Team ID	PNT2022TMID08775
ProjectName	Efficient Water Quality Analysis and Prediction using Machine Learning

Python code

```python
import numpy as np
from flask import Flask,render_template,request
import pickle
import requests

API_KEY = "THZC3nURvpmMSRhpBOMUNeqJsJ6p40DZ4pp2FSnNkWfY"
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__)
model = pickle.load(open('wqi.pkl','rb'))
@app.route('/',methods=['GET'])
def home():
    return render_template("index.html")
@app.route('/login',methods = ['POST'])
def login():
    year = request.form["year"]
    do = request.form["do"]
    ph = request.form["ph"]
    co = request.form["co"]
    bod = request.form["bod"]
    na = request.form["na"]
    tc = request.form["tc"]
    total = [[int(year),float(do),float(ph),float(co),float(bod),float(na),float(tc)]]

    payload_scoring = {"input_data": [{"fields": [['year','do','ph','co','bod','na','tc']], "values": total}]}

    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/735973ab-d35c-4182-90f9-ca418497ced0/predictions?versio
    headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    print(response_scoring.json())
    predictions=response_scoring.json()
    y_pred=predictions['predictions'][0]['values'][0][0]
```

```python
    if(y_pred >= 95 and y_pred <=100):
        return render_template("index.html",showcase = "Excellent, The Predicted Value is "+str(y_pred))
    elif(y_pred >=89 and y_pred <=94):
        return render_template("index.html",showcase = "Very Good, The Predicted Value is "+str(y_pred))
    elif(y_pred >=80 and y_pred <=88):
        return render_template("index.html",showcase = "Good, The Predicted Value is "+str(y_pred))
    elif(y_pred>=65 and y_pred<=79):
        return render_template("index.html",showcase = "Fair, The Predicted Value is "+str(y_pred))
    elif(y_pred>=45 and y_pred<=64):
        return render_template("index.html",showcase = "Marginal, The Predicted Value is "+str(y_pred))
    else:
        return render_template("index.html",showcase = "Poor, The Predicted Value is "+str(y_pred))

if __name__ == '__main__':
    app.run(debug = True,port = 5000)
```