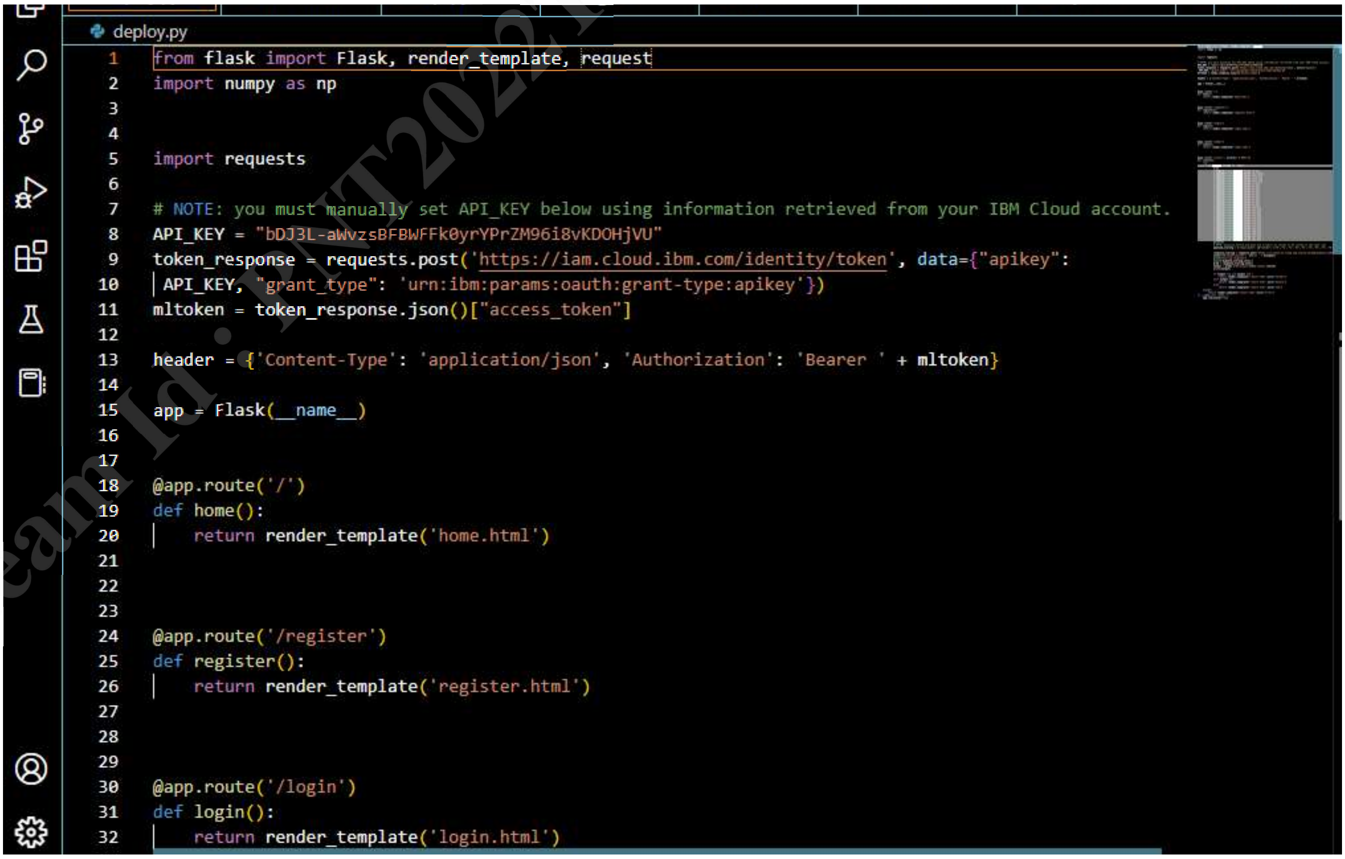
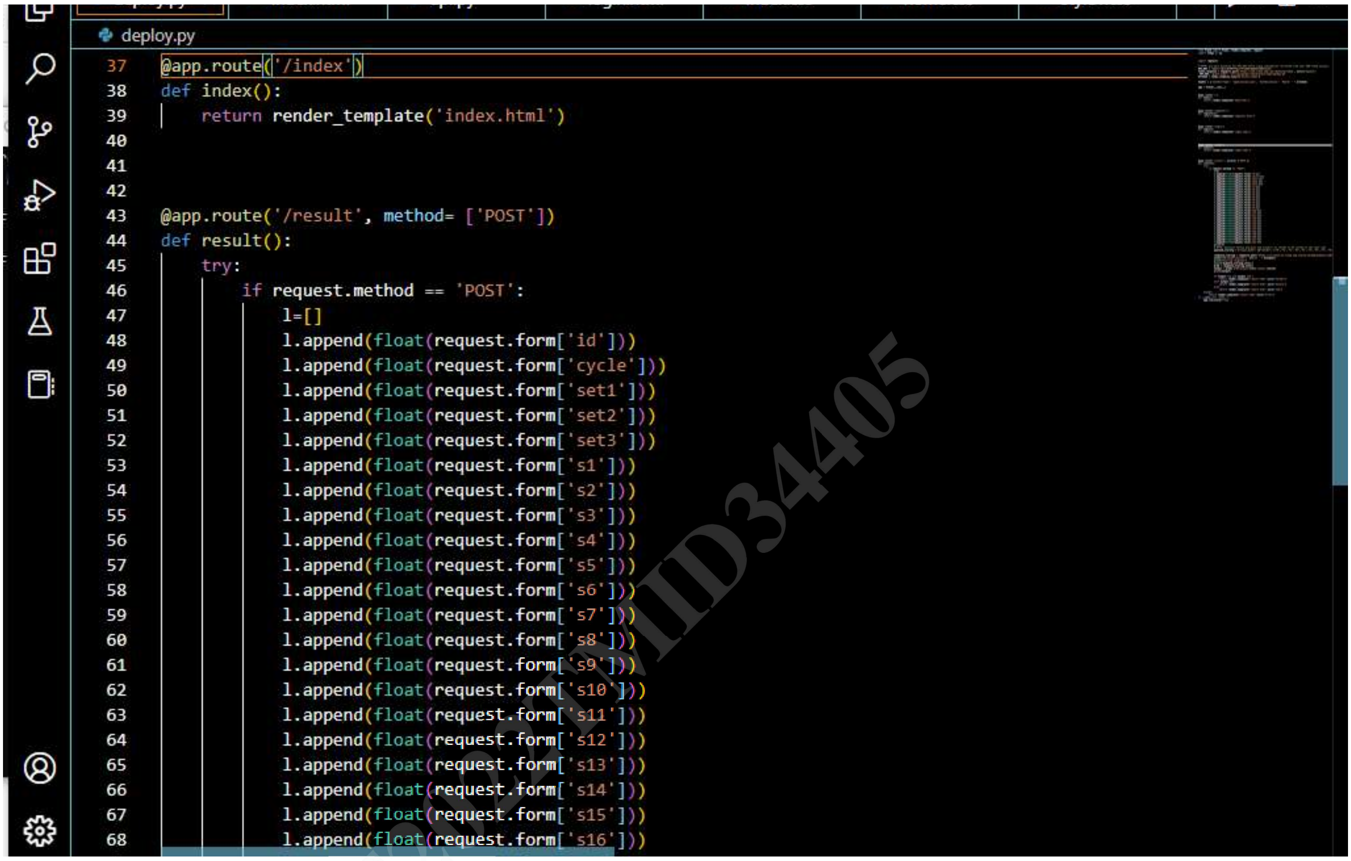
Project Development Phase
Sprint 4

Date	17 November 2022
Team ID	PNT2022TMID34405
Project Name	Machine Learning-Based Predictive Analytics for Aircraft Engine
Team Members	Abishek S (TeamLeader) Jishnu J.B (Member - 1) Madhavan M (Member - 2) Jithu Prasennan (Member - 3)

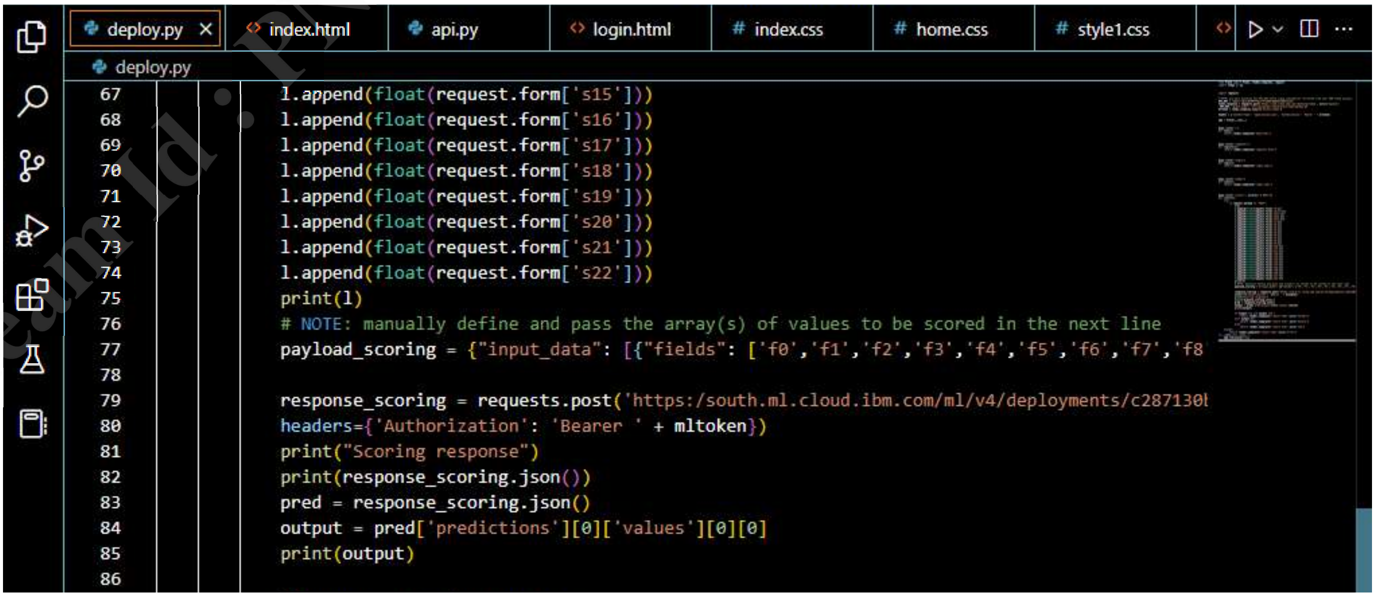
Deploy.py



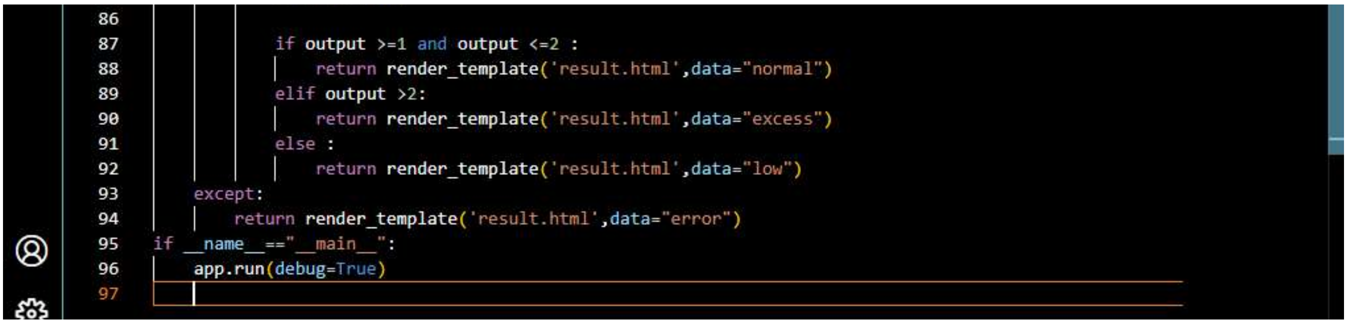
```
from flask import Flask, render_template, request
import numpy as np


import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "bDJ3L-aWvzsBFBWFFk0yrYPrZM96i8vKDOHjVU"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

app = Flask(__name__)


@app.route('/')
def home():
    return render_template('home.html')



@app.route('/register')
def register():
    return render_template('register.html')



@app.route('/login')
def login():
    return render_template('login.html')
```



```python
@app.route('/index')
def index():
    return render_template('index.html')



@app.route('/result', method= ['POST'])
def result():
    try:
        if request.method == 'POST':
            l=[]
            l.append(float(request.form['id']))
            l.append(float(request.form['cycle']))
            l.append(float(request.form['set1']))
            l.append(float(request.form['set2']))
            l.append(float(request.form['set3']))
            l.append(float(request.form['s1']))
            l.append(float(request.form['s2']))
            l.append(float(request.form['s3']))
            l.append(float(request.form['s4']))
            l.append(float(request.form['s5']))
            l.append(float(request.form['s6']))
            l.append(float(request.form['s7']))
            l.append(float(request.form['s8']))
            l.append(float(request.form['s9']))
            l.append(float(request.form['s10']))
            l.append(float(request.form['s11']))
            l.append(float(request.form['s12']))
            l.append(float(request.form['s13']))
            l.append(float(request.form['s14']))
            l.append(float(request.form['s15']))
            l.append(float(request.form['s16']))
```



```python
            l.append(float(request.form['s15']))
            l.append(float(request.form['s16']))
            l.append(float(request.form['s17']))
            l.append(float(request.form['s18']))
            l.append(float(request.form['s19']))
            l.append(float(request.form['s20']))
            l.append(float(request.form['s21']))
            l.append(float(request.form['s22']))
            print(l)
            # NOTE: manually define and pass the array(s) of values to be scored in the next line
            payload_scoring = {"input_data": [{"fields": ['f0','f1','f2','f3','f4','f5','f6','f7','f8

            response_scoring = requests.post('https:/south.ml.cloud.ibm.com/ml/v4/deployments/c287130
            headers={'Authorization': 'Bearer ' + mltoken})
            print("Scoring response")
            print(response_scoring.json())
            pred = response_scoring.json()
            output = pred['predictions'][0]['values'][0][0]
            print(output)

```



```

            if output >=1 and output <=2 :
                return render_template('result.html',data="normal")
            elif output >2:
                return render_template('result.html',data="excess")
            else :
                return render_template('result.html',data="low")
    except:
        return render_template('result.html',data="error")
if __name__=="__main__":
    app.run(debug=True)

```