

TEAM ID : PNT2022TMID25018

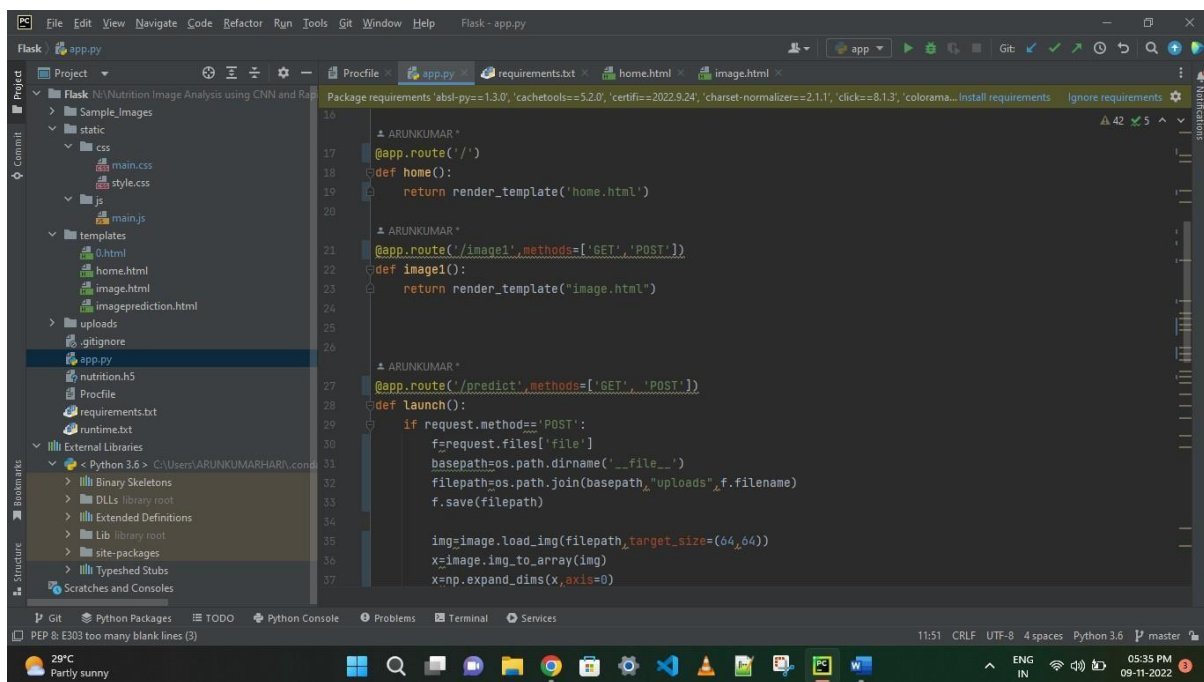
PROJECT NAME : AI-powered Nutrition Analyzer for Fitness Enthusiasts

Routing To The Html Page

Here, the declared constructor is used to route to the HTML page created earlier.

In the above example, the '/' URL is bound with the home.html function. Hence, when the home page of the webserver is opened in the browser, the HTML page is rendered. Whenever you enter the values from the HTML page the values can be retrieved using the POST Method.

Here, "home.html" is rendered when the home button is clicked on the UI

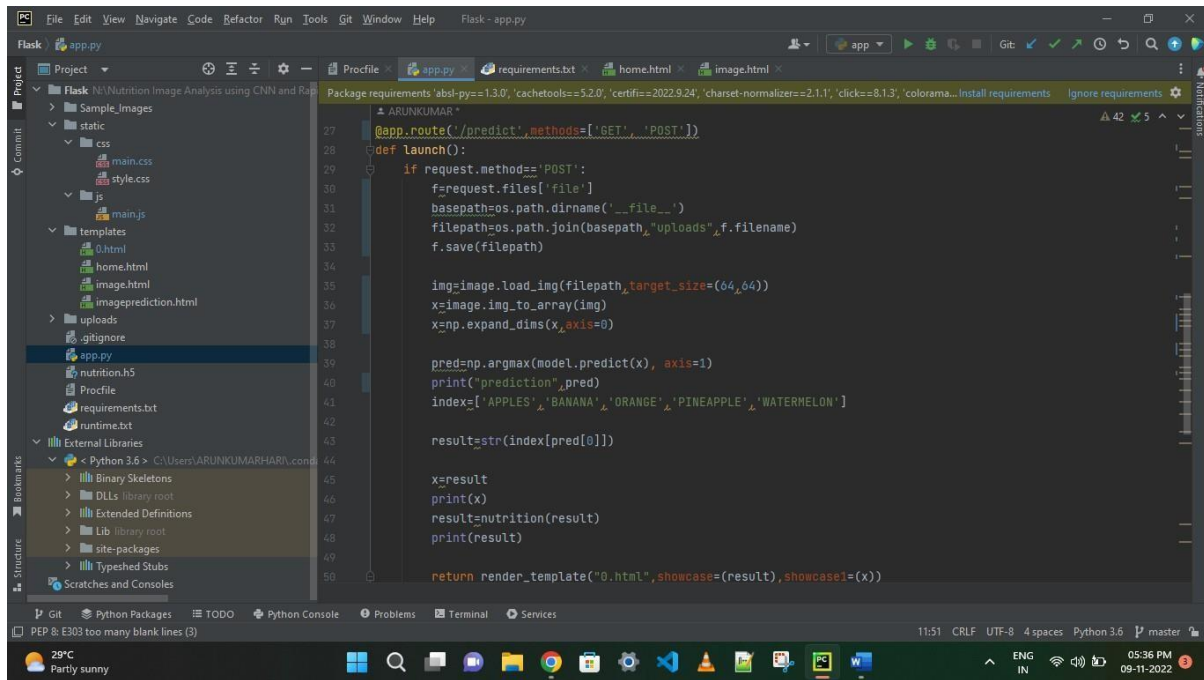
A screenshot of a code editor window titled 'Flask - app.py'. The editor shows a Python file with Flask routing logic. The left sidebar displays a project structure with folders like 'static', 'templates', and 'uploads'. The main code area contains the following Python code:

```
16 from flask import Flask, render_template, request, jsonify
17
18 app = Flask(__name__)
19
20 @app.route('/')
21 def home():
22     return render_template("home.html")
23
24 @app.route('/image', methods=['GET', 'POST'])
25 def image():
26     return render_template("image.html")
27
28 @app.route('/predict', methods=['GET', 'POST'])
29 def launch():
30     if request.method == 'POST':
31         f = request.files['file']
32         basepath = os.path.dirname(__file__)
33         filepath = os.path.join(basepath, "uploads", f.filename)
34         f.save(filepath)
35
36         img = image.load_img(filepath, target_size=(64, 64))
37         x = image.img_to_array(img)
38         x = np.expand_dims(x, axis=0)
```

The status bar at the bottom indicates the file is 'PEP 8: E303 too many blank lines (3)', the time is 11:51, and the editor is using Python 3.6.

When "image is uploaded "on the UI, the launch function is executed

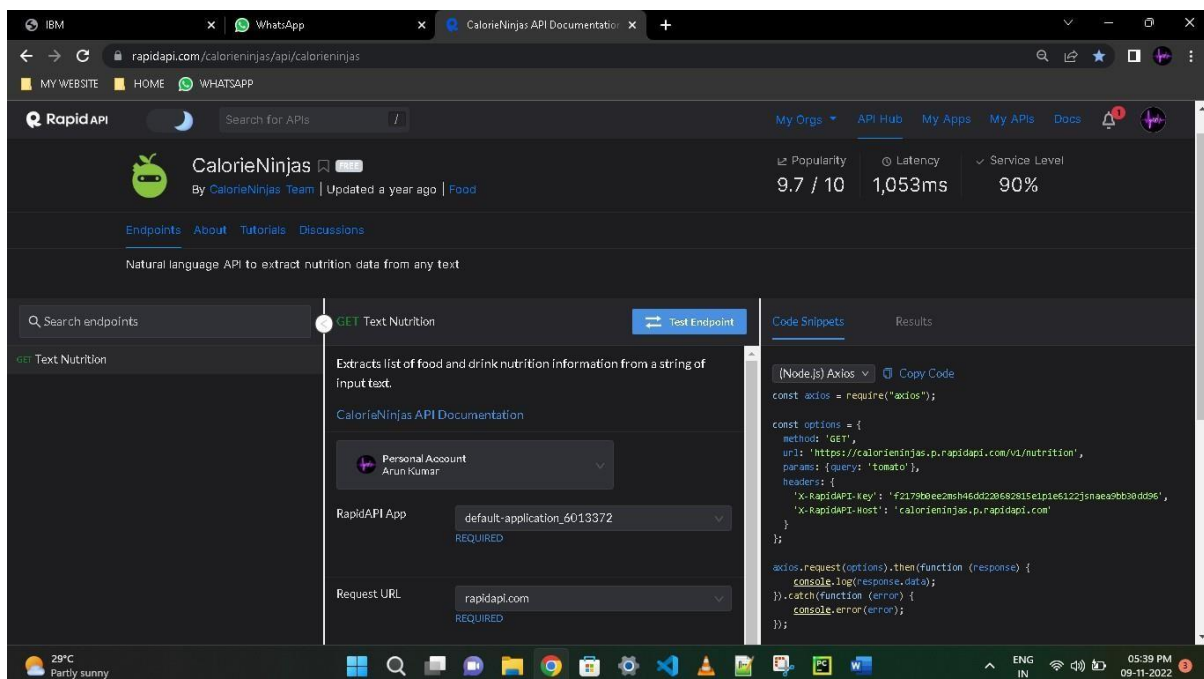
It will take the image request and we will be storing that image in our local system then we will convert the image into our required size and finally, we will be predicting the results with the help of our model which we trained and depending upon the class identified we will showcase the class name and its properties by rendering the respective html pages.



API Integration:

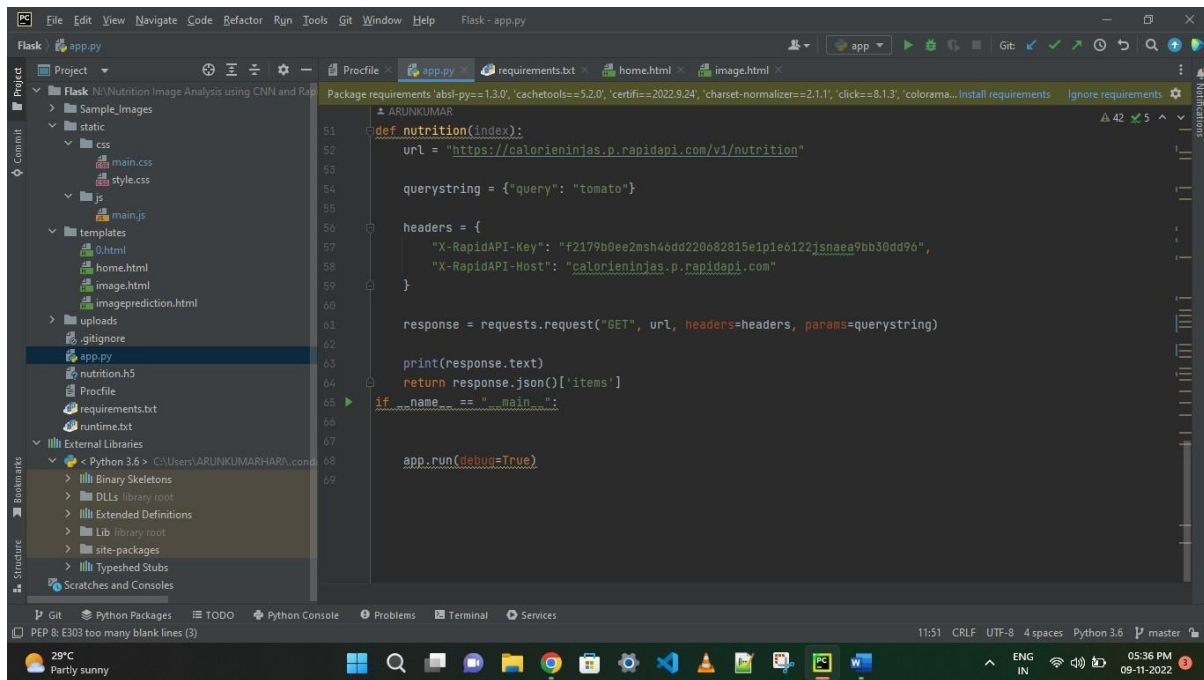
Here we will be using Rapid API

Using RapidAPI, developers can search and test the APIs, subscribe, and connect to the APIs — all with a single account, single API key and single SDK. Engineering teams also use RapidAPI to share internal APIs and microservice documentation.



The link above will allow us to test the food item and will result the nutrition content present in the food item.

NOTE: When we keep hitting the API the limit of it might expire. So making a smart use of it will be an efficient way.



```
File Edit View Navigate Code Refactor Run Tools Git Window Help Flask - app.py
Project
Flask N:\Nutrition Image Analysis using CNN and Rap
Sample_Images
static
css
main.css
style.css
js
main.js
templates
0.html
home.html
image.html
imageprediction.html
uploads
.gitignore
app.py
nutrition.hs
Profile
requirements.txt
runtime.txt
External Libraries
Python 3.6 > C:\Users\ARUNKUMARHARI\cond
Binary Skeletons
DLLs library root
Extended Definitions
Lib library root
site-packages
Typestbed Stubs
Scratches and Consoles
PEP 8: E303 too many blank lines (3)
11:51 CRLF UTF-8 4 spaces Python 3.6 master
```

```
def nutrition(index):
    url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"

    querystring = {"query": "tomato"}

    headers = {
        "X-RapidAPI-Key": "f2179b0ee2msh46dd226e82815e1p1e6122jsnaea9bb36dd96",
        "X-RapidAPI-Host": "calorieninjas.p.rapidapi.com"
    }

    response = requests.request("GET", url, headers=headers, params=querystring)

    print(response.text)
    return response.json()['items']

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

Finally, Run the application

This is used to run the application in a localhost. The local host runs on port number 5000.(We can give different port numbers)