# Assignment3

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.

Answer: -

from flask import Flask, jsonify, render\_template

import json

app = Flask(\_\_name\_\_)

@app.route('/', methods=['GET'])

def home():

# Homepage route

return "<h1>Welcome to Flask API</h1><p>Go to <a href='/api'>/api</a> to see the JSON data</p>"

@app.route('/api', methods=['GET'])

def get\_data():

try:

# Open and read the JSON file

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

data = json.load(file) # Load JSON data as a Python list

return jsonify(data) # Return as JSON response

except Exception as e:

return jsonify({"error": str(e)}), 500 # Error handling

@app.errorhandler(404)

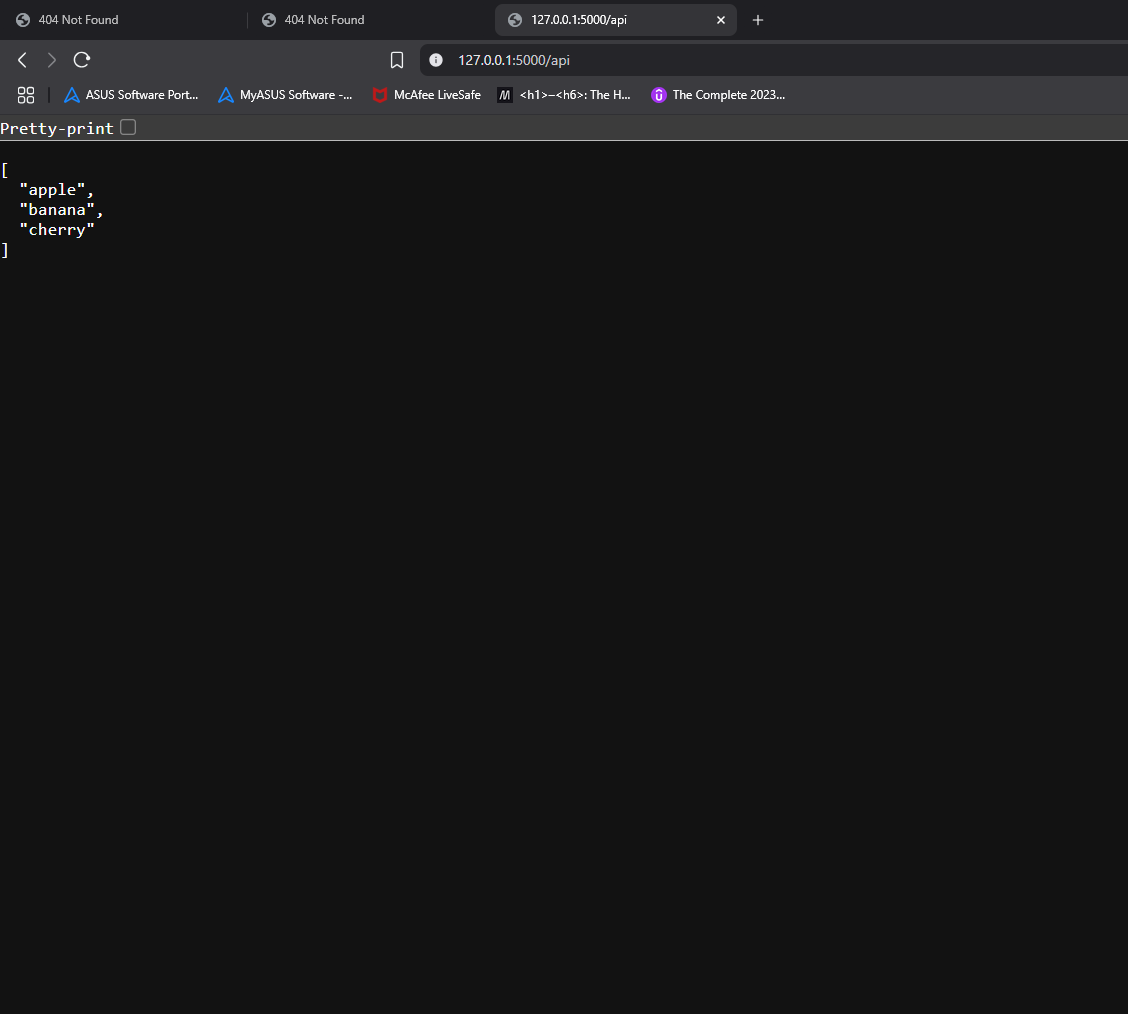
def page\_not\_found(e):

# Custom 404 handler

return "<h1>404 - Page Not Found</h1><p>The requested URL was not found on the server.</p><p>Please check the URL or go to <a href='/'>Homepage</a> or <a href='/api'>API</a>.</p>", 404

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)



. 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 w ithout redirection.

Answer: -

from flask import Flask, render\_template, request, redirect, url\_for, flash

from pymongo import MongoClient

from pymongo.errors import PyMongoError

app = Flask(\_\_name\_\_)

app.secret\_key = "super\_secret\_key"

# MongoDB Atlas connection setup

MONGO\_URI = "mongodb+srv://wobece6207:eGL7TgW3ZOV6N3Ci@cluster0.nygjpwz.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0"

try:

    client = MongoClient(MONGO\_URI)

    # Test connection

    client.admin.command('ping')

    print("MongoDB connection successful")

    db = client['user\_database']  # you can rename it as needed

    collection = db['user\_data']  # collection/table

except PyMongoError as e:

    print(f"MongoDB connection error: {str(e)}")

    # We'll continue execution but the app will display errors when trying to access the database

@app.route("/", methods=["GET", "POST"])

def form():

    if request.method == "POST":

        name = request.form.get("name")

        email = request.form.get("email")

        # Validate fields

        if not name or not email:

            flash("Name and Email are required.")

            return render\_template("form.html")

        try:

            # Insert into MongoDB

            collection.insert\_one({"name": name, "email": email})

            return redirect(url\_for("success"))

        except PyMongoError as e:

            flash(f"Database Error: {str(e)}")

            return render\_template("form.html")

    return render\_template("form.html")

@app.route("/success")

def success():

    return render\_template("success.html")

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

