TEAM ID: PNT2022TMID46445

BUILD PYTHON CODE

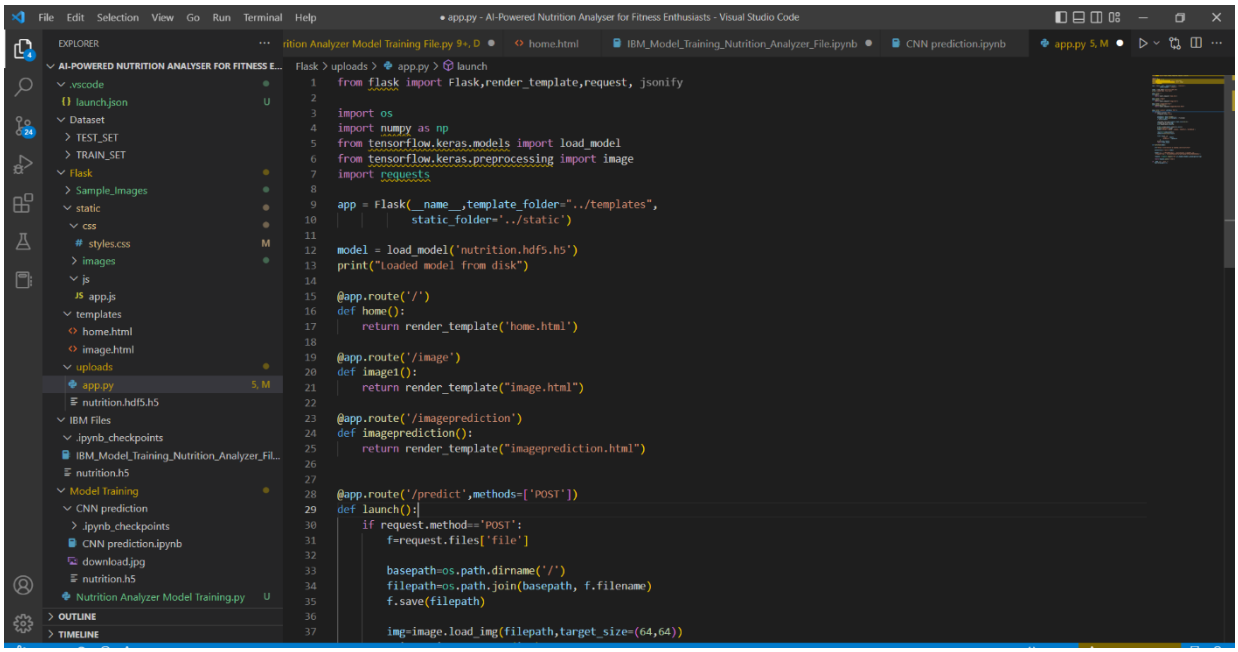
Importing Libraries

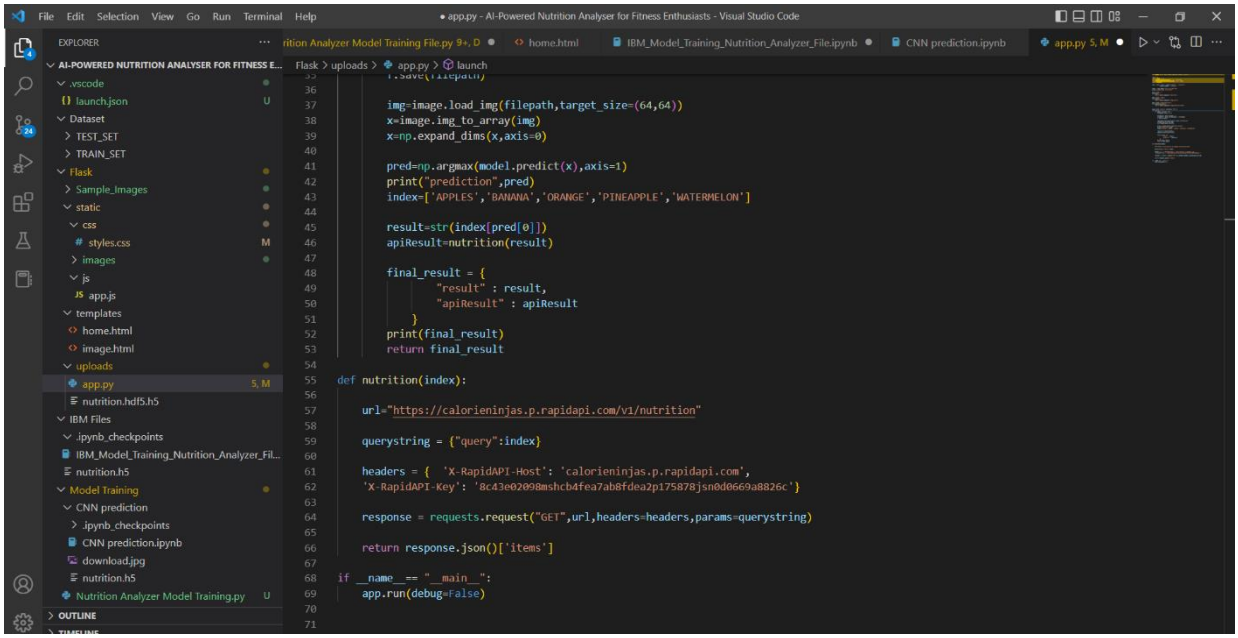
The first step is usually importing the libraries that will be needed in the program.

```
from flask import Flask,render_template,request
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
import os
import numpy as np #used for numerical analysis
from tensorflow.keras.models import load_model#to load our trained model
from tensorflow.keras.preprocessing import image
import requests
```

Importing the flask module into the project is mandatory. An object of the Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as an argument Pickle library to load the model file.

SCREENSHOTS:




File Edit Selection View Go Run Terminal Help
• app.py - AI-Powered Nutrition Analyser for Fitness Enthusiasts - Visual Studio Code
EXPLORER
AI-POWERED NUTRITION ANALYSER FOR FITNESS E...
.vscode
launch.json
Dataset
TEST_SET
TRAIN_SET
Flask
Sample_Images
static
css
styles.css
images
js
app.js
templates
home.html
image.html
uploads
app.py
nutrition.hdf5.h5
IBM Files
.ipynb_checkpoints
IBM_Model_Training_Nutrition_Analyzer_Fil...
nutrition.h5
Model Training
CNN prediction
.ipynb_checkpoints
CNN prediction.ipynb
download.jpg
nutrition.h5
Nutrition Analyzer Model Training.py
OUTLINE
TIMELINE
ition Analyzer Model Training File.py 9+, D
home.html
IBM_Model_Training_Nutrition_Analyzer_File.ipynb
CNN prediction.ipynb
app.py 5, M
Flask > uploads > app.py > launch
img=image.load_img(filepath,target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=np.argmax(model.predict(x),axis=1)
print("prediction",pred)
index=['APPLES','BANANA','ORANGE','PINEAPPLE','WATERMELON']
result=str(index[pred[0]])
apiResult=nutrition(result)
final_result = {
"result" : result,
"apiResult" : apiResult
}
print(final_result)
return final_result
def nutrition(index):
url="https://calorieninjas.p.rapidapi.com/v1/nutrition"
querystring = {"query":index}
headers = { 'X-RapidAPI-Host': 'calorieninjas.p.rapidapi.com',
'X-RapidAPI-Key': '8c43e02098mshcb4fea7ab8fdea2p175878jsn0d0669a8826c'}
response = requests.request("GET",url,headers=headers,params=querystring)
return response.json()['items']
if __name__== "__main__":
app.run(debug=False)