

Week4: Deployment on Flask

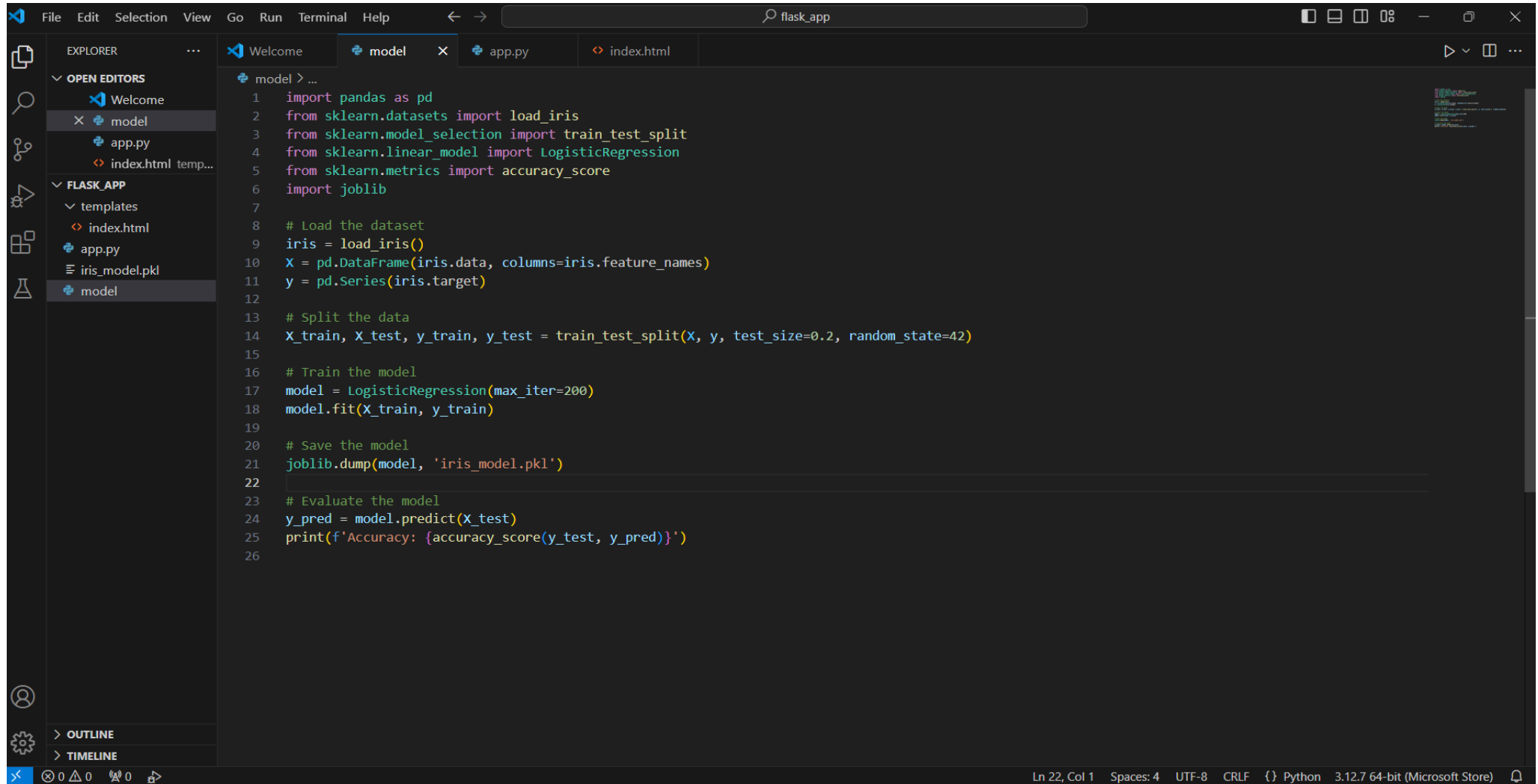
Name: Amrin Shaikh

Batch code: LISUM38

Submission date:10/10/2024

Submission to: Data Glacier

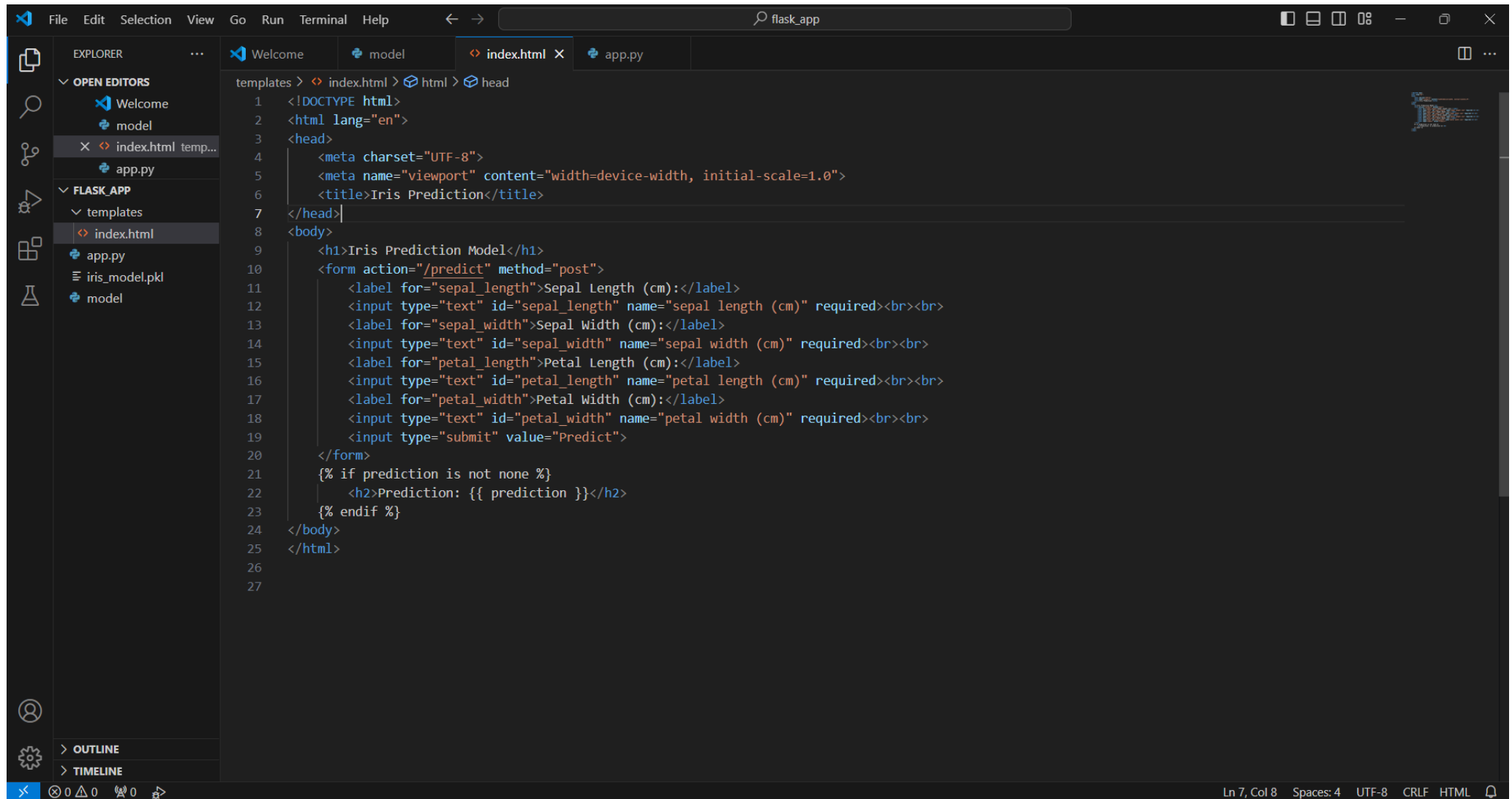
1. Load the dataset and train the model:



The screenshot shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project named 'FLASK_APP' with files 'index.html', 'app.py', 'iris_model.pkl', and 'model'. The 'model' file is selected and open in the editor. The code in the editor is a Python script that loads the Iris dataset, splits it into training and testing sets, trains a Logistic Regression model, and evaluates its accuracy. The status bar at the bottom indicates the current position is Line 22, Column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.12.7 64-bit (Microsoft Store) interpreter.

```
1 import pandas as pd
2 from sklearn.datasets import load_iris
3 from sklearn.model_selection import train_test_split
4 from sklearn.linear_model import LogisticRegression
5 from sklearn.metrics import accuracy_score
6 import joblib
7
8 # Load the dataset
9 iris = load_iris()
10 X = pd.DataFrame(iris.data, columns=iris.feature_names)
11 y = pd.Series(iris.target)
12
13 # Split the data
14 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
15
16 # Train the model
17 model = LogisticRegression(max_iter=200)
18 model.fit(X_train, y_train)
19
20 # Save the model
21 joblib.dump(model, 'iris_model.pkl')
22
23 # Evaluate the model
24 y_pred = model.predict(X_test)
25 print(f'Accuracy: {accuracy_score(y_test, y_pred)}')
```

2.Create index.html:

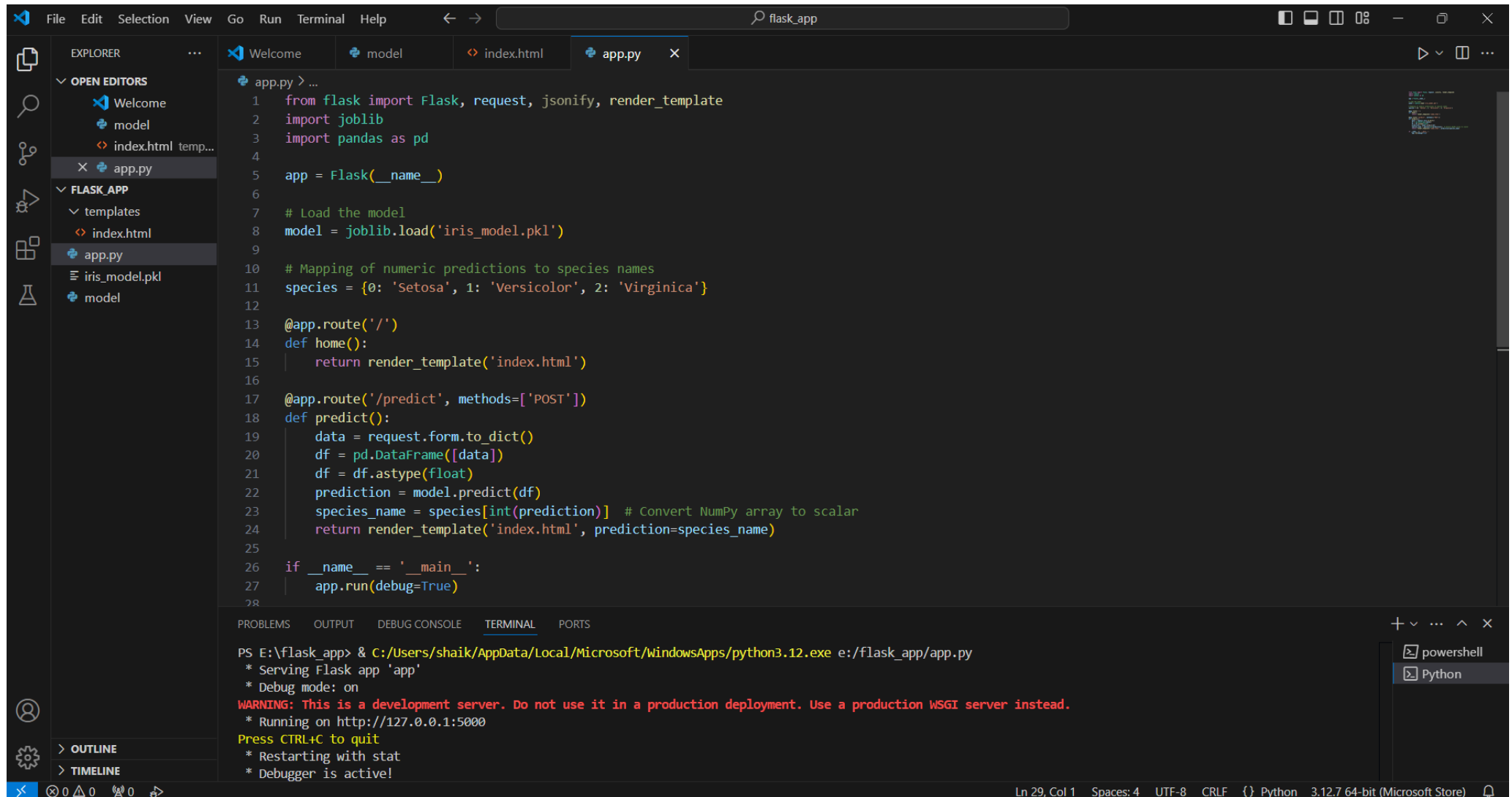


The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays the project structure for 'FLASK_APP', including 'templates' and 'index.html'. The main editor window shows the content of 'index.html' with the following code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Iris Prediction</title>
7 </head>
8 <body>
9   <h1>Iris Prediction Model</h1>
10  <form action="/predict" method="post">
11    <label for="sepal_length">Sepal Length (cm):</label>
12    <input type="text" id="sepal_length" name="sepal length (cm)" required><br><br>
13    <label for="sepal_width">Sepal Width (cm):</label>
14    <input type="text" id="sepal_width" name="sepal width (cm)" required><br><br>
15    <label for="petal_length">Petal Length (cm):</label>
16    <input type="text" id="petal_length" name="petal length (cm)" required><br><br>
17    <label for="petal_width">Petal Width (cm):</label>
18    <input type="text" id="petal_width" name="petal width (cm)" required><br><br>
19    <input type="submit" value="Predict">
20  </form>
21  {% if prediction is not none %}
22    <h2>Prediction: {{ prediction }}</h2>
23  {% endif %}
24 </body>
25 </html>
```

The status bar at the bottom indicates the current position is Line 7, Column 8, with 4 spaces, UTF-8 encoding, CRLF line endings, and the HTML file type.

3. Deploy the Model on Flask:



```
File Edit Selection View Go Run Terminal Help
flask_app

EXPLORER
OPEN EDITORS
Welcome
model
index.html temp...
app.py
FLASK_APP
templates
index.html
app.py
iris_model.pkl
model

app.py > ...
1 from flask import Flask, request, jsonify, render_template
2 import joblib
3 import pandas as pd
4
5 app = Flask(__name__)
6
7 # Load the model
8 model = joblib.load('iris_model.pkl')
9
10 # Mapping of numeric predictions to species names
11 species = {0: 'Setosa', 1: 'Versicolor', 2: 'Virginica'}
12
13 @app.route('/')
14 def home():
15     return render_template('index.html')
16
17 @app.route('/predict', methods=['POST'])
18 def predict():
19     data = request.form.to_dict()
20     df = pd.DataFrame([data])
21     df = df.astype(float)
22     prediction = model.predict(df)
23     species_name = species[int(prediction)] # Convert NumPy array to scalar
24     return render_template('index.html', prediction=species_name)
25
26 if __name__ == '__main__':
27     app.run(debug=True)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
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!
```

Ln 29, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.7 64-bit (Microsoft Store)

5. Testing the Flask Deployment

Iris Prediction

×

+

←

↺

127.0.0.1:5000

☆

⚙

|

📄

⌵

🔖

🔗

🛡

⋮

🌐

Iris Prediction Model

Sepal Length (cm):

Sepal Width (cm):

Petal Length (cm):

Petal Width (cm):

Predict

Prediction:

6. Enter the values & get the prediction result

Iris Prediction

×

+

←

↺

127.0.0.1:5000/predict

☆

⚙️

|

📄

🔖

🔒

🔗

⋮

🟢

Iris Prediction Model

Sepal Length (cm):

Sepal Width (cm):

Petal Length (cm):

Petal Width (cm):

Predict

Prediction: Versicolor