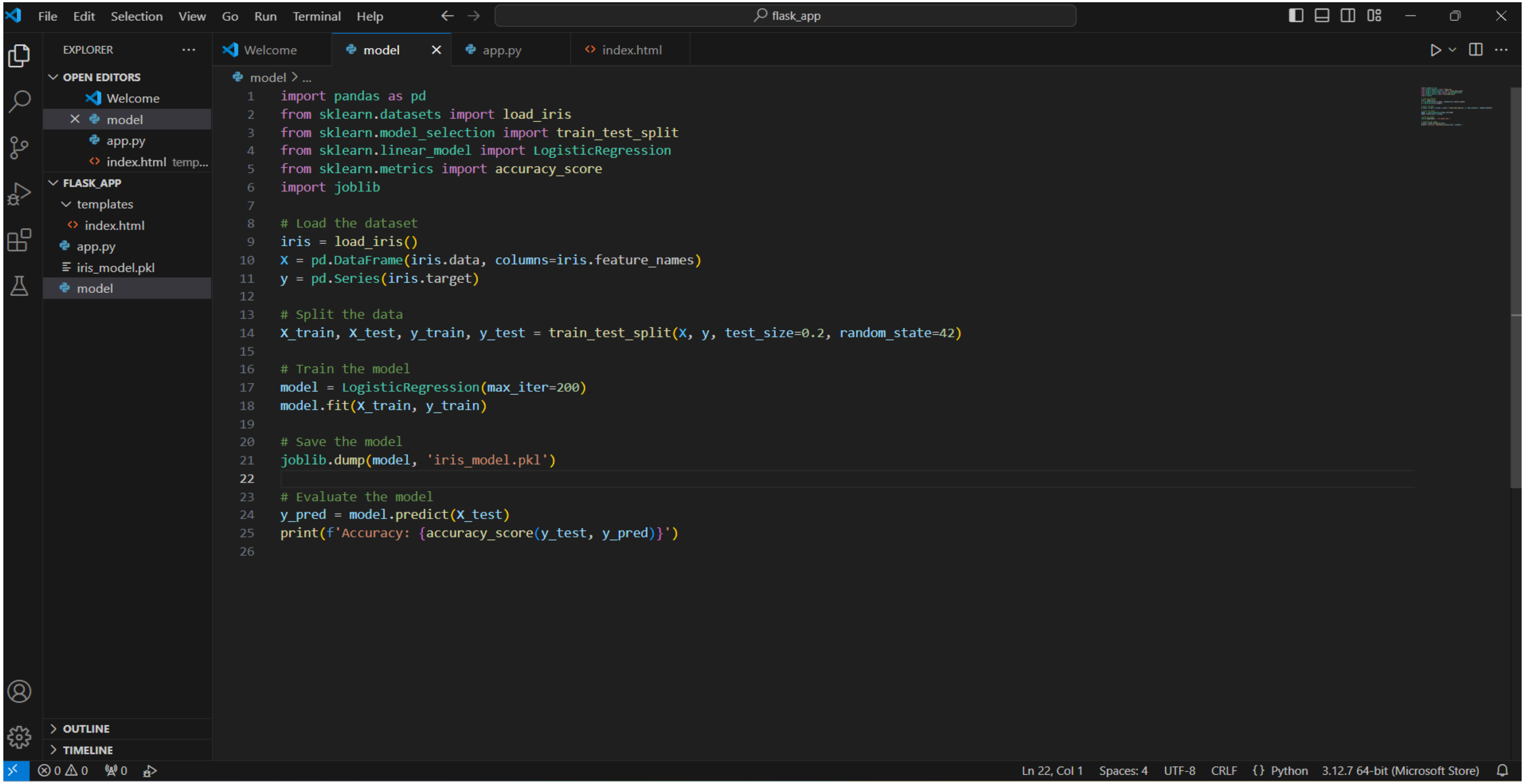
Week4: Deployment on Flask

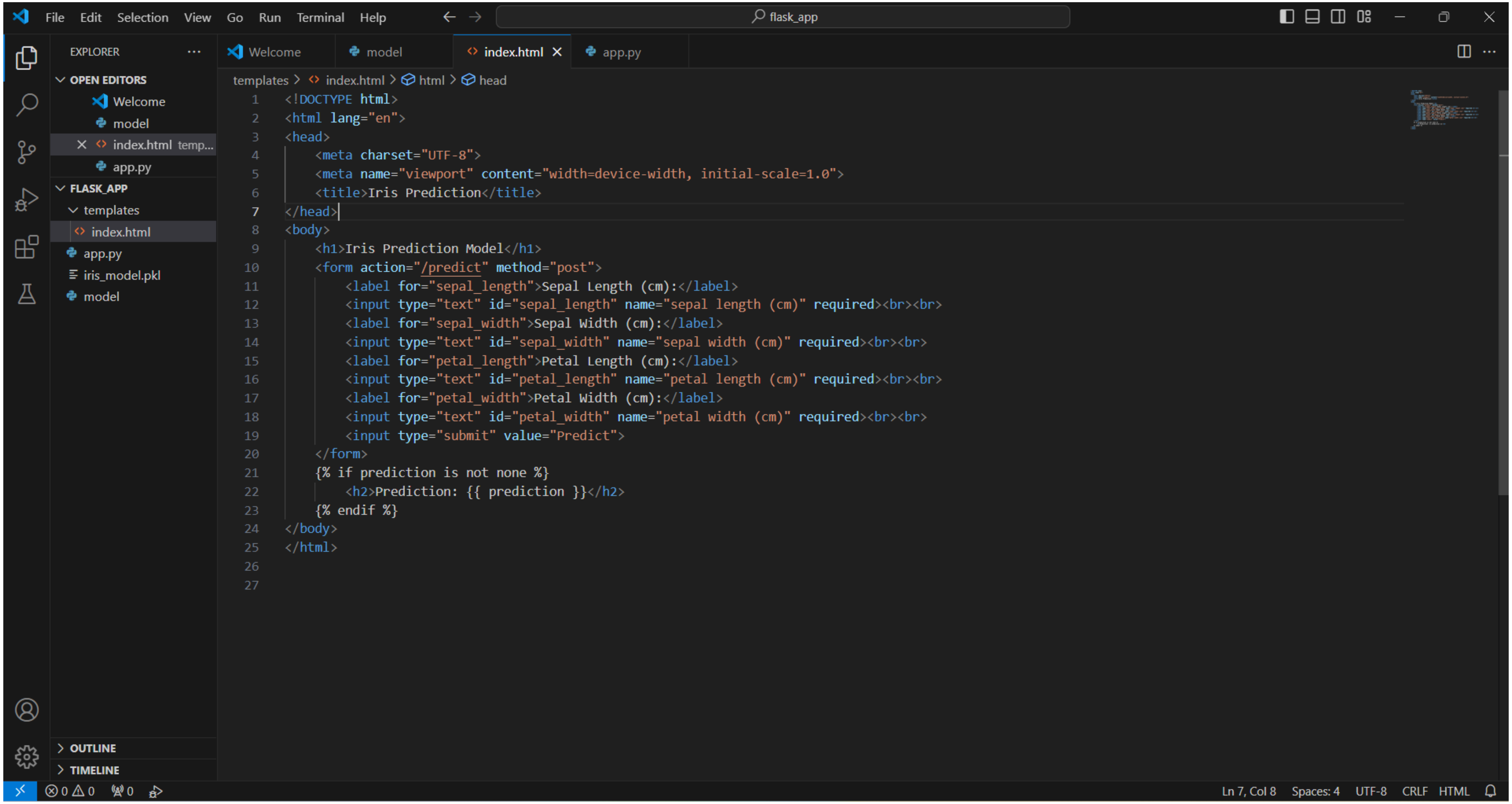
Name: Amrin Shaikh
Batch code: LISUM38
Submission date:10/10/2024
Submission to: Data Glacier

1. **Load the dataset and train the model:**



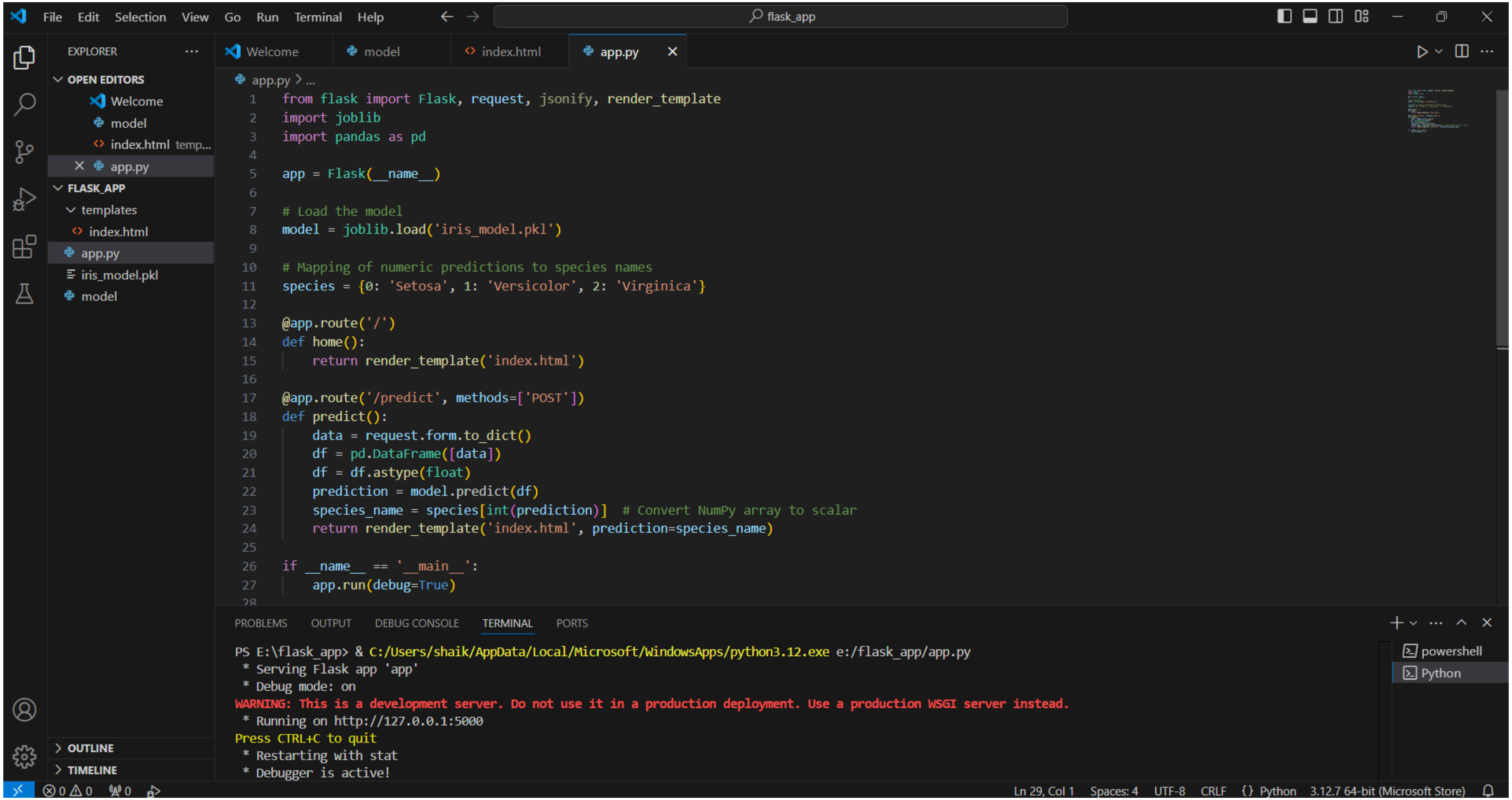
```python
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import joblib

# Load the dataset
iris = load_iris()
X = pd.DataFrame(iris.data, columns=iris.feature_names)
y = pd.Series(iris.target)

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the model
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)

# Save the model
joblib.dump(model, 'iris_model.pkl')

# Evaluate the model
y_pred = model.predict(X_test)
print(f'Accuracy: {accuracy_score(y_test, y_pred)}')
```

2.Create index.html:



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Iris Prediction</title>
</head>
<body>
    <h1>Iris Prediction Model</h1>
    <form action="/predict" method="post">
        <label for="sepal_length">Sepal Length (cm):</label>
        <input type="text" id="sepal_length" name="sepal length (cm)" required><br><br>
        <label for="sepal_width">Sepal Width (cm):</label>
        <input type="text" id="sepal_width" name="sepal width (cm)" required><br><br>
        <label for="petal_length">Petal Length (cm):</label>
        <input type="text" id="petal_length" name="petal length (cm)" required><br><br>
        <label for="petal_width">Petal Width (cm):</label>
        <input type="text" id="petal_width" name="petal width (cm)" required><br><br>
        <input type="submit" value="Predict">
    </form>
    {% if prediction is not none %}
        <h2>Prediction: {{ prediction }}</h2>
    {% endif %}
</body>
</html>


```

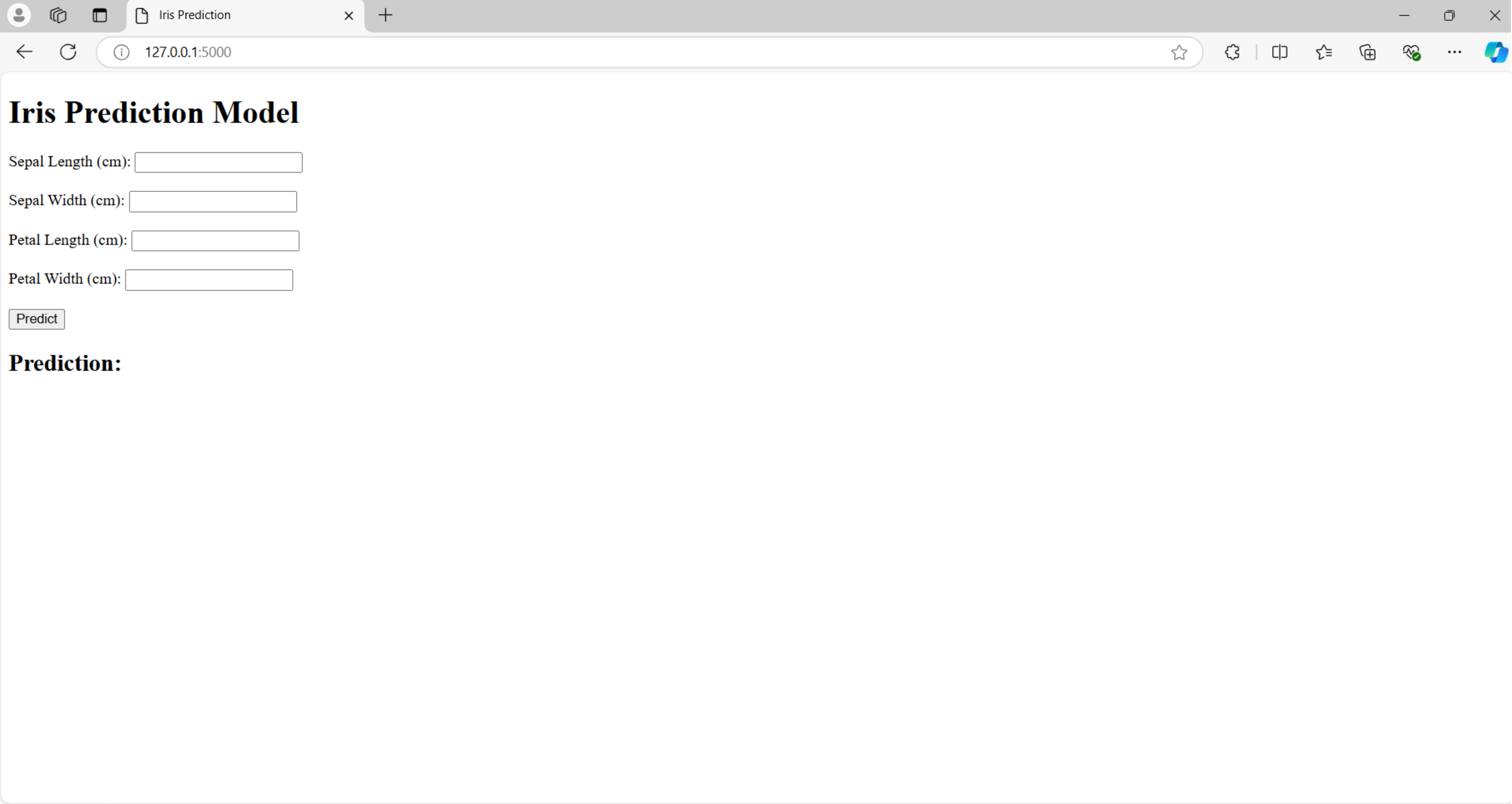
3.Deploy the Model on Flask:



```python
from flask import Flask, request, jsonify, render_template
import joblib
import pandas as pd

app = Flask(__name__)

# Load the model
model = joblib.load('iris_model.pkl')

# Mapping of numeric predictions to species names
species = {0: 'Setosa', 1: 'Versicolor', 2: 'Virginica'}

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    data = request.form.to_dict()
    df = pd.DataFrame([data])
    df = df.astype(float)
    prediction = model.predict(df)
    species_name = species[int(prediction)]  # Convert NumPy array to scalar
    return render_template('index.html', prediction=species_name)

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

```
PS E:\flask_app> & C:/Users/shaik/AppData/Local/Microsoft/WindowsApps/python3.12.exe e:/flask_app/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 stat
 * Debugger is active!
```

5. Testing the Flask Deployment



Iris Prediction Model

Sepal Length (cm):

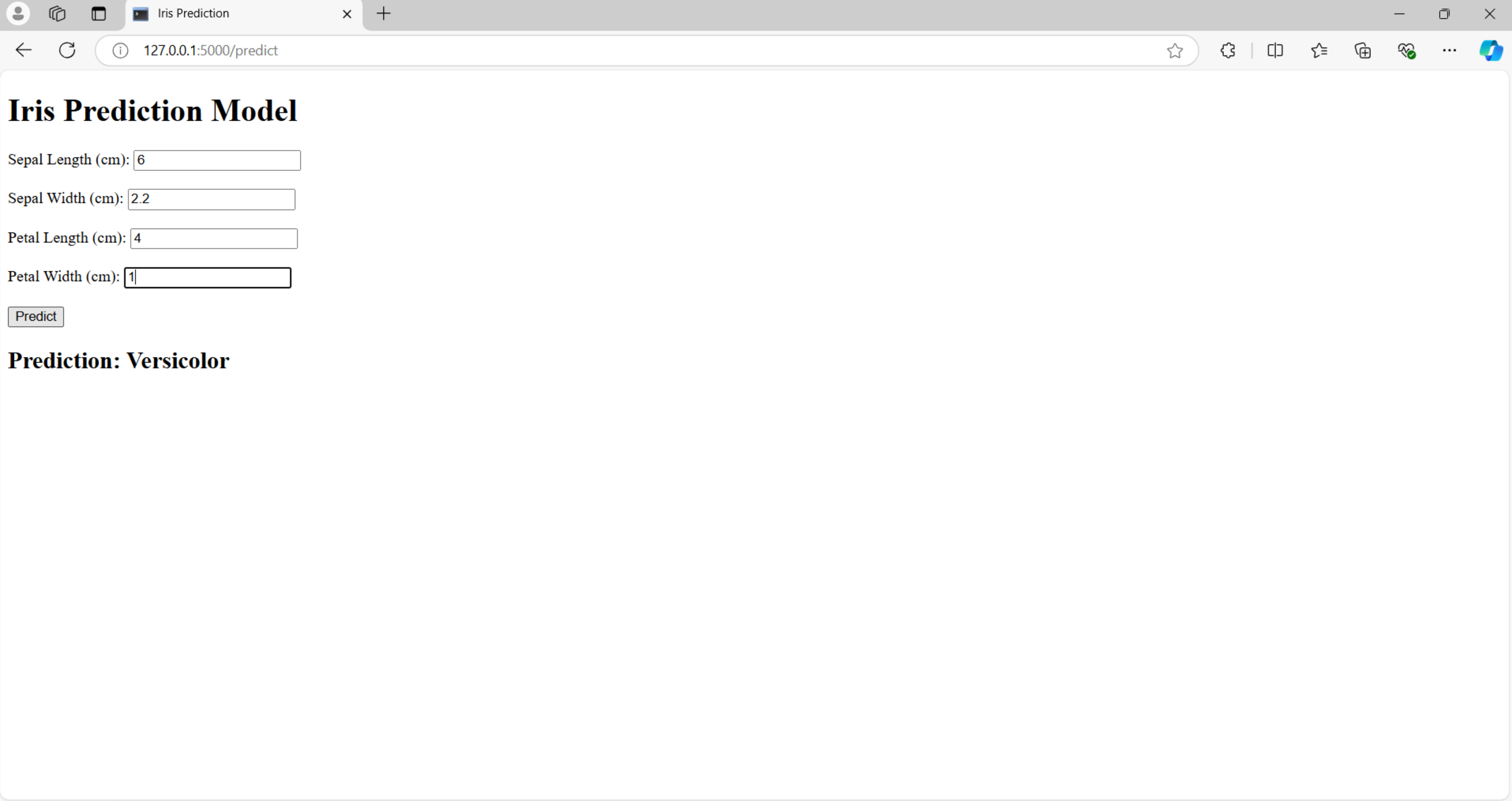
Sepal Width (cm):

Petal Length (cm):

Petal Width (cm):

Prediction:

6. Enter the values & get the prediction result



Iris Prediction Model

Sepal Length (cm): 6

Sepal Width (cm): 2.2

Petal Length (cm): 4

Petal Width (cm): 1

Prediction: Versicolor