Name: **Cynthia Onyia**

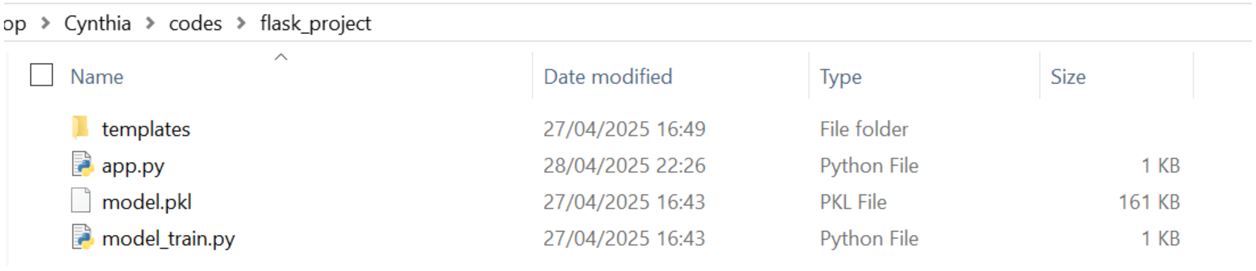
Batch Code: **LISUM44**

Submission Date: **28th April 2025**

Submitted To: **Data Glacier Team**

Details: The following contains screenshots of the processes I used to create my **first Flask app** (Flask deployment) using the Iris datasets (toy data).

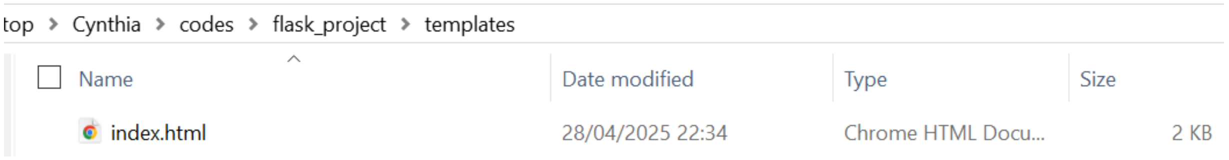
1. **Folder structure (showing model.pkl, app.py, templates/ folder)**



op > Cynthia > codes > flask_project

Name	Date modified	Type	Size
templates	27/04/2025 16:49	File folder	
app.py	28/04/2025 22:26	Python File	1 KB
model.pkl	27/04/2025 16:43	PKL File	161 KB
model_train.py	27/04/2025 16:43	Python File	1 KB

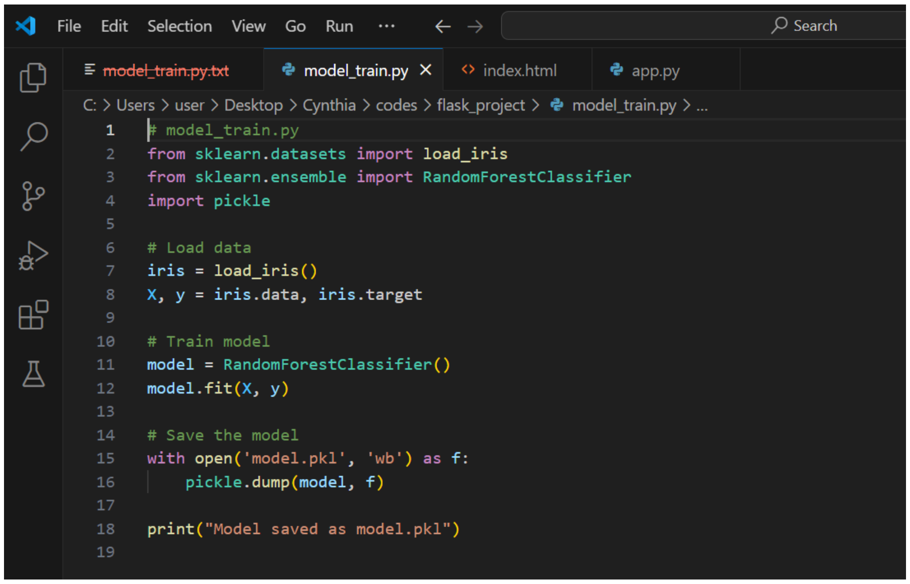
The templates folder contains my index.html file



top > Cynthia > codes > flask_project > templates

Name	Date modified	Type	Size
index.html	28/04/2025 22:34	Chrome HTML Docu...	2 KB

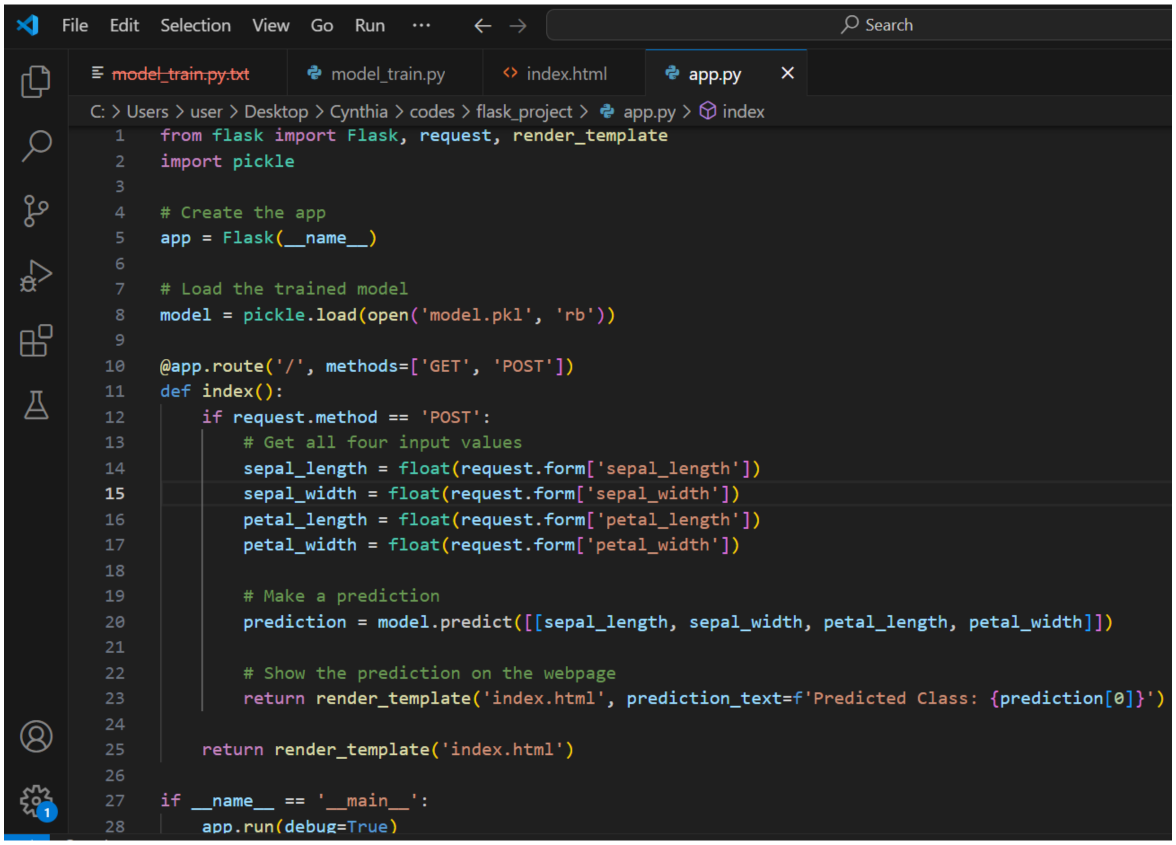
2. **Code for model_train.py**



File Edit Selection View Go Run ···
Search
~~model_train.py.txt~~
model_train.py
index.html
app.py
C: > Users > user > Desktop > Cynthia > codes > flask_project > model_train.py > ...

```python
# model_train.py
from sklearn.datasets import load_iris
from sklearn.ensemble import RandomForestClassifier
import pickle

# Load data
iris = load_iris()
X, y = iris.data, iris.target

# Train model
model = RandomForestClassifier()
model.fit(X, y)

# Save the model
with open('model.pkl', 'wb') as f:
    pickle.dump(model, f)

print("Model saved as model.pkl")
```

3. Code for app.py



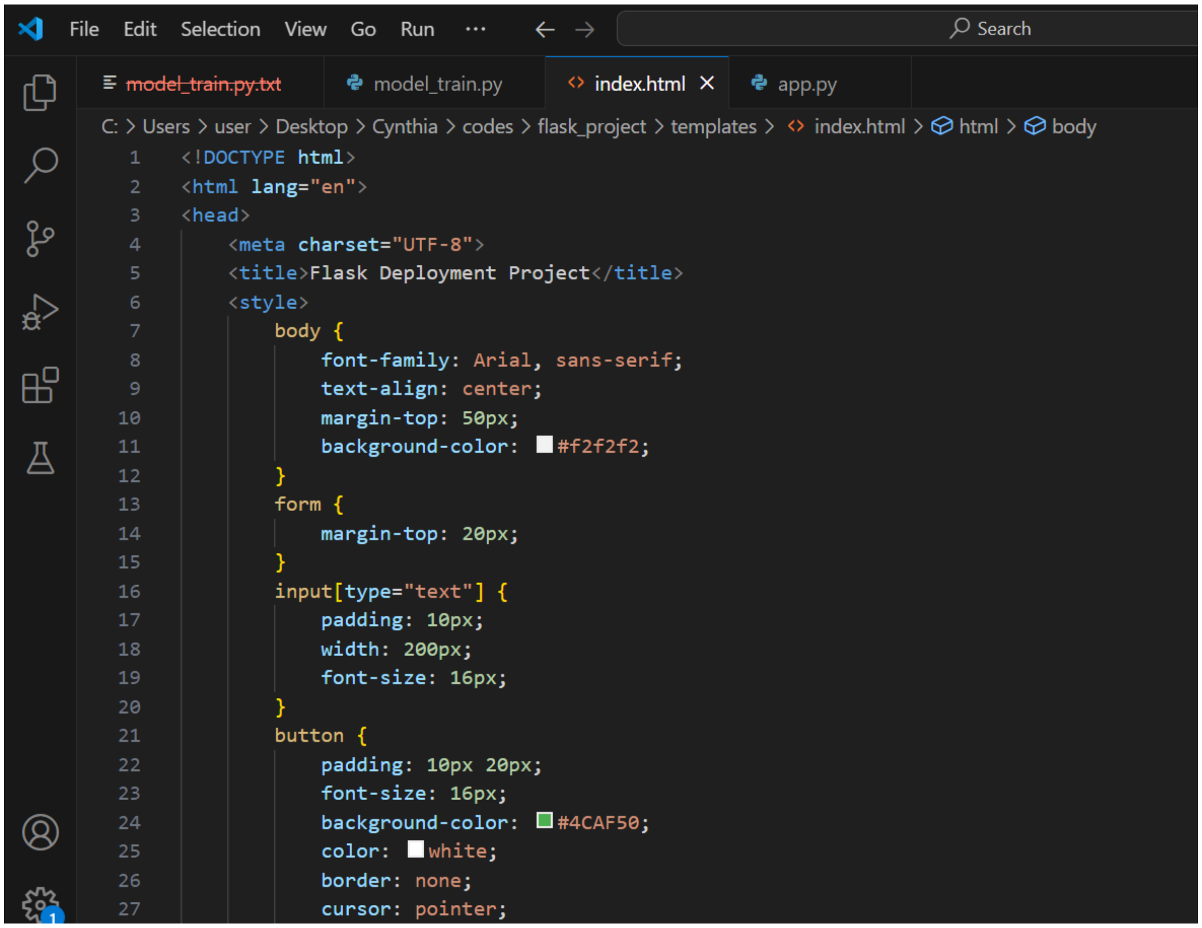
File Edit Selection View Go Run ··· Search

~~model_train.py.txt~~ model_train.py index.html app.py

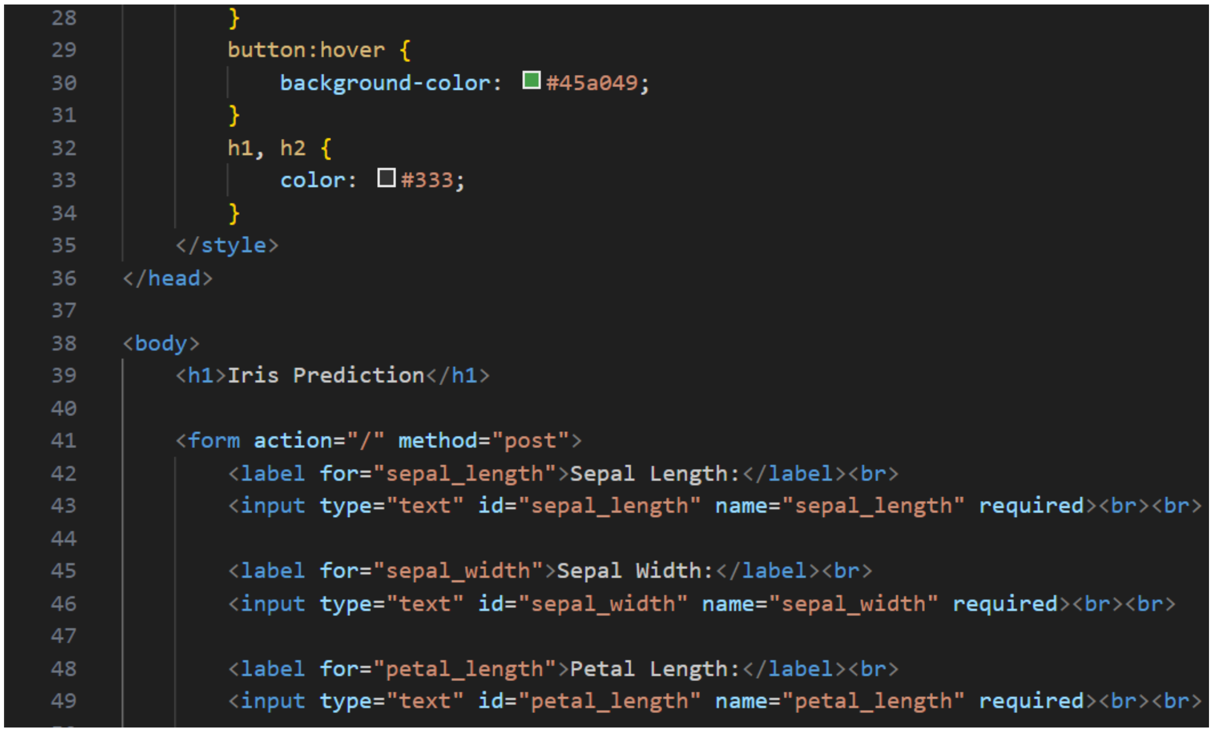
C: > Users > user > Desktop > Cynthia > codes > flask_project > app.py > index

```python
from flask import Flask, request, render_template
import pickle

# Create the app
app = Flask(__name__)

# Load the trained model
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        # Get all four input values
        sepal_length = float(request.form['sepal_length'])
        sepal_width = float(request.form['sepal_width'])
        petal_length = float(request.form['petal_length'])
        petal_width = float(request.form['petal_width'])

        # Make a prediction
        prediction = model.predict([[sepal_length, sepal_width, petal_length, petal_width]])

        # Show the prediction on the webpage
        return render_template('index.html', prediction_text=f'Predicted Class: {prediction[0]}')

    return render_template('index.html')

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

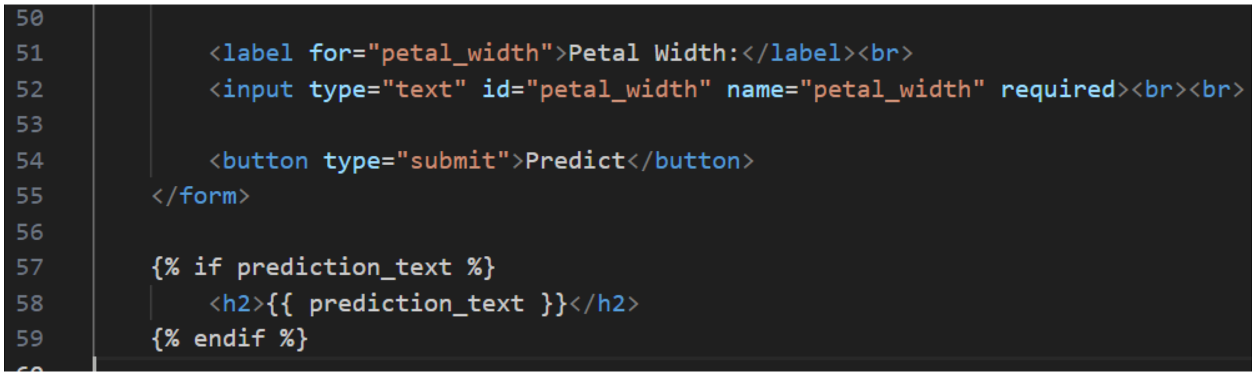
4. Code for index.html


File
Edit
Selection
View
Go
Run
Search
~~model_train.py.txt~~
model_train.py
index.html
app.py
C: > Users > user > Desktop > Cynthia > codes > flask_project > templates > index.html > html > body

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Flask Deployment Project</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin-top: 50px;
            background-color: #f2f2f2;
        }
        form {
            margin-top: 20px;
        }
        input[type="text"] {
            padding: 10px;
            width: 200px;
            font-size: 16px;
        }
        button {
            padding: 10px 20px;
            font-size: 16px;
            background-color: #4CAF50;
            color: white;
            border: none;
            cursor: pointer;
```

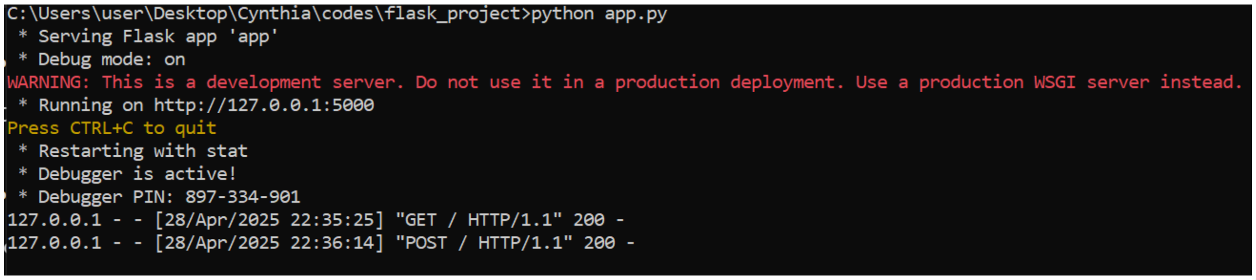


```
        }
        button:hover {
            background-color: #45a049;
        }
        h1, h2 {
            color: #333;
        }
    </style>
</head>

<body>
    <h1>Iris Prediction</h1>

    <form action="/" method="post">
        <label for="sepal_length">Sepal Length:</label><br>
        <input type="text" id="sepal_length" name="sepal_length" required><br><br>

        <label for="sepal_width">Sepal Width:</label><br>
        <input type="text" id="sepal_width" name="sepal_width" required><br><br>

        <label for="petal_length">Petal Length:</label><br>
        <input type="text" id="petal_length" name="petal_length" required><br><br>
```



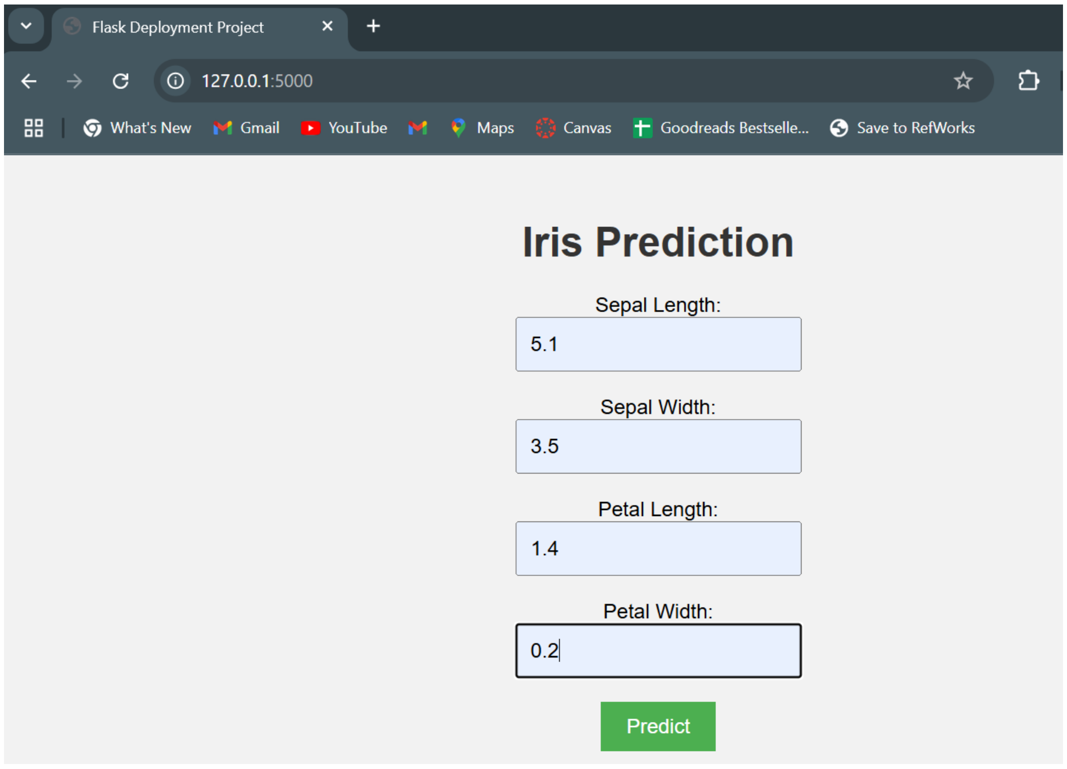
```
        <label for="petal_width">Petal Width:</label><br>
        <input type="text" id="petal_width" name="petal_width" required><br><br>

        <button type="submit">Predict</button>
    </form>

    {% if prediction_text %}
        <h2>{{ prediction_text }}</h2>
    {% endif %}
```

5. Terminal running Flask (python app.py)



```
C:\Users\user\Desktop\Cynthia\codes\flask_project>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 stat
 * Debugger is active!
 * Debugger PIN: 897-334-901
127.0.0.1 - - [28/Apr/2025 22:35:25] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Apr/2025 22:36:14] "POST / HTTP/1.1" 200 -
```

6. Browser showing your form (before prediction)



Iris Prediction

Sepal Length: 5.1

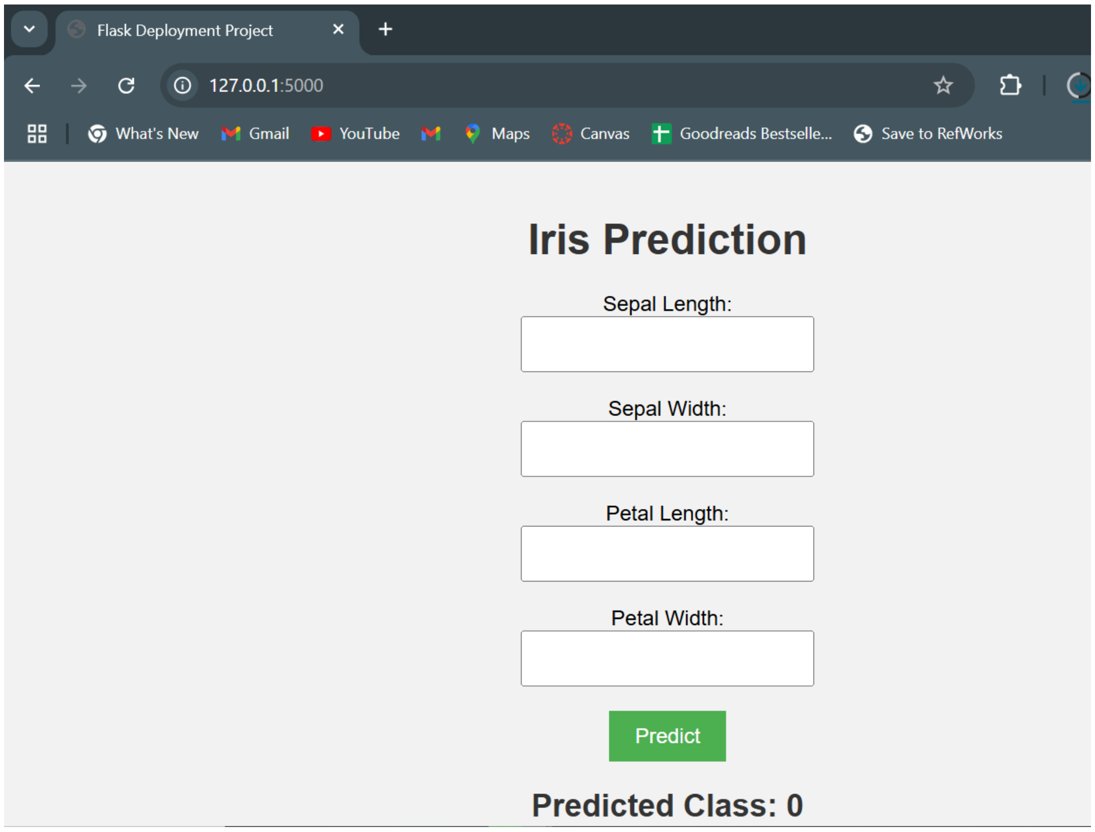
Sepal Width: 3.5

Petal Length: 1.4

Petal Width: 0.2

Predict

7. Browser showing the prediction result



Iris Prediction

Sepal Length:

Sepal Width:

Petal Length:

Petal Width:

Predict

Predicted Class: 0