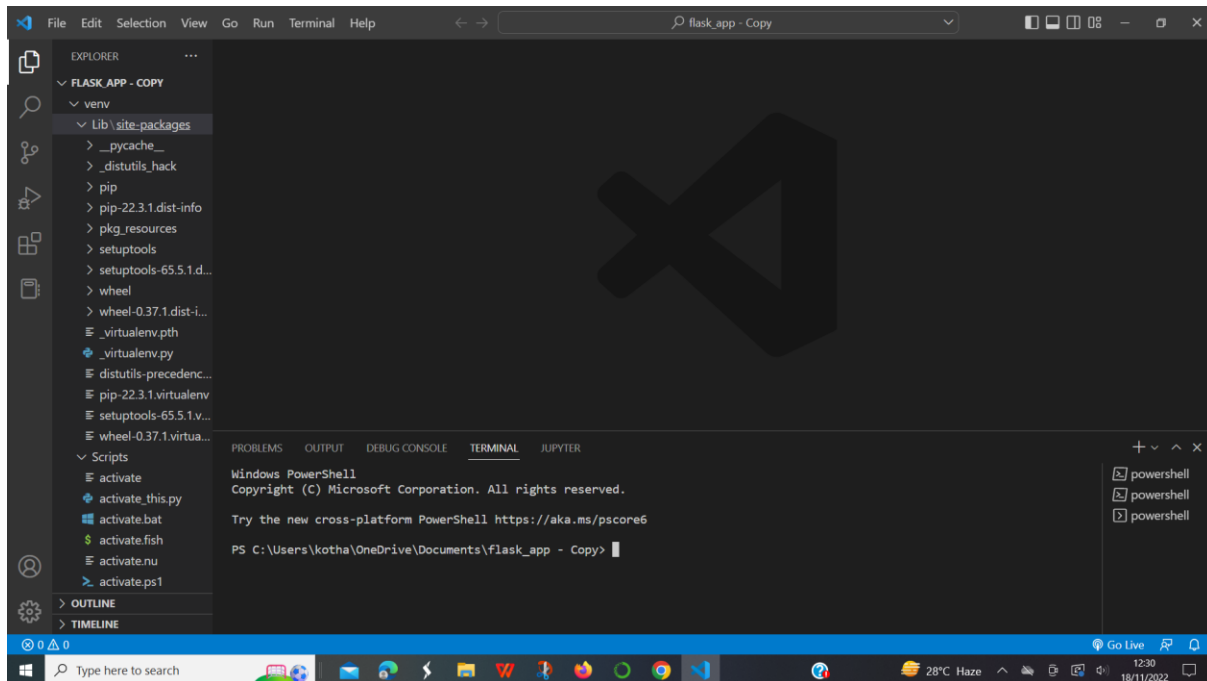


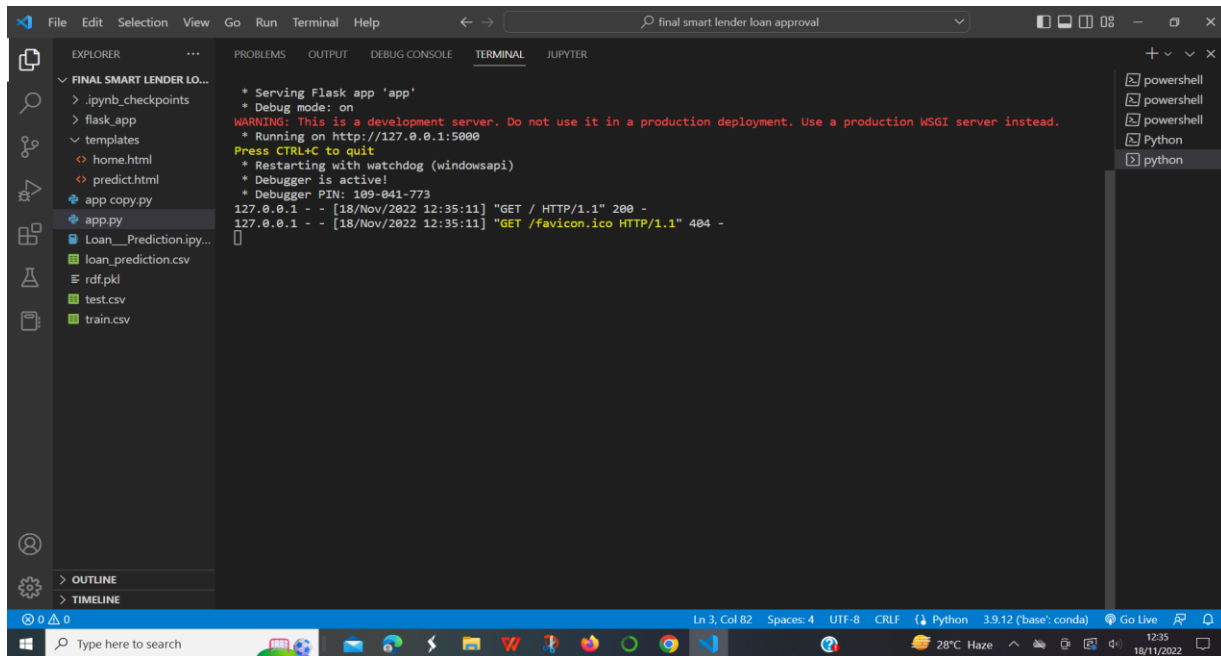
Integrate Flask with Scoring End Point

Team ID	PNT2022TMID10667
Project Name	Smart Lender - Applicant Credibility Prediction for Loan Approval

Create the virtual environment using the visual studio code



Run the python file



```
File Edit Selection View Go Run Terminal Help
final smart lender loan approval

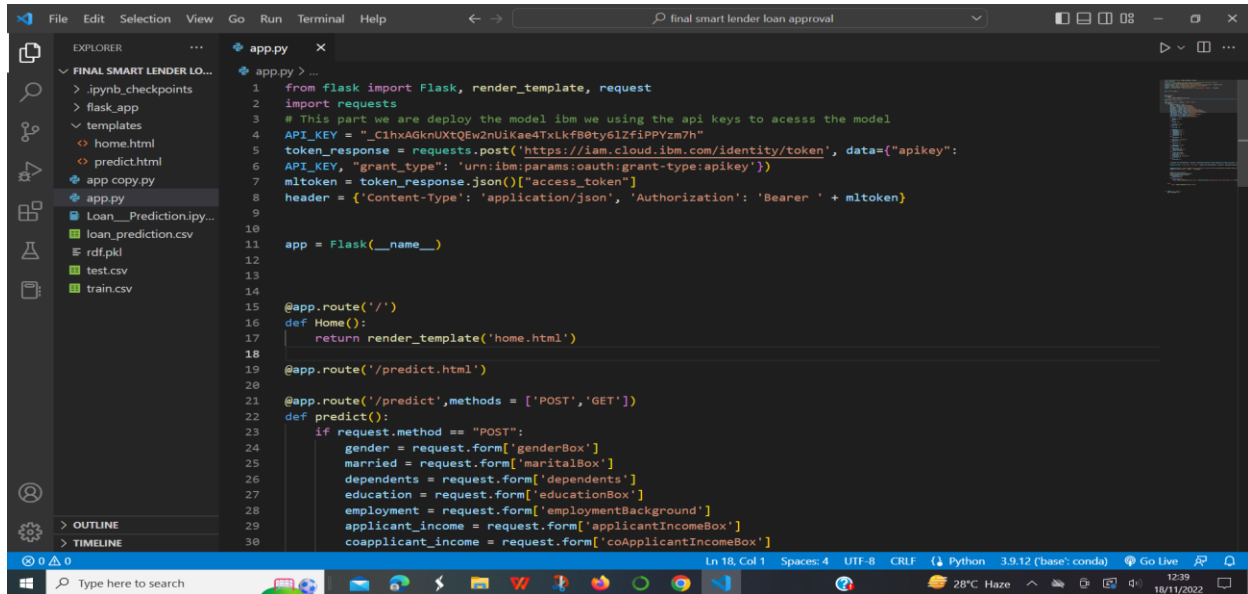
EXPLORER
FINAL SMART LENDER LO...
  .ipynb_checkpoints
  flask_app
  templates
    home.html
    predict.html
  app copy.py
  app.py
  Loan_Prediction.ipynb
  loan_prediction.csv
  rdf.pkl
  test.csv
  train.csv

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
JUPYTER

* 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: 109-041-773
127.0.0.1 - - [18/Nov/2022 12:35:11] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Nov/2022 12:35:11] "GET /favicon.ico HTTP/1.1" 404 -

powershell
powershell
Python
python
```

Use the API key that have generated in the IBM cloud in the python code



```
1 from flask import Flask, render_template, request
2 import requests
3 # This part we are deploy the model ibm we using the api keys to access the model
4 API_KEY = "_C1hxAGknUXtQew2nUiKae4TxLkF80ty61ZfIPPyzm7h"
5 token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
6 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
7 mltoken = token_response.json()["access_token"]
8 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
9
10
11 app = Flask(__name__)
12
13
14
15 @app.route('/')
16 def Home():
17     return render_template('home.html')
18
19
20 @app.route('/predict.html')
21
22 @app.route('/predict',methods = ['POST','GET'])
23 def predict():
24     if request.method == "POST":
25         gender = request.form['genderBox']
26         married = request.form['maritalBox']
27         dependents = request.form['dependents']
28         education = request.form['educationBox']
29         employment = request.form['employmentBackground']
30         applicant_income = request.form['applicantIncomeBox']
31         coapplicant_income = request.form['coApplicantIncomeBox']
```

Using the Api key, we are integrating the flask application