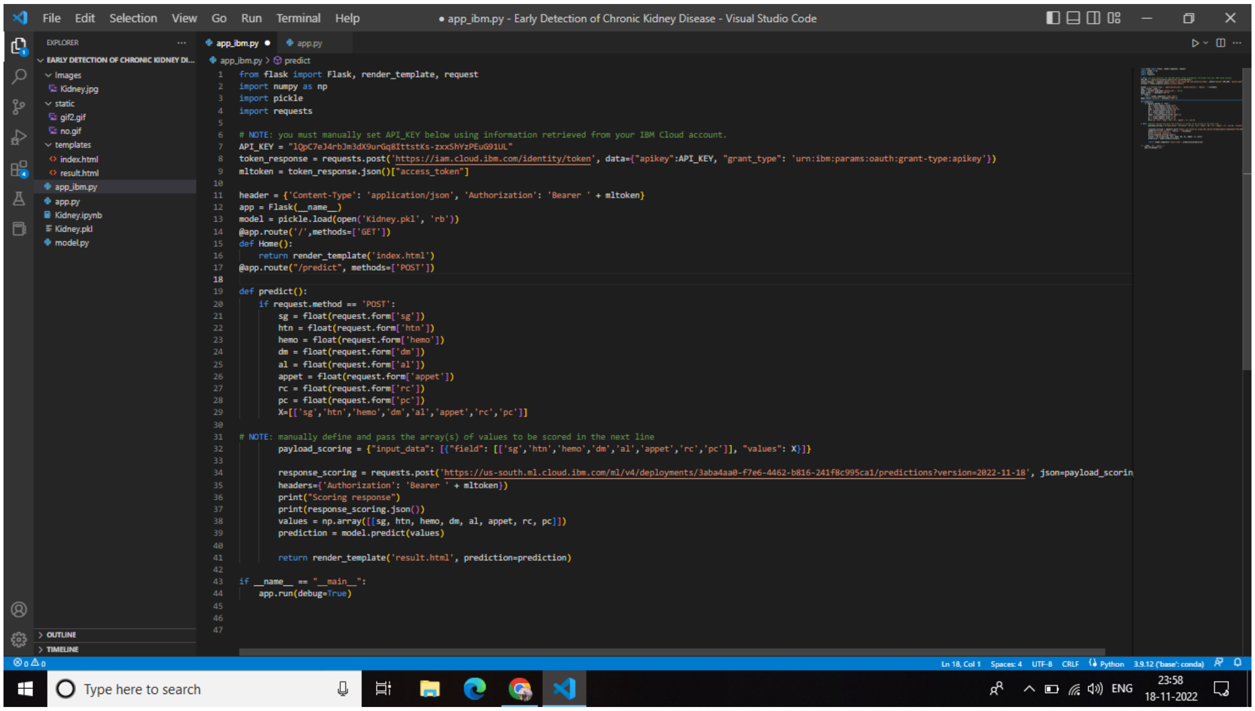
INTEGRATE FLASK WITH SCORING ENDPOINTS

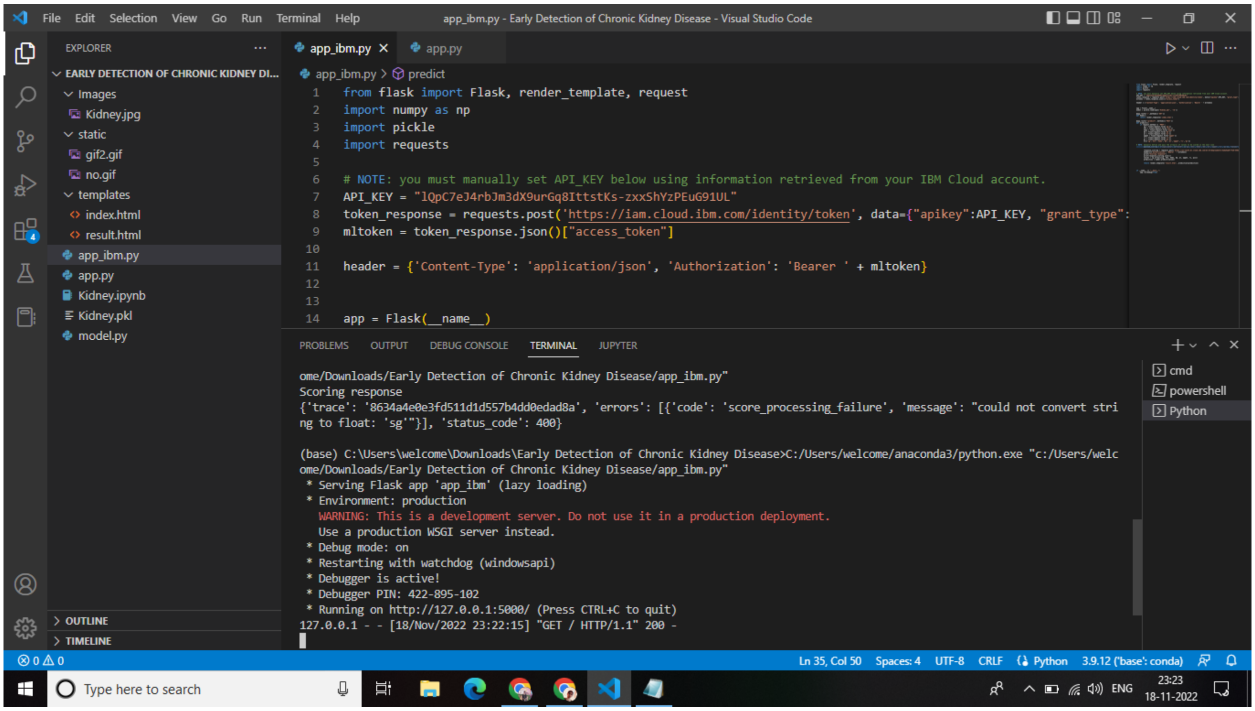
Date	19 November 2022
Team ID	PNT2022TMID04381
Project Name	Project - Early Detection of Chronic Kidney Disease Using Machine Learning

APP_IBM.PY



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

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "1QpC7eJ4rbJm3dX9urGq8IttstKs-zxxShYzPEuG91UL"
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('Kidney.pkl', 'rb'))
@app.route('/',methods=['GET'])
def Home():
    return render_template('index.html')
@app.route("/predict", methods=['POST'])

def predict():
    if request.method == 'POST':
        sg = float(request.form['sg'])
        htn = float(request.form['htn'])
        hemo = float(request.form['hemo'])
        dm = float(request.form['dm'])
        al = float(request.form['al'])
        appet = float(request.form['appet'])
        rc = float(request.form['rc'])
        pc = float(request.form['pc'])
        X=[['sg','htn','hemo','dm','al','appet','rc','pc']]

# NOTE: manually define and pass the array(s) of values to be scored in the next line
        payload_scoring = {"input_data": [{"field": [['sg','htn','hemo','dm','al','appet','rc','pc']], "values": X}]}

        response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/3aba4aa0-f7e6-4462-b816-241f8c995ca1/predictions?version=2022-11-18', json=payload_scorin
        headers={'Authorization': 'Bearer ' + mltoken})
        print("Scoring response")
        print(response_scoring.json())
        values = np.array([[sg, htn, hemo, dm, al, appet, rc, pc]])
        prediction = model.predict(values)

        return render_template('result.html', prediction=prediction)

if __name__ == "__main__":
    app.run(debug=True)
```

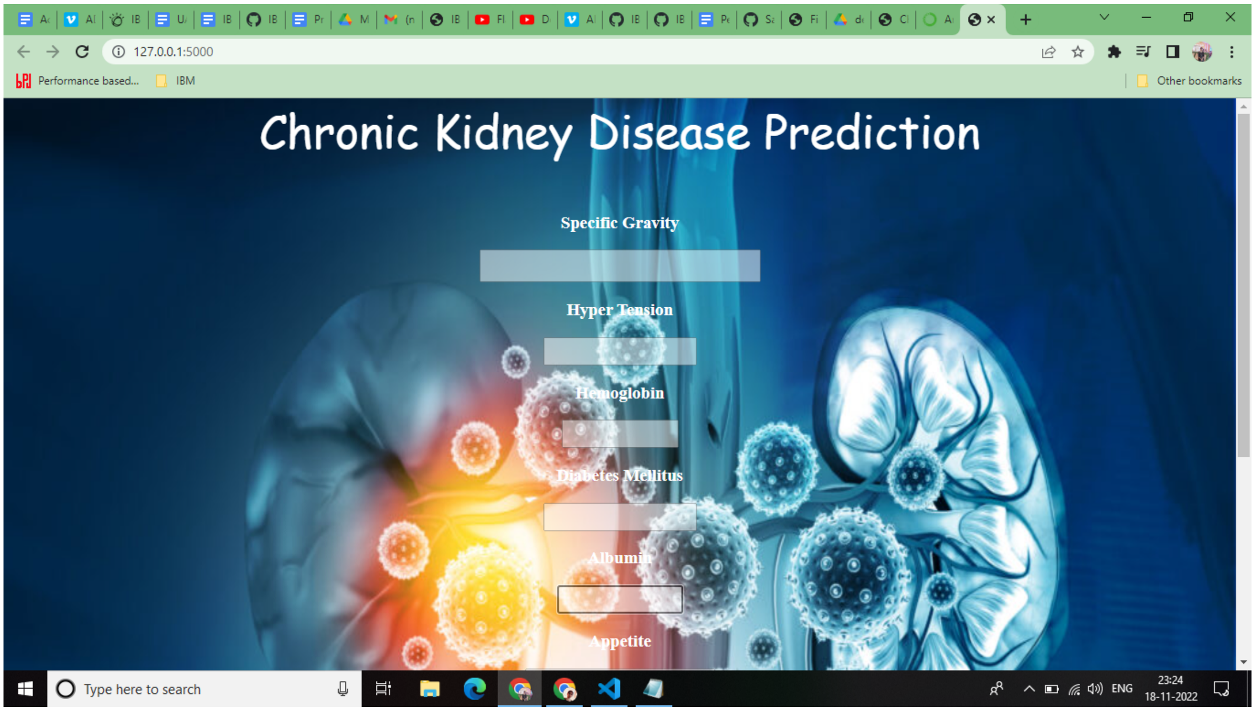
RUN THE APP:



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

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "lQpC7eJ4rbJm3dX9urGq8IttstKs-zxxShYzPEuG91UL"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":API_KEY, "grant_type":
mltoken = token_response.json()["access_token"]

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


app = Flask(__name__)
```

```
ome/Downloads/Early Detection of Chronic Kidney Disease/app_ibm.py"
Scoring response
{'trace': '8634a4e0e3fd511d1d557b4dd0edad8a', 'errors': [{'code': 'score_processing_failure', 'message': "could not convert stri
ng to float: 'sg'"}], 'status_code': 400}

(base) C:\Users\welcome\Downloads\Early Detection of Chronic Kidney Disease>C:/Users/welcome/anaconda3/python.exe "c:/Users/welc
ome/Downloads/Early Detection of Chronic Kidney Disease/app_ibm.py"
 * Serving Flask app 'app_ibm' (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: 422-895-102
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [18/Nov/2022 23:22:15] "GET / HTTP/1.1" 200 -
```

OUTPUT:



Chronic Kidney Disease Prediction

Specific Gravity

Hyper Tension

Hemoglobin

Diabetes Mellitus

Albumin

Appetite