

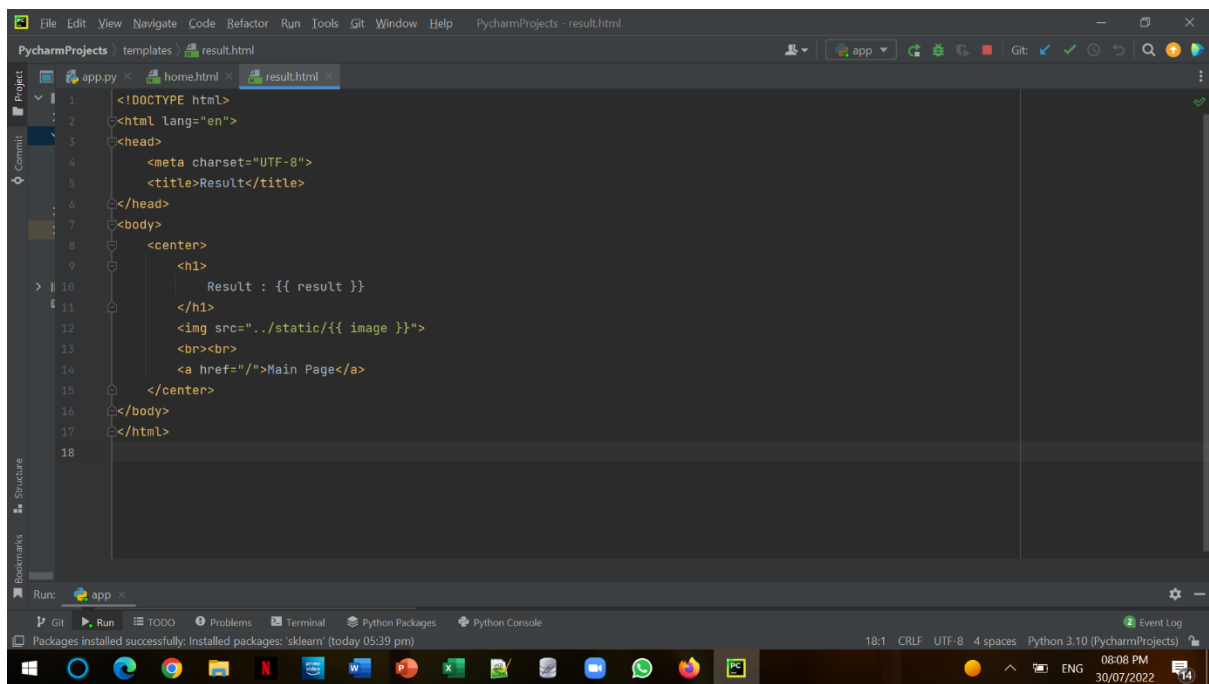
Name: Isha Panjwani

Batch Code: LISUM11: 30

Submission Date: 30th July 2022

Submitted to: Data Glacier

(i) Html file: result.html



The screenshot shows the PyCharm IDE interface with the file 'result.html' open. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Result</title>
6   </head>
7   <body>
8     <center>
9       <h1>
10        Result : {{ result }}
11      </h1>
12      
13      <br><br>
14      <a href="/">Main Page</a>
15    </center>
16  </body>
17 </html>
18
```

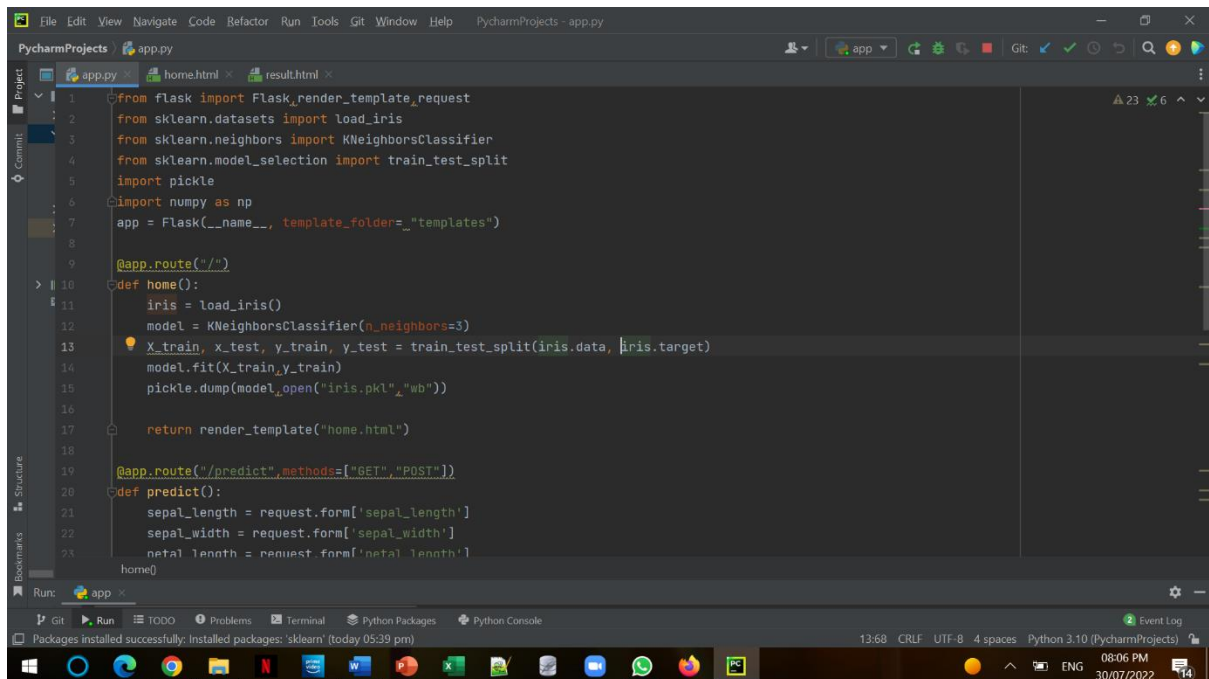
The bottom status bar indicates the file encoding is UTF-8, line endings are CRLF, and 4 spaces are used for indentation. The Python version is 3.10. The system clock shows 08:08 PM on 30/07/2022.

(ii) Html file: home.html

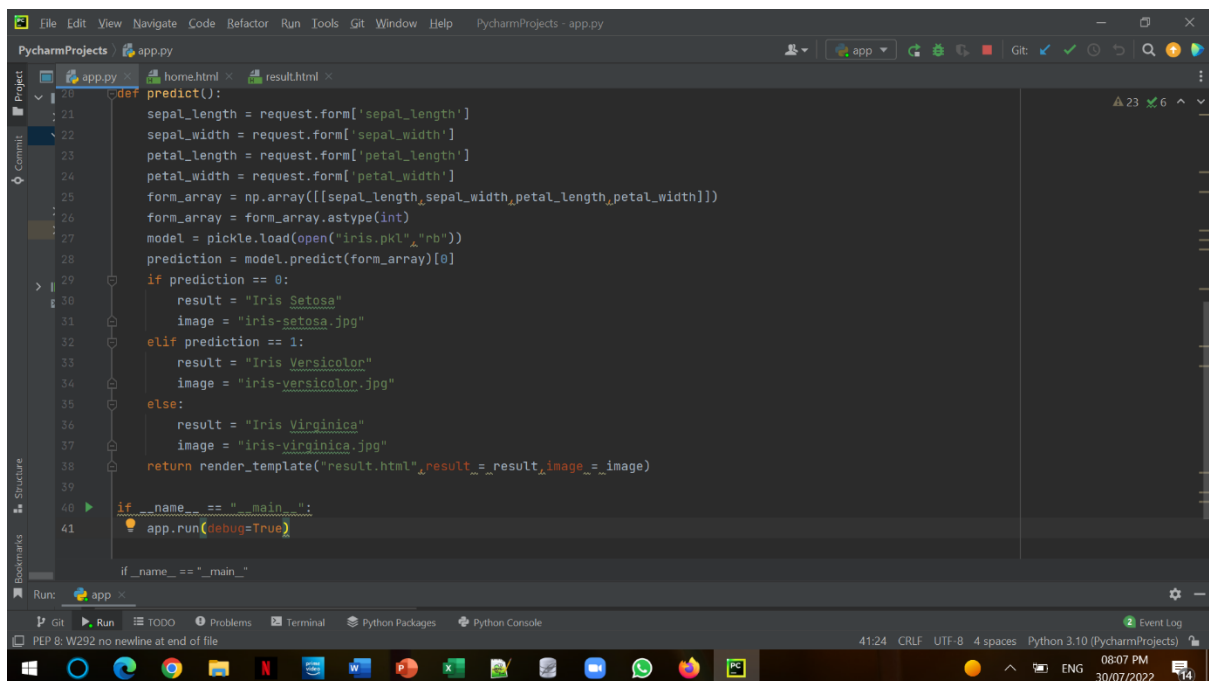
The image displays two screenshots of the PyCharm IDE, showing the development of a web form for Iris Flower Detection. The top screenshot shows the initial HTML structure, including the head, body, and a form group. The bottom screenshot shows the completed form with input fields for sepal length, sepal width, petal length, and petal width, along with a submit button and external script links.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7
8   <!-- Bootstrap CSS -->
9   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-6n5384xqQ1aoWXA+058RXPx"
10
11   <title>Iris Dataset</title>
12 </head>
13 <body style="background-color: #b4aee8">
14   <center>
15
16     <h1>
17       Iris Flower Detection
18     </h1>
19     <div style="border: thick black solid ; width: fit-content; background-color: aliceblue; ">
20       <form action = "{{ url_for('predict') }}" method="POST" class="text-center border border-light p-5" >
21         <div class="form-group" style="padding:10px;">
22           <input style="padding:5px; height:45px" name = "sepal_length" type="number" class="form-control mb-4" placeholder="Enter sepal length">
23           <input style="padding:5px; height:45px" name = "sepal_width" type="number" class="form-control mb-4" placeholder="Enter sepal width">
24           <input style="padding:5px; height:45px" name = "petal_length" type="number" class="form-control mb-4" placeholder="Enter petal length">
25           <input style="padding:5px; height:45px" name = "petal_width" type="number" class="form-control mb-4" placeholder="Enter petal width">
26           <button type="submit" class="btn btn-primary btn-block mb-3">Submit</button>
27         </div>
28       </form>
29     </div>
30   </center>
31
32   <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KcKrr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
33   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn3W3mgPxhU9K/ScQsA"
34   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZr6Spej4002d8j0t6vLEHfe/JQGiRRSQQxSfFWp"
35 </body>
36 </html>
```

(iii) Python file: app.py

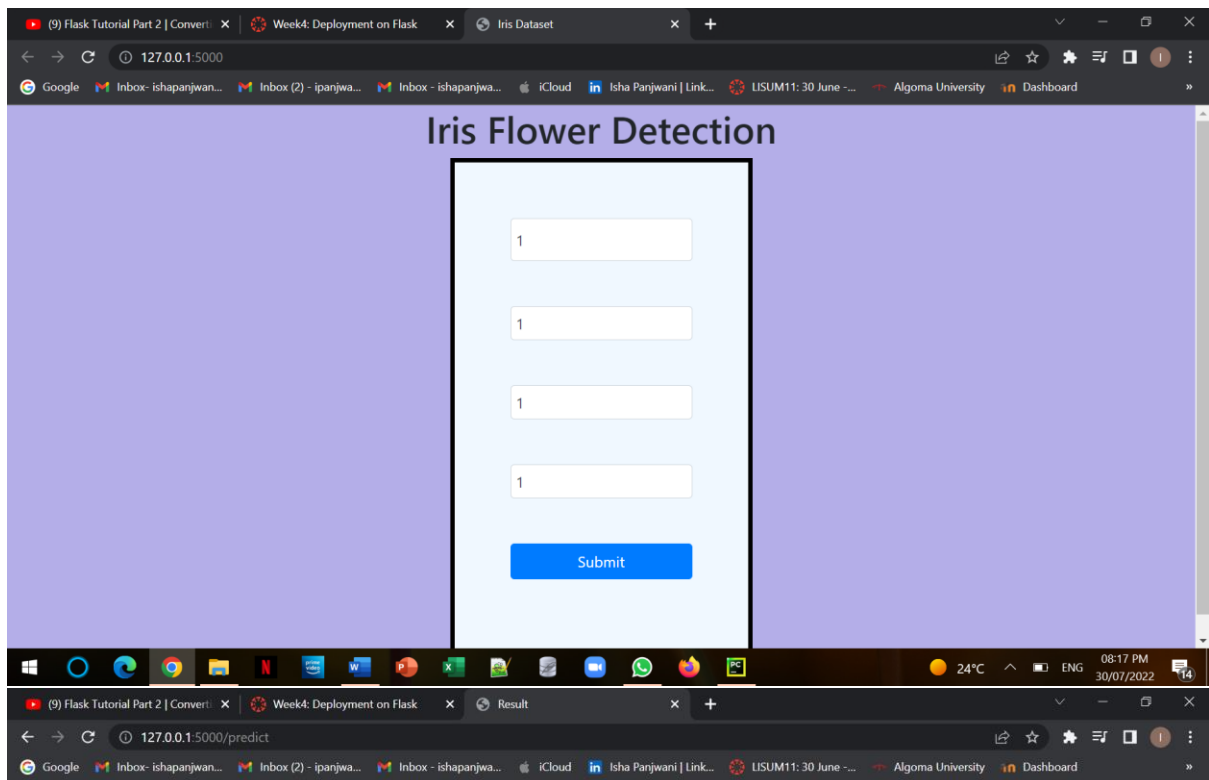


```
1 from flask import Flask, render_template, request
2 from sklearn.datasets import load_iris
3 from sklearn.neighbors import KNeighborsClassifier
4 from sklearn.model_selection import train_test_split
5 import pickle
6
7 import numpy as np
8 app = Flask(__name__, template_folder='templates')
9
10 @app.route("/")
11 def home():
12     iris = load_iris()
13     model = KNeighborsClassifier(n_neighbors=3)
14     X_train, x_test, y_train, y_test = train_test_split(iris.data, iris.target)
15     model.fit(X_train, y_train)
16     pickle.dump(model, open("iris.pkl", "wb"))
17
18     return render_template("home.html")
19
20 @app.route("/predict", methods=["GET", "POST"])
21 def predict():
22     sepal_length = request.form['sepal_length']
23     sepal_width = request.form['sepal_width']
24     petal_length = request.form['petal_length']
25     petal_width = request.form['petal_width']
26
27     form_array = np.array([[sepal_length, sepal_width, petal_length, petal_width]])
28     form_array = form_array.astype(int)
29     model = pickle.load(open("iris.pkl", "rb"))
30     prediction = model.predict(form_array)[0]
31
32     if prediction == 0:
33         result = "Iris Setosa"
34         image = "iris-setosa.jpg"
35     elif prediction == 1:
36         result = "Iris Versicolor"
37         image = "iris-versicolor.jpg"
38     else:
39         result = "Iris Virginica"
40         image = "iris-virginica.jpg"
41
42     return render_template("result.html", result=result, image=image)
43
44 if __name__ == "__main__":
45     app.run(debug=True)
```



```
21 sepal_length = request.form['sepal_length']
22 sepal_width = request.form['sepal_width']
23 petal_length = request.form['petal_length']
24 petal_width = request.form['petal_width']
25
26 form_array = np.array([[sepal_length, sepal_width, petal_length, petal_width]])
27 form_array = form_array.astype(int)
28 model = pickle.load(open("iris.pkl", "rb"))
29 prediction = model.predict(form_array)[0]
30
31 if prediction == 0:
32     result = "Iris Setosa"
33     image = "iris-setosa.jpg"
34 elif prediction == 1:
35     result = "Iris Versicolor"
36     image = "iris-versicolor.jpg"
37 else:
38     result = "Iris Virginica"
39     image = "iris-virginica.jpg"
40
41 return render_template("result.html", result=result, image=image)
42
43 if __name__ == "__main__":
44     app.run(debug=True)
```

(iv) Web page after deployment



Result : Iris Setosa



[Main Page](#)

