



INSURANCE FRAUD DETECTION USING

# MACHINE LEARNING

Project Hand-out, Faculty Development Program – NaanMudhalvan

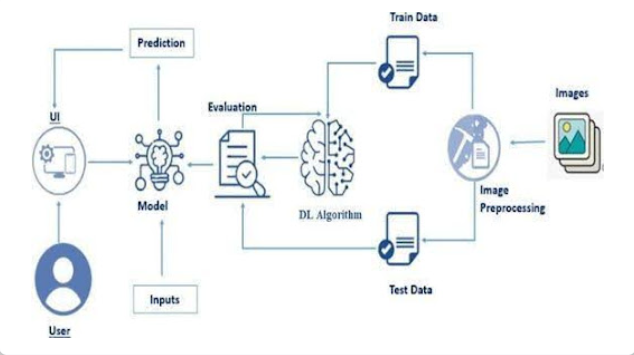
SmartInternz

www.smartinternz.com

**Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables**

Smart Sorting is an innovative project focused on enhancing the precision and efficiency of detecting rotten fruits and vegetables using cutting-edge transfer learning techniques. By leveraging pre-trained deep learning models and adapting them to specific datasets of fruits and vegetables, this project aims to revolutionize the process of sorting and quality control in the agricultural and food industry.

**Technical Architecture:**



**Project Flow:**

* The user interacts with the UI (User Interface) to choose the image.
* The chosen image is analyzed by the model which is integrated with the flask application.
* Once the model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

* Data Collection: Collect or download the dataset that you want to train.
* Data pre-processing
  + Data Augmentation
  + Splitting data into train and test
* Model building
  + Import the model-building libraries
  + Initializing the model
  + Training and testing the model
  + Evaluating the performance of the model
  + Save the model
* Application Building
  + Create an HTML file
  + Build python code

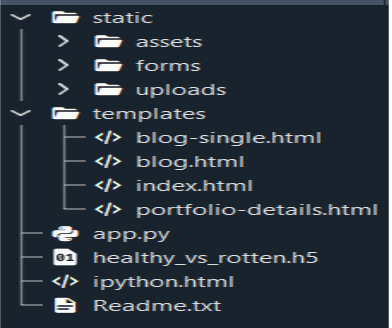
**Prior Knowledge:**

You must have prior knowledge of the following topics to complete this project.

* DL Concepts
  + Neural Networks:: [https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/](https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/%20)
* Deep Learning Frameworks:: [https://www.knowledgehut.com/blog/data-science/pytorch-vs-tensorflow](about:blank)
* Transfer Learning: [https://towardsdatascience.com/a-demonstration-of-transfer-learning-of-vgg-convolutional-neural-network-pre-trained-model-with-c9f5b8b1ab0a](https://towardsdatascience.com/a-demonstration-of-transfer-learning-of-vgg-convolutional-neural-network-pre-trained-model-with-c9f5b8b1ab0a%20)
* VGG16: https://www.geeksforgeeks.org/vgg-16-cnn-model/
* Convolutional Neural Networks (CNNs): [https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/ s://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning](https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/%20s:/www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning)
* Overfitting and Regularization: [https://www.analyticsvidhya.com/blog/2021/07/prevent-overfitting-using-regularization-techniques/](https://www.analyticsvidhya.com/blog/2021/07/prevent-overfitting-using-regularization-techniques/%20)
* Optimizers: [https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/](https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/%20ttps:/www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/)
* Flask Basics: <https://www.youtube.com/watch?v=lj4I_CvBnt0>

**Project Structure:**

Create the Project folder which contains files as shown below



### We are building a Flask application with HTML pages stored in the templates folder and a Python script app.py for scripting.

### Healthy\_vs\_rotten.h5 is our saved model. Further, we will use this model for flask integration.

### Milestone 1: Define Problem / Problem Understanding

**Activity 1: Specify the business problem**

Refer [Problem Definition](https://github.com/vijaay10/Smart-Sorting-Transfer-Learning-For-Identifying-Rotten-Fruits-And-Vegetables/tree/main/2.%20Project%20Initialization%20and%20Planning%20Phase)

#### Activity 2: Business requirements

A drug classification project can have a variety of business requirements, depending on the specific goals and objectives of the project. Some potential requirements may include:

∙**Accurate Detection:** The system should correctly identify:

* Rotten fruits or vegetables
* Fresh fruits or vegetables
* The type of fruit or vegetable (e.g., apple, tomato, potato, etc.)

∙ **Flexibility:** The system should be able to work with different types of fruits and vegetables without complicated setup or major changes.

∙ **Compliance:** The sorting system should follow food safety standards and hygiene rules to ensure safe handling of produce.

∙ **Easy to Use:** The system should be simple to operate, with clear instructions and easy-to-understand controls for the workers.

**Activity 4: Social or Business Impact.**

**Social Impact:-** Improved food quality and safetyby accurately detecting rotten fruits and vegetables, the sorting project ensures that only fresh produce reaches consumers. This leads to improved food quality, better health outcomes, and reduced chances of contaminated or spoiled food being sold in markets.

**Business Model/Impact :-** Reduced waste and increased efficiency by automatically identifying rotten items and sorting produce by type, the project helps reduce manual labor, minimize waste, and improve overall efficiency. This can lead to cost savings, higher productivity, and better profit margins for businesses involved in food processing or retail.

### Milestone 2: Data Collection & Preparation

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

#### Activity 1: Collect the dataset

#### It is the most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

#### Activity 1: Download the dataset

#### There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

#### In this project, we have used 28 classes of fruits and vegetables data. This data is downloaded from kaggle.com or can be connected by using API. Please refer to the link given below to download the dataset.

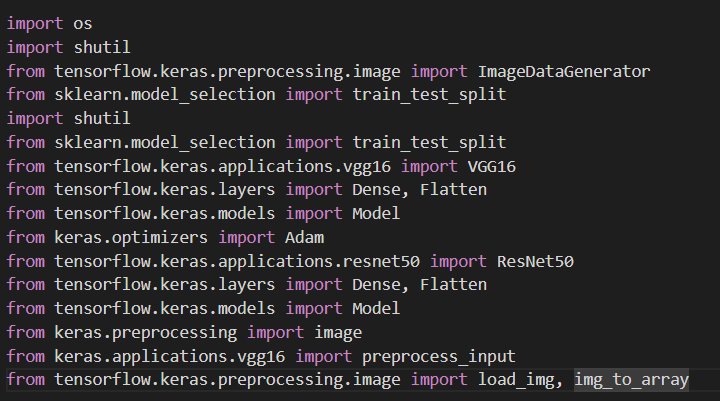
#### Link: Dataset

#### As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analyzing techniques.

#### Note: There are several techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

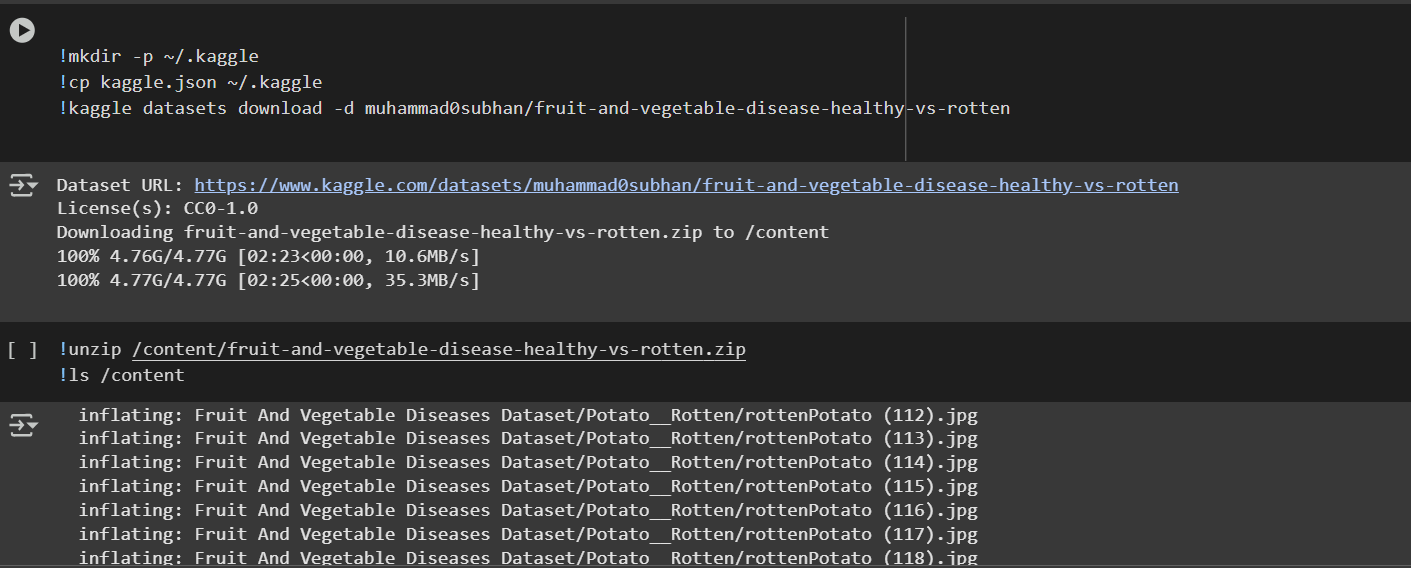
#### Activity 1.1: Importing the libraries

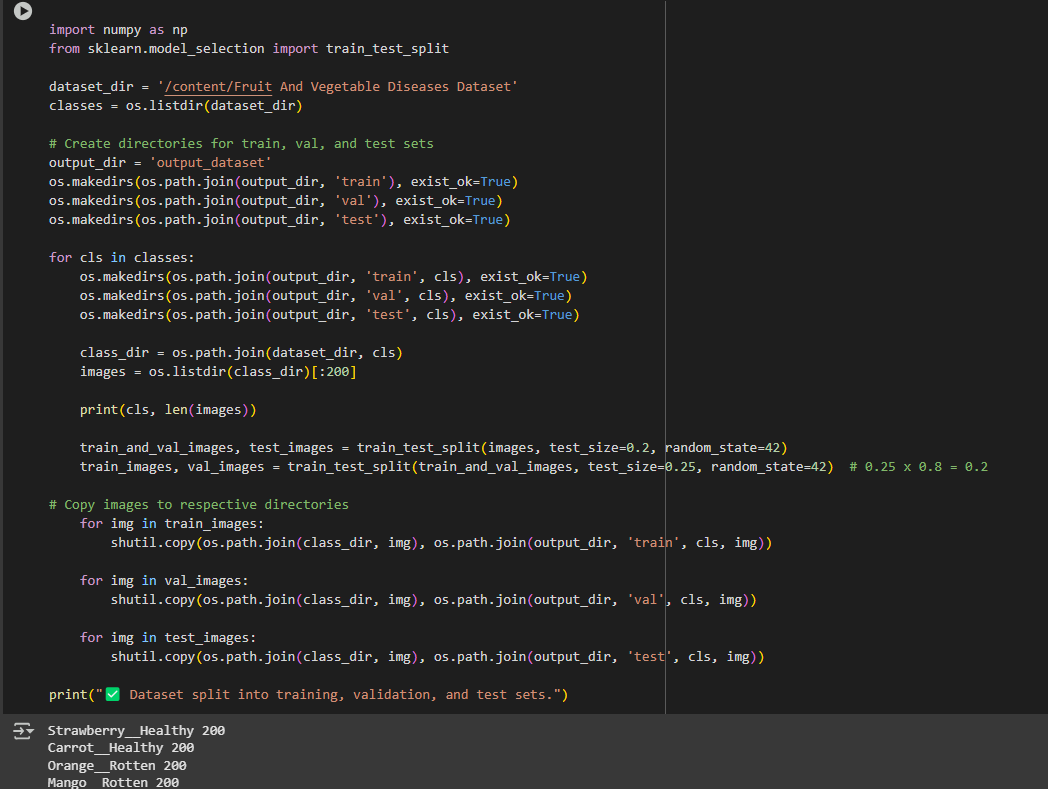
Import the necessary libraries as shown in the image.

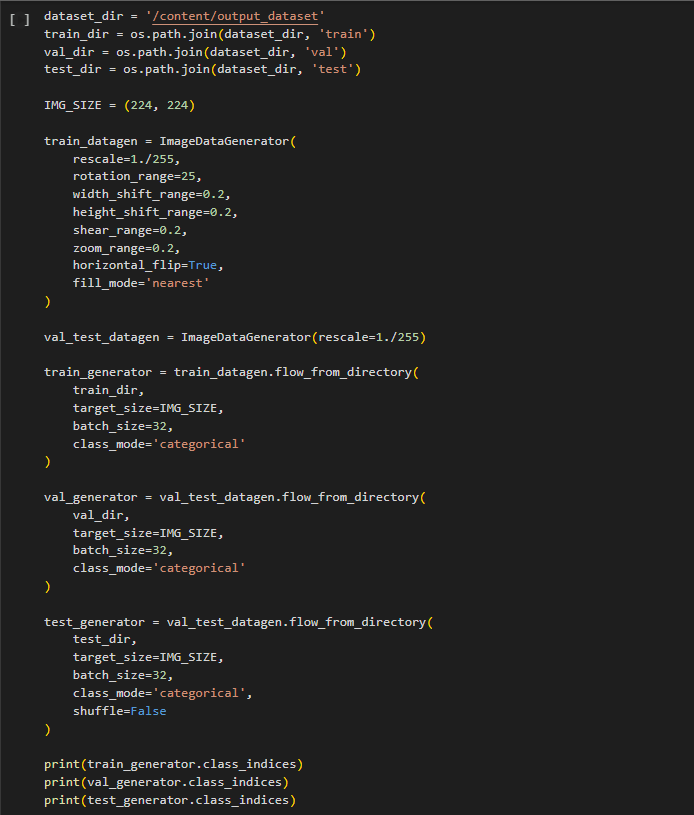


#### Activity 1.2: Read the Dataset

Our dataset format might be in .csv, excel files, .txt, .json, or zip files, etc. We can read the dataset with the help of pandas.

At first, unzip the data and convert it into a pandas data frame. 

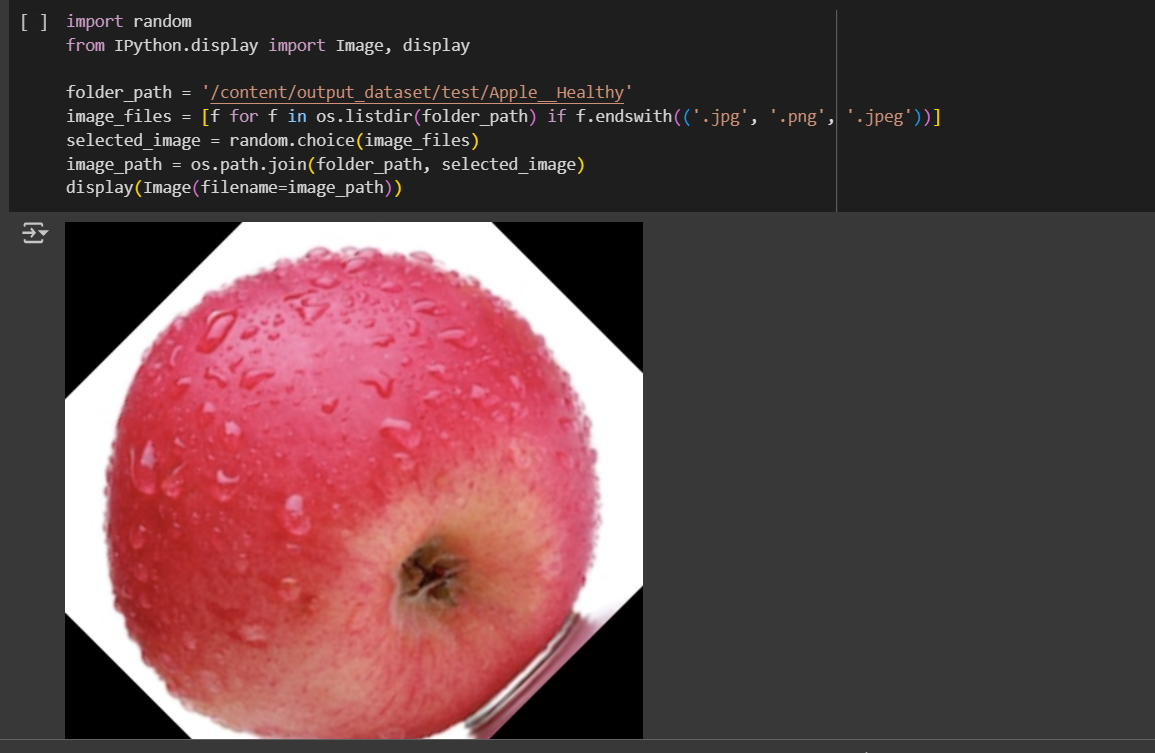




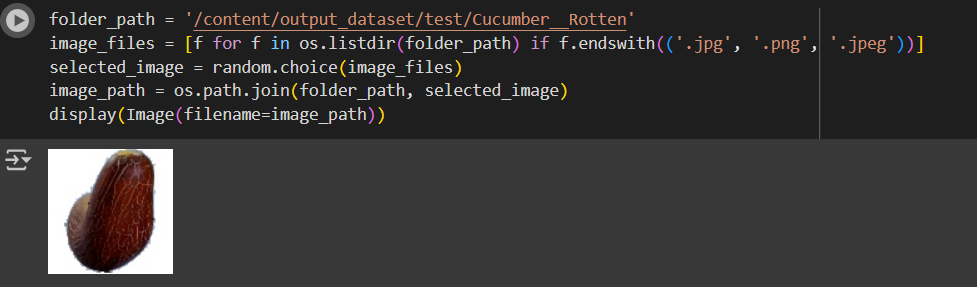
### Milestone 2: Exploratory Data Analysis

**Activity 2.1: Data Visualization**

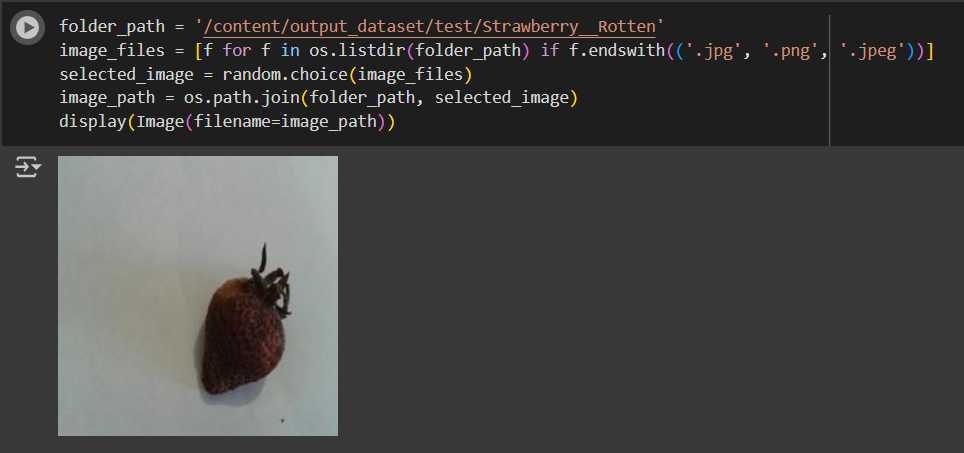
The provided Python code imports necessary libraries and modules for image manipulation. It selects a random image file from a specified folder path. Then, it displays the randomly selected image using IPython's Image module. This code is useful for showcasing random images from a directory for various purposes like data exploration or testing image processing algorithms.



In the above code, I used class Apple\_healthy 0 for prediction, This code randomly selects an image file from a specified folder (folder\_path) containing JPEG, PNG, or JPEG files, and then displays the selected image using IPython's display function. It utilizes Python's OS and random modules for file manipulation and random selection, respectively. And It has predicted correctly as Apple\_healthy 0.



In the above code, I used class Apple\_healthy 0 for prediction, This code randomly selects an image file from a specified folder (folder\_path) containing JPEG, PNG, or JPEG files, and then displays the selected image using IPython's display function. It utilizes Python's OS and random modules for file manipulation and random selection, respectively. And It has predicted correctly as Apple\_healthy 0.



In the above code, I used class strawberry\_rotten for prediction, This code randomly selects an image file from a specified folder (folder\_path) containing JPEG, PNG, or JPEG files, and then displays the selected image using IPython's display function. It utilizes Python's OS and random modules for file manipulation and random selection, respectively. And It has predicted correctly as strawberry\_rotten.

**Activity 2.2: Data Augmentation**

Data augmentation is a technique commonly employed in machine learning, particularly in computer vision tasks such as image classification, including projects like the healthy vs rotten Classification  in fruits and vegetables. The primary objective of data augmentation is to artificially expand the size of the training dataset by applying various transformations to the existing images, thereby increasing the diversity and robustness of the data available for model training. This approach is particularly beneficial when working with limited labeled data.

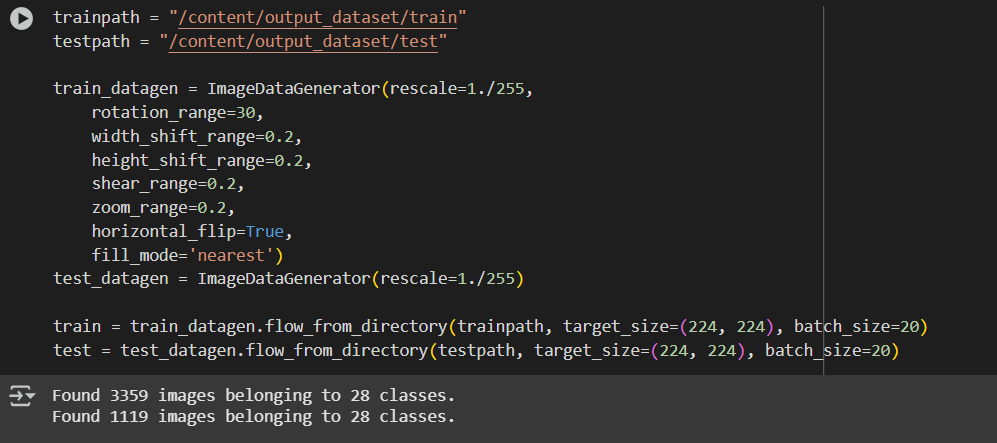
In the context of the 28  class Classification, data augmentation can involve applying transformations such as rotation, scaling, flipping, and changes in brightness or contrast to the original images of fossils. These transformations help the model generalize better to variations and potential distortions present in real-world images, enhancing its ability to accurately classify unseen data.

This is a crucial step but this data is already cropped from the augmented data so. this time it is skipped accuracy is not much affected but the training time increased

**Milestone 3: Split Data**

**Train-Test-Split:**

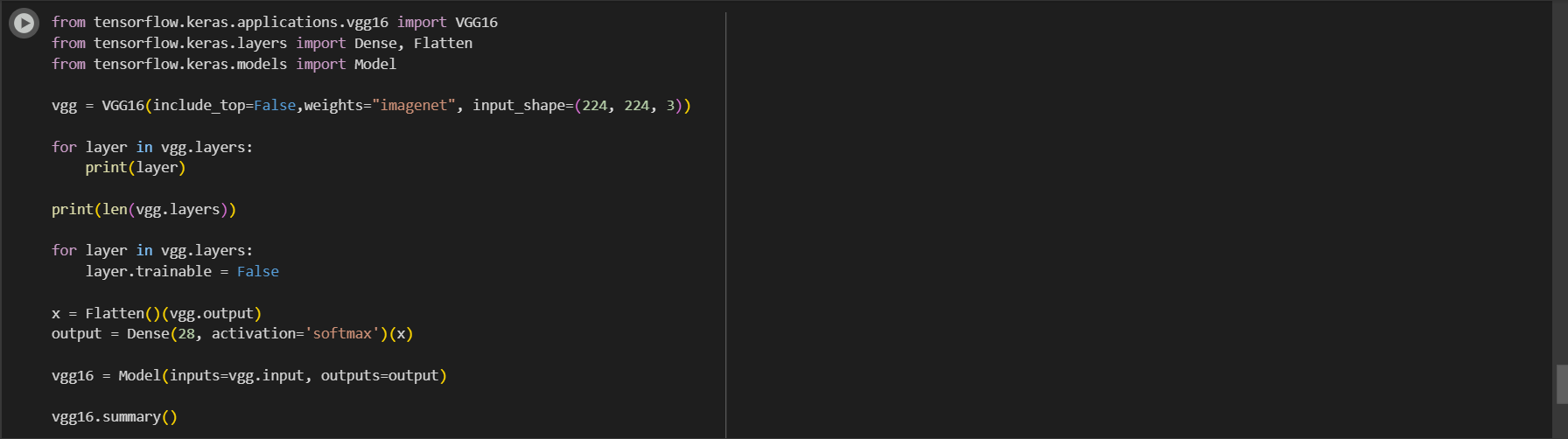
In this project, we have already separated data for training and testing.

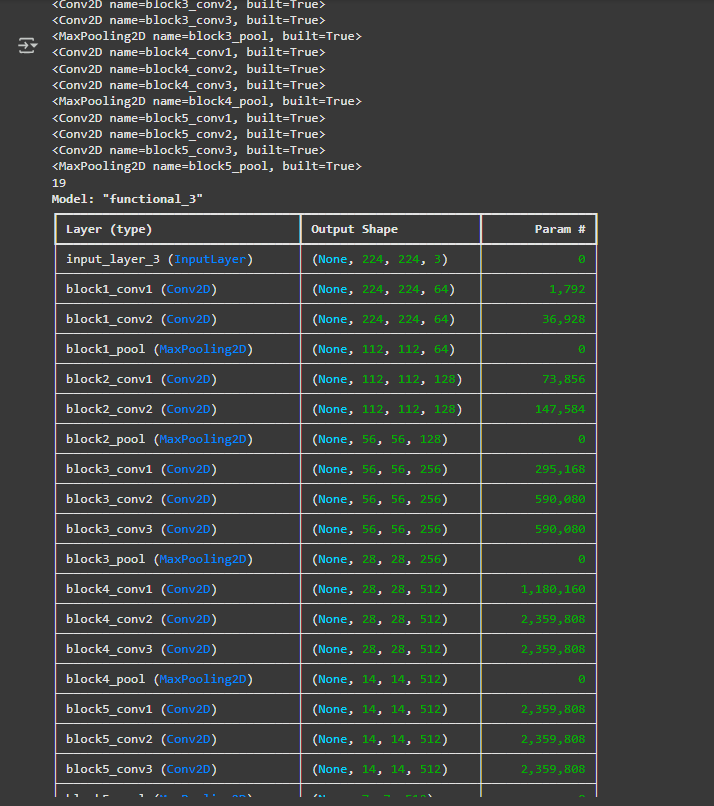


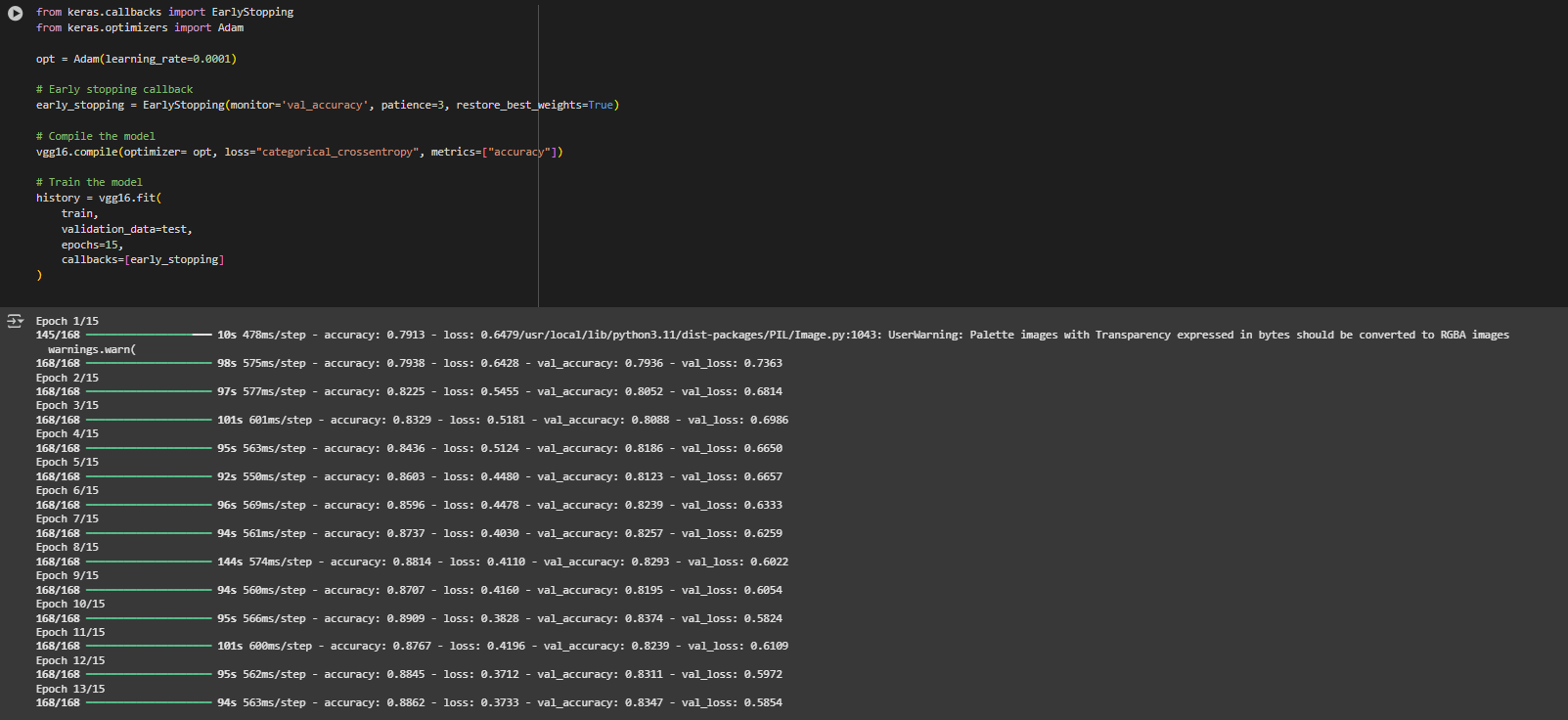
### Milestone 4: Model Building

**Vgg16 Transfer-Learning Model:**

The VGG16-based neural network is created using a pre-trained VGG16 architecture with frozen weights. The model is built sequentially, incorporating the VGG16 base, a flattening layer, dropout for regularization, and a dense layer with SoftMax activation for classification into five categories. The model is compiled using the Adam optimizer and sparse categorical cross-entropy loss. During training, which spans 10 epochs, a generator is employed for the training data, and validation is conducted, incorporating call-backs such as Model Checkpoint and Early Stopping. The best-performing model is saved as "healthy\_vs\_rotten.h5 " for potential future use. The model summary provides an overview of the architecture, showcasing the layers and parameters involved.

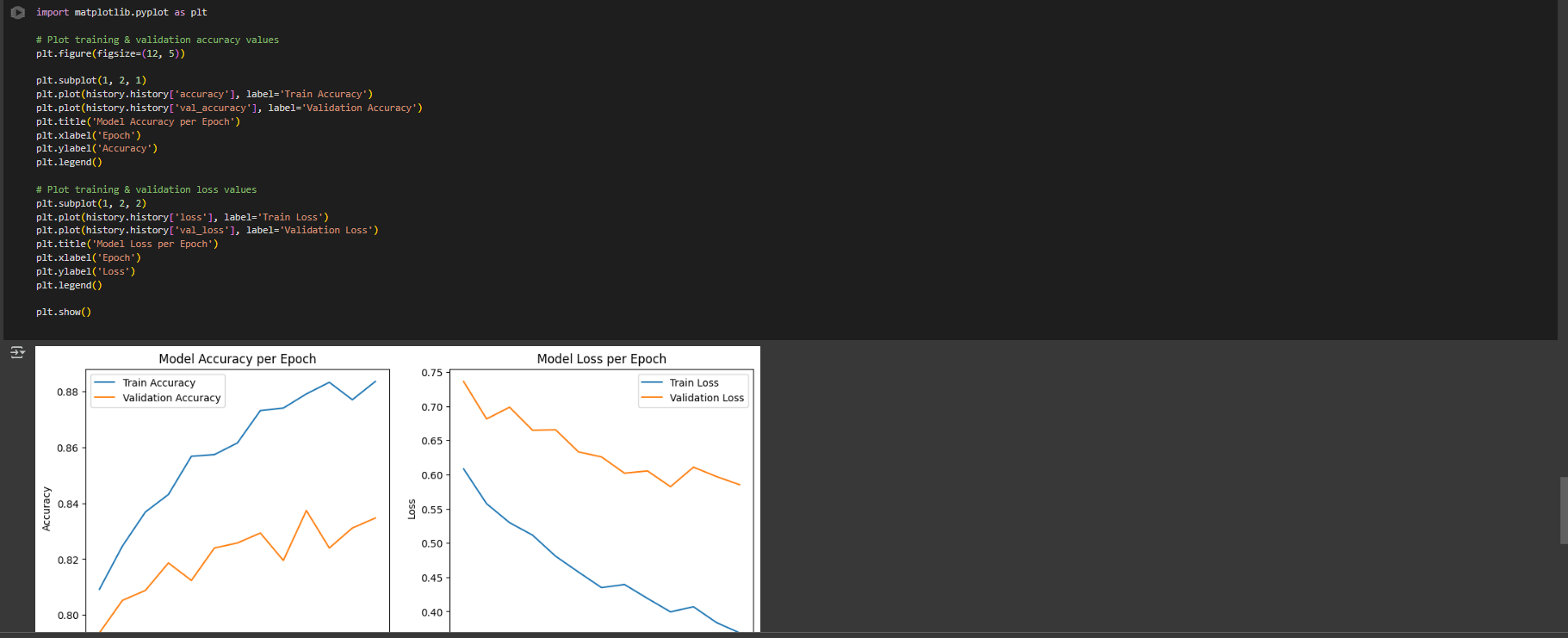


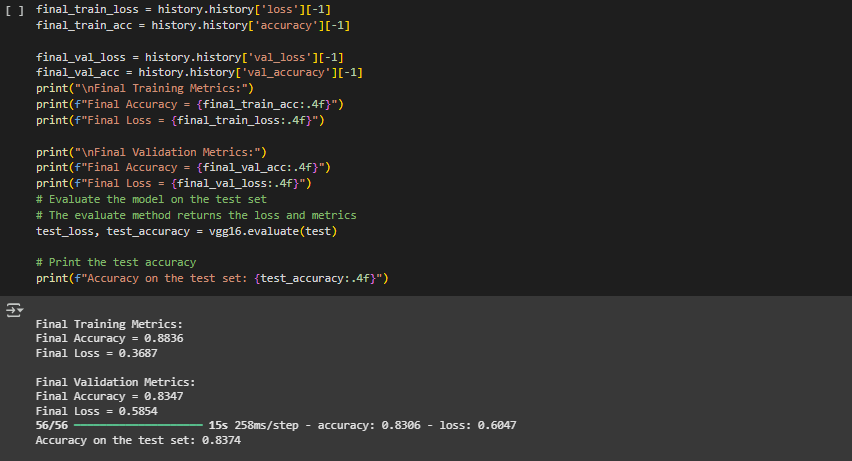




### Milestone 5: Performance Testing & Hyperparameter Tuning

First we check how well our model has been training in each epoch by plotting model loss and accuracy per epoch.

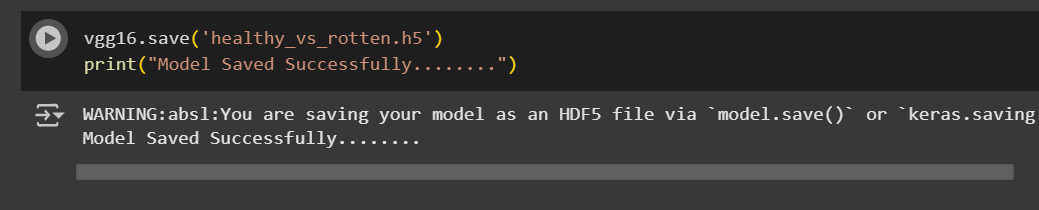


Here we have tested with the Vgg16 Model With the help of the predict () function.

We got **83.74%** accuracy on test set !

**Saving the model**

Finally, we have chosen the best model now saving that model



### Milestone 6: Model Deployment

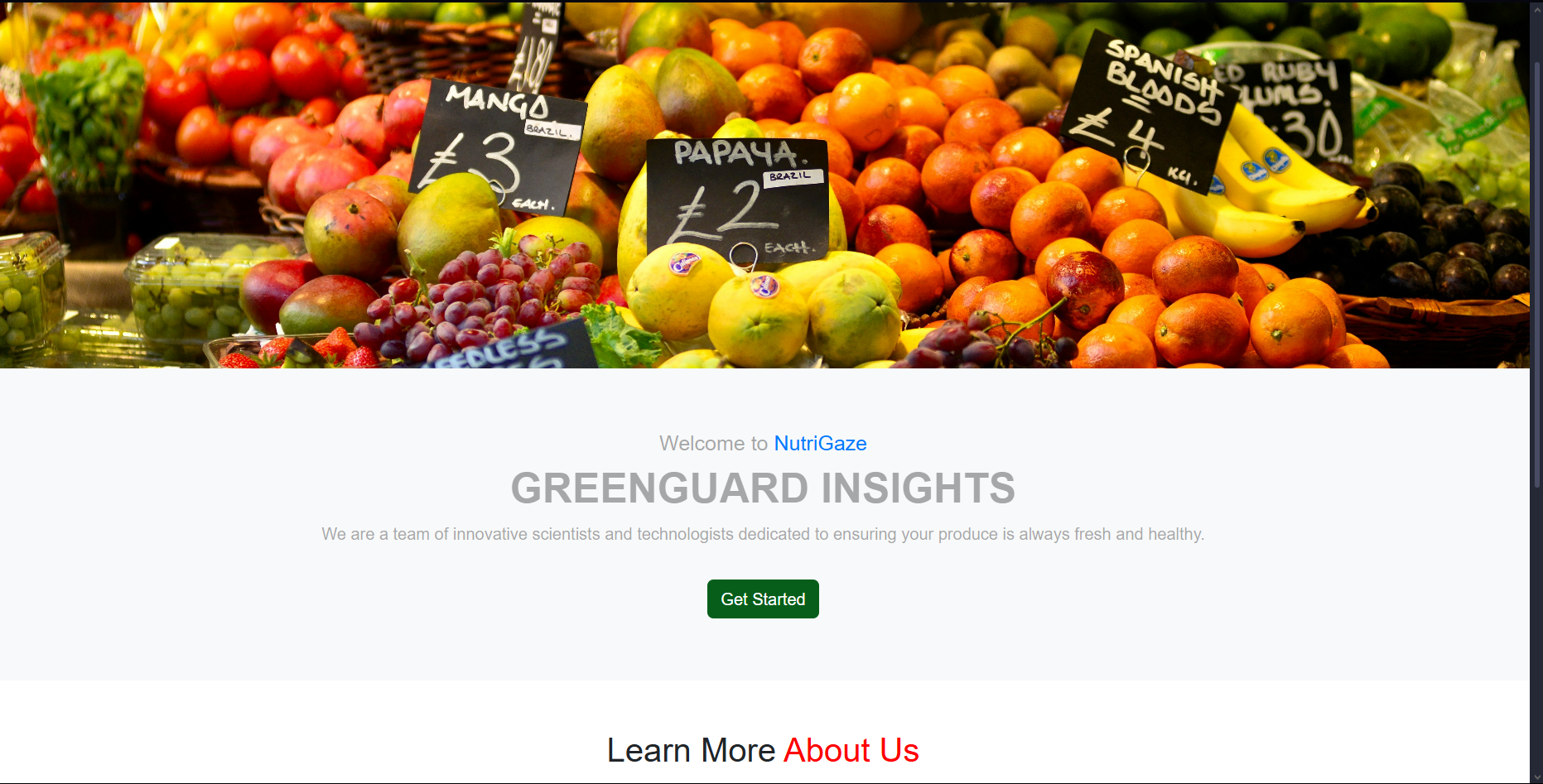
**Application Building**

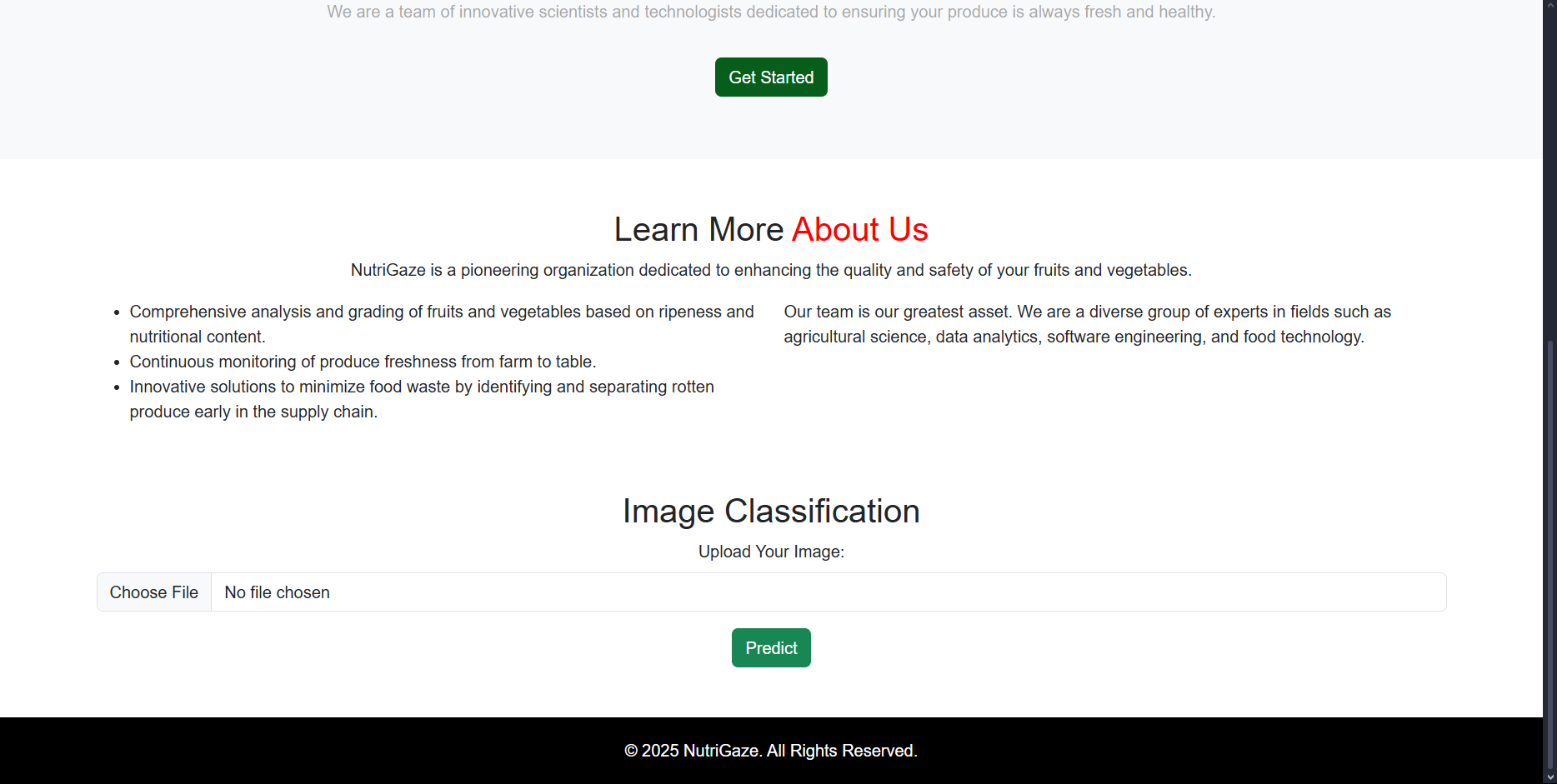
In this section, we will be building a web application that is integrated into the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

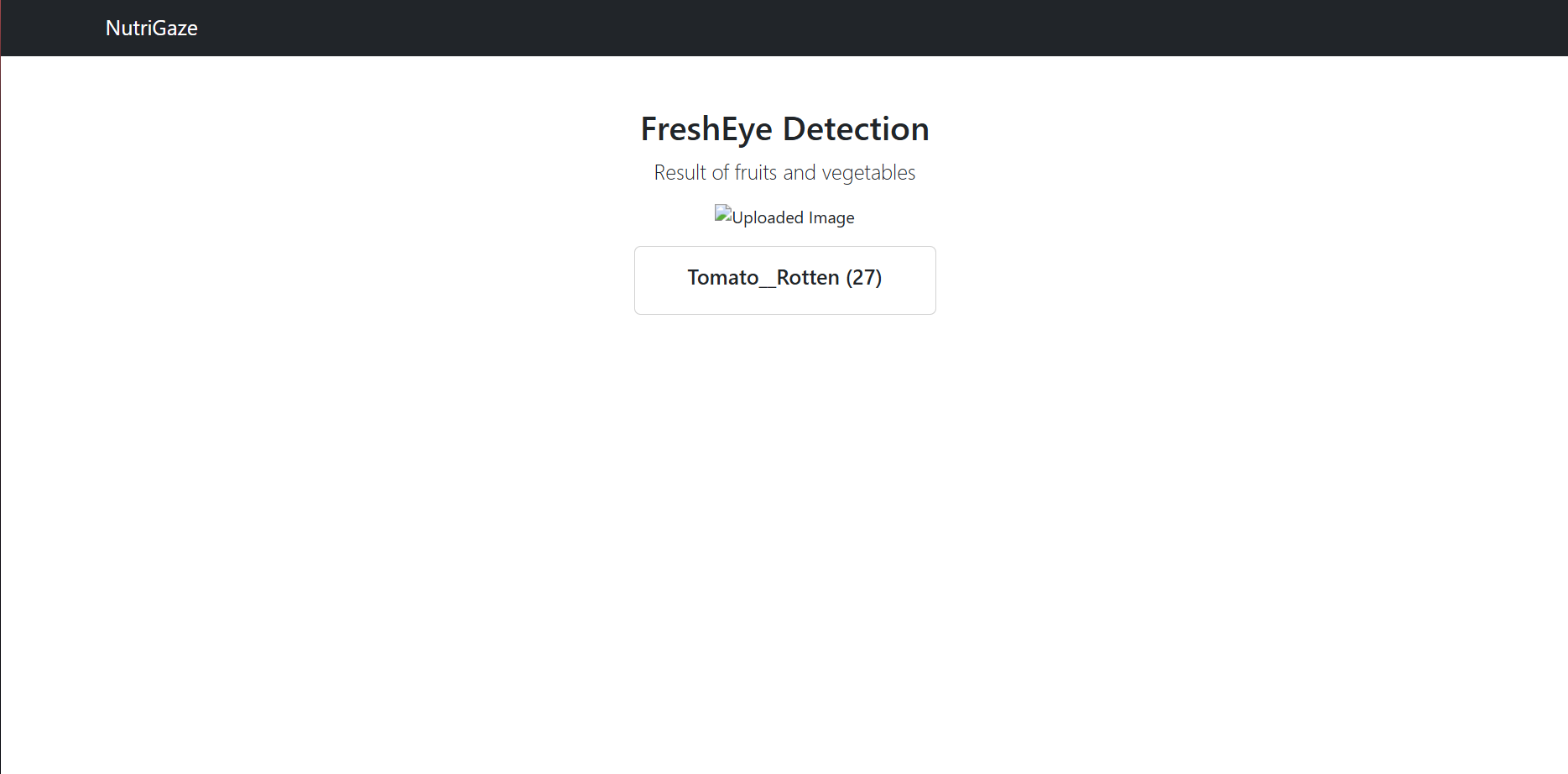
This section has the following tasks

Building HTML Pages  
Building server-side script

**Activity 6.1: Building HTML Pages**

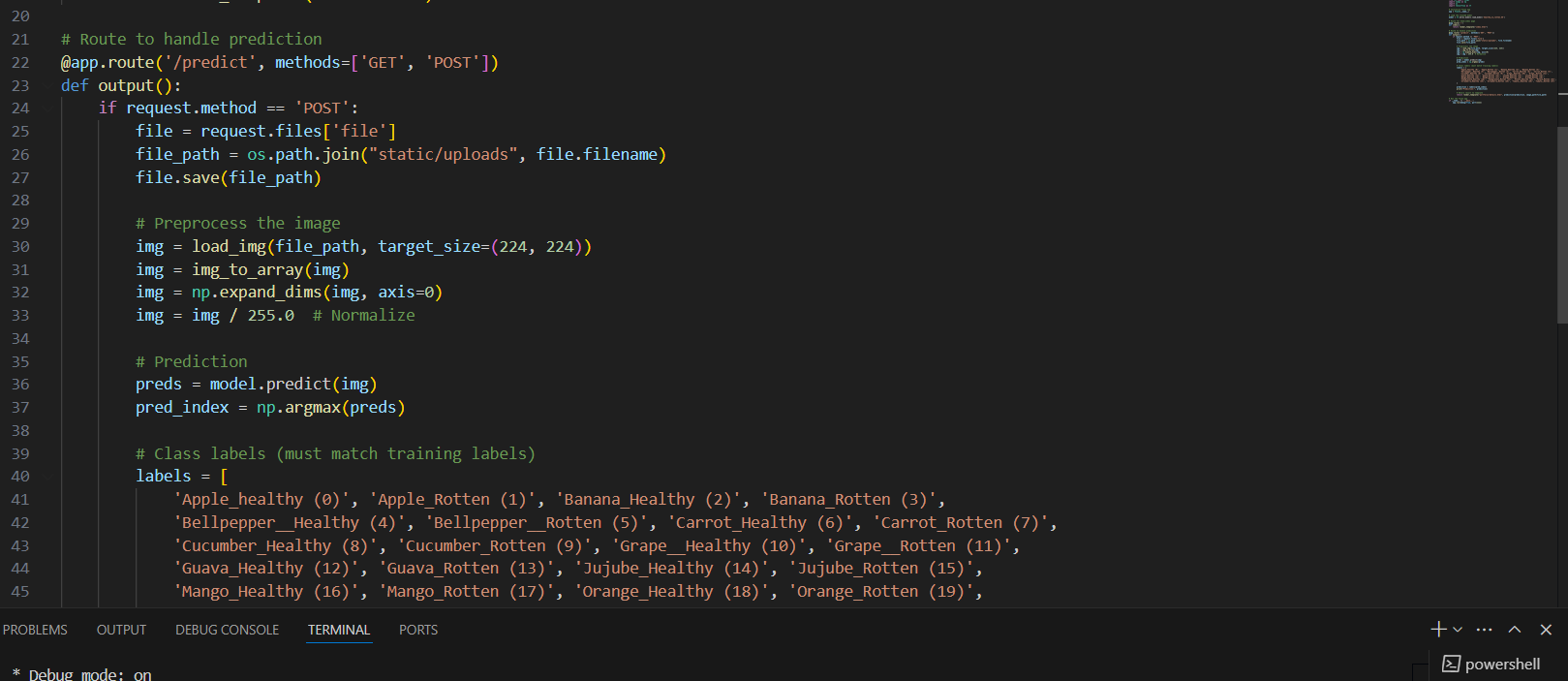


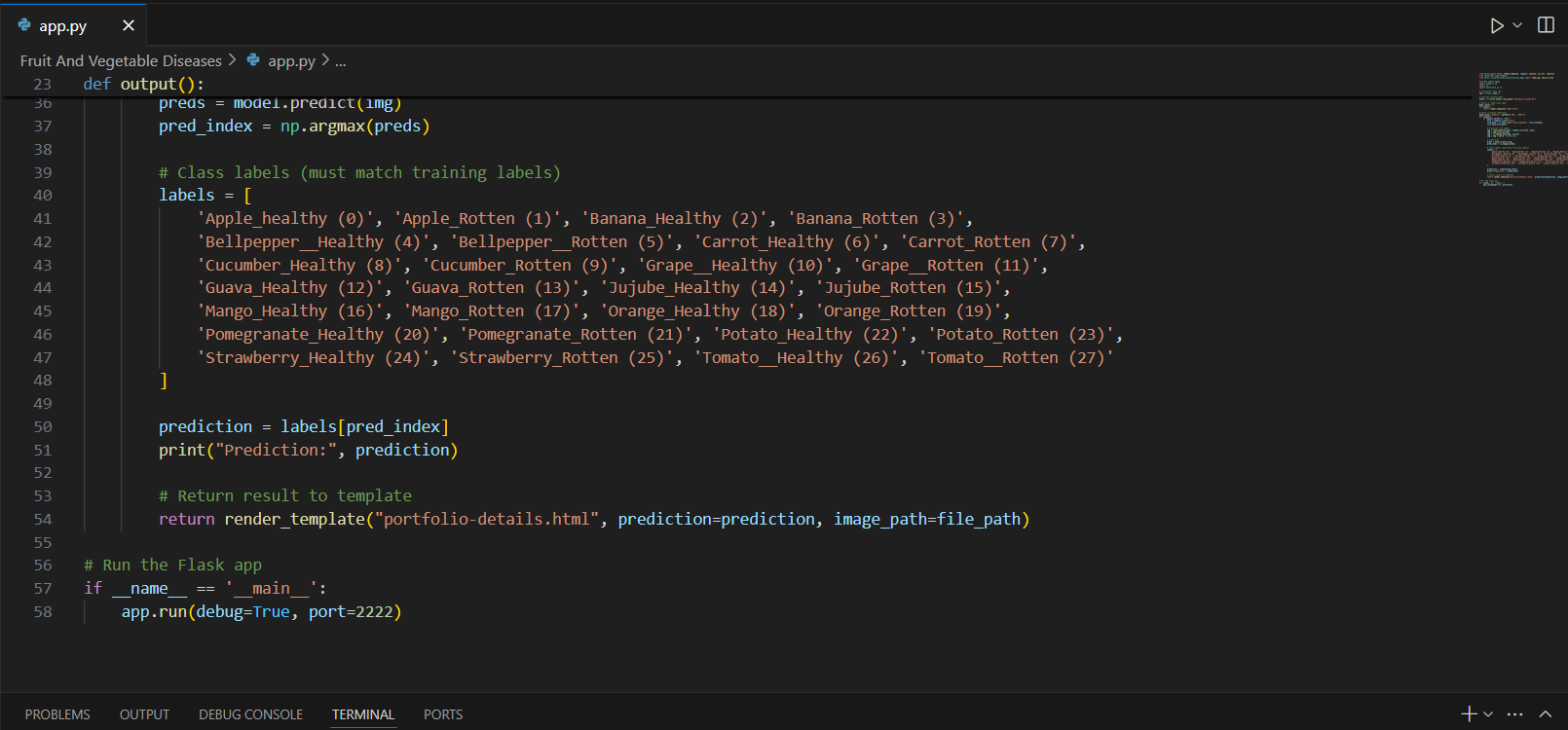


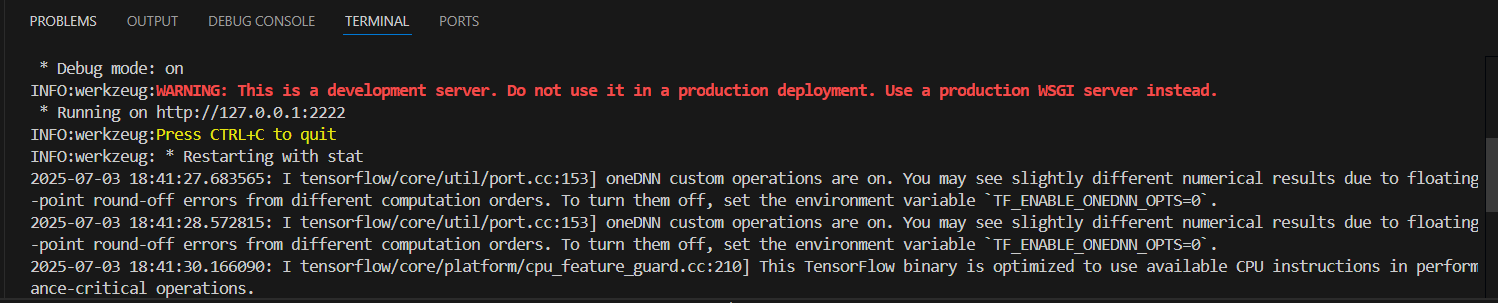


**Python Code:**

****

****

****

****

#### Milestone 7: Project Demonstration & Documentation

**Activity 1:-** [Video Demo Link](https://github.com/vijaay10/Smart-Sorting-Transfer-Learning-For-Identifying-Rotten-Fruits-And-Vegetables/tree/main/7.%20Documentation%20%26%20Demonstration)

##### Activity 2:- [Project Documentation-Step by step project development procedure](https://github.com/vijaay10/Smart-Sorting-Transfer-Learning-For-Identifying-Rotten-Fruits-And-Vegetables/tree/main/7.%20Documentation%20%26%20Demonstration)