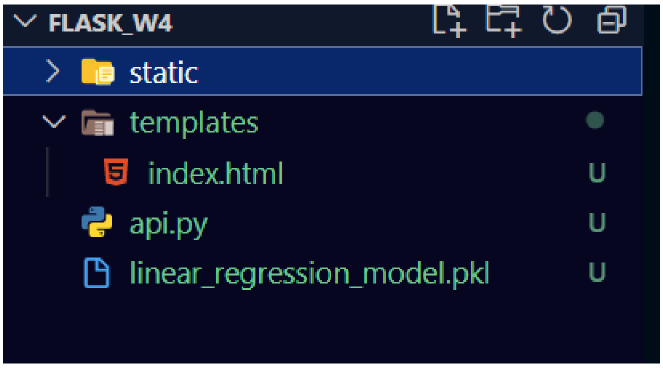
Name: Rahul Saini

Batch Code: LISUM18

Submission Date: 28/02/2023

Step1:

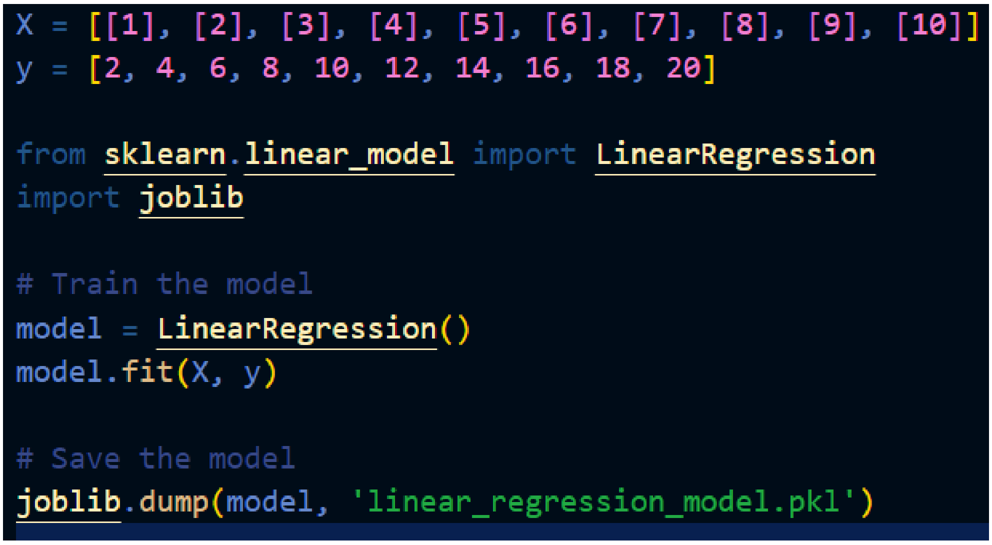
- Created a directory for the Flask application which includes the subfolder templates and static to store other files.



```
FLASK_W4
  > static
  v templates
      index.html                U
    api.py                      U
    linear_regression_model.pkl U
```

Step2:

- Create a python file api.py to store run the ml model and to run the flask application.

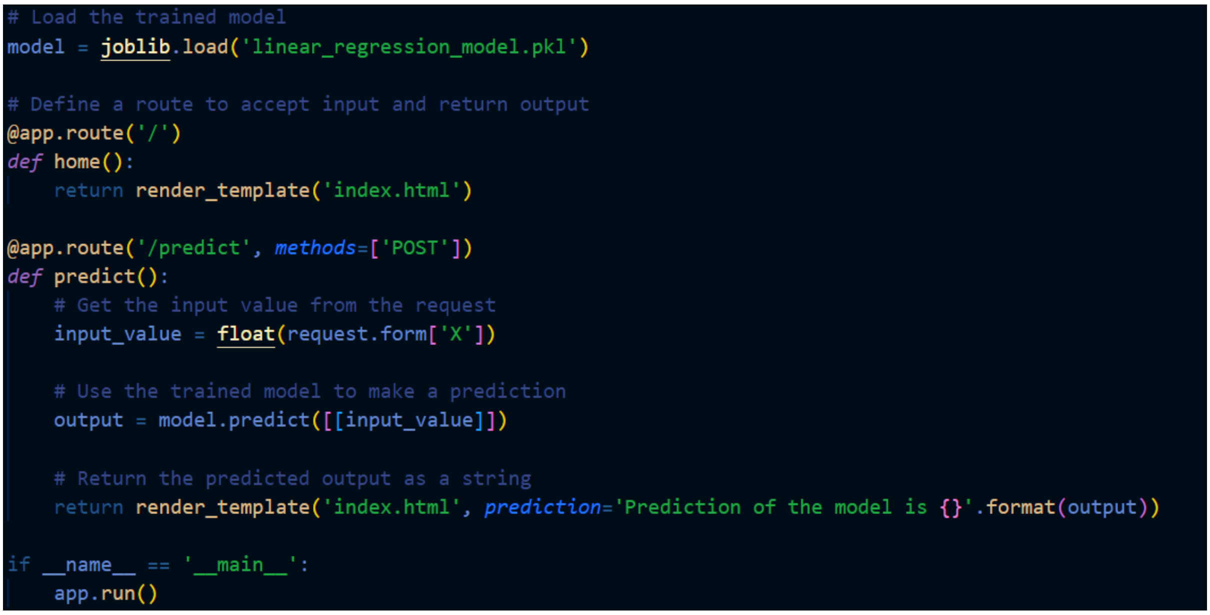


```python
X = [[1], [2], [3], [4], [5], [6], [7], [8], [9], [10]]
y = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

from sklearn.linear_model import LinearRegression
import joblib

# Train the model
model = LinearRegression()
model.fit(X, y)

# Save the model
joblib.dump(model, 'linear_regression_model.pkl')
```

Here we used the dummy dataset X and Y which have some underlying pattern, which we will use to predict the value of y.
We have used Linear Regression and then fit on x and y.
We have saved the model for further use.

Step3:

- Load the previous saved model and create a flask instance

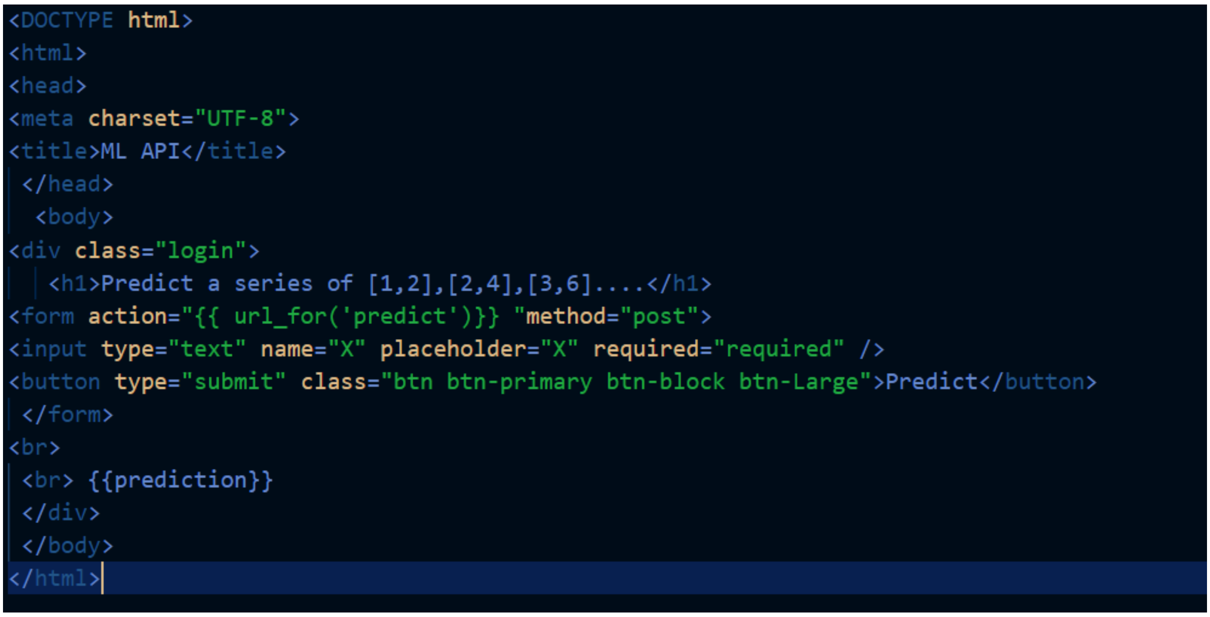


```
# Load the trained model
model = joblib.load('linear_regression_model.pkl')

# Define a route to accept input and return output
@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    # Get the input value from the request
    input_value = float(request.form['X'])

    # Use the trained model to make a prediction
    output = model.predict([[input_value]])

    # Return the predicted output as a string
    return render_template('index.html', prediction='Prediction of the model is {}'.format(output))

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

We have used two function home() to render the home page i.e index.html and then predict() function to predict the value using the saved model.

Step4:

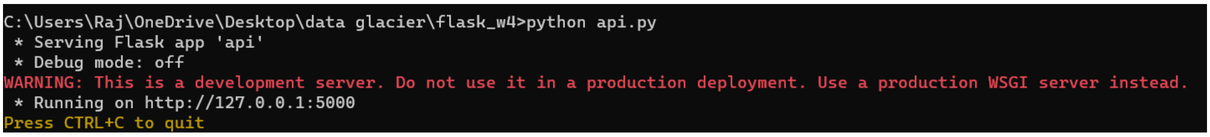
- Create a HTML page named index.html to take input X and then to predict the value from the api.py page.



```
<DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>ML API</title>
 </head>
  <body>
<div class="login">
   <h1>Predict a series of [1,2],[2,4],[3,6]....</h1>
<form action="{{ url_for('predict')}} "method="post">
<input type="text" name="X" placeholder="X" required="required" />
<button type="submit" class="btn btn-primary btn-block btn-Large">Predict</button>
 </form>
<br>
 <br> {{prediction}}
 </div>
 </body>
</html>
```

Step5:

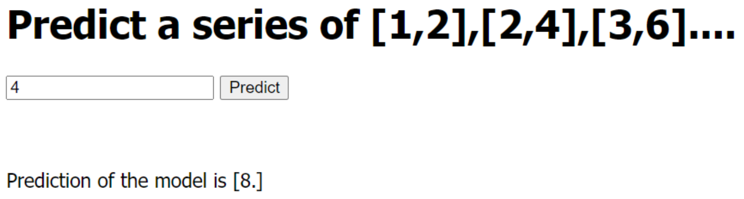
- Open the directory of the project and run the .py file from command prompt



```
C:\Users\Raj\OneDrive\Desktop\data glacier\flask_w4>python api.py
 * Serving Flask app 'api'
 * Debug mode: off
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
```

Step6:

- On your browser run: http://127.0.0.1:5000/



Predict a series of [1,2],[2,4],[3,6]....

4

Prediction of the model is [8.]

Our final web page will look like this. I have not added any CSS or styling as the main purpose was to deploy the ML model onto web.

Thank You