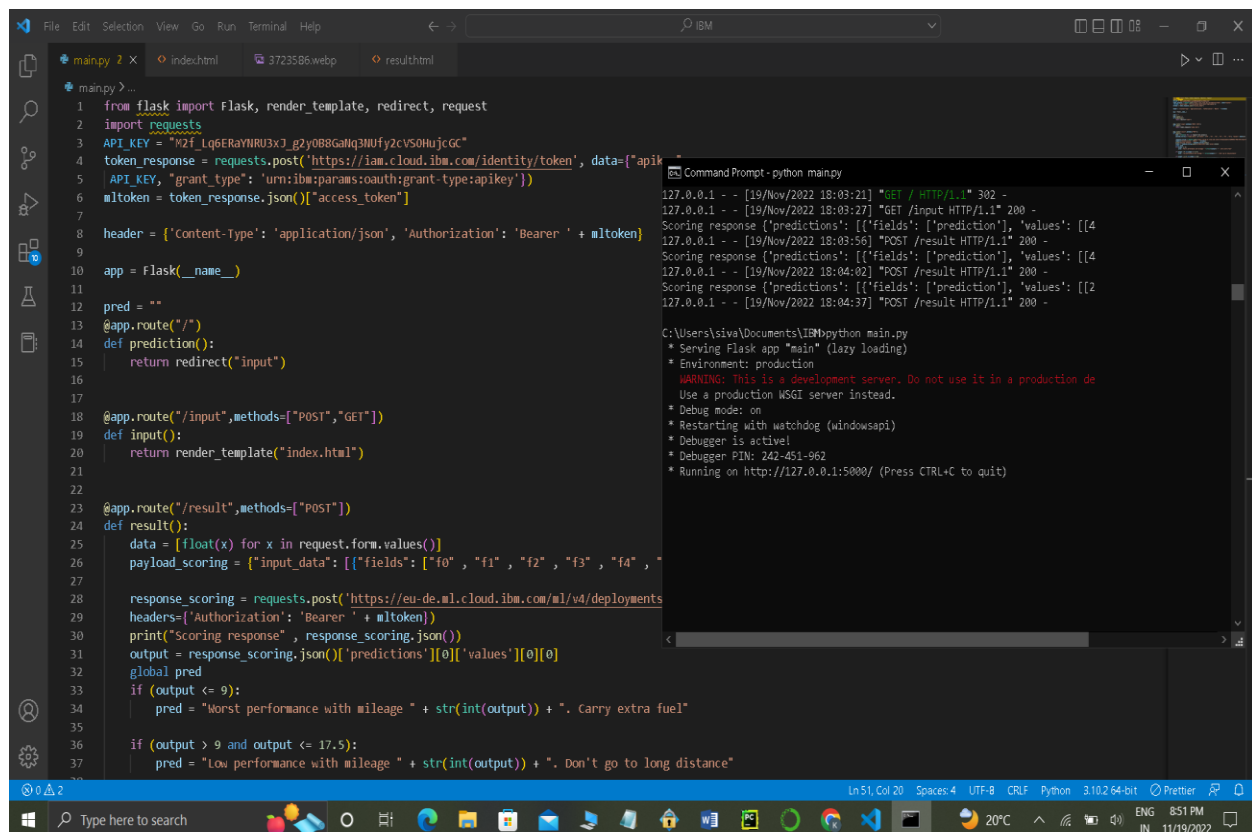


OUTPUT

After Integrate Flask with Scoring End Point :

TeamID — PNT2022TMID22357

Running After Integrate Flask with Scoring End Point :



The screenshot displays a Visual Studio Code editor with a Python file named `mainpy` and a terminal window. The code in `mainpy` is a Flask application that integrates with IBM ML Cloud for scoring. It includes imports for `Flask`, `render_template`, `redirect`, and `request`. It uses an API key to obtain an access token from IBM Cloud IAM. The application has two endpoints: `/input` (GET) which renders `index.html`, and `/result` (POST) which sends a request to the IBM ML Cloud scoring endpoint. The `/result` endpoint processes the response and returns a prediction based on the mileage value.

```
1 from flask import Flask, render_template, redirect, request
2 import requests
3 API_KEY = "M2fLq6ERAYNRU3xJ_g2y0B8GaMq3MUfy2cv50HujcGC"
4 token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
5   API_KEY, "grant_type": "urn:ibm:params:oauth:grant-type:apikey"})
6 mltoken = token_response.json()["access_token"]
7
8 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
9
10 app = Flask(__name__)
11
12 pred = ""
13 @app.route("/")
14 def prediction():
15     return redirect("/input")
16
17
18 @app.route("/input", methods=["POST", "GET"])
19 def input():
20     return render_template("index.html")
21
22
23 @app.route("/result", methods=["POST"])
24 def result():
25     data = {float(x) for x in request.form.values()}
26     payload_scoring = {"input_data": [{"fields": ["f0", "f1", "f2", "f3", "f4",
27
28     response_scoring = requests.post('https://eu-de.ml.cloud.ibm.com/ml/v4/deployments
29     headers={"Authorization": "Bearer " + mltoken})
30     print("scoring response", response_scoring.json())
31     output = response_scoring.json()["predictions"][0]["values"][0][0]
32     global pred
33     if (output <= 9):
34         pred = "Worst performance with mileage " + str(int(output)) + ". Carry extra fuel"
35
36     if (output > 9 and output <= 17.5):
37         pred = "Low performance with mileage " + str(int(output)) + ". Don't go to long distance"
```

The terminal window shows the following output:

```
127.0.0.1 - - [19/Nov/2022 18:03:21] "GET / HTTP/1.1" 302 -
127.0.0.1 - - [19/Nov/2022 18:03:27] "GET /input HTTP/1.1" 200 -
Scoring response {'predictions': [{'fields': ['prediction'], 'values': [[4
127.0.0.1 - - [19/Nov/2022 18:03:56] "POST /result HTTP/1.1" 200 -
Scoring response {'predictions': [{'fields': ['prediction'], 'values': [[4
127.0.0.1 - - [19/Nov/2022 18:04:02] "POST /result HTTP/1.1" 200 -
Scoring response {'predictions': [{'fields': ['prediction'], 'values': [[2
127.0.0.1 - - [19/Nov/2022 18:04:37] "POST /result HTTP/1.1" 200 -

C:\Users\siva\Documents\IBM\python main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production de
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 242-451-962
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```