from flask import Flask, render\_template, request

import os

from werkzeug.utils import secure\_filename

from PIL import Image

import label\_image

import image\_fuzzy\_clustering as fem

import numpy as np

from keras.preprocessing.image import img\_to\_array

from keras.applications import imagenet\_utils

# Function to load the image and make predictions

def load\_image(image):

    text = label\_image.main(image)

    return text

def prepare\_image(image, target):

    # Convert image mode if necessary

    if image.mode != "RGB":

        image = image.convert("RGB")

    # Resize and preprocess the image

    image = image.resize(target)

    image = img\_to\_array(image)

    image = np.expand\_dims(image, axis=0)

    image = imagenet\_utils.preprocess\_input(image)

    return image

app = Flask(\_\_name\_\_)

UPLOAD\_FOLDER = os.path.join(app.root\_path, 'static', 'img')

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

# Routes

@app.route('/')

@app.route('/first')

def first():

    return render\_template('first.html')

@app.route('/login')

def login():

    return render\_template('login.html')

@app.route('/chart')

def chart():

    return render\_template('chart.html')

@app.route('/upload')

def upload():

    return render\_template('index1.html')

@app.route('/success', methods=['POST'])

def success():

    if request.method == 'POST':

        i = int(request.form.get('cluster'))

        f = request.files['file']

        fname, f\_ext = os.path.splitext(f.filename)

        original\_pic\_path = save\_img(f, f.filename)

        fem.plot\_cluster\_img(original\_pic\_path, i)

        return render\_template('success.html')

def save\_img(img, filename):

    picture\_path = os.path.join(current\_app.root\_path, 'static/images', filename)

    i = Image.open(img)

    i.save(picture\_path)

    return picture\_path

@app.route('/index')

def index():

    return render\_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])

def upload1():

    if request.method == 'POST':

        # Get the file from the POST request

        f = request.files['file']

        file\_path = secure\_filename(f.filename)

        f.save(file\_path)

        # Make prediction

        result = load\_image(file\_path)

        result = result.title()

        d = {

            "1": " → Age = 30-35 , SBP = 140-160 mmHg, DBP = 80-90 mmHg, BMI = 27-29, HbA1c = 4-5.6, Risk of Heart Attack = Very Low Risk 20% ",

            "2": " → → Age = 35-40 , SBP = 150-166 mmHg, DBP = 85-95 mmHg, BMI = 29-31, HbA1c = 7.5 - 10.5 , Risk of Heart Attack = Mild Risk 40%",

            "3": " → Age = 35-40 , SBP = 120-136 mmHg, DBP = 75-55 mmHg, BMI = 18-25, HbA1c = 5.5 - 6.5 , Risk of Heart Attack = No Risk You are Healthy",

            "4": " → Age = 45-60 , SBP = 160-176 mmHg, DBP = 95-100 mmHg, BMI = 30-35, HbA1c = 13.4 -14.9 , Risk of Heart Attack = High Chance of Heart Attack 60%",

            "0": " → Age = 20-25 , SBP = 111-126 mmHg, DBP = 80-85 mmHg, BMI = 18-25, HbA1c = 5.4 -7.0 , Risk of Heart Attack = No Risk You are Healthy"

        }

        result = result + d[result]

        print(result)

        os.remove(file\_path)

        return result

    return None

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)