```
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


App = Flask(__name__,template_folder="templates") # initializing a flask app
# Loading the model
Model=load_model('nutrition.h5')
Print("Loaded model from disk")


@app.route('/')# route to display the home page
Def home():
    Return render_template('home.html')#rendering the home page

@app.route('/image1',methods=['GET','POST'])# routes to the index html
Def image1():
    Return render_template("image.html")


@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
Def launch():
    If request.method=='POST':
```

```
    F=request.files['file'] #requesting the file
    Basepath=os.path.dirname('__file__')#storing the file directory
    Filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
    f.save(filepath)#saving the file

    img=image.load_img(filepath,target_size=(64,64)) #load and reshaping the image
    x=image.img_to_array(img)#converting image to an array
    x=np.expand_dims(x,axis=0)#changing the dimensions of the image

    pred=np.argmax(model.predict(x), axis=1)
    print("prediction",pred)#printing the prediction
    index=['APPLES','BANANA','ORANGE','PINEAPPLE','WATERMELON']

    result=str(index[pred[0]])

    x=result
    print(x)
    result=nutrition(result)
    print(result)

    return render_template("0.html",showcase=(result),showcase1=(x))
def nutrition(index):


  url = https://calorieninjas.p.rapidapi.com/v1/nutrition

  querystring = {"query":index}

  headers = {
```

```
        'x-rapidapi-key': "5d797ab107mshe668f26bd044e64p1ffd34jsnf47bfa9a8ee4",
        '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__":
   # running the app
    App.run(debug=False)
```