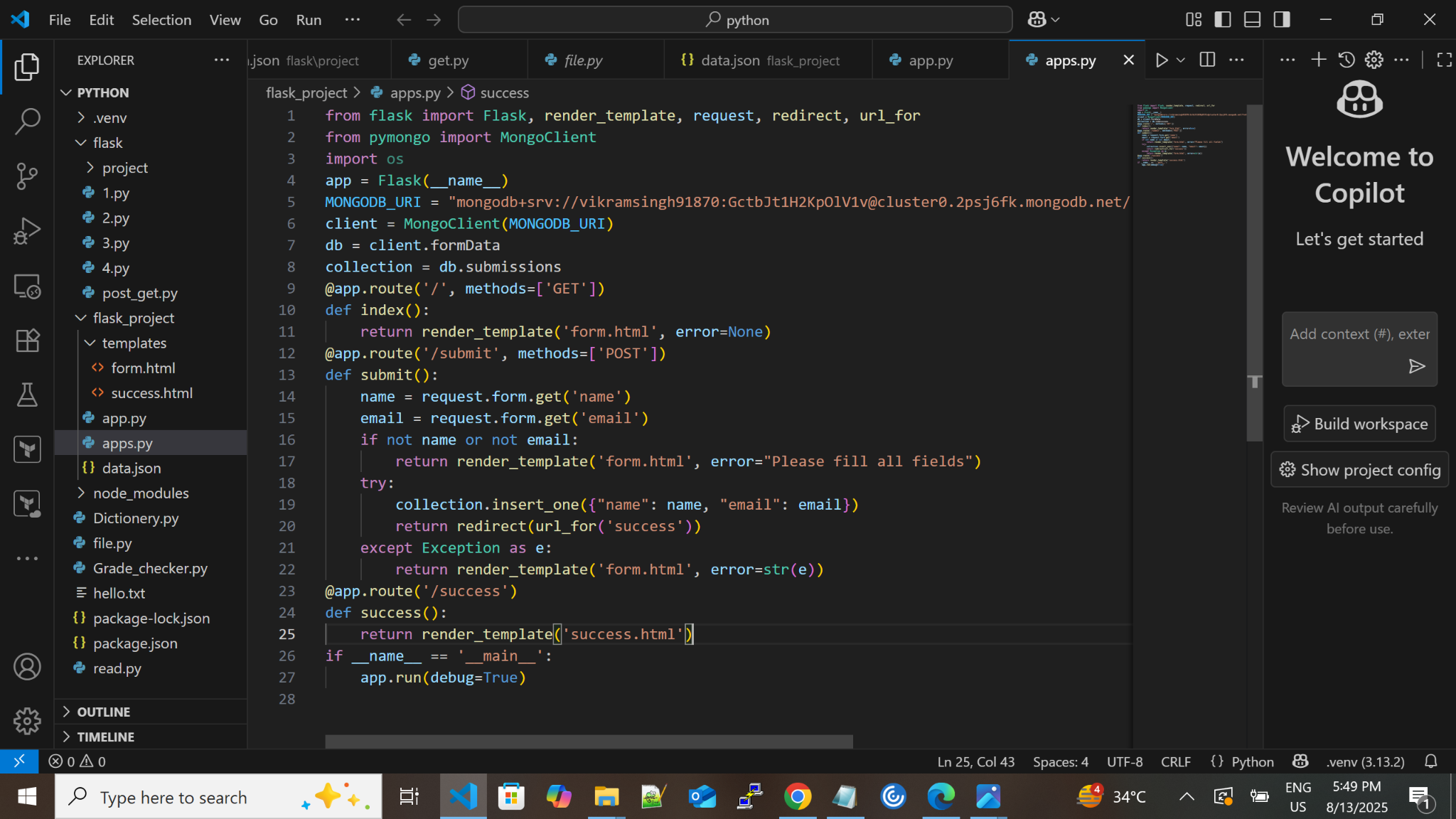
• if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

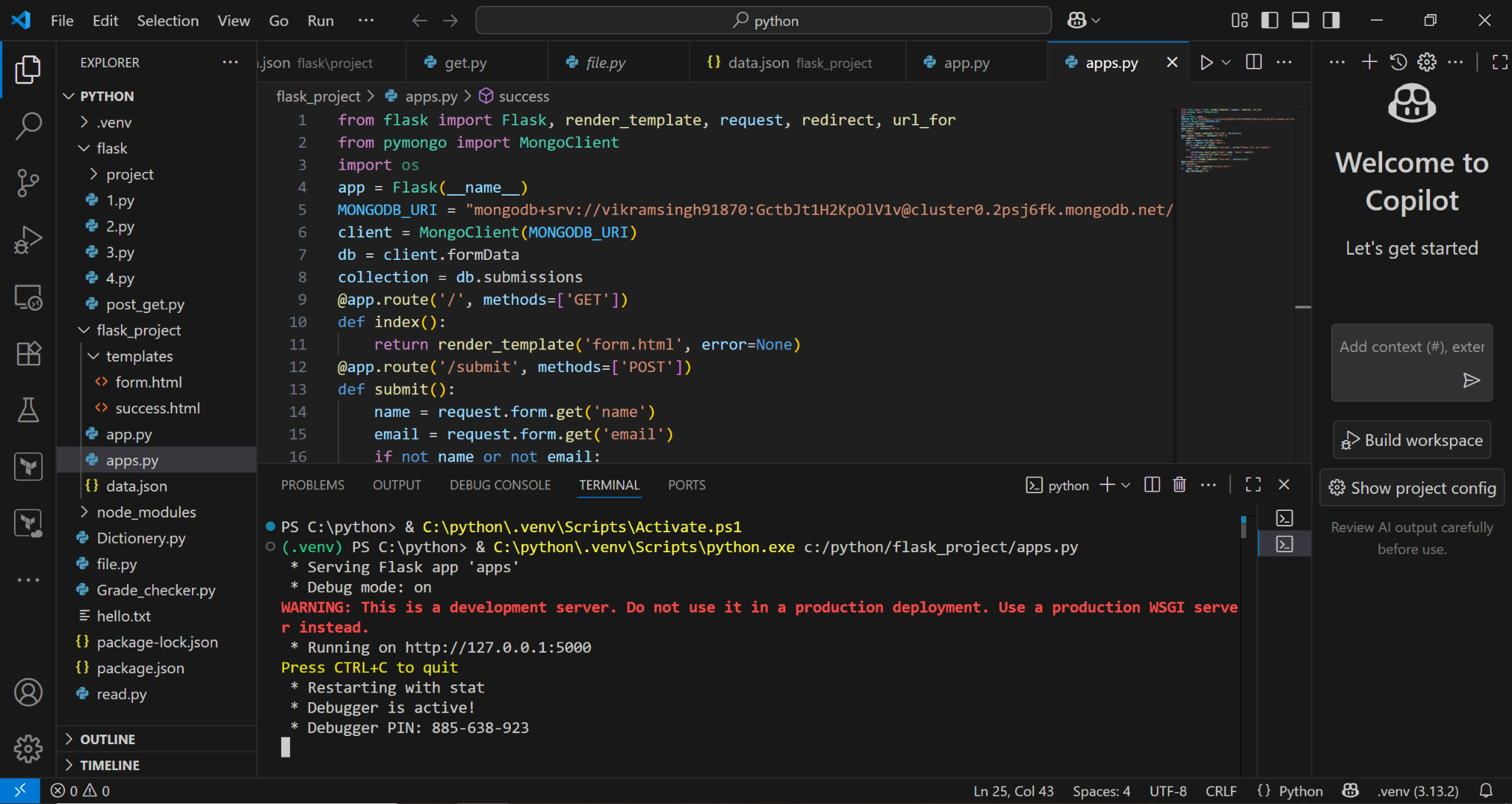
This starts the Flask app in debug mode, so you can see error logs and live reload during development.

2. Create a form on the frontend that, when submitted, inserts data into MongoDB Atlas. Upon successful submission, the user should be redirected to another page displaying the message **"Data submitted successfully"**. If there's an error during submission, display the error on the same page without redirection.

CODE FOR THE PROGRAM:

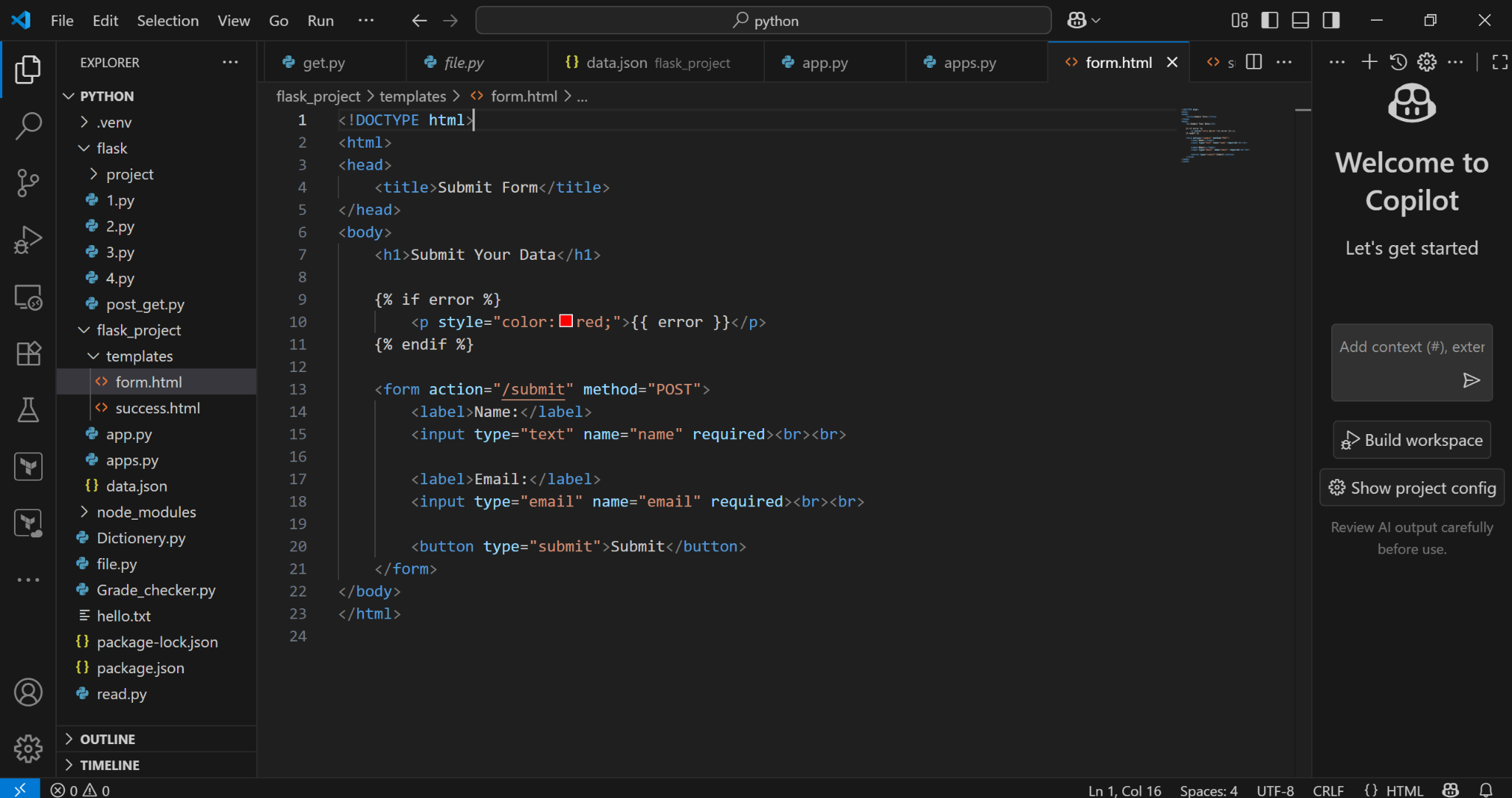


OUTPUT OF THE PROGRAM IN THE VSCODE TERMINAL:

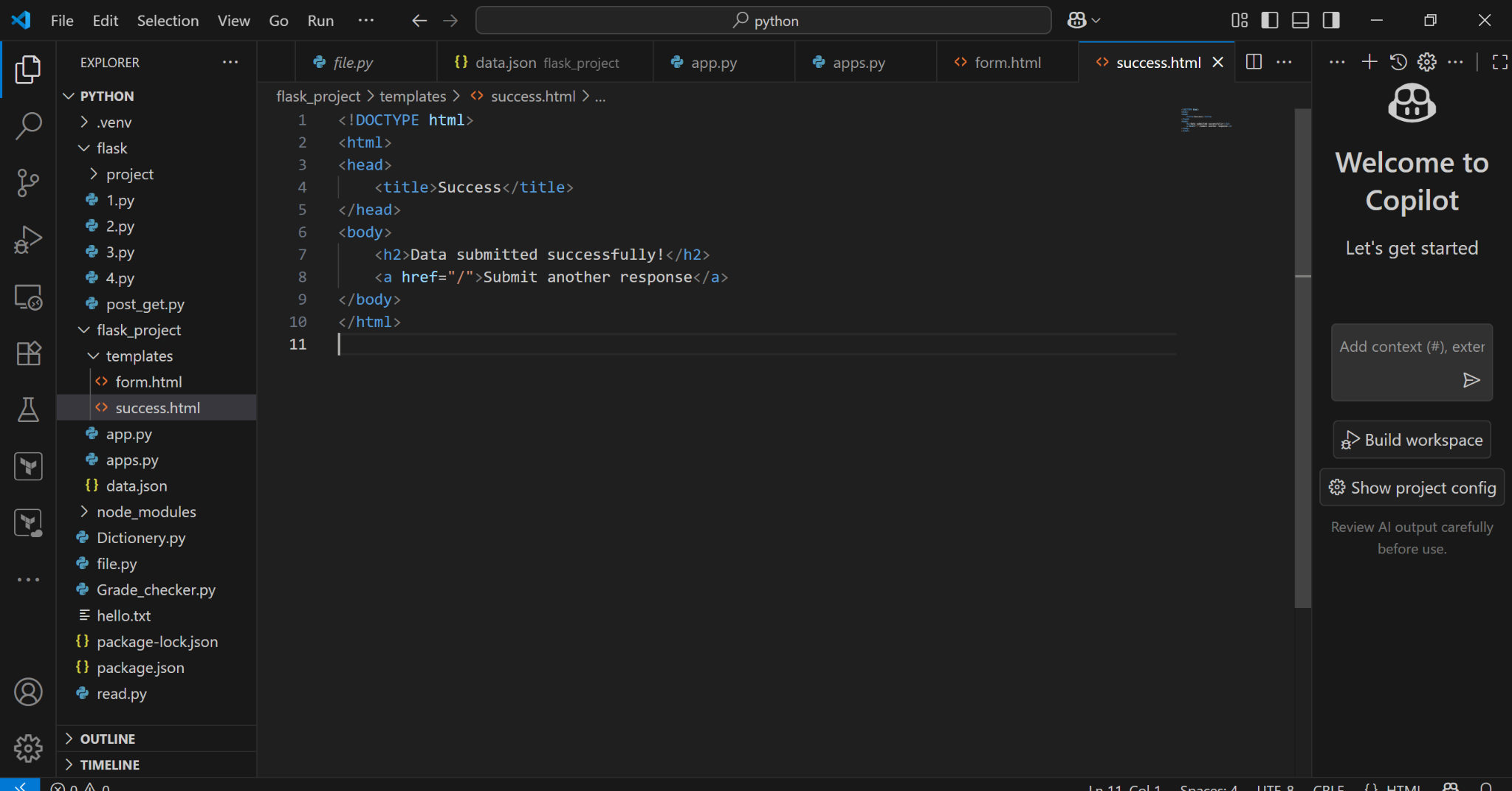


FOR FURTHER PROCESS WE HAVE TO WRITE DIFFERENT HTML CODE IN SAME FOLDER

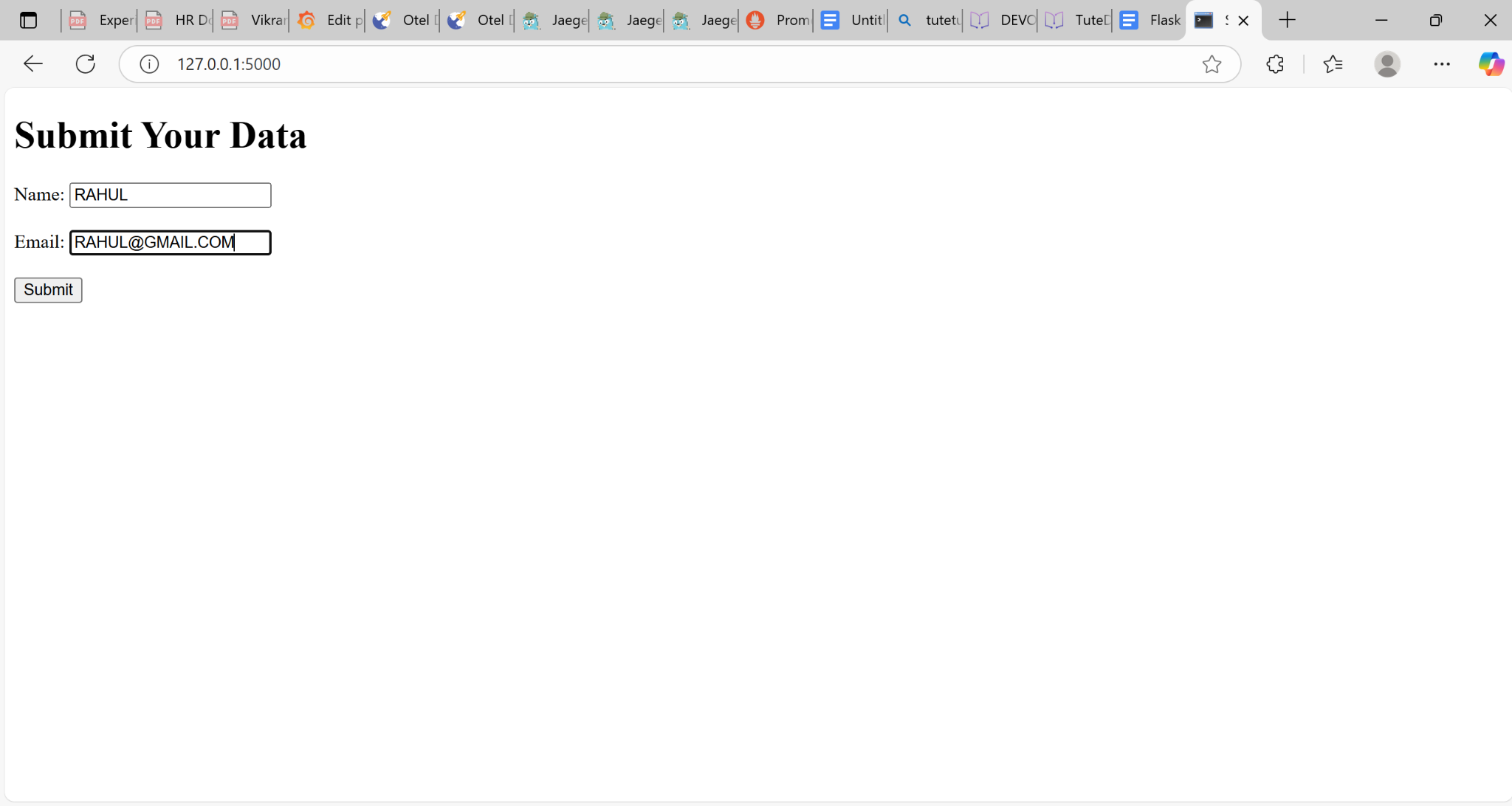
HTML CODE FOR THE PROGRAM FOR MAKING A FORM IN THE FRONTEND:



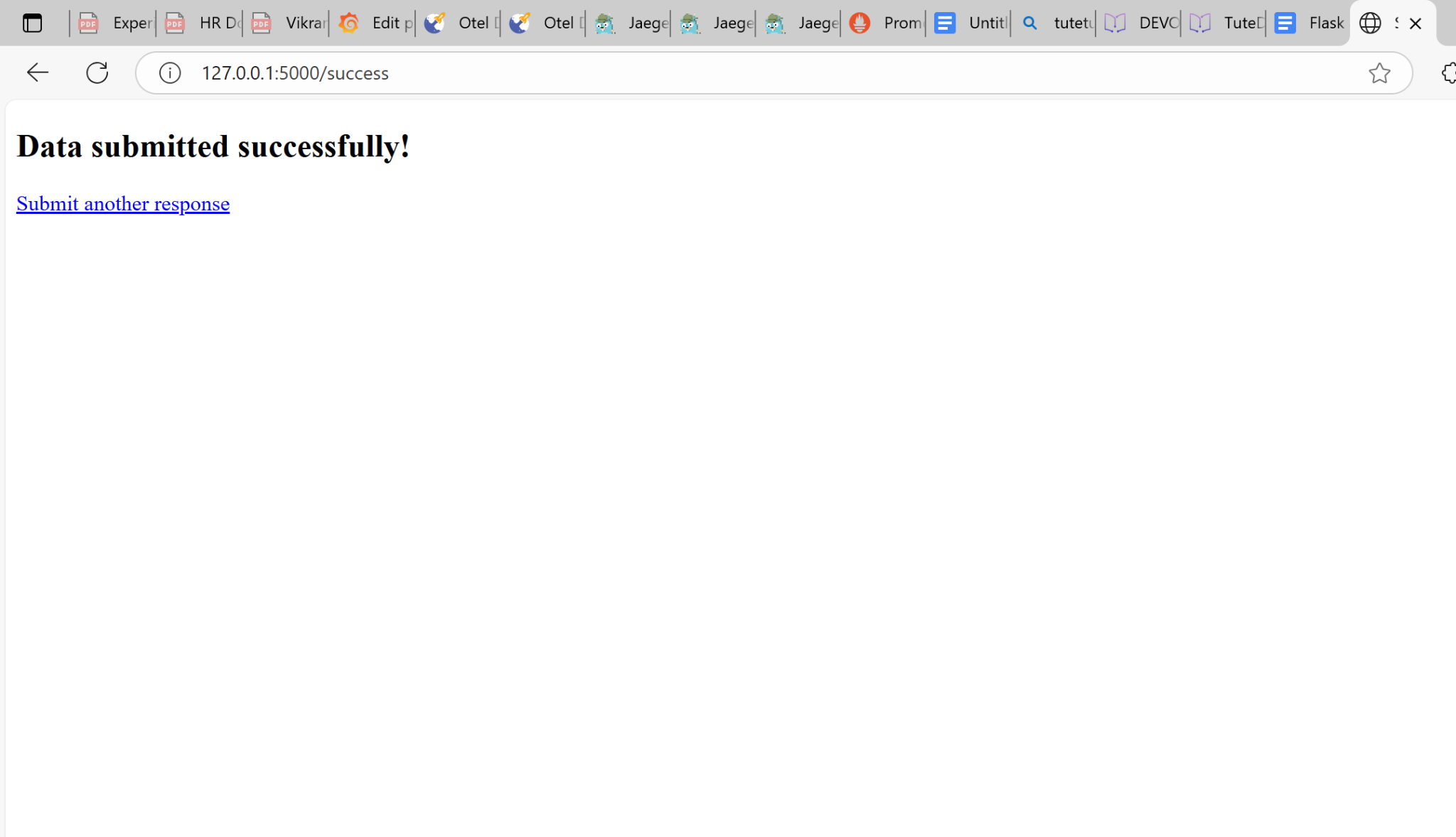
HTML CODE FOR THE PROGRAM FOR SAYING THAT YOUR DATA IS SUBMITTED SUCCESSFULLY:



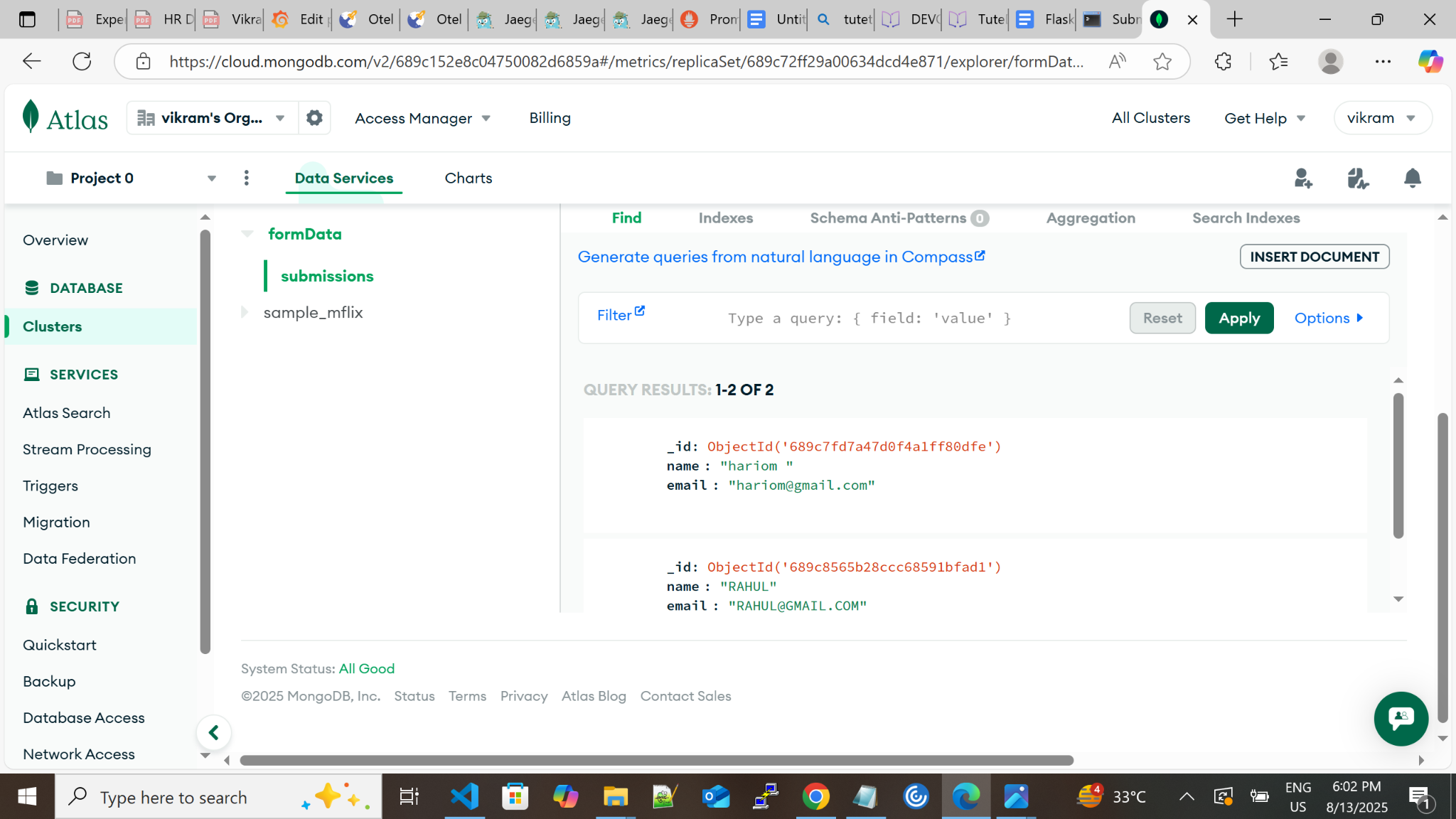
THE OUTPUT OF THE PROGRAM IN THE URL:



AFTER GIVING THE DATA IN THE FIELD IT SHOWS:



NOW CHECK THIS DATA IN THE MONGODB ATLAS:



YES THE DATA IS SAVED IN THE MONGODB ATLAS.

EXPLANATION OF THE PROGRAM: