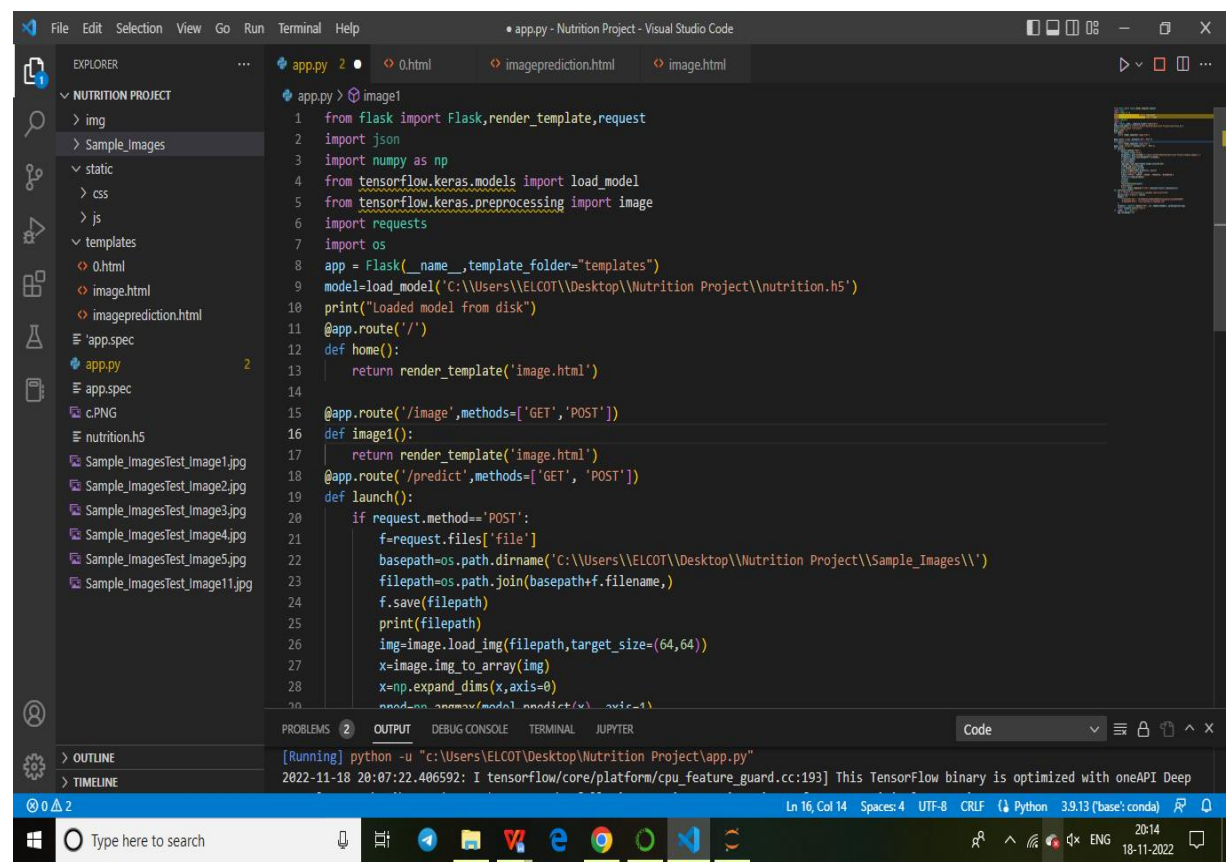


## RUN THE APPLICATION

**TEAM ID** : PNT2022TMID10679

**PROJECT NAME** : AI-Powered Nutrition Analyzer for fitness enthusiasts

## RUNNING THE APPLICATION

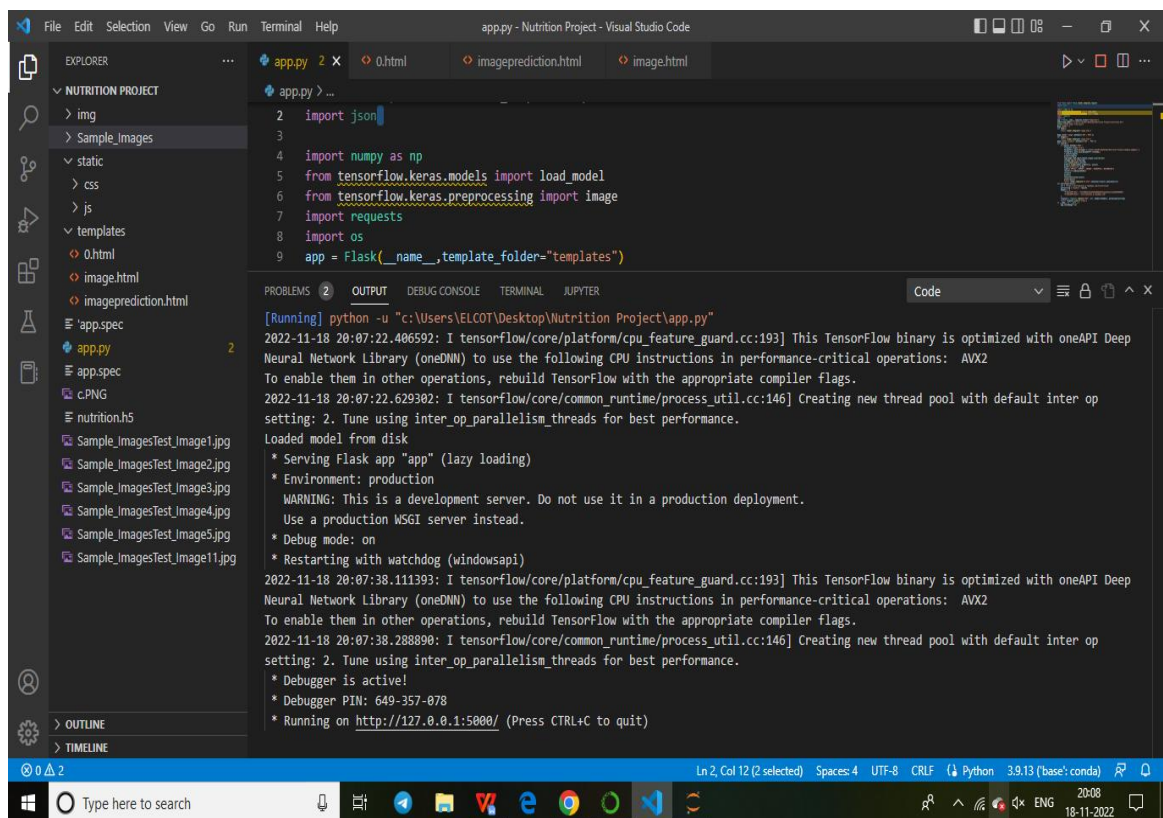
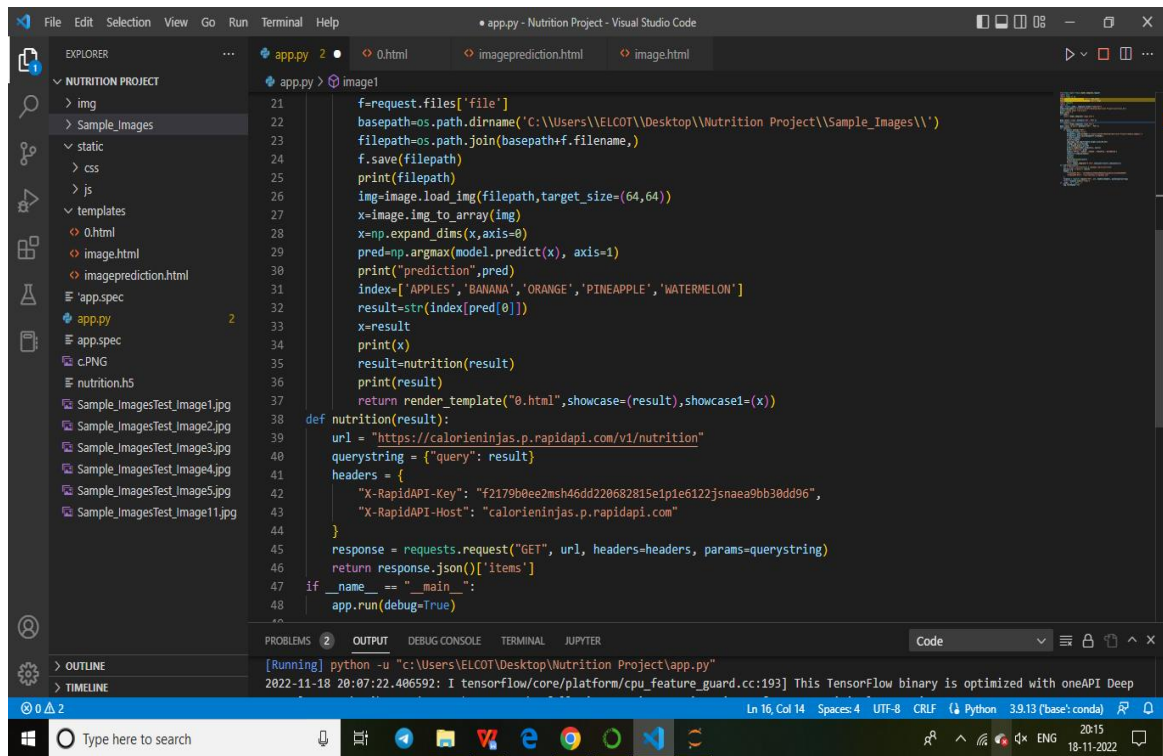


The screenshot displays the Visual Studio Code interface for a project named 'Nutrition Project'. The Explorer sidebar on the left shows the project structure, including folders for 'img', 'static', 'css', 'js', and 'templates', along with files like 'app.py', 'app.spec', 'nutrition.hs', and several sample images. The main editor window shows the 'app.py' file, which is a Flask application. The code includes imports for Flask, json, numpy, TensorFlow Keras models, TensorFlow Keras preprocessing, requests, and os. It defines a Flask app, loads a model from a local file, and sets up routes for a home page, an image upload endpoint, and a prediction endpoint. The prediction endpoint uses the loaded model to process uploaded images. The bottom status bar indicates the application is running, showing the command 'python -u "c:\Users\ELCOT\Desktop\Nutrition Project\app.py"' and the current file path 'Ln 16, Col 14'.

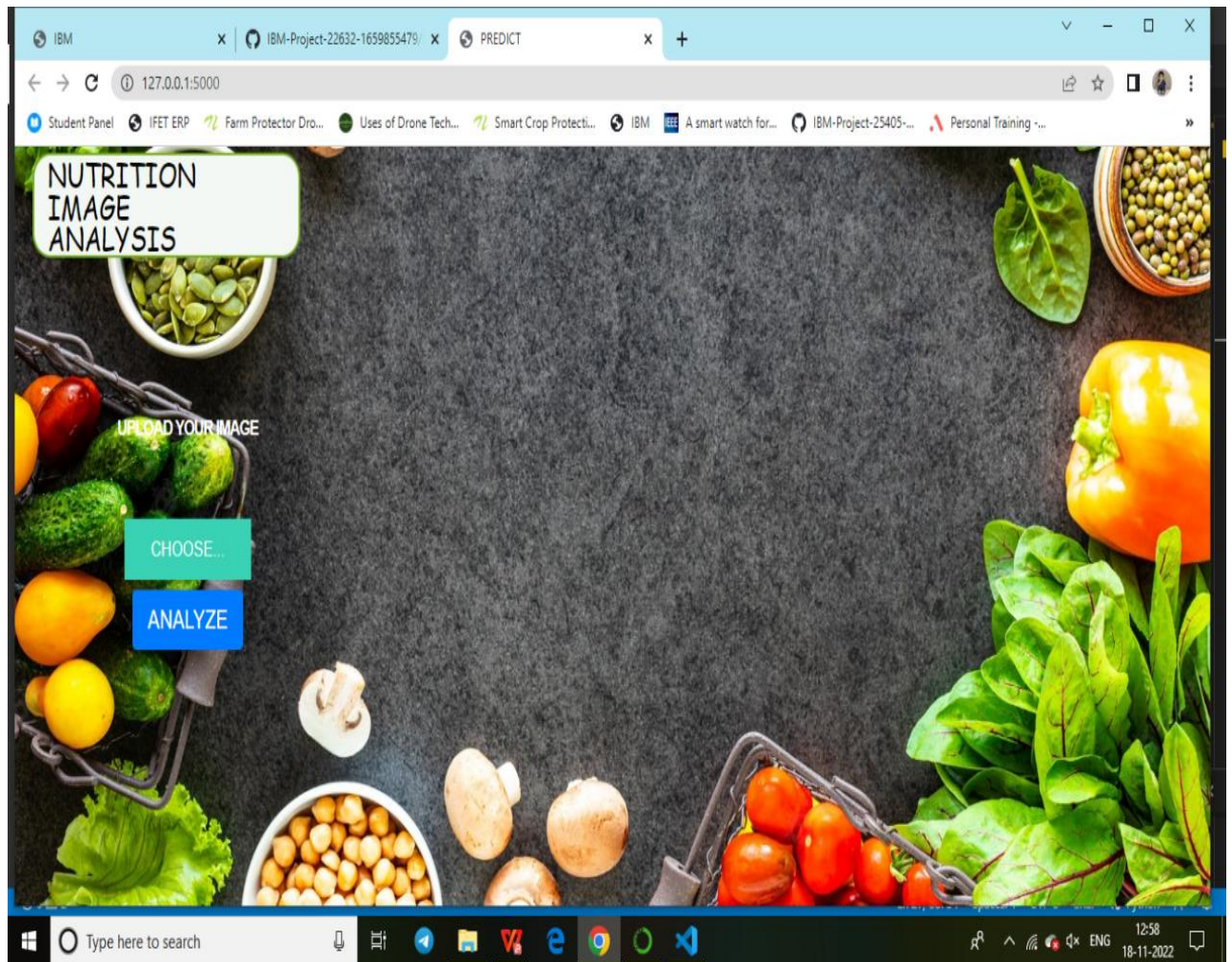
```
app.py > image1
1 from flask import Flask,render_template,request
2 import json
3 import numpy as np
4 from tensorflow.keras.models import load_model
5 from tensorflow.keras.preprocessing import image
6 import requests
7 import os
8 app = Flask(__name__,template_folder="templates")
9 model=load_model('C:\\Users\\ELCOT\\Desktop\\Nutrition Project\\nutrition.hs')
10 print("Loaded model from disk")
11 @app.route('/')
12 def home():
13     return render_template('image.html')
14
15 @app.route('/image',methods=['GET','POST'])
16 def image1():
17     return render_template('image.html')
18 @app.route('/predict',methods=['GET','POST'])
19 def launch():
20     if request.method=="POST":
21         f=request.files['file']
22         basepath=os.path.dirname('C:\\Users\\ELCOT\\Desktop\\Nutrition Project\\Sample_Images\\')
23         filepath=os.path.join(basepath+f.filename,)
24         f.save(filepath)
25         print(filepath)
26         img=image.load_img(filepath,target_size=(64,64))
27         x=image.img_to_array(img)
28         x=np.expand_dims(x,axis=0)
29         #pred=on.onames(model.predict(x))
30         #pred=on.onames(model.predict(x))
```

[Running] python -u "c:\Users\ELCOT\Desktop\Nutrition Project\app.py"

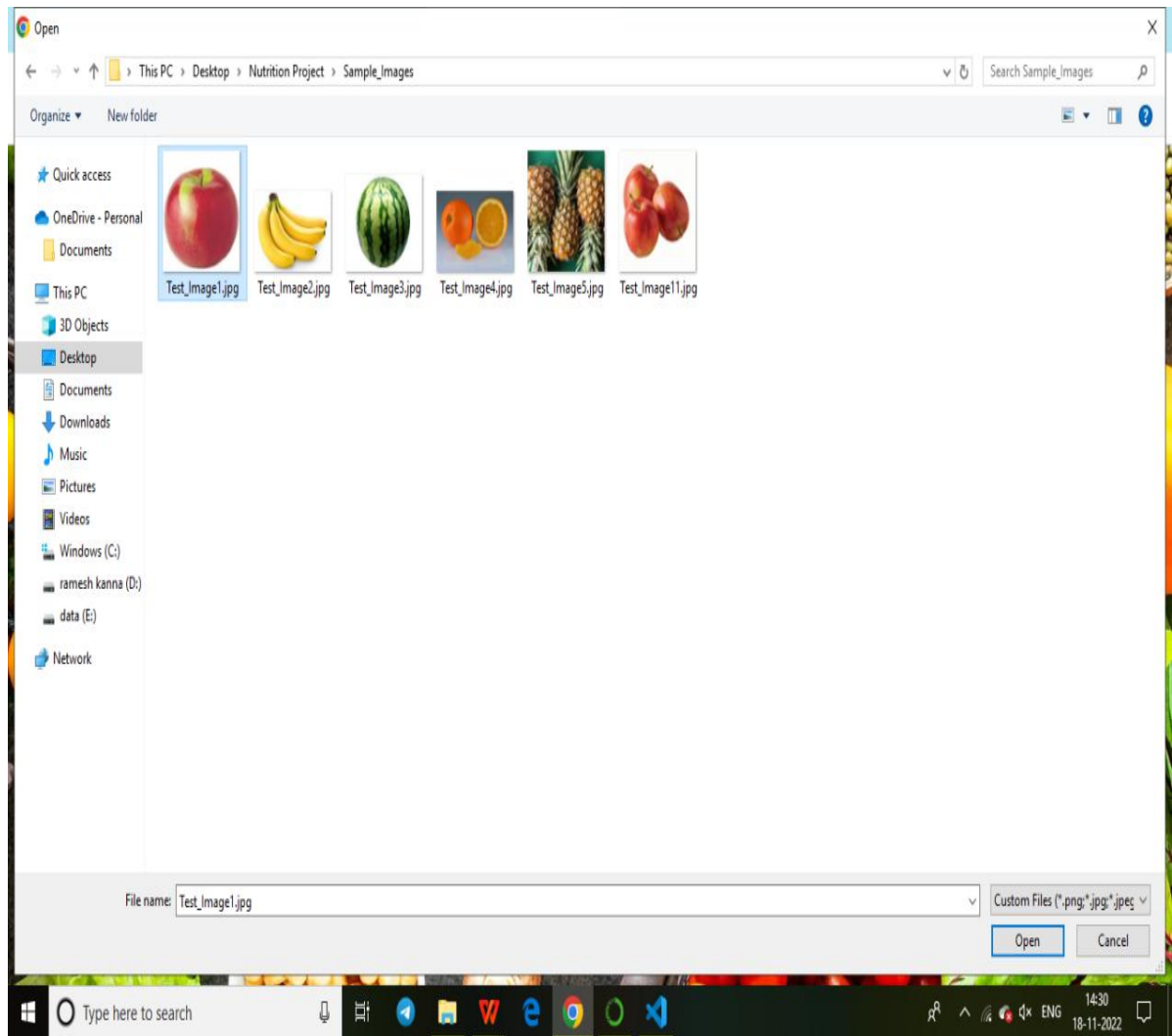
2022-11-18 20:07:22.406592: I tensorflow/core/platform/cpu\_feature\_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep

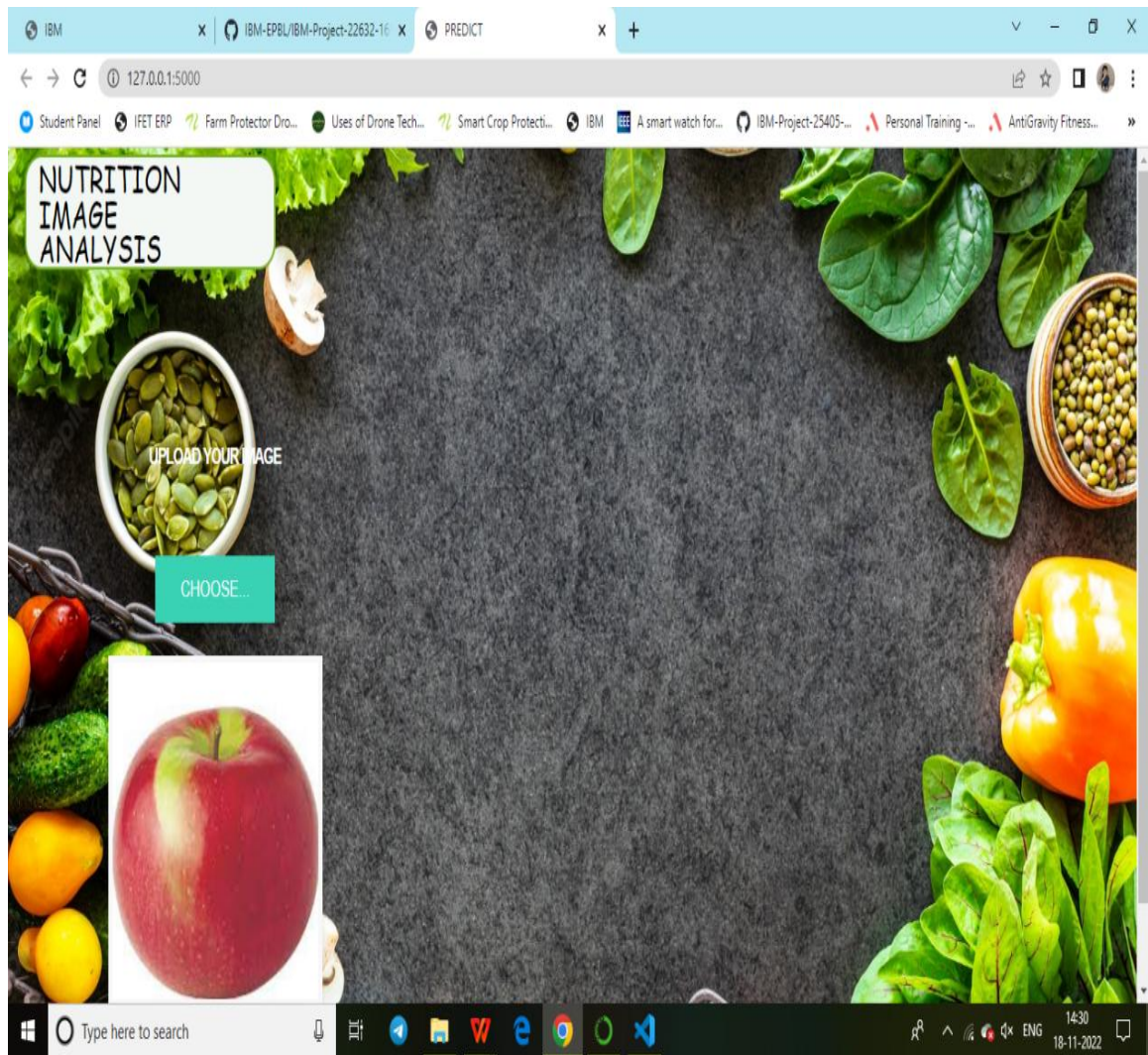


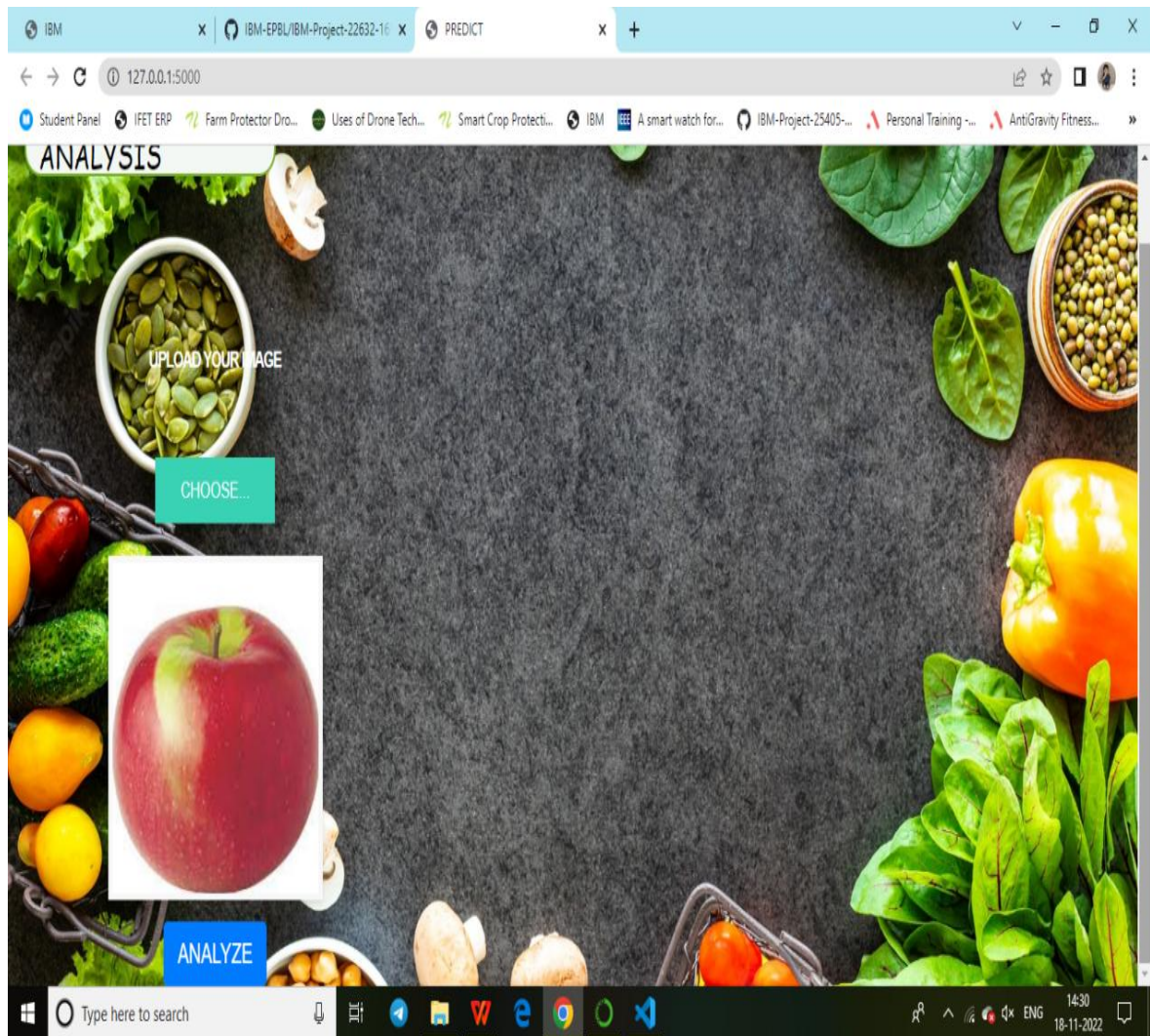
## OUTPUT SCREENSHOTS













CHOOSE...



## APPLES

[{'sugar g': 10.3, 'fiber g': 2.4, 'serving size g': 100.0,  
'sodium mg': 1, 'name': 'apples', 'potassium mg': 11,  
'fat saturated g': 0.0, 'fat total g': 0.2, 'calories': 53.4,  
'cholesterol mg': 0, 'protein g': 0.3, 'carbohydrates total g':  
13.8}]

[{'sugar g': 10.3, 'fiber g': 2.4, 'serving size g': 100.0,  
'sodium mg': 1, 'name': 'apples', 'potassium mg': 11, 'fat saturated g': 0.0,  
'fat total g': 0.2, 'calories': 53.4, 'cholesterol mg': 0, 'protein g': 0.3,  
'carbohydrates total g': 13.8}]