

```
import os

os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

import requests

from keras.preprocessing import image

from keras.models import load_model

import numpy as np

import pandas as pd

import tensorflow as tf

from tensorflow.python.keras.backend import set_session

from werkzeug.utils import secure_filename

from flask import Flask, redirect, render_template, request, url_for


app= Flask(__name__)

model1 = load_model('fruit.h5')

model = load_model('vegetable.h5')


@app.route('/')

def home():

    return render_template('home.html')

@app.route('/Predict')

def prediction():

    return render_template('predict.html')


@app.route('/Prediction',methods=['GET','POST'])

def upload():

    if request.method == 'POST':

        f= request.files['images']
```

```

basepath = os.path.dirname(__file__)

file_path=os.path.join(basepath, 'uploads',secure_filename(f.filename))

f.save(file_path)

print("file save")

img = image.load_img(file_path, target_size=(128,128))

x=image.img_to_array(img)

print("image to gray")

x=np.expand_dims(x, axis=0)

plant=request.form['plant']

if (plant=="fruit"):

    model1.predict_classess(x)

    print(preds)

    df=pd.read_excel('precautions - fruits.xlsx')

    print (df.iloc[preds[0]]['cautions'])

else:

    preds=model.predict_classes(x)

    df=pd.read_excel('precautions - veg.xlsx')

    print(df.iloc[preds[0]]['caution'])

    return df.iloc[preds[0]]['caution']


if __name__=="__main__":

    app.run(debug=True)

```