Name: Shreya Ramachandra

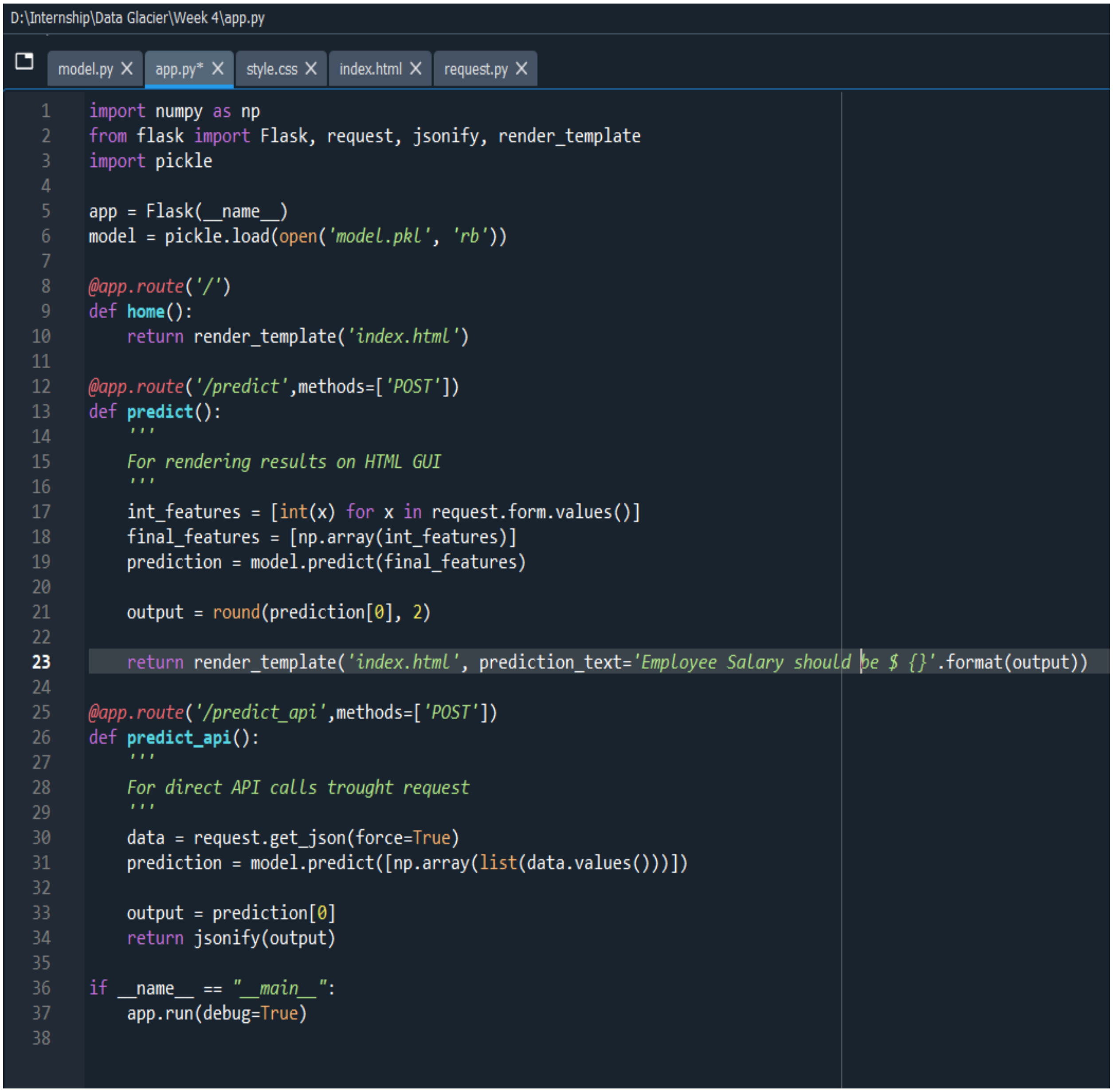
Batch No: LISUM16

Submission Date: 26th December 2022

Submitted to: Data Glacier

Title: Model Deployment on Flask: Prediction Salary Analysis

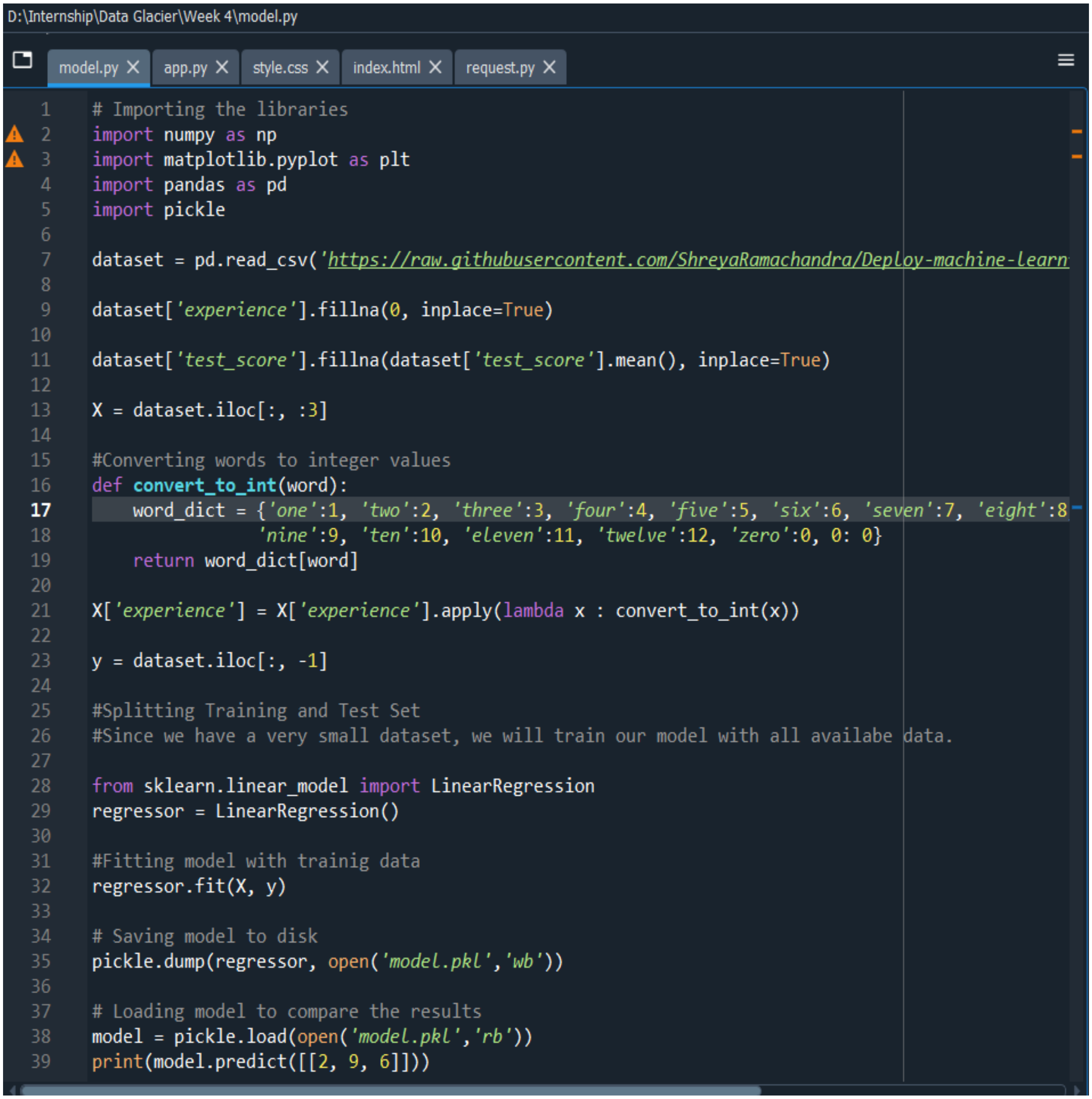
Writing program for Flask Application



```python
import numpy as np
from flask import Flask, request, jsonify, 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='Employee Salary should be $ {}'.format(output))

@app.route('/predict_api',methods=['POST'])
def predict_api():
    '''
    For direct API calls trought request
    '''
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

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

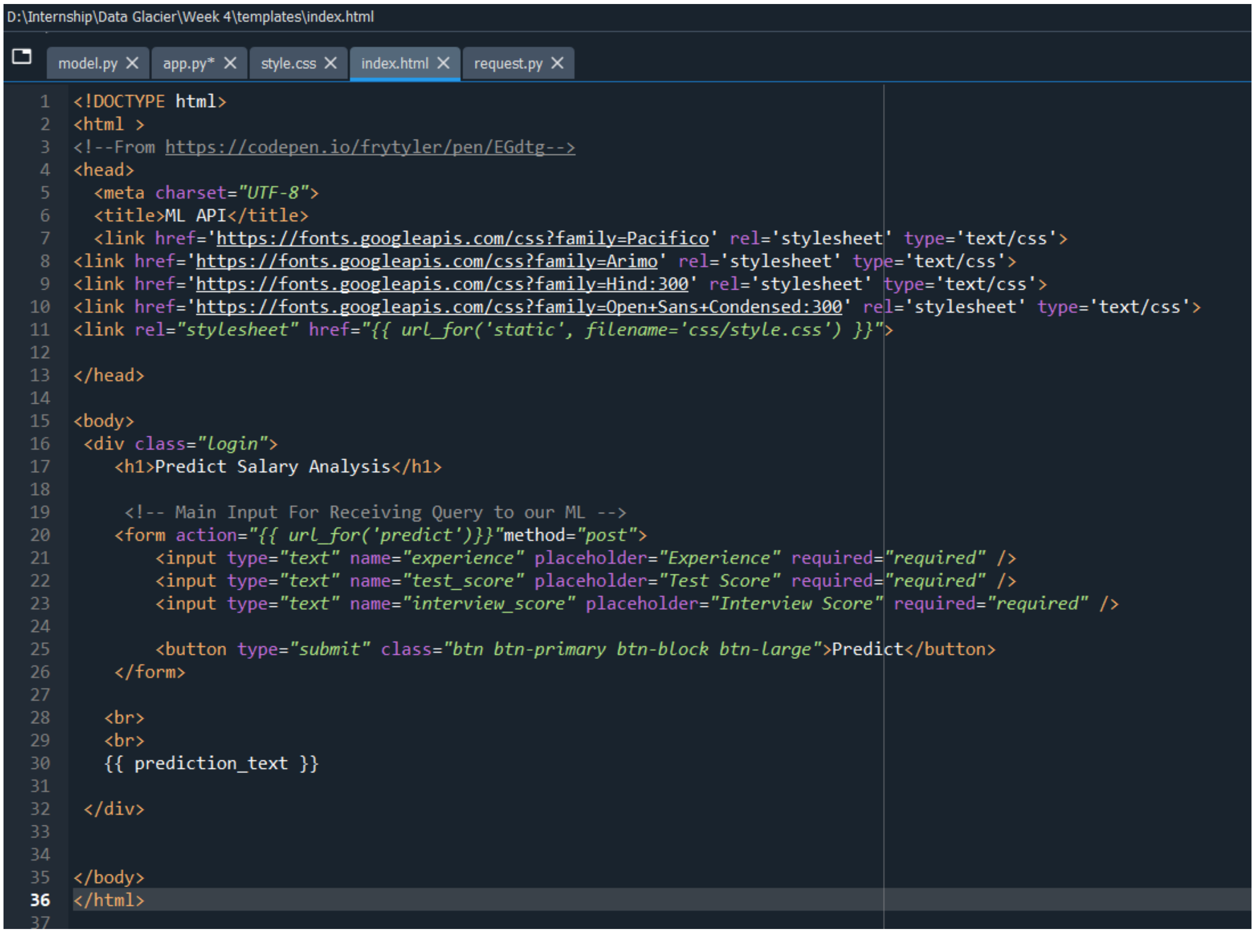
Saving the model in the Flask

Model is built for predicting the salaries of the employees based on the experience, test score and interview score



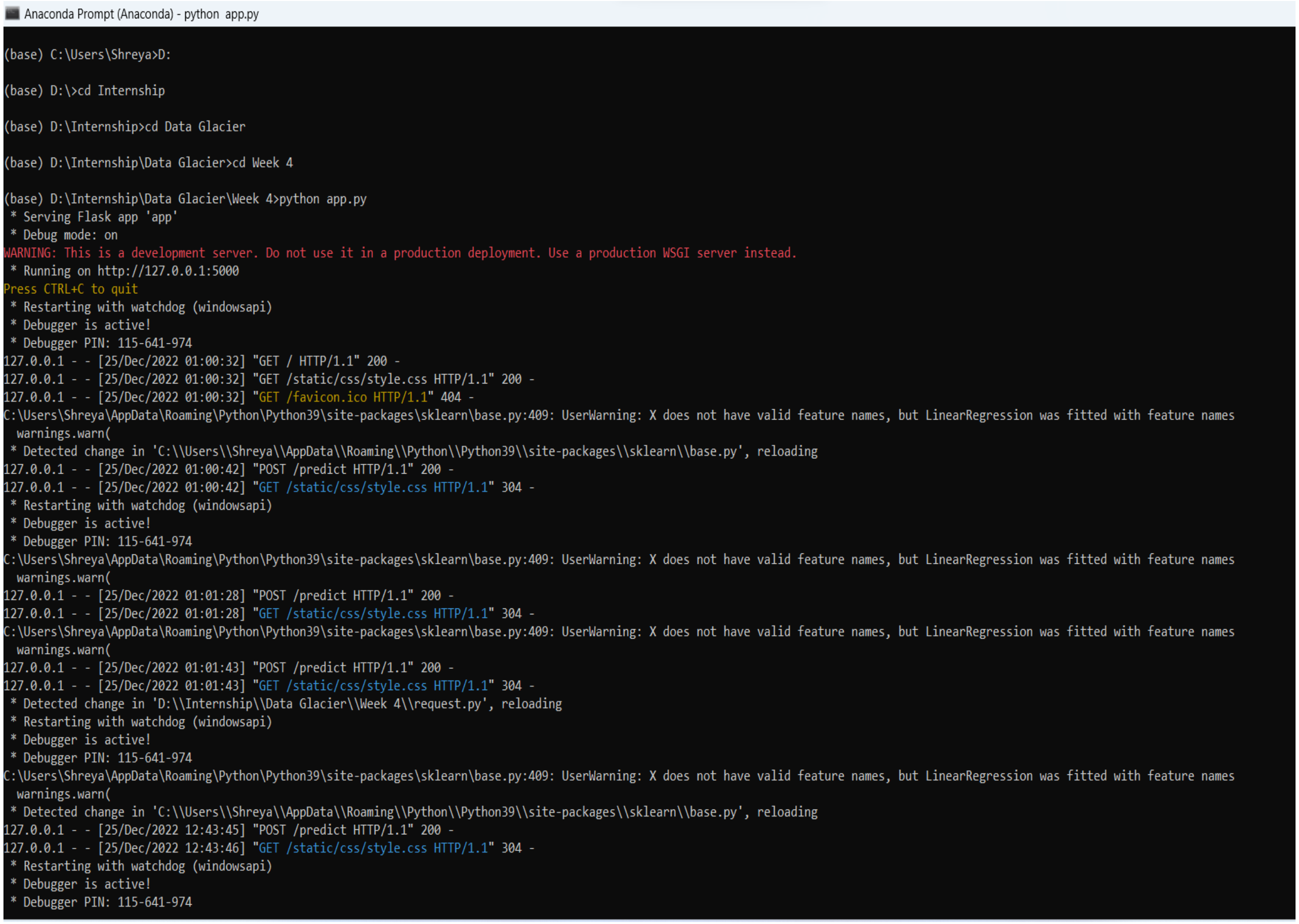
```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle

dataset = pd.read_csv('https://raw.githubusercontent.com/ShreyaRamachandra/Deploy-machine-learn

dataset['experience'].fillna(0, inplace=True)

dataset['test_score'].fillna(dataset['test_score'].mean(), inplace=True)

X = dataset.iloc[:, :3]

#Converting words to integer values
def convert_to_int(word):
    word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8
                'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
    return word_dict[word]

X['experience'] = X['experience'].apply(lambda x : convert_to_int(x))

y = dataset.iloc[:, -1]

#Splitting Training and Test Set
#Since we have a very small dataset, we will train our model with all availabe data.

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()

#Fitting model with trainig data
regressor.fit(X, y)

# Saving model to disk
pickle.dump(regressor, open('model.pkl','wb'))

# Loading model to compare the results
model = pickle.load(open('model.pkl','rb'))
print(model.predict([[2, 9, 6]]))
```

HTML file



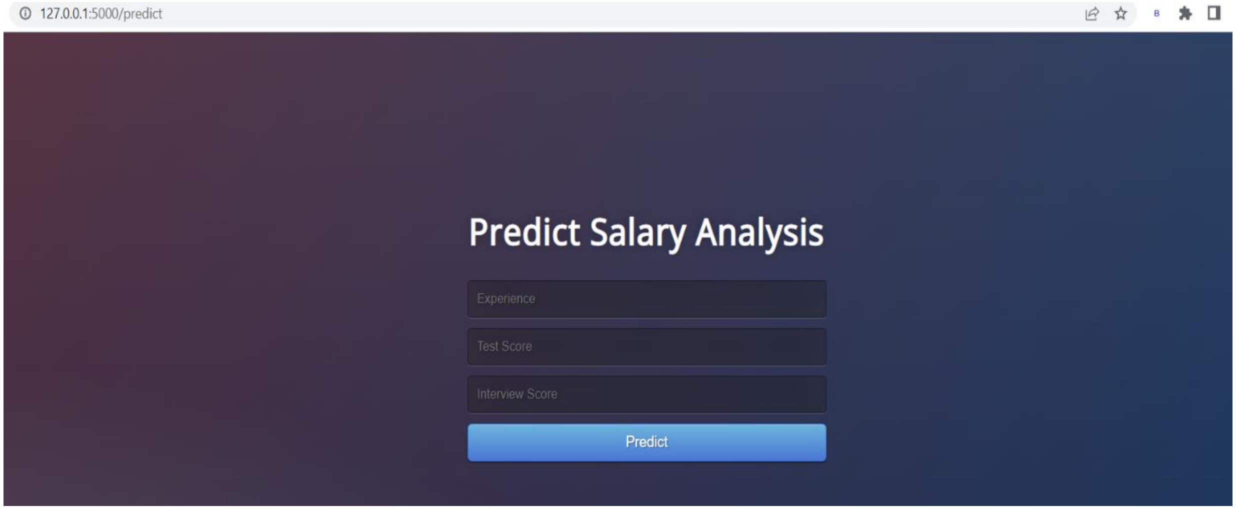
```html
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' 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'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

</head>

<body>
 <div class="login">
    <h1>Predict Salary Analysis</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
        <input type="text" name="experience" placeholder="Experience" required="required" />
        <input type="text" name="test_score" placeholder="Test Score" required="required" />
        <input type="text" name="interview_score" placeholder="Interview Score" required="required" />

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

   <br>
   <br>
   {{ prediction_text }}

 </div>


</body>
</html>
```

Running the Flask Application in Anaconda Prompt

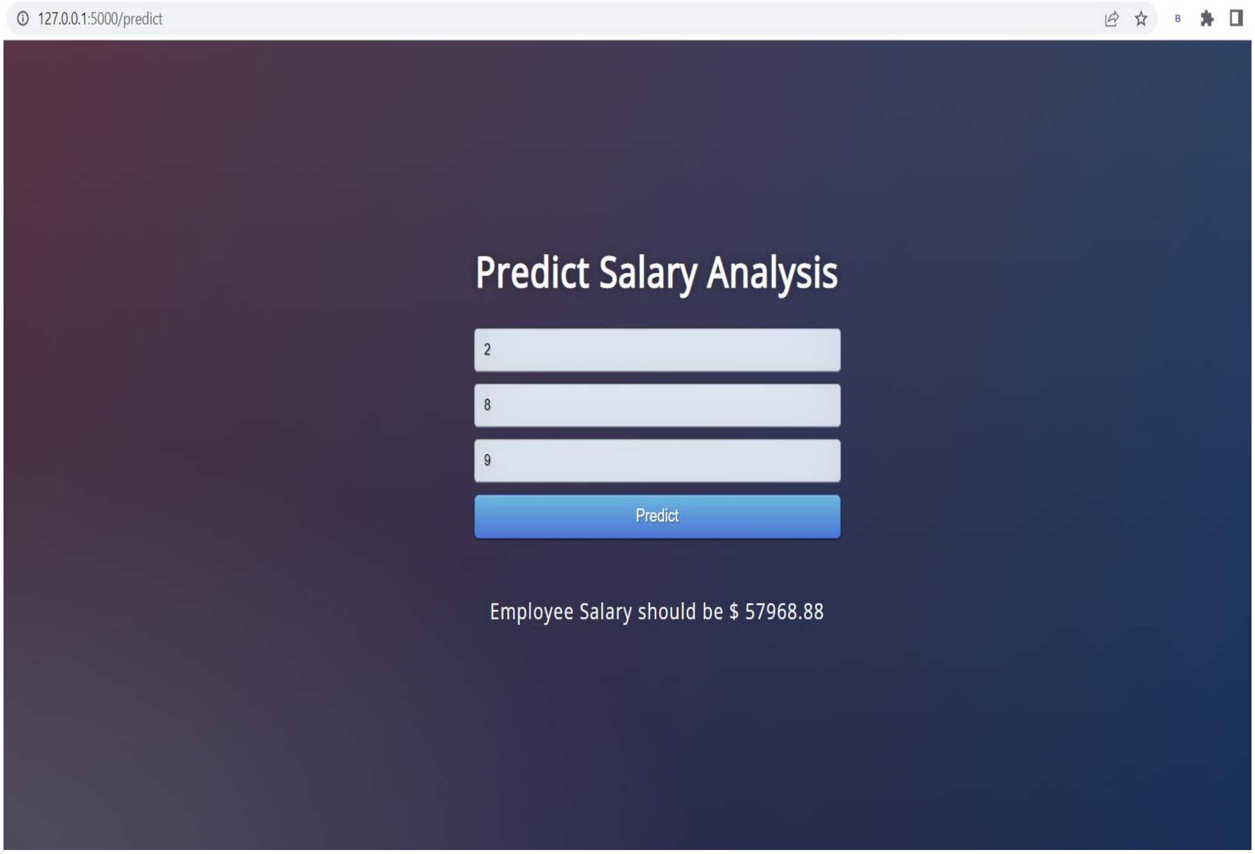


```
(base) C:\Users\Shreya>D:

(base) D:\>cd Internship

(base) D:\Internship>cd Data Glacier

(base) D:\Internship\Data Glacier>cd Week 4

(base) D:\Internship\Data Glacier\Week 4>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 watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 115-641-974
127.0.0.1 - - [25/Dec/2022 01:00:32] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 01:00:32] "GET /static/css/style.css HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 01:00:32] "GET /favicon.ico HTTP/1.1" 404 -
C:\Users\Shreya\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py:409: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
 * Detected change in 'C:\\Users\\Shreya\\AppData\\Roaming\\Python\\Python39\\site-packages\\sklearn\\base.py', reloading
127.0.0.1 - - [25/Dec/2022 01:00:42] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 01:00:42] "GET /static/css/style.css HTTP/1.1" 304 -
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 115-641-974
C:\Users\Shreya\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py:409: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
127.0.0.1 - - [25/Dec/2022 01:01:28] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 01:01:28] "GET /static/css/style.css HTTP/1.1" 304 -
C:\Users\Shreya\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py:409: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
127.0.0.1 - - [25/Dec/2022 01:01:43] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 01:01:43] "GET /static/css/style.css HTTP/1.1" 304 -
 * Detected change in 'D:\\Internship\\Data Glacier\\Week 4\\request.py', reloading
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 115-641-974
C:\Users\Shreya\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py:409: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
 * Detected change in 'C:\\Users\\Shreya\\AppData\\Roaming\\Python\\Python39\\site-packages\\sklearn\\base.py', reloading
127.0.0.1 - - [25/Dec/2022 12:43:45] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 12:43:46] "GET /static/css/style.css HTTP/1.1" 304 -
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 115-641-974
```

Testing the model



Result: Predict Salary Analysis Model



2

8

9

Employee Salary should be $ 57968.88