Sprint-3

Python Code Building - Routing To HTML Pages

Date	19 November 2022
Team ID	PNT2022TMID04240
Project Name	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Maximum Marks	10 Marks

Building Python Code :

Importing Flask

```
from flask import Flask, render_template, request,session import os
from werkzeug.utils import secure_filename import numpy as np
from keras.models import load_model
from keras.utils import load_img,img_to_array import sqlite3



UPLOAD_FOLDER=os.path.join('static','uploads')
ALLOWED_EXTENSIONS = {'jpg','png','jpeg'}


app = Flask( name , template_folder="templates")
app.config['UPLOAD_FOLDER']=UPLOAD_FOLDER
app.secret_key = "nutrition"
```

Routing To HTML Pages

@app.route('/') def home():

return render_template('home.html')

@app.route('/aboutus') def aboutus():

return render_template("aboutus.html")

@app.route('/image') def image():

return render_template("image.html") @app.route('/imageprediction', methods=['GET', 'POST'])

Image Prediction

def imageprediction():

if request.method=="POST": img = request.files["image"]

img_filename = secure_filename(img.filename) img.save(os.path.join(app.config['UPLOAD_FOLDER'],img_filename)) session['uploaded_img_filepath'] =

os.path.join(app.config['UPLOAD_FOLDER'],img_filename) img_filepath = session.get('uploaded_img_filepath',None) image_pred = launch(img_filepath)

print(image_pred) print("image_pred",image_pred[1]) fruit = 'FRUIT: ' + image_pred[1]['FRUIT']

serving_size = 'SERVING_SIZE: ' + image_pred[1]['SERVING_SIZE'] energy = 'ENERGY: ' + image_pred[1]['ENERGY']

fat = 'FAT: ' + image_pred[1]['FAT']

saturated_fat = 'SATURATED FAT: ' + image_pred[1]['SATURATED FAT'] mono_unsaturated_fat = 'MONO_UNSATURATED_FAT: ' +

image_pred[1]['MONO_UNSATURATED_FAT'] poly_unsaturated_fat = 'POLY_UNSATURATED_FAT: ' +

image_pred[1]['POLY_UNSATURATED_FAT']

carbohydrates = 'CARBOHYDRATES: ' + image_pred[1]['CARBOHYDRATES'] sugar = 'SUGAR: ' + image_pred[1]['SUGAR']

fiber = 'FIBER: ' + image_pred[1]['FIBER']

protein = 'PROTEIN: ' + image_pred[1]['PROTEIN'] sodium = 'SODIUM: ' + image_pred[1]['SODIUM']

cholesterol = 'CHOLESTEROL: ' + image_pred[1]['CHOLESTEROL'] potassium = 'POTASSIUM: ' + image_pred[1]['POTASSIUM'] output = 'OUTPUT: ' + image_pred[1]['OUTPUT']

return render_template("imageprediction.html", value=img_filepath, pred=image_pred[0],

fruit=fruit, serving_size=serving_size, energy=energy, fat=fat, saturated_fat=saturated_fat,

mono_unsaturated_fat=mono_unsaturated_fat, poly_unsaturated_fat=poly_unsaturated_fat,

carbohydrates=carbohydrates, sugar=sugar, fiber=fiber, protein=protein, sodium=sodium,

cholesterol=cholesterol, potassium=potassium, output=output, flag=True)

def launch(img_filepath):

model = load_model('nutrition.h5')

img = load_img(img_filepath, target_size=(64, 64)) x = img_to_array(img)

x = np.expand_dims(x, axis=0) predict_x = model.predict(x) classes_x = np.argmax(predict_x)

index = ['Apple', 'Banana', 'Orange', 'Pineapple', 'Watermelon'] values = nutrition(index[classes_x])

return [index[classes_x], values]

def nutrition(x):

```
conn = sqlite3.connect('nutri.db')

cursor = conn.execute(f'''SELECT * FROM NUTRI WHERE FRUIT=="{x}"''')
for row in cursor:

rec =

{"FRUIT":row[0],"SERVING_SIZE":row[1],"ENERGY":row[2],"FAT":row[3
],"SATURATED FAT":row[4], "MONO_UNSATURATED_FAT":row[5],
"POLY_UNSATURATED_FAT":row[6],"CARBOHYDRATES":row[7],"SUG
AR":row[8],"FIBER":ro
w[9],"PROTEIN":row[10],"SODIUM":row[11],"CHOLESTEROL":row[12],"P
OTASSIUM":row[13

],"OUTPUT":row[14]}

return rec



if     name     == " main ":

          app.run(debug=False)
```