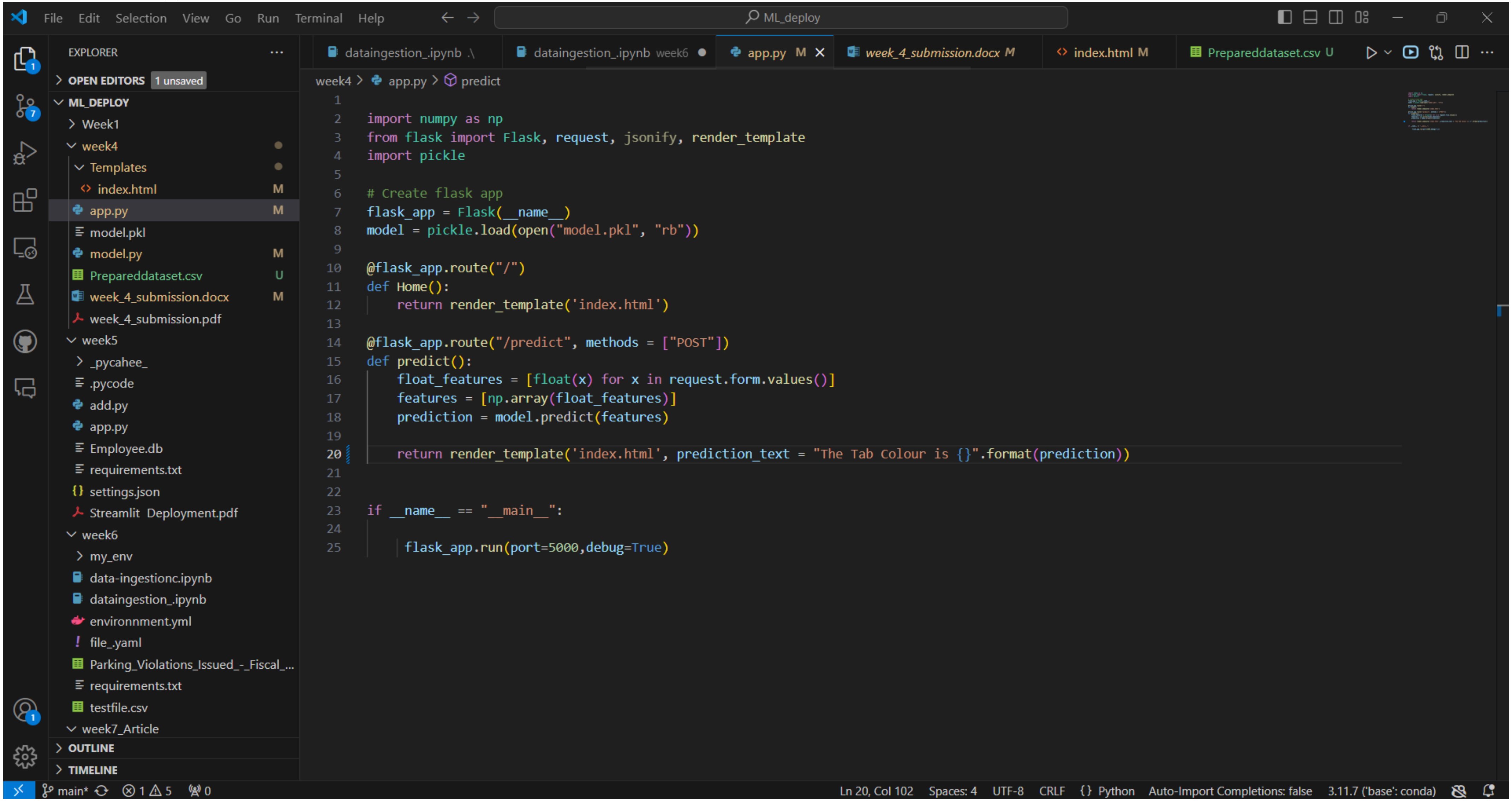
Week 4 Assignment - Week4: Deployment on Flask

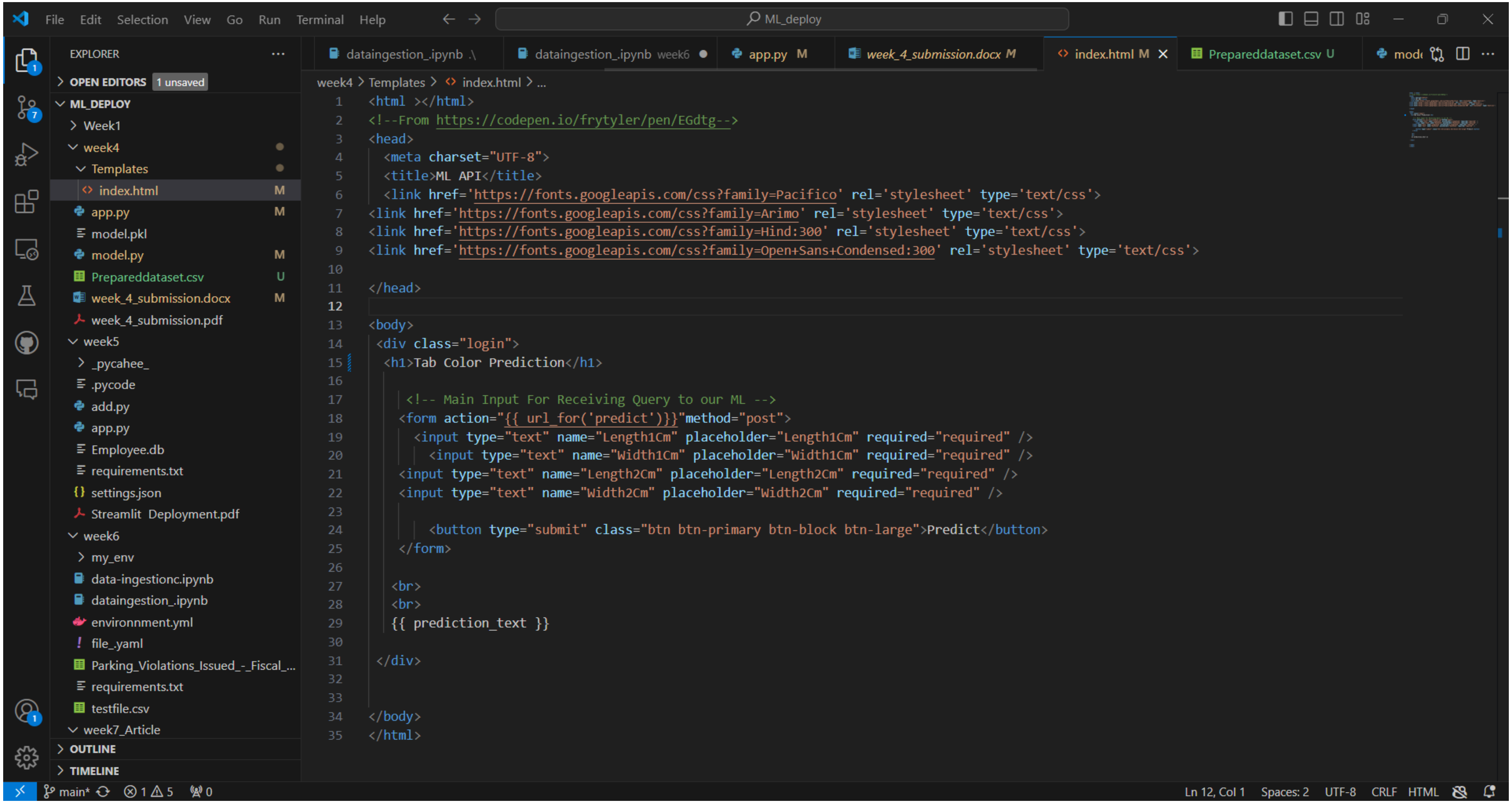
Name: Vijayarajan Vijaya Jothi
Batch code: LISUM37
Submission date: Sep 23, 2024
Submitted to: Data Glacier

1. Code of Flask app



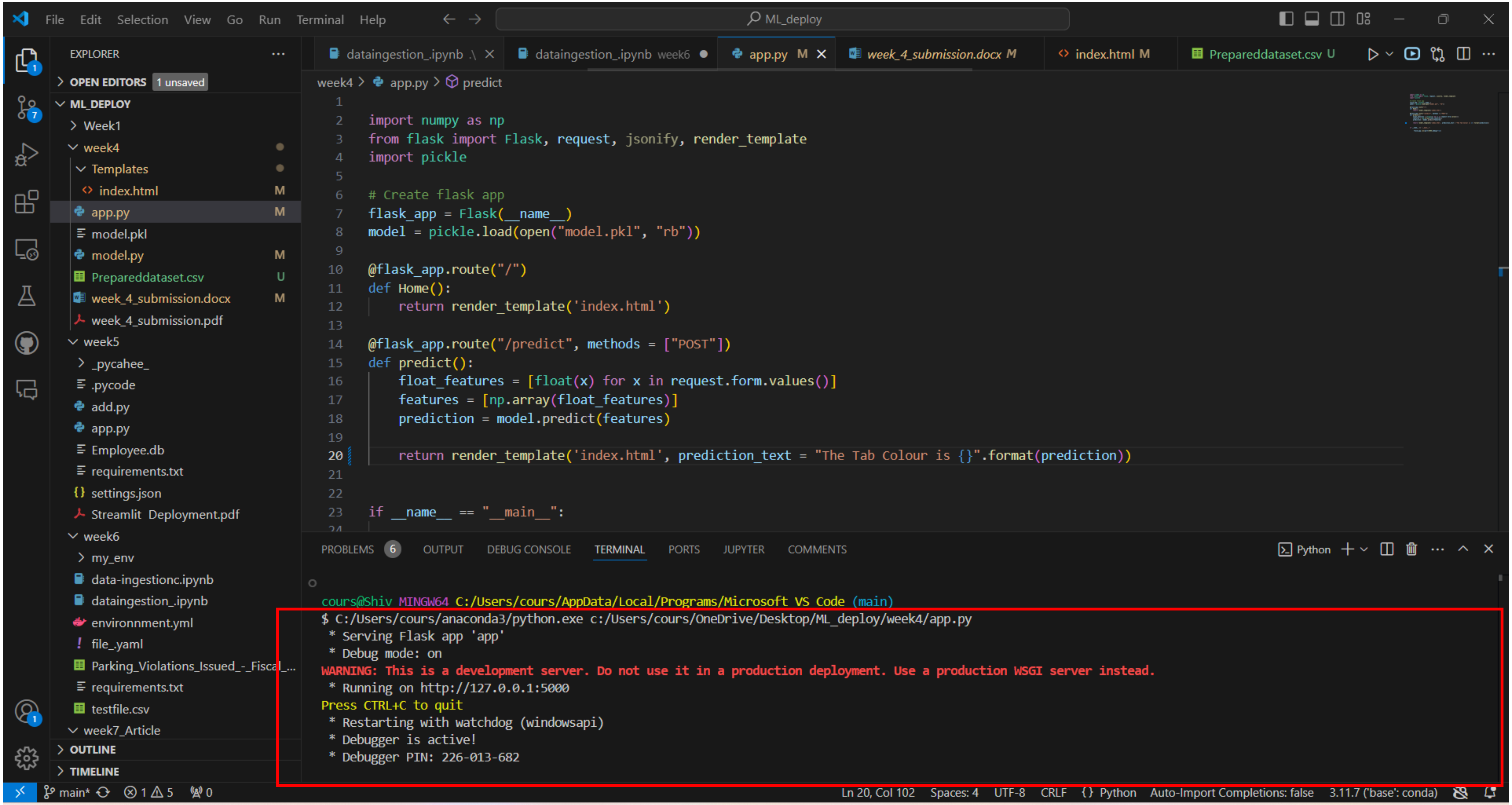
```python

import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

# Create flask app
flask_app = Flask(__name__)
model = pickle.load(open("model.pkl", "rb"))

@flask_app.route("/")
def Home():
    return render_template('index.html')

@flask_app.route("/predict", methods = ["POST"])
def predict():
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    prediction = model.predict(features)

    return render_template('index.html', prediction_text = "The Tab Colour is {}".format(prediction))


if __name__ == "__main__":

     flask_app.run(port=5000,debug=True)
```

2. index.html



```html
<html ></html>
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

</head>

<body>
 <div class="login">
  <h1>Tab Color Prediction</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
      <input type="text" name="Length1Cm" placeholder="Length1Cm" required="required" />
        <input type="text" name="Width1Cm" placeholder="Width1Cm" required="required" />
    <input type="text" name="Length2Cm" placeholder="Length2Cm" required="required" />
    <input type="text" name="Width2Cm" placeholder="Width2Cm" required="required" />

        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

   <br>
   <br>
   {{ prediction_text }}

 </div>


</body>
</html>
```

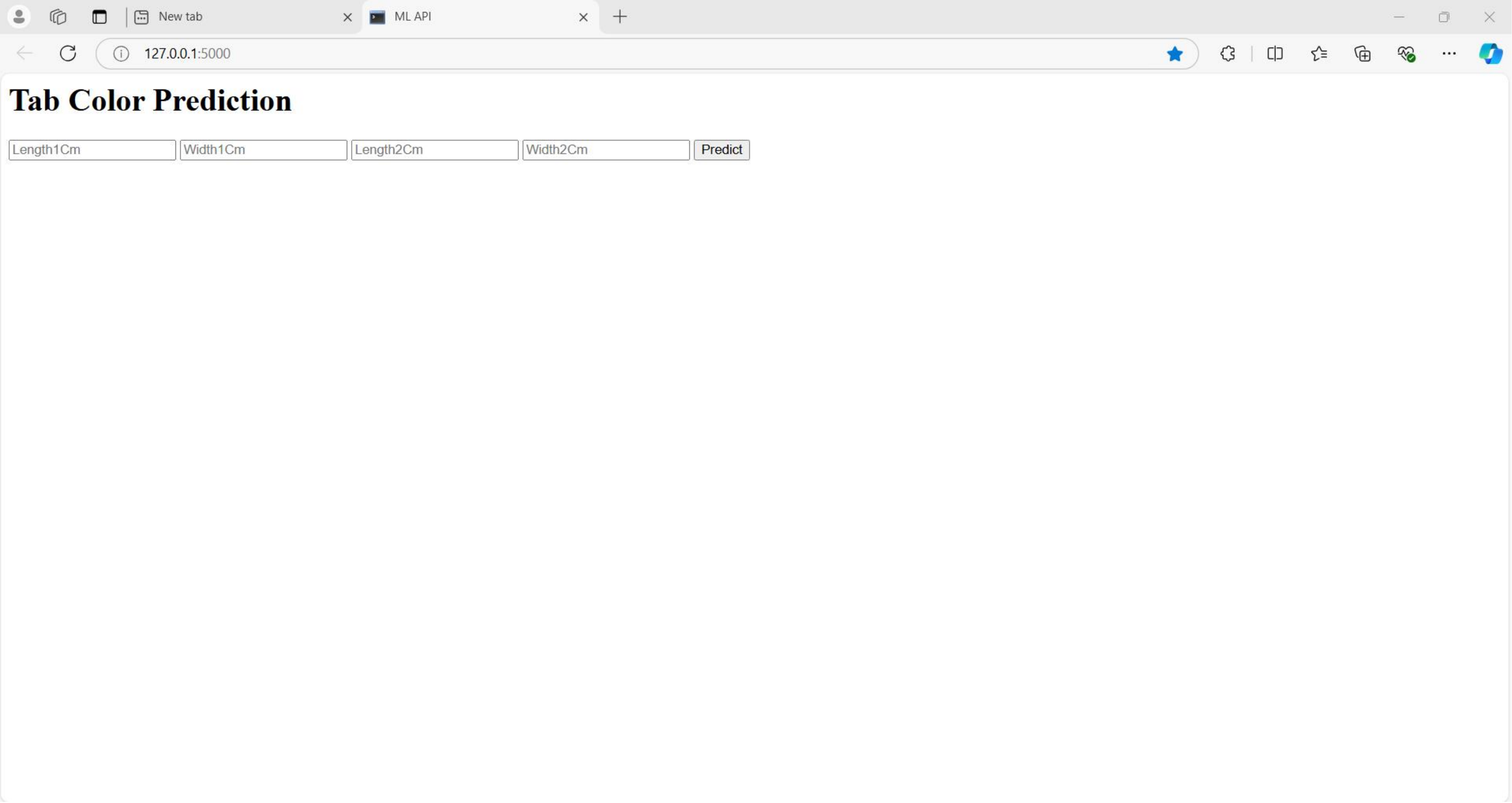
3. Running Flask in debug mode in VisualStudioCode



```python

import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

# Create flask app
flask_app = Flask(__name__)
model = pickle.load(open("model.pkl", "rb"))

@flask_app.route("/")
def Home():
    return render_template('index.html')

@flask_app.route("/predict", methods = ["POST"])
def predict():
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    prediction = model.predict(features)

    return render_template('index.html', prediction_text = "The Tab Colour is {}".format(prediction))


if __name__ == "__main__":
```

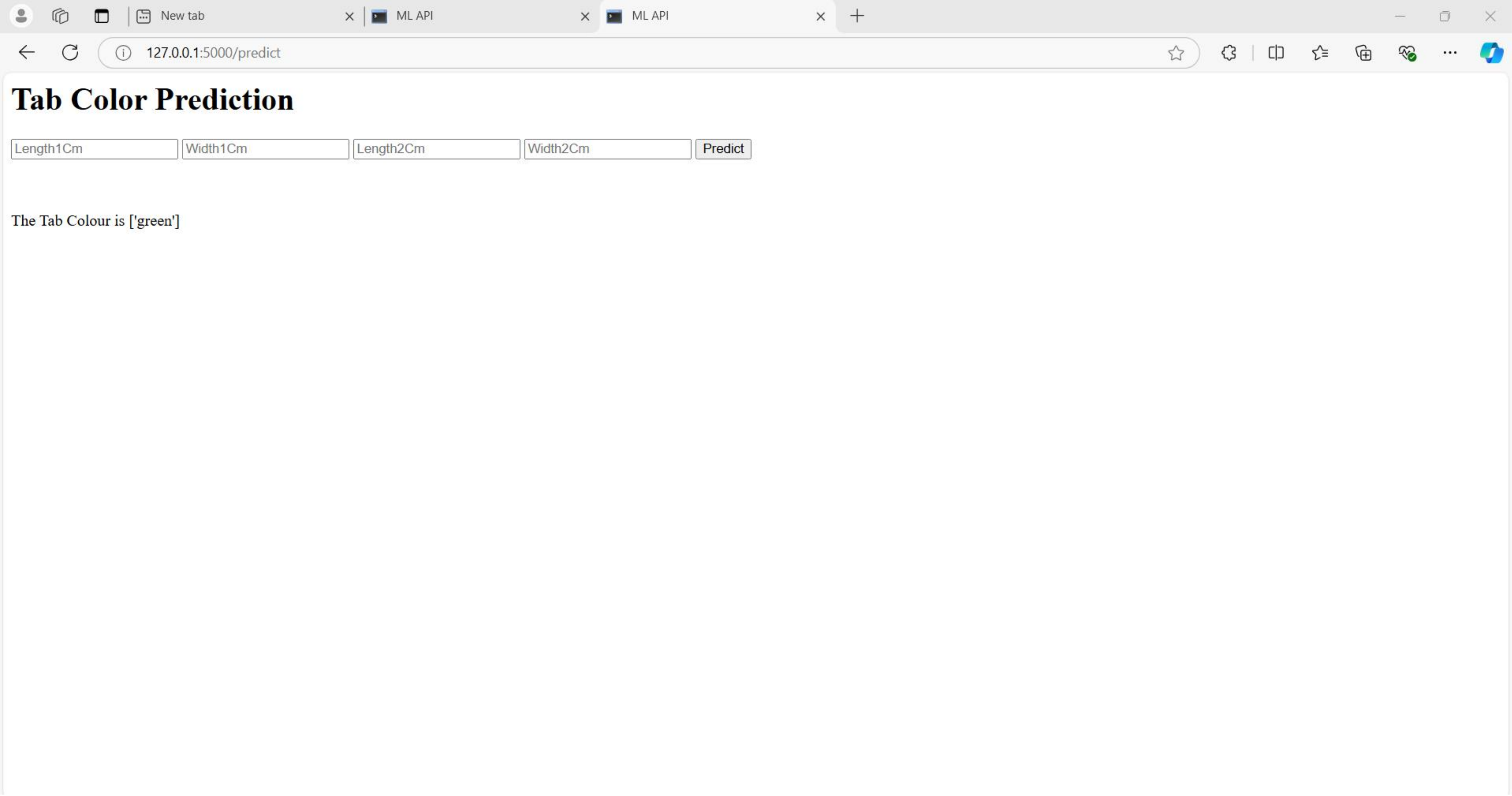
```
cours@Shiv MINGW64 C:/Users/cours/AppData/Local/Programs/Microsoft VS Code (main)
$ C:/Users/cours/anaconda3/python.exe c:/Users/cours/OneDrive/Desktop/ML_deploy/week4/app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. 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: 226-013-682
```

4. Running app homepage



Tab Color Prediction

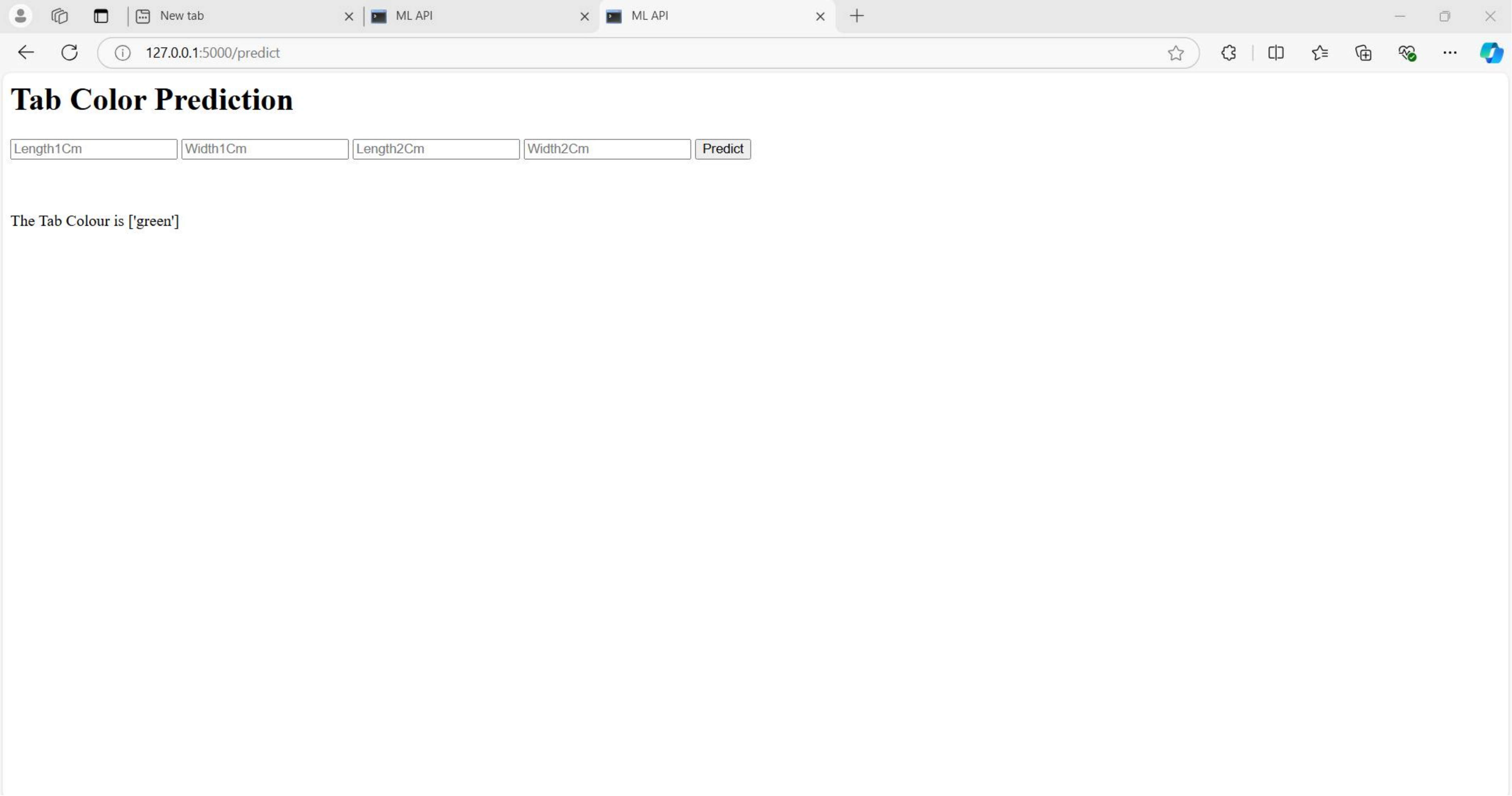
5. Filling test data



Tab Color Prediction

The Tab Colour is ['green']

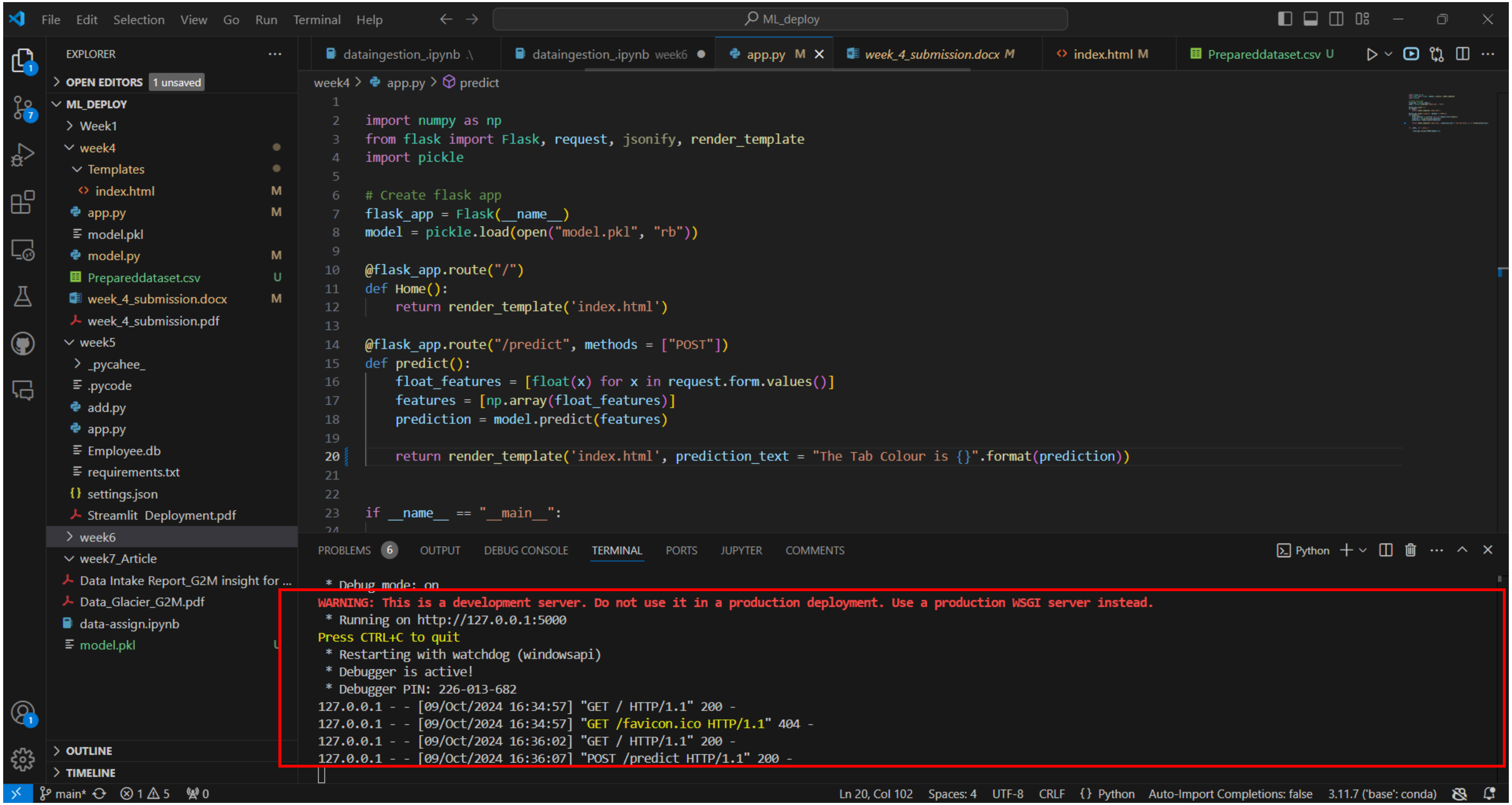
6. Prediction result shown on web page



Tab Color Prediction

The Tab Colour is ['green']

7. Flask output in terminal window



```python

import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

# Create flask app
flask_app = Flask(__name__)
model = pickle.load(open("model.pkl", "rb"))

@flask_app.route("/")
def Home():
    return render_template('index.html')

@flask_app.route("/predict", methods = ["POST"])
def predict():
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    prediction = model.predict(features)

    return render_template('index.html', prediction_text = "The Tab Colour is {}".format(prediction))


if __name__ == "__main__":
```

```
 * 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: 226-013-682
127.0.0.1 - - [09/Oct/2024 16:34:57] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Oct/2024 16:34:57] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [09/Oct/2024 16:36:02] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Oct/2024 16:36:07] "POST /predict HTTP/1.1" 200 -
```