Team ID	PNT2022TMID52708
Project Name	Early Detection of Chronic Kidney Disease using Machine Learning

app.py

C: > Users > Sasikumar > Downloads > IBM-Project-11080-1659260039-main (1) > IBM-Project-11080-1659260039-main > Project Development Phase > Sprint 3 > CKD Prediction > app.py

```python
import pandas as pd
from flask import Flask, request, render_template, url_for, session, redirect
import pickle
from flask_mysqldb import MySQL
import MySQLdb.cursors
import re

import requests
API_KEY = "0_HBSpdG-lQimCMKdA8WhsPoIAIvtjAtLI2FUetE_8uo"
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.Flask(__name__,static_folder='')
model = pickle.load(open('./CKD.pkl','rb'))
decision_model=pickle.load(open('./DTCKD.pkl','rb'))
app.secret_key = '123'

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'user'

mysql = MySQL(app)

@app.route('/')
def homePage():
```

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

@app.route('/dashboard')
def dashboard():
    print(session)
    return render_template('dashboard.html',name=session['username'])

@app.route('/prediction', methods=['POST','GET'])
def predictCKD():
    return render_template('predict.html')
'''
@app.route('/home', methods=['POST'])
def home():
    return render_template('index.html')
'''

@app.route('/predict', methods=['POST','GET'])
def predict():
    input_features = [float(x) for x in request.form.values()]
    print(input_features)
    features_value = [np.array(input_features)]

    features_name = ['red_blood_cells', 'pus_cell', 'blood_glucose_random', 'blood_urea','pedal_edema','anemia','diabetes_mellitus','corona
    df = pd.DataFrame(features_value, columns=features_name)
    output = model.predict(df)
```

```
        session['loggedin'] = True
        #session['id'] = account['id']
        session['username'] = account['username']
        msg = 'Logged in successfully !'
        return render_template('dashboard.html', msg = msg, name=username)
    else:
        msg = 'Incorrect username / password !'
    return render_template('index.html', msg = msg)

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    #session.pop('id', None)
    session.pop('username', None)
    return redirect('/')

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    #if request.method == 'POST' and 'username' in request.form and 'password' in request.form and 'email' in request.form :
    username = request.form['username']
    password = request.form['password']
    email = request.form['email']
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute('SELECT * FROM accounts WHERE username = % s', (username, ))
    account = cursor.fetchone()
    if account:
        msg = 'Username already exists !'
```

```
    account = cursor.fetchone()
    if account:
        msg = 'Username already exists !'
        return render_template('index.html', msg = msg)
    else:
        cursor.execute('INSERT INTO accounts VALUES (% s, % s, % s)', (username, password, email ))
        mysql.connection.commit()
        msg = 'You have successfully registered !'
        return render_template('dashboard.html', msg = msg, name=username)

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


"""
```