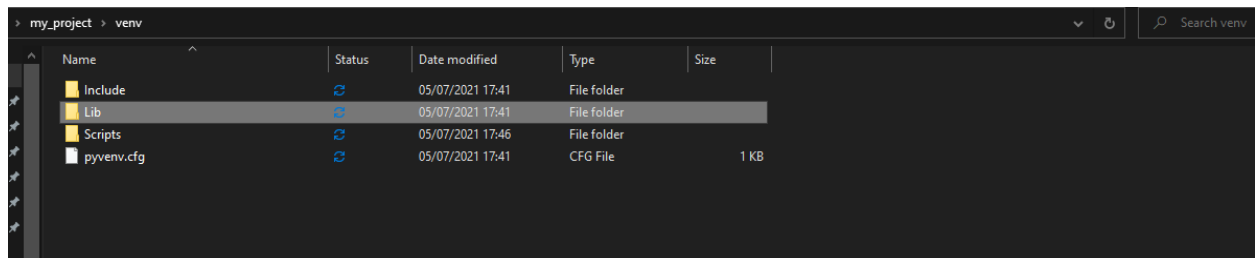


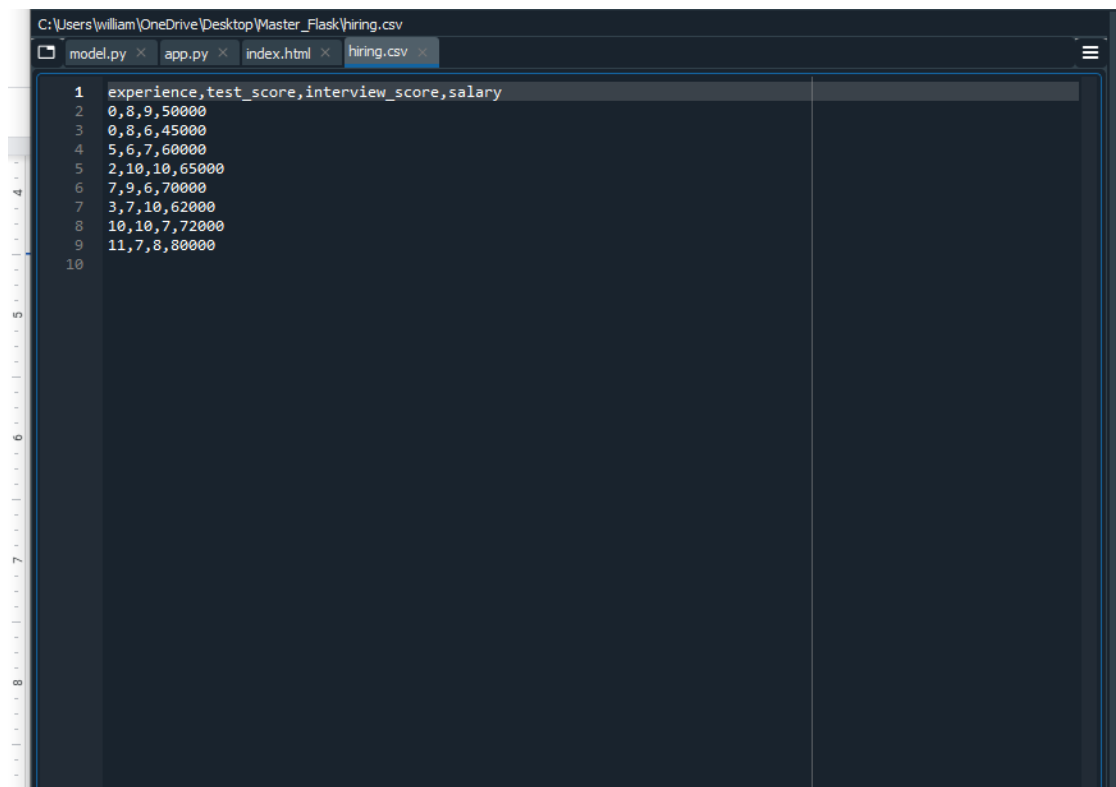
**Name:** William Okomba  
**Batch code:** [LISUM0](#)  
**Submission date:** 05/07/2021  
**Submitted to:** Data Glacier

## Snapshots of deployment

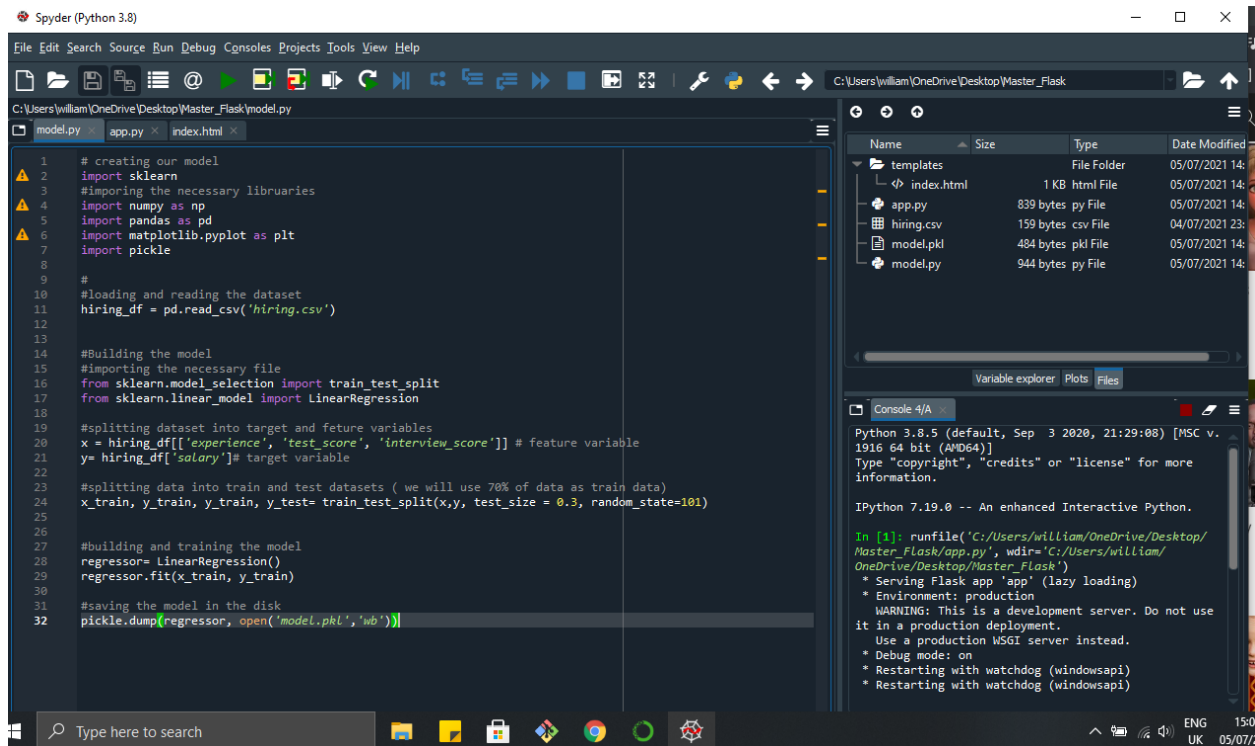
### 1. Created virtual environment called venv



### 2. Dataset for the model training and deployment



### 3. Building the model and saving in the flask



The screenshot shows the Spyder Python IDE with the following code in `model.py`:

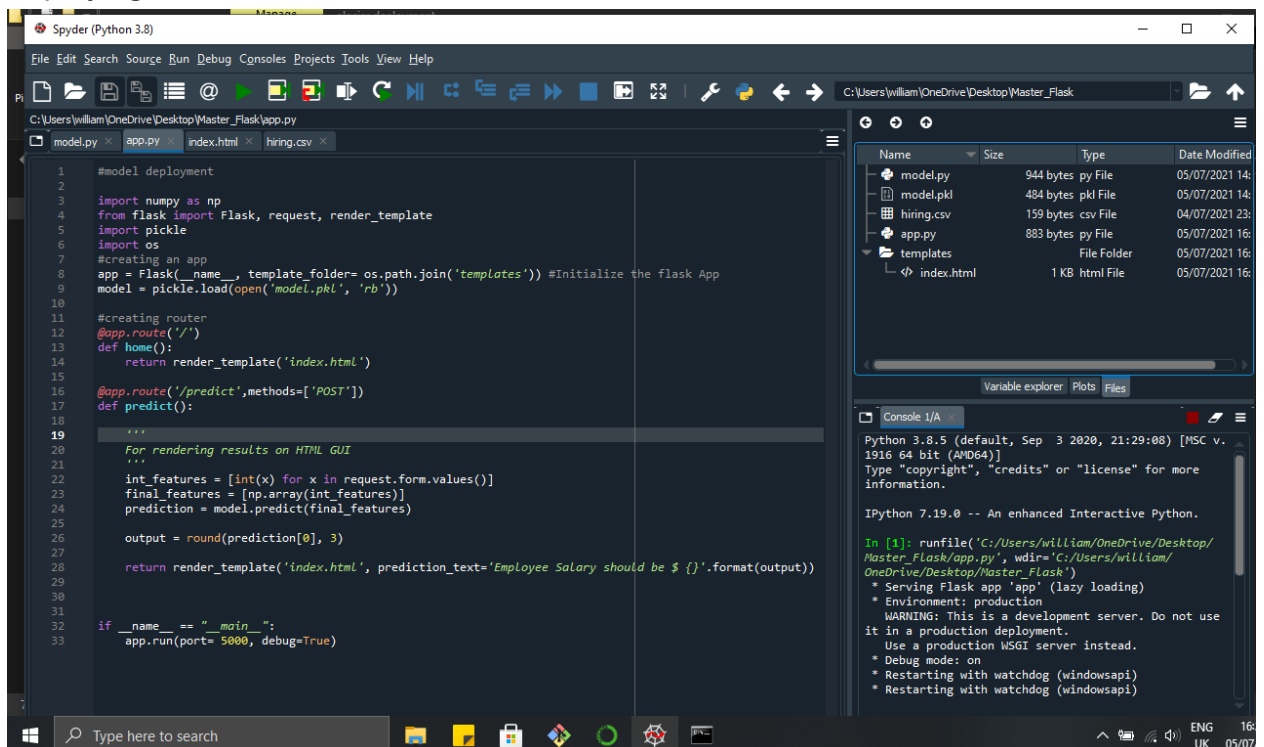
```
1 # creating our model
2 import sklearn
3 #importing the necessary libraries
4 import numpy as np
5 import pandas as pd
6 import matplotlib.pyplot as plt
7 import pickle
8
9 #
10 #loading and reading the dataset
11 hiring_df = pd.read_csv('hiring.csv')
12
13 #Building the model
14 #importing the necessary file
15 from sklearn.model_selection import train_test_split
16 from sklearn.linear_model import LinearRegression
17
18 #splitting dataset into target and feature variables
19 x = hiring_df[['experience', 'test_score', 'interview_score']] # feature variable
20 y = hiring_df['salary'] # target variable
21
22 #splitting data into train and test datasets ( we will use 70% of data as train data)
23 x_train, y_train, x_test, y_test = train_test_split(x, y, test_size = 0.3, random_state=101)
24
25 #building and training the model
26 regressor = LinearRegression()
27 regressor.fit(x_train, y_train)
28
29 #saving the model in the disk
30 pickle.dump(regressor, open('model.pkl', 'wb'))
```

The file explorer on the right shows the project structure:

| Name       | Size      | Type        | Date Modified  |
|------------|-----------|-------------|----------------|
| templates  |           | File Folder | 05/07/2021 14: |
| index.html | 1 KB      | html File   | 05/07/2021 14: |
| app.py     | 839 bytes | py File     | 05/07/2021 14: |
| hiring.csv | 159 bytes | csv File    | 04/07/2021 23: |
| model.pkl  | 484 bytes | pkl File    | 05/07/2021 14: |
| model.py   | 944 bytes | py File     | 05/07/2021 14: |

The console output shows the IPython environment and the execution of `runfile` for `Master_Flask/app.py`.

### 4. Deploying the model



The screenshot shows the Spyder Python IDE with the following code in `app.py`:

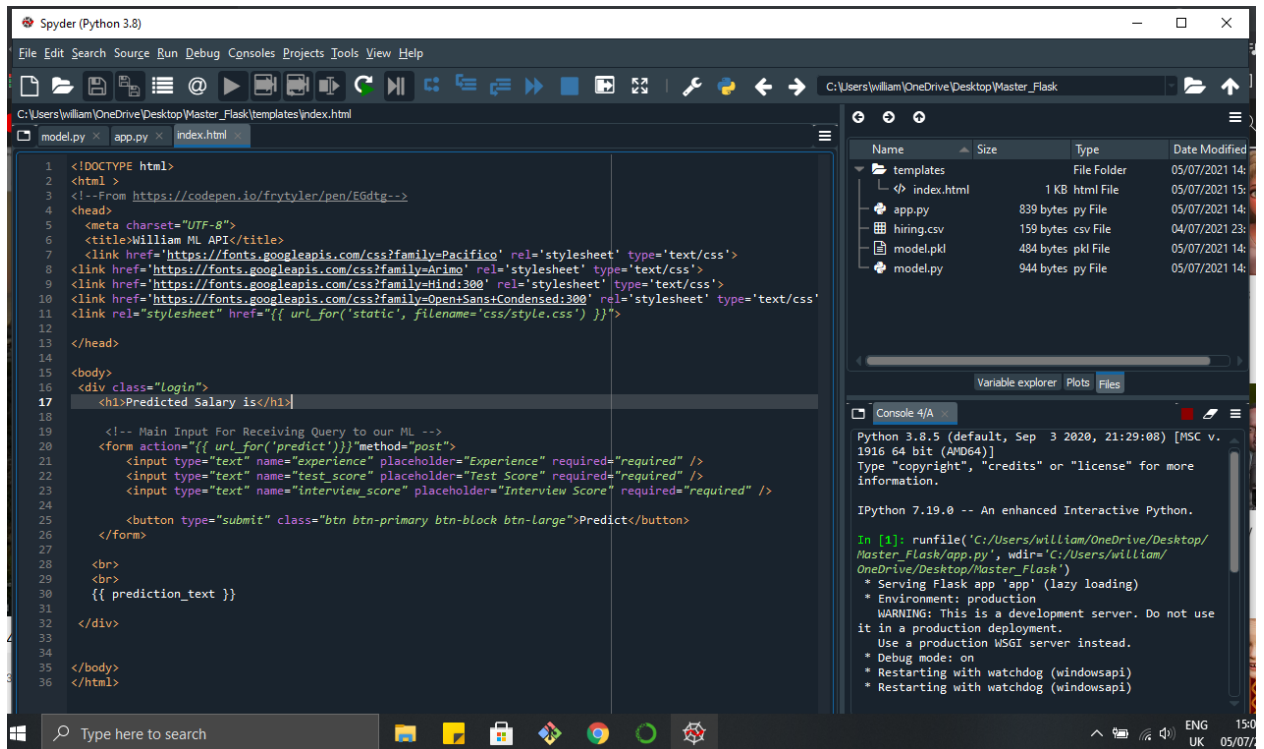
```
1 #model deployment
2
3 import numpy as np
4 from flask import Flask, request, render_template
5 import pickle
6 import os
7 #creating an app
8 app = Flask(__name__, template_folder=os.path.join('templates')) #Initialize the flask App
9 model = pickle.load(open('model.pkl', 'rb'))
10
11 #creating router
12 @app.route('/')
13 def home():
14     return render_template('index.html')
15
16 @app.route('/predict', methods=['POST'])
17 def predict():
18
19     '''
20     For rendering results on HTML GUI
21     '''
22     int_features = [int(x) for x in request.form.values()]
23     final_features = [np.array(int_features)]
24     prediction = model.predict(final_features)
25
26     output = round(prediction[0], 3)
27
28     return render_template('index.html', prediction_text='Employee Salary should be $ {}'.format(output))
29
30 if __name__ == '__main__':
31     app.run(port= 5000, debug=True)
```

The file explorer on the right shows the project structure:

| Name       | Size      | Type        | Date Modified  |
|------------|-----------|-------------|----------------|
| model.py   | 944 bytes | py File     | 05/07/2021 14: |
| model.pkl  | 484 bytes | pkl File    | 05/07/2021 14: |
| hiring.csv | 159 bytes | csv File    | 04/07/2021 23: |
| app.py     | 883 bytes | py File     | 05/07/2021 16: |
| templates  |           | File Folder | 05/07/2021 16: |
| index.html | 1 KB      | html File   | 05/07/2021 16: |

The console output shows the IPython environment and the execution of `runfile` for `Master_Flask/app.py`.

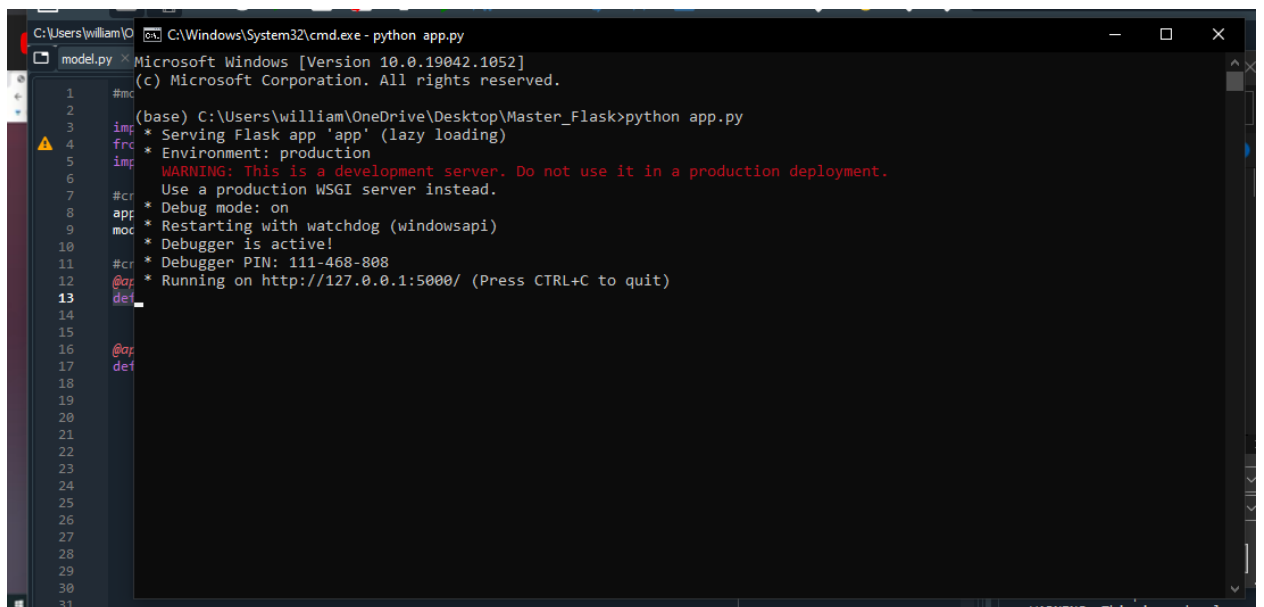
## 5. Creating html file



The screenshot shows the Spyder Python IDE interface. The main editor displays the content of `index.html`, which includes a DOCTYPE declaration, HTML tags, a meta charset, a title, and several links to Google Fonts. The body contains a login form with three input fields for experience, test score, and interview score, and a submit button. The right sidebar shows the file explorer with a list of files in the `templates` directory. The console at the bottom shows the output of running the application, including warnings about the development server and debug mode.

```
<!DOCTYPE html>
<html>
<!-- From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>William 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>Predicted Salary is</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>
    {{ prediction_text }}
  </div>
</body>
</html>
```

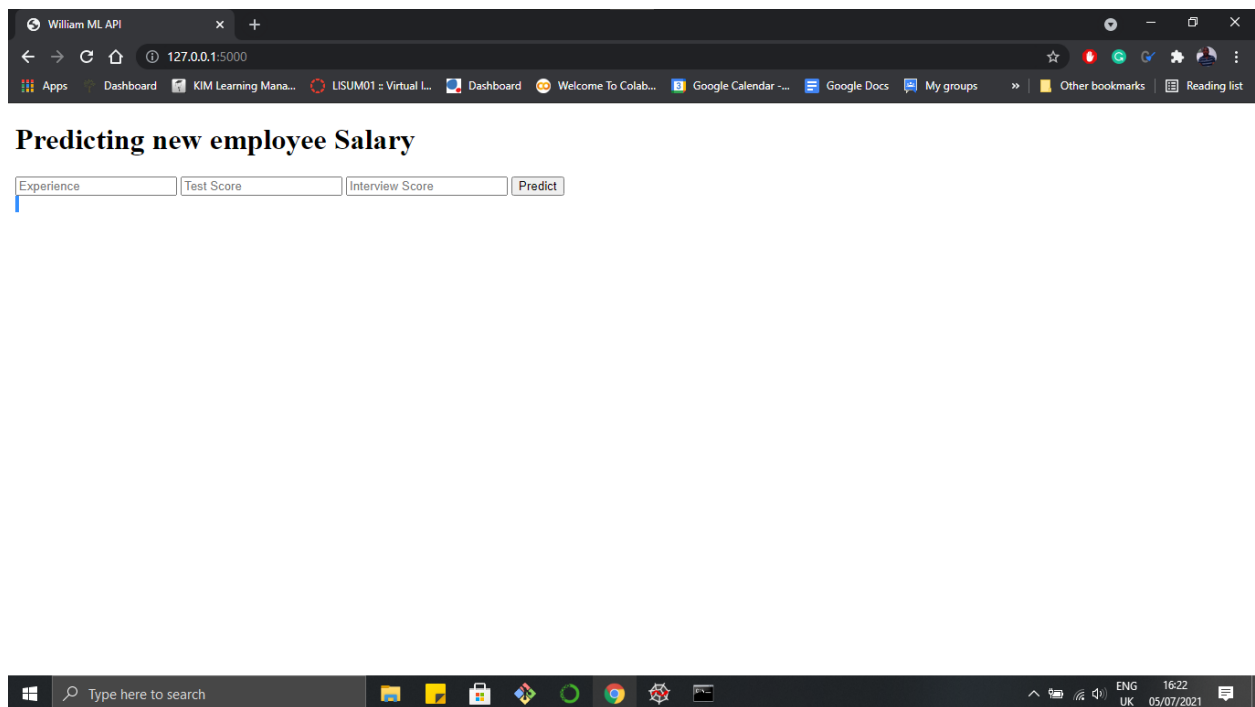
## 6. Getting the API



The screenshot shows a Windows command prompt window. The user has navigated to the directory `C:\Users\william\OneDrive\Desktop\Master_Flask` and executed the command `python app.py`. The output shows the application starting, serving Flask app 'app' (lazy loading), and running on `http://127.0.0.1:5000/`. The output also includes warnings about the development server and debug mode.

```
(base) C:\Users\william\OneDrive\Desktop\Master_Flask>python 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
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 111-468-808
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

## 7. Checking the model



## Summary

1. model.py: this file contains the machine learning model codes.
2. app.py: contains a flask API that receive the variables through GUI, compute prediction values based on the model and return the result.
3. templates folder: contains htm file (index.html) which allows user to enter employee details and display predicted salary.
4. hiring.csv file: this file was sourced from kaggle. It contains employee details such as year of experience, test score out of 10, interview score out of ten, and then the expected salary.