**Application Building**

Now that we have trained our model, let us build our flask application which will be running in our local browser with a user interface.

In the flask application, the input parameters are taken from the HTML page These factors are then given to the model  to predict the type of food and to know the nutrition content in it. In order to know the nutrition content we will be using an API in this project.

**Create HTML Pages**

* We use HTML to create the front-end part of the web page.
* Here, we have created 3 HTML pages- home.html, image.html,imageprediction.html, and 0.html.
* home.html displays the home page.
* image.html is used for uploading the image
* imageprediction.html will showcase the output
* 0.html is to showcase the result. It tells the action to be performed on imageprediction.html while showcasing the result.

For more information regarding HTML

<https://www.w3schools.com/html/>

* We also use JavaScript-main.js and CSS-main.css to enhance our functionality and view of HTML pages.
* Link :[CSS](https://www.w3schools.com/css/) , [JS](https://www.w3schools.com/js/DEFAULT.asp)

Home.html looks like this

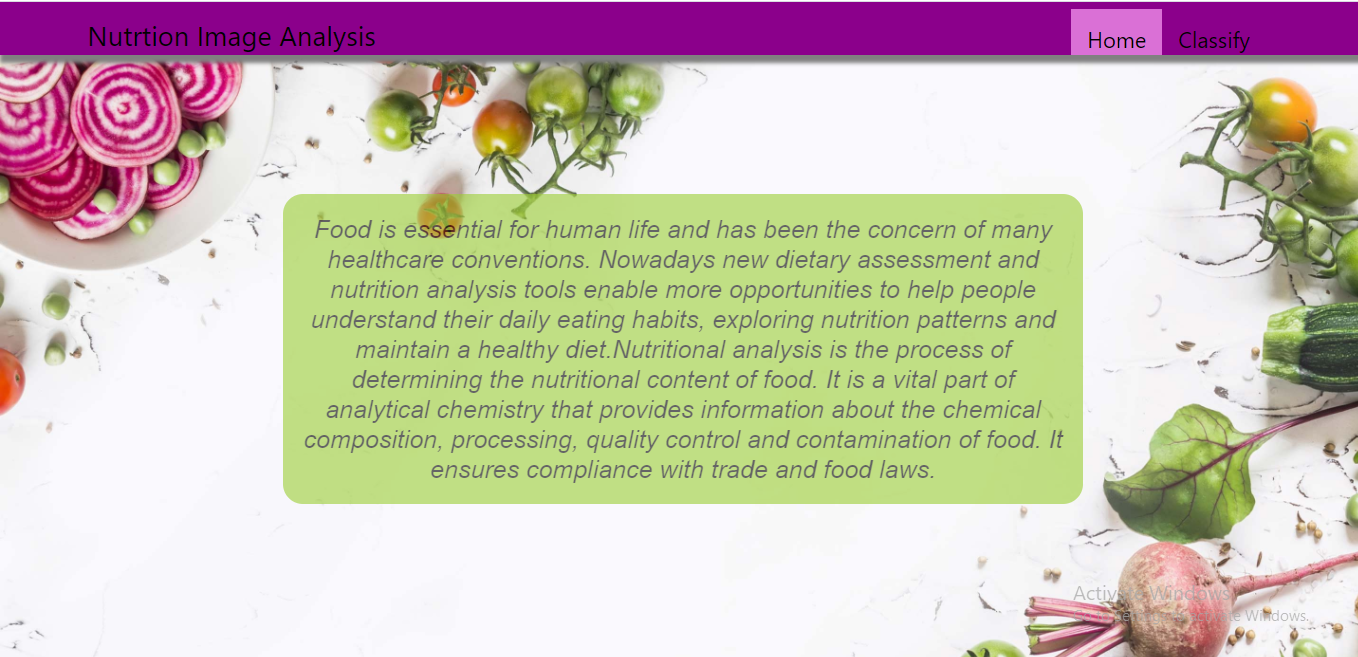
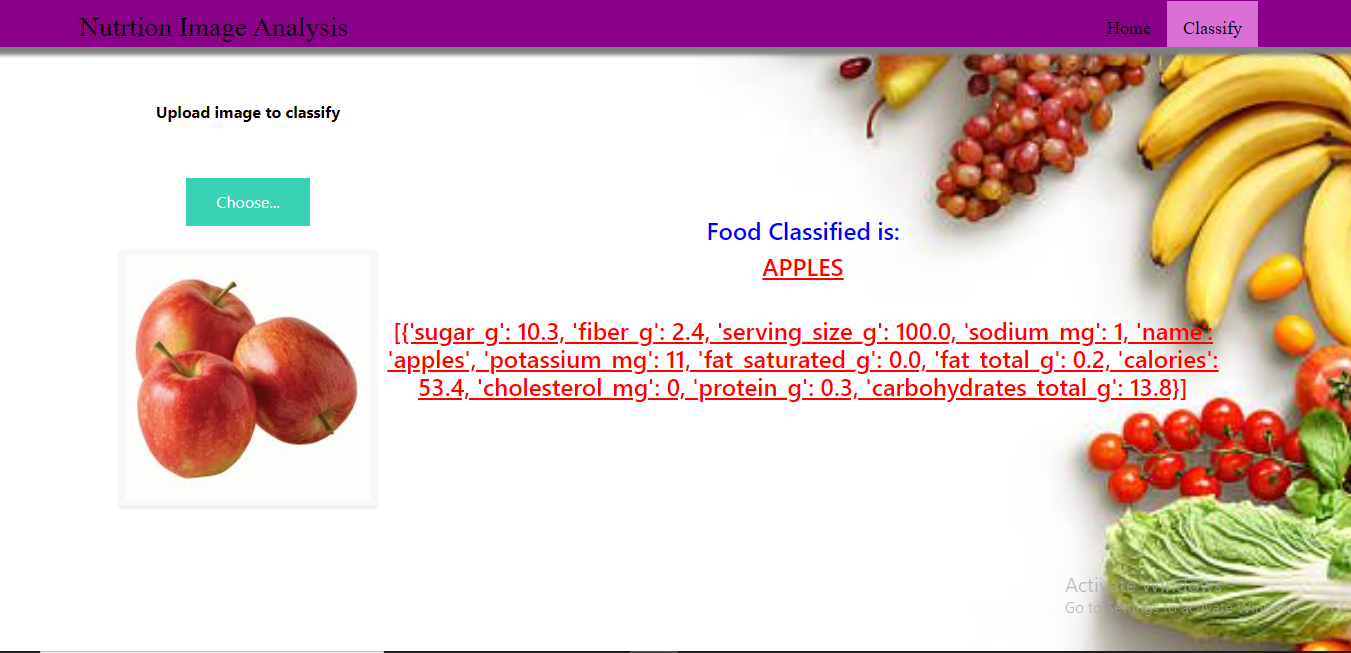


image.html



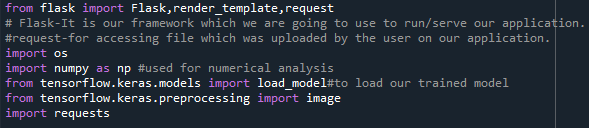
Imageprediction.html



**Build Python Code**

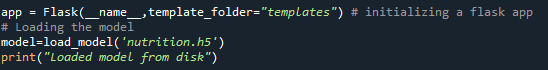
**Importing Libraries**

The first step is usually importing the libraries that will be needed in the program.

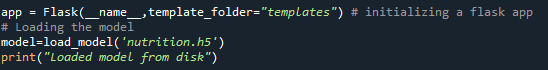


Importing the flask module into the project is mandatory. An object of the Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_) as an argument Pickle library to load the model file.

**Creating Our Flask Application And Loading Our Model By Using Load\_model Method**

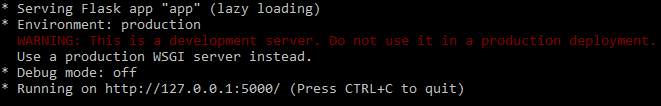
**Creating our flask application and loading our model by using the load\_model method**

**Creating Our Flask Application And Loading Our Model By Using Load\_model Method**

**Creating our flask application and loading our model by using the load\_model method**

### Run The Application

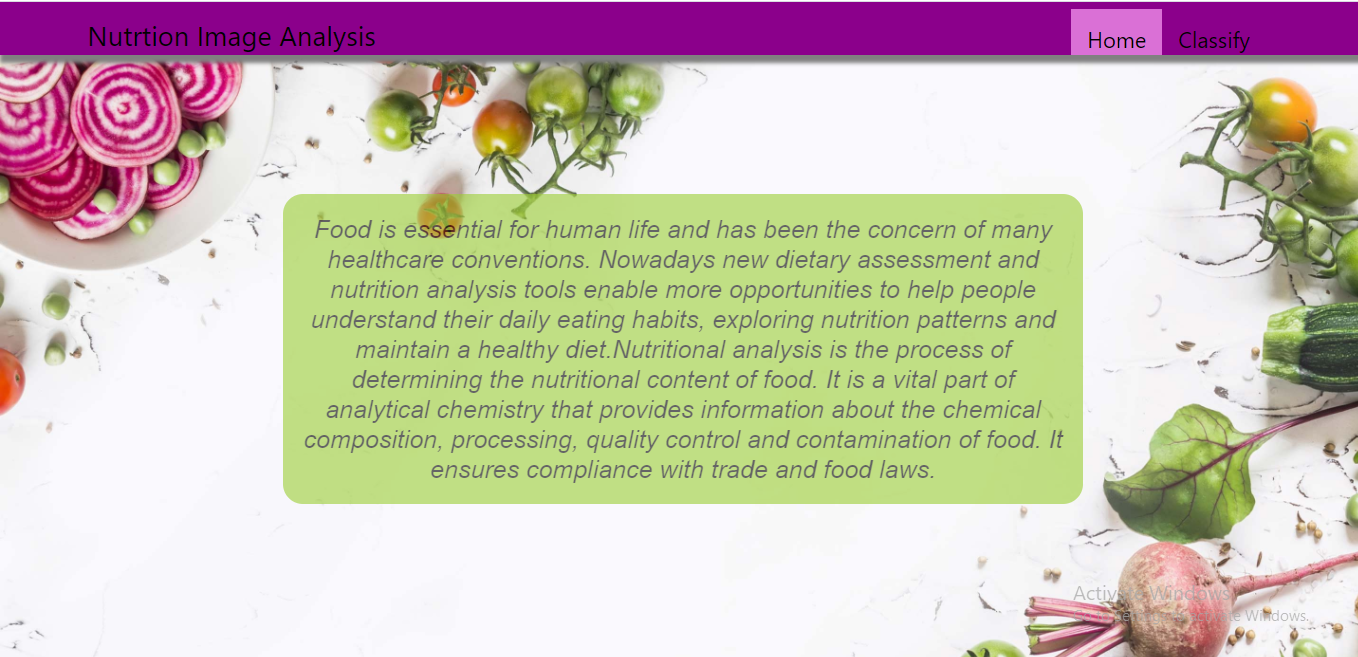
* Open the anaconda prompt from the start menu.
* Navigate to the folder where your app.py resides.
* Now type the “python app.py” command.
* It will show the local host where your app is running on http://127.0.0.1.5000/
* Copy that localhost URL and open that URL in the browser. It does navigate to where you can view your web page.
* **Enter the values, click on the predict button and see the result/prediction on the web page.**https://lh6.googleusercontent.com/wNTIPe3fZqmCEaxgRyYHW4qeChuLRTXxNhxR09Z0_bDVTdWrjj6RhBvxHIZszGIbGFD9VXkdxN_a3wWD7cVbxMDavRsOIVK_A1Wn1F5LHZ51SWiDeJxyT0W2hv-0ewoy2TPSb9bV
* Then it will run on localhost:5000



Navigate to the localhost (<http://127.0.0.1:5000/>)where you can view your web page.

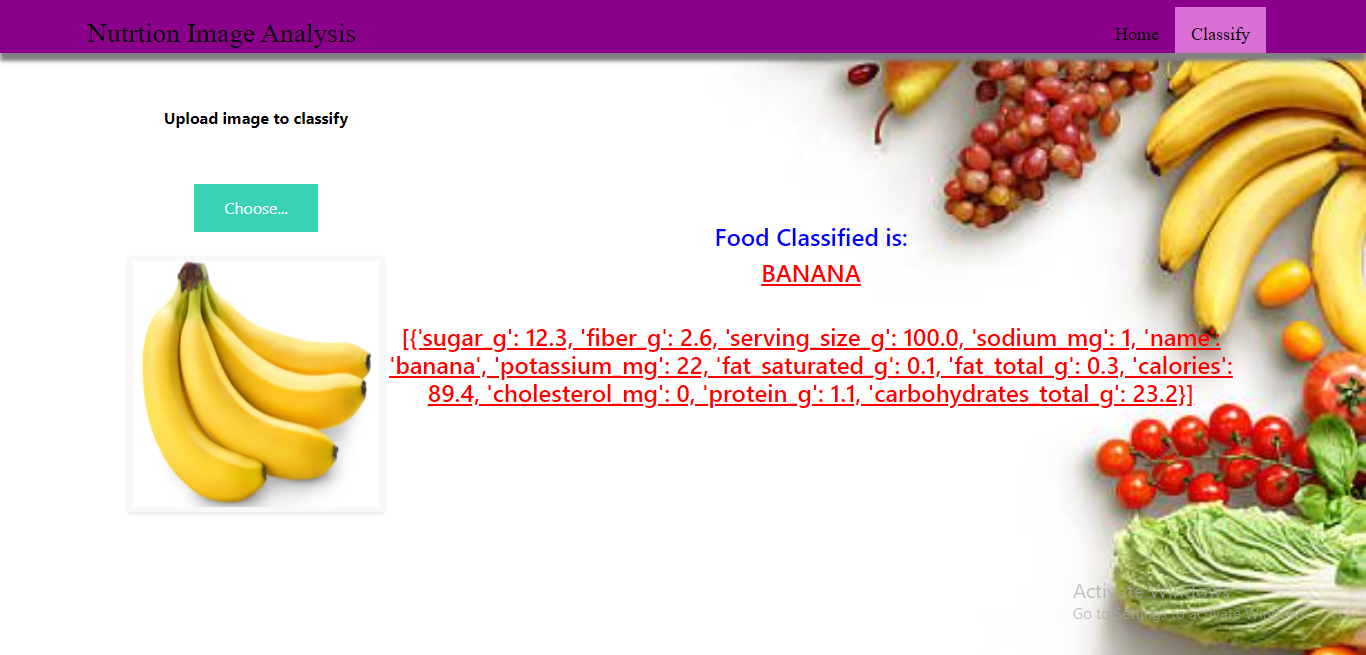
Click on classify button to see the results.

**Output screenshots:**









**Train The Model On IBM**

In this milestone, you will learn how to build Deep Learning Model Using the IBM cloud.

**Register For IBM Cloud**

**IBM Account:**

* Please click[**here**](https://www.ibm.com/academic/home) to register for IBM
* Please click [**here**](https://cloud.ibm.com/login)to log in to IBM Account

**Watch the below video to register and login into your IBM account**

**Train Model On IBM**

Please watch the below video to train the model on IBM  and integrate it with the flask Application

<https://youtu.be/BzouqMGJ41k>