

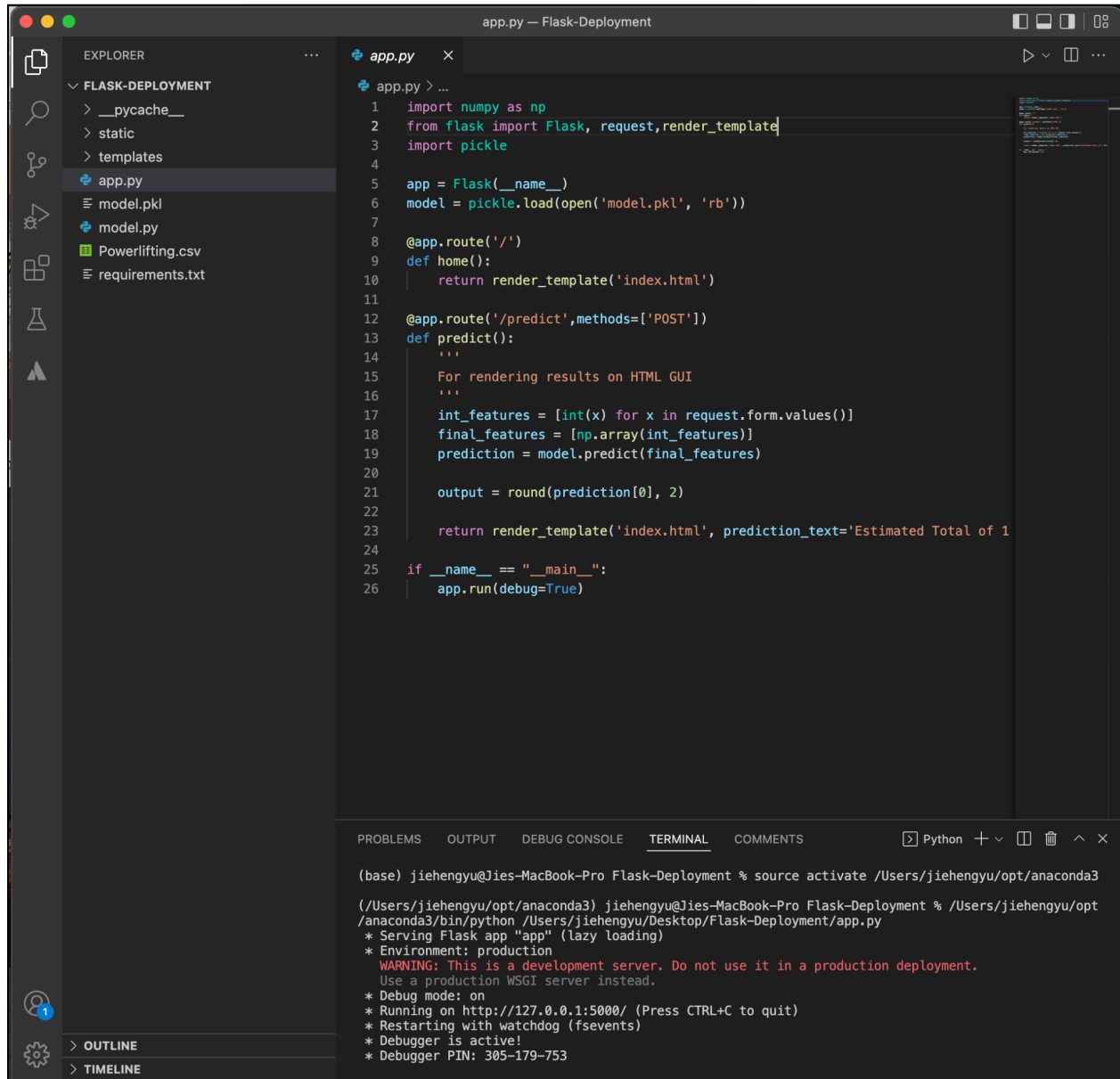
Name: Jie Heng Yu

Internship Batch: LISUM26

Submission Date: October 28, 2023

Submitted to: DataGlacier

Step 1: Run app in VSCode



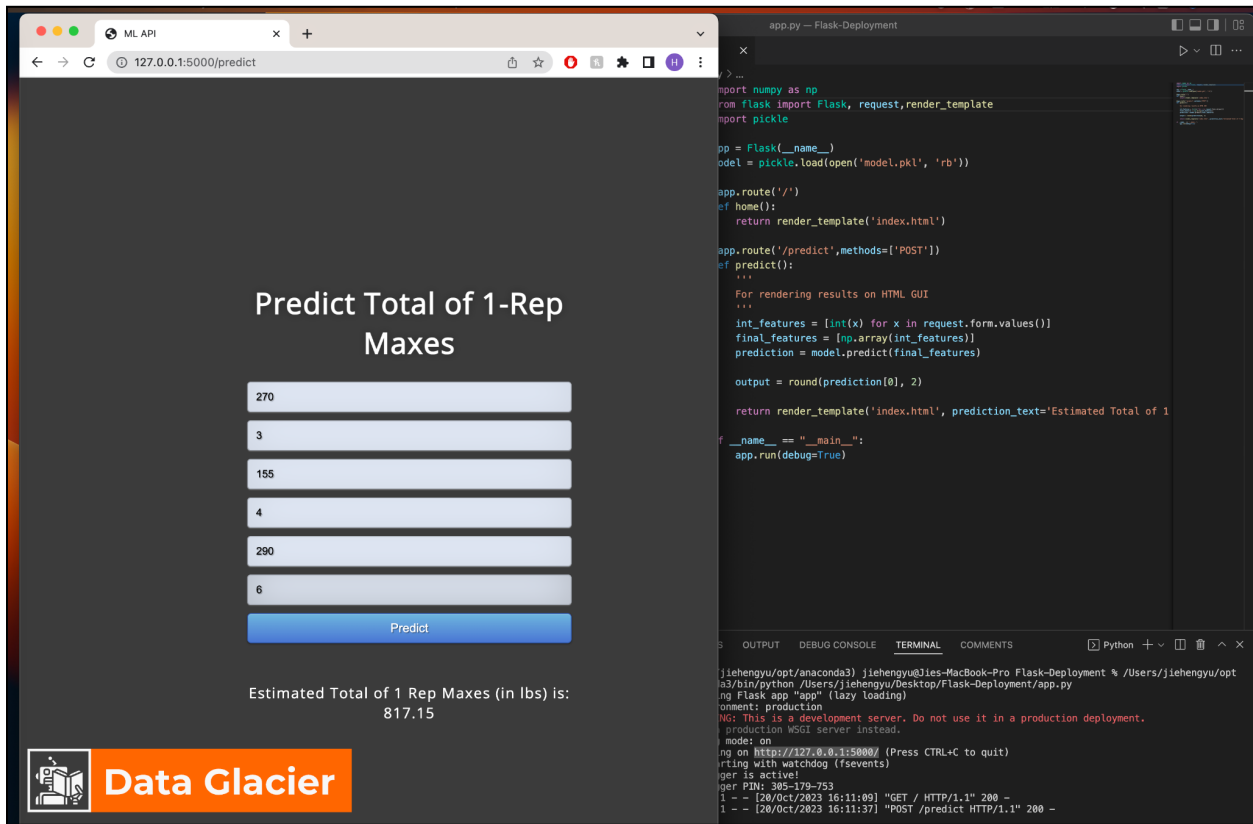
The screenshot displays the Visual Studio Code interface with a project named 'Flask-Deployment'. The Explorer sidebar on the left shows the file structure: `__pycache__`, `static`, `templates`, `app.py`, `model.pkl`, `model.py`, `Powerlifting.csv`, and `requirements.txt`. The `app.py` file is open in the editor, showing the following code:

```
1 import numpy as np
2 from flask import Flask, request, render_template
3 import pickle
4
5 app = Flask(__name__)
6 model = pickle.load(open('model.pkl', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14     """
15     For rendering results on HTML GUI
16     """
17     int_features = [int(x) for x in request.form.values()]
18     final_features = [np.array(int_features)]
19     prediction = model.predict(final_features)
20
21     output = round(prediction[0], 2)
22
23     return render_template('index.html', prediction_text='Estimated Total of 1
24
25 if __name__ == "__main__":
26     app.run(debug=True)
```

The TERMINAL panel at the bottom shows the command prompt output for running the application:

```
(base) jiehengyu@Jies-MacBook-Pro Flask-Deployment % source activate /Users/jiehengyu/opt/anaconda3
(/Users/jiehengyu/opt/anaconda3) jiehengyu@Jies-MacBook-Pro Flask-Deployment % /Users/jiehengyu/opt/anaconda3/bin/python /Users/jiehengyu/Desktop/Flask-Deployment/app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with watchdog (fsevents)
* Debugger is active!
* Debugger PIN: 305-179-753
```

Step 2: Open it in browser




The image shows a web browser window on the left and a code editor window on the right. The browser window displays a web application titled "Predict Total of 1-Rep Maxes". It features a form with six input fields containing the values 270, 3, 155, 4, 290, and 6. Below the form is a blue "Predict" button. The output of the prediction is displayed as "Estimated Total of 1 Rep Maxes (in lbs) is: 817.15". The code editor window shows the Python code for the Flask application, which includes imports for numpy, Flask, request, render_template, and pickle. The code defines a Flask app, loads a model from a pickle file, and implements routes for a home page and a prediction endpoint. The prediction endpoint uses the input values from the form to calculate the predicted total of 1-Rep Maxes. The terminal window at the bottom of the code editor shows the command to run the application and the output, including the URL to access the application in a browser.

Predict Total of 1-Rep Maxes

270
3
155
4
290
6

Predict

Estimated Total of 1 Rep Maxes (in lbs) is:
817.15

 **Data Glacier**

```
app.py - Flask-Deployment
> ...
import numpy as np
from flask import Flask, request, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

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

app.route('/predict', methods=['POST'])
def predict():
    ...
    For rendering results on HTML GUI
    ...
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Estimated Total of 1

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

OUTPUT DEBUG CONSOLE TERMINAL COMMENTS Python + - x

jiehengyu@jies-MacBook-Pro Flask-Deployment % ./Users/jiehengyu/opt/anaconda3/bin/python /Users/jiehengyu/Desktop/Flask-Deployment/app.py
Running Flask app "app" (lazy loading)
Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
Debug mode: on
Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
Pressing with watchdog (fsevents)
Server is active!
Server PIN: 305-179-753
1 - - [28/Oct/2023 16:11:09] "GET / HTTP/1.1" 200 -
1 - - [28/Oct/2023 16:11:37] "POST /predict HTTP/1.1" 200 -

Step 3: Get the prediction.

The image shows a web browser window on the left and a code editor on the right. The browser window displays a web application titled "Predict Total of 1-Rep Maxes". It features a form with six input fields containing the values 270, 3, 155, 4, 290, and 6. Below the form is a blue "Predict" button. Underneath the button, the text "Estimated Total of 1 Rep Maxes (in lbs) is: 817.15" is displayed. At the bottom left of the browser window is the "Data Glacier" logo. The code editor on the right shows the Python code for the Flask application, titled "app.py - Flask-Deployment". The code imports necessary libraries, loads a model, and defines routes for the web application. The terminal at the bottom of the code editor shows the command to run the application and the resulting output, including the URL "http://127.0.0.1:5000/" and the POST request to the "/predict" endpoint.

Predict Total of 1-Rep Maxes

270
3
155
4
290
6

Predict

Estimated Total of 1 Rep Maxes (in lbs) is:
817.15

Data Glacier

```
app.py - Flask-Deployment
> ...
import numpy as np
from flask import Flask, request, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

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

app.route('/predict', methods=['POST'])
def predict():
    """
    For rendering results on HTML GUI
    """
    int_features = [int(x) for x in request.form.values()]
    final_features = np.array(int_features)
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Estimated Total of 1

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

OUTPUT DEBUG CONSOLE TERMINAL COMMENTS Python + - x

jiehengyu/opt/anaconda3) jiehengyu@Jies-MacBook-Pro Flask-Deployment % ./Users/jiehengyu/opt
33/bin/python /Users/jiehengyu/Desktop/Flask-Deployment/app.py
ng Flask app "app" (lazy loading)
onment: production
NG: This is a development server. Do not use it in a production deployment.
production WSGI server instead.
mode: on
ng on http://127.0.0.1:5000/ (Press CTRL+C to quit)
ring with watchdog (fsevents)
ger is active!
ger PIN: 305-179-753
1 - - [20/Oct/2023 16:11:00] "GET / HTTP/1.1" 200 -
1 - - [20/Oct/2023 16:11:37] "POST /predict HTTP/1.1" 200 -