AI-Powered Web Application Using Flask and NumPy

This Python script sets up a **Flask-based web application** that integrates **AI capabilities**. Flask is a lightweight web framework that allows AI models to interact with users via a web interface. The script includes:

- **Importing Flask:** Enables the creation of web applications that process user input and display outputs dynamically.
- **Importing NumPy:** Provides numerical computing capabilities, which can be used for AI-related operations such as data transformations and predictions.
- **Defining** predict_next **(Dummy AI Function):** This placeholder function can later be replaced with an actual AI model, likely for predictive tasks such as forecasting trends or making intelligent recommendations.

This approach is commonly used to **deploy AI models** as interactive web applications, allowing users to input data, receive AI-generated insights, and interact with intelligent systems in real-time.

File No. 1: app.py

This file is necessary to start Flask Application as Interactive Web UI. Execute the command 'python app.py' to start with this application.

Code:

```
from flask import Flask, render_template, request
import numpy as np

app = Flask(__name__)

# Dummy AI function (to be replaced with your trained model)
def predict_next_number(sequence):
    return sequence[-1] + (sequence[1] - sequence[0])  # Basic pattern detection

@app.route("/", methods=["GET", "POST"])
def index():
    prediction = None
    if request.method == "POST":
        sequence = list(map(int, request.form["sequence"].split(",")))
        prediction = predict_next_number(sequence)

    return render_template("index.html", prediction=prediction)

if __name__ == "__main__":
    app.run(debug=True)
```

File No. 2: index.html

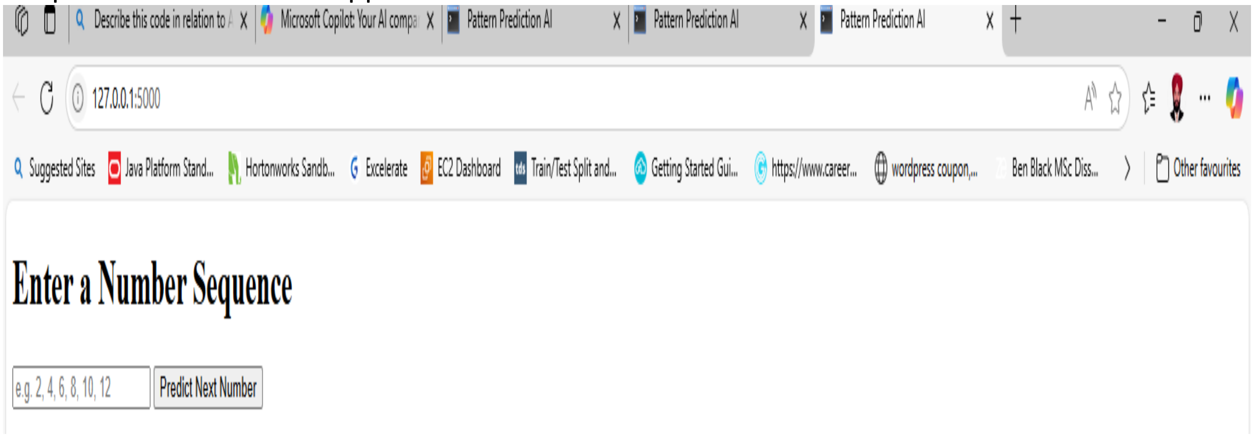
This file resides with templates folder and is called by app.py during its execution. You can access the Web User Interface from inside a web browser of you choice using the url given below:

```
(base) PS C:\Users\your-loggedin_user_name\NumberPredictionFromSequence> python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 915-525-297
```

Code:

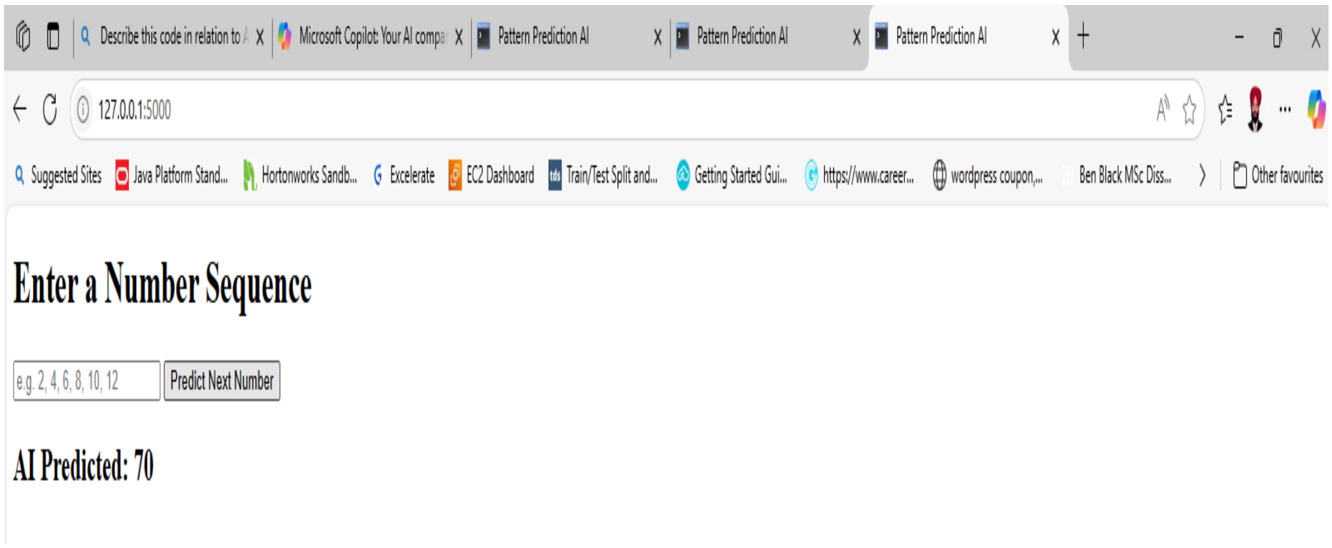
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Pattern Prediction AI</title>
</head>
<body>
   <h1>Enter a Number Sequence</h1>
   <form method="POST">
      <input type="text" name="sequence" placeholder="e.g. 2, 4, 6, 8, 10, 12">
      <button type="submit">Predict Next Number</button>
   </form>

   {% if prediction is not none %}
      <h2>AI Predicted: {{ prediction }}</h2>
   {% endif %}
</body>
</html>
```

Output from this AI Web App



Enter a Number Sequence

Result given by the App for first User Input



Enter a Number Sequence

e.g. 2, 4, 6, 8, 10, 12

Predict Next Number

AI Predicted: 70

User’s First Input	Output for the First input
1 10, 25, 40, 55	**AI Predicted: 70**
User’s Second Input	Output for the Second input
2, 8, 14, 20, 26	**AI Predicted: 32**
User’s Third Input	Output for the Third input
3, 8, 11, 19	**AI Predicted: 24**

You can try further for number sequence of your choice for which you know the correct answer. For example try out for the answers to Arithmetic, Geometric and Exponential Number Sequences at your own.

----------x------------

Flask is not inside the scope of tenth and ninth standards, but you like to know something more about this cutting-edge technology for designing User Interfaces. Here is a brief note about it.

Flask is an excellent choice for building web-based user interfaces (UIs) due to its lightweight and flexible nature. It is a micro web framework for Python that allows developers to create web applications quickly and efficiently. Here's why Flask is a great platform for web UIs:

1. Lightweight and Minimalistic
Flask is designed to be simple and unopinionated, meaning you can build your application exactly the way you want without unnecessary overhead. It provides the essential tools to get started, and you can add extensions as needed.

2. Easy Integration with Frontend Technologies

Flask works seamlessly with frontend frameworks like **React**, **Vue.js**, or **Angular**, as well as templating engines like **Jinja2** (built into Flask). You can use Jinja2 to dynamically render HTML pages with Python variables, making it easy to create interactive UIs.

Example of a simple Flask route rendering a Jinja2 template:

Copy the codefrom flask import Flask, render_template

```
app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html', title="Welcome to Flask UI")

if __name__ == '__main__':
    app.run(debug=True)
```

3. RESTful API Support

Flask makes it easy to build RESTful APIs, which can serve as the backend for your web UI. This is particularly useful if you're building a single-page application (SPA) or need to connect to external services.

Example of a simple API endpoint:

```
Copy the code@app.route('/api/data', methods=['GET'])
def get_data():
    return {"message": "Hello, Flask UI!"}
```

4. Extensibility

Flask has a rich ecosystem of extensions for adding functionality like:

- **Flask-WTF** for form handling.
- **Flask-SQLAlchemy** for database integration.
- **Flask-Login** for user authentication.
- **Flask-Bootstrap** for integrating Bootstrap UI components.

5. Debugging and Development Tools

Flask comes with a built-in development server and debugger, making it easy to test and debug your web UI during development. The debug=True mode provides detailed error messages and an interactive debugger.

6. Scalability

While Flask is lightweight, it can scale to handle complex applications by structuring your project into blueprints or modules. This makes it suitable for both small projects and large-scale applications.

Use Cases

- **Dashboards**: Flask is often used to create data visualization dashboards with libraries like **Plotly** or **Matplotlib**.
- **Admin Panels**: Flask can power admin interfaces with tools like **Flask-Admin**.
- **Prototyping**: Its simplicity makes it ideal for quickly prototyping web UIs.

Flask is a versatile and approachable framework for web UI development, whether you're building a simple website or a complex application. Its flexibility allows you to tailor it to your specific needs, making it a popular choice among developers.