

## Application Building

|              |  |
|--------------|--|
| Team ID      | PNT2022TMID30319   |
| Project Name | Fertilizers Recommendation System For Disease Prediction |

### Activity 1:

#### Build Python Code:

The backend code using python and flask framework used. It will integrate the cnn model to the frontend.

#### Step1:

Import the libraries

```
#Import necessary libraries
from flask import Flask, render_template, request, url_for

import numpy as np
import os
import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from werkzeug.utils import secure_filename
import pandas as pd
from PIL import Image
```

#### Step2:

Initialize the flask app and load model

```
model = load_model(r"Uploads\Vegetable .h5")
model1 = load_model(r"Uploads\fruitdata.h5")
print(model)

print("Model Loaded Successfully")
```

```
# Create flask instance
app = Flask(__name__)
```

Step3:

Configure the pages

```
@app.route("/")
def home():
    return render_template('home.html')

@app.route("/predict2", methods=['GET', 'POST'])
def predict2():
    return render_template('predict2.html')

# render index.html page
@app.route("/predict1", methods=['GET', 'POST'])
def predict1():
    return render_template('predict1.html')
```

Step4:

Preprocess the Frame run

```
@app.route("/predict", methods = ['GET', 'POST'])
def predict():
    if request.method == 'POST':
        file = request.files['image1'] # fet input
        filename = secure_filename(file.filename)
        basepath = os.path.abspath(os.path.dirname(__file__))
        file_path = os.path.join(basepath, r'static\upload', filename)
        img_path = os.path.join(r'static\upload', filename)
        file.save(file_path)
        print(filename)
        Prediction_image, output=leaves(Plant_image=file_path)
        return
    render_template('predict1.html', pred_output=Prediction_image, Disease=output, value
    =img_path, flag=True)

def leaves(Plant_image):
    img=image.load_img(Plant_image, target_size=(128,128))
    x=image.img_to_array(img)
    x=np.expand_dims(x, axis=0)
```

```

plant = request.form.get('Plant')
print(plant)
if(plant == "vegetable"):
    prediction=np.argmax(model.predict(x),axis=1)
    print(prediction)
    index=['Pepper,_bell__Bacterial_spot','Pepper,_bell__healthy','Pota
to__Early_blight','Potato__Late_blight','Potato__healthy','Tomato__Bacterial_
spot','Tomato__Late_blight','Tomato__Leaf_Mold','Tomato__Septoria_leaf_spot']
    print(index[prediction[0]])
    df=pd.read_excel(r'Uploads\precautions - veg.xlsx')
    print(df.iloc[prediction[0]]['caution'])

else:
    prediction=np.argmax(model1.predict(x),axis=1)
    index=['Apple__Black_rot','Apple__healthy','Corn_(maize)__Northern
_Leaf_Blight','Corn_(maize)__healthy','Peach__Bacterial_spot','Peach__healthy'
]

    print(index[prediction[0]])
    df=pd.read_excel(r'Uploads\precautions - fruits.xlsx')
    print(df.iloc[prediction[0]]['caution'])

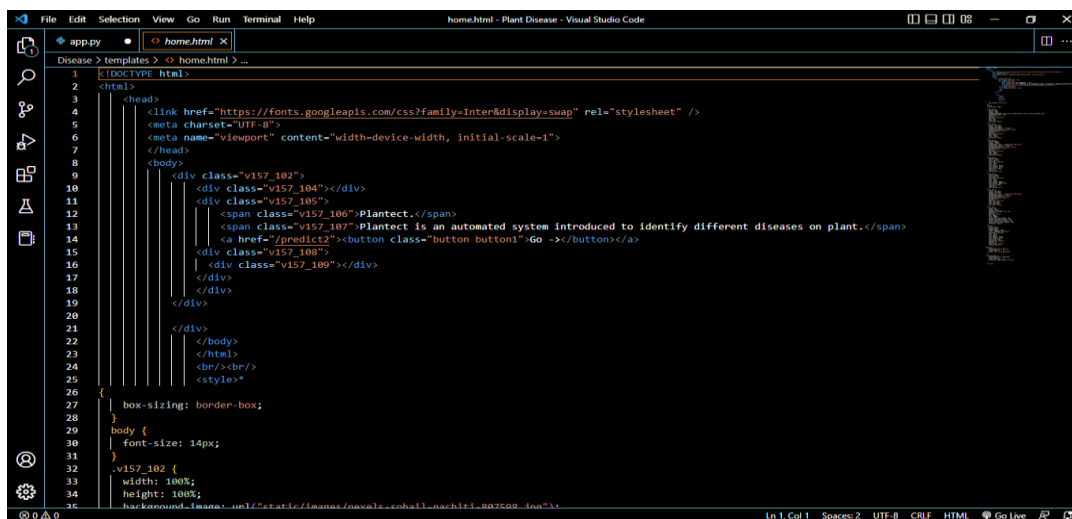
return df.iloc[prediction[0]]['caution'],index[prediction[0]]

if __name__ == "__main__":
    app.run(threaded=False,debug=True)

```

## Activity 2:

### Build HTML Pages



## Predict Pages

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <link href="https://fonts.googleapis.com/css?family=Inter&display=swap" rel="stylesheet" />
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <!-- Bootstrap CSS -->
8 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css" integrity="sha384-VCmXjyyRehH4PwouA" />
9 </head>
10 <body>
11
12 <div class="v195_2">
13 <div class="container1">
14 <div class="v195_6">
15 | 
16 </div>
17 <div class="v213_22"></div>
18 <span class="v214_23">{{Disease}}</span>
19 </div>
20 <div class="container">
21 <div class="v215_24">
22 | <span class="v215_27">Disease</span>
23 <div class="v195_4">{{Disease}}</div>
24 <div class="v215_37">
25 | 
26 </div>
27 </div>
28 <div class="v215_30">
29 <span class="v215_33">Fertilizer</span>
30 <span class="v215_34">
31 | {{pred_output}}
32 </span>
33 
34 </div>
35 </div>

```

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <link href="F:\Visual\Plantect\image 12.png" rel="stylesheet" />
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css" integrity="sha384-VCmXjywRehH4PwovA" />
8
9 </head>
10 <body>
11 <div class="class">
12 <div class="Follow link (ctrl + click)">
13 
14 <span class="v157_76">Detect if your plant is infected/</span>
15 <span class="v157_77">Check whether your plant is affected by any diseases.</span>
16 </div>
17 <div class="box1">
18 <div class="Upload_button">
19 <form action="/predict" method="post" enctype="multipart/form-data">
20 <div class="leaf">
21 <label for="disease" style="font-size: 20px;">Choose Leaf:</label>
22 <select name="Plant" style="font-size: 20px;">
23 <option value="fruit">Fruit</option>
24 <option value="vegetable">Vegetable</option>
25 </select>
26 </div>
27 <br><br>
28 <input class="upload" type="file" name="image1" accept="Image/*" style="width: max-content; font-size: 18px;" />
29 </div>
30 <br><br>
31 <a href="/predict">
32 <button class="button">Predict</button>
33 </a>
34 </div>
35 <span class="v157_81">v/cnans

```

### Run the Code:

