IBM PROJECT REPORT

TEAM ID : PNT2022TMID30524

PROJECT TITLE : NUTRITION ASSISTANT APPLICATION

AGILA.P	**Reg. No: (613019104001)**
FARJANA BANU.S	**Reg. No: (613019104023)**
JOTHILASHMI. R	**Reg. No: (613019104029)**
KRIHTIKA. R.S	**Reg. No: (613019104042)**

ABSTRACT

The importance of food for human survival has been discussed in several medical conferences. Consumers now have more opportunities to learn about nutrition patterns, understand their daily eating habits, and maintain a balanced diet owing to modern dietary evaluation and nutrition analysis tools. Due to the ignorance of healthy food habits, obesity rates are increasing at an alarming speed, and this is reflective of the risks to people's health. People need to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity. However, although food packaging comes with nutrition (and calorie) labels, it's still not very convenient for people to refer to App-based nutrient dashboard systems which can analyze real-time images of a meal and analyze it for nutritional content which can be very handy and improves the dietary habits, and therefore, helps in maintaining a healthy lifestyle.This project aims at building application that automatically estimates food attributes such as ingredients and nutritional value by classifying the input image of food. This method employs deep learning model (CNN) for accurate food identification and Food API's to give the nutritional value of the identified food.

1. INTRODUCTION

1.1 PROJECT OVERVIEW

The style of life in the modern world is evolving daily, and with it, the requirements of the human body's composition, which includes a variety of consumables or prepared foods, are changing as well. Obesity, or being overweight, is one issue brought on by an abundance of food in our daily life. The body's excessive calorie consumption is the root cause of this issue. In today's world, obesity is becoming a major problem. Therefore, we need a system that can influence people's eating habits and give them guidance leading to healthy lifestyle maintenance. If a system alerts users to the nutritional information of a food item and categorises it as healthy or unhealthy as well as the nutrition content to the user, they can establish their daily intake of calories from their diet. First, we must determine the type of food, and then, after predicting the type of food (fruit or vegetable), our system must determine the type of that image (if the image is in the category of food or vegetable). A combination of deep learning techniques is used to recognise the image and determine the category based on it. Our approach incorporates a wide range of segmentation and picture features.

1.2 PURPOSE

The type of food must be determined first, and then our system must determine the type of that image after predicting the type of food (fruit or vegetable) (if the image is in the category of food or vegetable). The image is recognised, and a category is derived from it using a combination of deep learning algorithms. Our method includes a variety of segmentation and image properties.

Due to the ignorance of healthy food habits, obesity rates are increasing at an alarming speed, and this is reflective of the risks to people's health. People need to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity. However, although food packaging comes with nutrition (and calorie) labels, it's still not very convenient for people to refer to App-based nutrient dashboard systems which can analyze real-time images of a meal and analyze it for nutritional content which can be very handy and improves the dietary habits, and therefore, helps in maintaining a healthy lifestyle.

2. LITERATURE REVIEW

2.1 EXISTING PROBLEM

2.1.1 TITILE : Study for Food Recognition System Using Deep Learning
AUHTOR: Nareen O. M. Salim

The evaluation of weight reduction therapy has been shown in the literature to greatly benefit from accurate dietary assessment. The majority of contemporary nutritional evaluation techniques, however, rely on memory. Current computer-based food identification system development for accurate food evaluation is now possible via rich Cloud services and complete mobile devices. Addressing the issue of food detection and identification in images of various foods. The issue is worsened by the wide range of food products with low inter- and large intra-class differences and the scant information in a single image. By outlining the general application of numerous fusion-trained classifiers, it is possible to improve the identification and recognition of traits gleaned from different deep models. This essay investigated numerous methods for identifying foods. The Food identification is a challenging challenge since food products are presented; Sometimes, they are different within the same group. A sort of issue with categorizing fine-grained pictures as the identification of food pairwise local characteristics that take advantage of eight specific food ingredients' positional relationships. The proposed multi-food image recognition system that detects first food recognizes color, texture, gradient, and SIFT extracted by several detectors using multiple kernels learning regions. The food is divided into 300 blocks, and five classes are further classified, such as staple, main dish, side dish, fruit, and non-food from each block's extract color and DCT coefficients. Food identification and quantity estimation are part of the TADA dietary evaluation system.

2.1.2 TITILE : Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices

AUHTOR: Yao Liu , Hongbin Pu , Da-Wen Sun b

For the food sector, it is important to establish tools and processes for quickly and accurately identifying and analysing food quality and safety goods. traditional methods for machine learning Based on handcrafted traits, they typically perform poorly since they have a limited capacity to describe complex culinary properties. The convolutional neural network (CNN), which is the most widely used architecture of deep learning and has been increasingly used for the detection and analysis of complex food matrices, has recently emerged as an efficient and viable method for feature extraction. Goals and strategy: The current review introduces multi-feature aggregation techniques, 1-D, 2-D, and 3-D CNN models for feature extraction, and the structure of CNN. CNN's uses as a depth feature. With the improvement of the quality of life, people are increasingly conscious of high quality and safe food products in daily life, therefore the development of methods for reliably detecting and analysing food quality and safety is important for the industry. With the unique advantages of strong feature learning and good generalization ability, CNN is potential and attractive for effective and efficient analysis of complex food matrices. CNN can not only automatically locate important features, but can also obtain unparalleled performance under challenging conditions such as complex background, and different resolutions and orientations of the images. Despite the advantages of CNN in the provision of better performance, there still remain numerous challenges to its applications in the food domain.

2.1.3 TITILE : The Use of Different Image Recognition Techniques in Food Safety: A Study

AUHTOR: Rijwan Khan, Santosh Kumar

In order to prevent foodborne illness and harm, food must be properly prepared, transported, and stored. Food products may encounter a variety of health risks from farm to factory and factory to fork. Food safety is therefore essential from a financial and moral standpoint. The consequences of not complying with food safety regulations are diverse.)e demand for precise, rapid, and neutral quality assessments of these qualities in food products is increasing as dietary requirements and high-quality standards are demanded more frequently. To accomplish these goals, computer vision offers an automated, non-destructive, and cost-effective method. Its usefulness for fruit and vegetable assessment and classification has been proven by a significant body of research. It highlights the key elements of image processing technology and provides an overview of the most recent developments in the food industry. Public health is consistently and significantly burdened by foodborne illnesses. After more than a century Large-scale changes in food production, distribution, and regulations were pushed and fed into macrosocial pressures like population growth, urbanisation, and globalisation. Compared to other economic sectors, the food industry and distribution network, in particular, have created huge amounts of data in recent years. To increase the safety of the food supply, several types of data were imaginatively examined at various points along the agricultural value chain. For instance, toxic contaminations on farmlands were forecasted in preharvest, field, and weather forecasts; in the retail setting, contactless audits and record-keeping were carried out for 1.4 million months; and observations of Hindawi Journal of Food Quality Volume

2.1.4 TITILE : Barriers to and Facilitators for Using Nutrition Apps: Systematic Review and Conceptual Framework
AUHTOR: Laura Maria König, Christiane Attig

Diet-related health risk factors and eating habits can both be modified with the help of nutrition apps. Although they might slow the rising rates of overweight and obesity, they haven't yet been widely adopted. Therefore, comprehensive understanding of the development of design recommendations targeted at promoting adoption and sustained use of nutrition apps requires an understanding of the factors encouraging and discouraging (long-term) app use. The literature on obstacles to and enablers of the use of nutrition apps across disciplines, as well as empirical qualitative and quantitative studies with current, former, and non-users of nutrition apps, has been synthesised in this systematic review. PsychINFO, PSYNDEX, PsycArticles, PubMed, Web of Science, and SPORTDiscus were among the six databases used in a comprehensive literature search that also included backward and forward citation searches. The anticipated data extraction procedure, the inclusion and exclusion criteria, and the search strategy were all registered in advance. All empirical qualitative and quantitative publications in German or English that focused on adolescents (aged 13–18) or adults who were either present, former, or non-users of nutrition apps were eligible for inclusion. Individual barriers and facilitators were extracted and put into categories based on a qualitative content analysis. Multiple factors influencing participation with mobile weight reduction and weight maintenance therapies have been discovered by two systematic evaluations. These elements include social support, customisation, ease of use, entertainment, and the availability of tools like self-monitoring, prompts, and feedback.

2.1.5 TITILE : Identification of malnutrition and prediction of BMI from facial images using real-time image processing and machine learning

AUHTOR: Dhanamjayulu C, Nizhal U N

The usable information on human faces can be used to determine an individual's age, gender, weight, etc. Body mass index (BMI) and weight are two of these biometrics that are reliable predictors of health. Based on recent health science studies, this work proposes a regression approach based on the 50-layer Residual network architecture to investigate ways to identify malnourished individuals and obese individuals by evaluating body weight and BMI from facial photos. Multi-task Cascaded Convolutional Neural Networks have been used for face detection. A method is developed to assess BMI, age, and gender using real-time photographs of human faces. Obesity and malnutrition are frequently identified with the aid of BMI. The estimation of height, weight, and BMI using automated methods was done in earlier publications. Today's social networks, like Facebook, Instagram, and Snapchat, contain a variety of functions, including the trading of images, looking for a job, dating, and blogging. More and more people around the world are capturing their lives with digital cameras and publishing the records as images or videos on social media networks. The proposed method is useful in establishing the relation between the characteristics of the human face and the body, such as body height and weight. The proposed method to identify Malnutrition and obese children from human faces. The proposed system does not require the full body real image of a person. Face detection is done with the Multi-task Cascaded Convolution Neural Networks on pictures with single/multiple faces. BMI, age, and gender are estimated from a person's face using residual neural networks. The problems of BMI, age, and gender estimation are posed as three separate regression pattern classification problems. The dataset of facial images taken from the internet along with their metadata containing information like gender, age, and BMI.
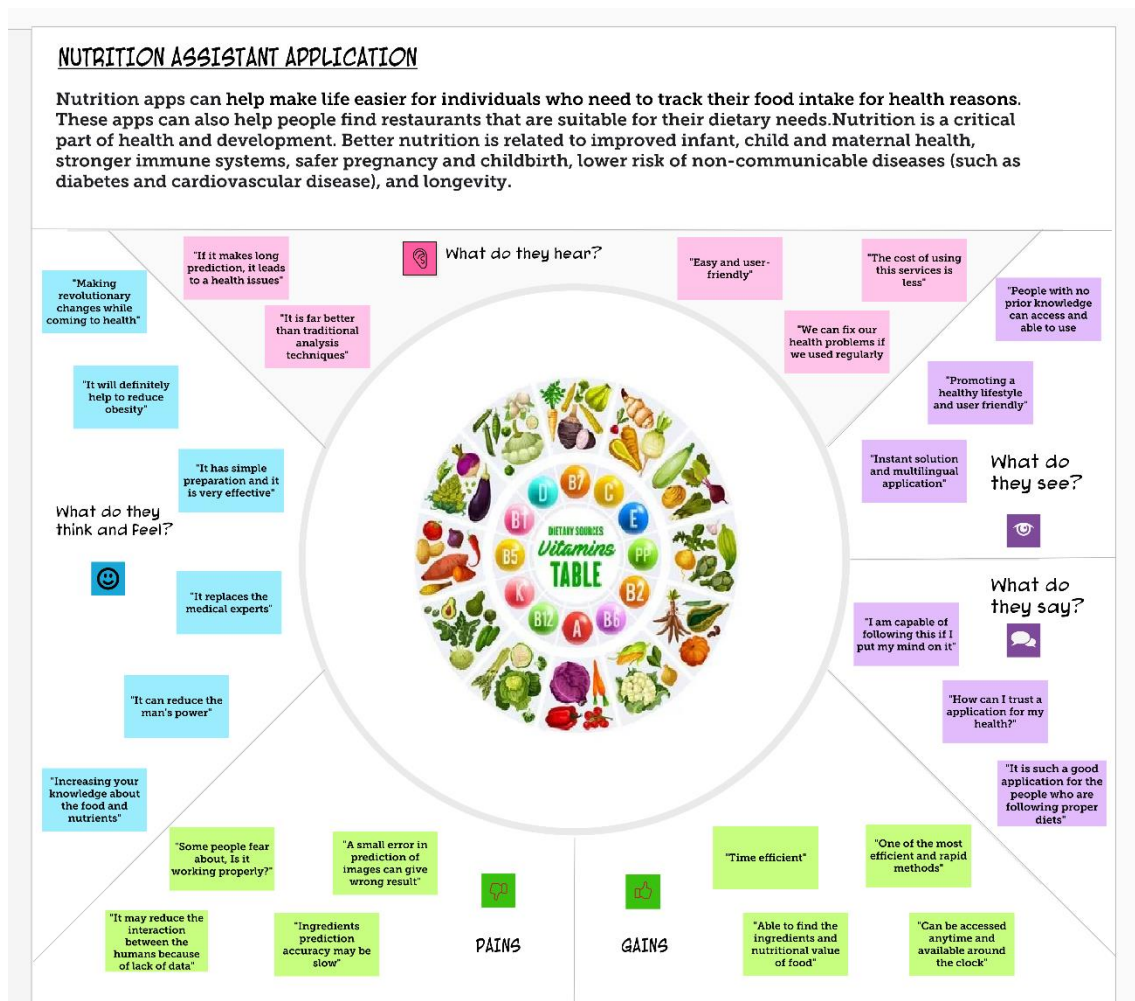
2.2 REFERENCES

1. Nareen O. M. Salim, Study for Food Recognition System Using Deep Learning,2021

2. Yao Liu , Hongbin Pu, Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing   complex food matrices, 2021

3. Rijwan Khan, Santosh Kumar, The Use of Different Image Recognition Techniques in Food Safety: A   Study, 2021

4. Laura Maria König, Barriers to and Facilitators for Using Nutrition Apps: Systematic Review and Conceptual Framework, 2021

5. Dhanamjayulu C, Identification of malnutrition and prediction of BMI from facial images using real-time image processing and machine learning, 2021

2.3 PROBLEM STATEMENT DEFINITION

In existing system, the way food data is analysed is evolving as a result of recent developments in computer vision and machine learning. Food-related photos, however, are frequently challenging to recognise and slow to detect due to the high quantity of food items and the ineffectiveness of the detection method. A diet plan is also fully manual to obtain. People may occasionally have to wait for many hours before learning about the suggested diet plan after contacting their dietician. For users, this makes getting their diet plan challenging. From the perspective of the customer, this is ineffective.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION & BRAINSTORMING

3.3 PROPOSED SOLUTION

Sufficient nutrition and energy consumption, which can only be obtained through proper eating habits, are essential for maintaining a healthy existence. In addition to maintaining a good standard of living, a healthy diet helps people prevent chronic conditions including diabetes, high blood pressure, mental illness, asthma, and others. Obesity is one of the conditions brought on by over eating that is most common. When a person is obese, excess body fat accumulates to the point where it endangers their health. For the suggested system to recognise and locate food items from the input images, develop an automated nutrition analysis system. By locating possible locations and classifying them with deep neural networks, develop a three-step procedure specifically for detecting various foods in photographs. In the first stage, the automated system generates numerous regions of proposals using the provided photos. Then, it collects each region of thoughts by locating them in the original pictures, putting them on feature maps, and classifying them according to different food groups. Finally, by analysing the nutritional elements in the images, determine the food's calories, fat, carbs, and protein contents as well as its ingredients to generate a dietary evaluation report. The system's effectiveness and accuracy will also be increased by expanding the dataset to cover a larger variety of food types.

3.4 PROBLEM SOLUTION FIT

There is no organised way to quickly collect dietary data. Waiting times for diet appointments might be hours long. Calculate the calories, fat, carbohydrates, and protein content after examining the nutritional components in the images to produce a report on the dietary quality. The system's effectiveness and accuracy will be increased by broadening the information to include a wider variety of food types.

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Upload Image

In this module, the nutrition datasets should be uploaded as CSV files. Additionally, the information is kept in a database for later use. The dataset includes information on the calories, protein, fat, and food ingredient values of fruits and vegetables. These numbers are preserved as integer values and acquired from the Kaggle website.

Filtering Noise

Ability to determine the nutritional content of diverse fruits and vegetables is made possible by the use of filter techniques that reduce noise in photos. The filter's goal is to get rid of noise, which makes photographs look bad. Statistics back up this statement. A filter's typical frequency response is constructed. Filtering is a nonlinear technique widely used in image processing to get rid of "salt and pepper" noise. When edge preservation and noise reduction are top concerns, a median filter is preferable to convolution. Practice photo binarization in a similar way. In the pre-processing stage of document analysis, document picture binarization is used to separate the text in the foreground from the backdrop of the document. The succeeding document image processing tasks require a quick and precise binarization approach.

Classification

Classification is the process of dividing data into various categories. The method starts by determining the class of the given data points. Classification is achievable for both structured and unstructured data. The terms target, label, and classes are occasionally used to describe the classes. The user-uploaded food image will be compared to the food items in the system database for the features obtained in the feature extraction stage in the classification process. The specific food item will be recognised once the ideal match is discovered based on the qualities matched. The detected food item's name with ingredients will be displayed over the food. Here, a convolution neural network approach is employed to classify data.

Nutrition Detection

The food nutrition API receives the image after the model has identified the food category or food type and extracts the food's nutritional data before sending it to the system. The system contrasts the nutritional information with the suggested dietary allowances. If the amount of a specific nutrient, let's say calories, exceeds the recommended dietary allowance, the user will receive a warning message to reduce nutrition intake. In that case, the user will see the food's nutritional value.

4.2 NON FUNCTIONAL REQUIREMENTS

Usability

The system shall allow the users to access the system with pc using web application. The system uses a web application as an interface. The system is user friendly which makes the system easy

Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Scalability

Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

Security

A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied.

Performance

The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

Reliability

The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week. 24 hours a day.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

LEVEL 0

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.
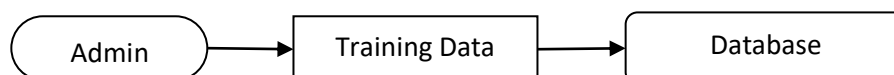
LEVEL 1

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable on screen or paper.

```
┌────────┐        ┌──────────────────┐
│  User  │ ─────▶ │   Testing Data   │
└────────┘        └──────────────────┘
                           │
                           ▼
                  ┌──────────────────┐
                  │    Input Data    │
                  └──────────────────┘
                           │
                           ▼
     ┌──────────────────┐        ┌──────────────────┐
     │  Pre-processing  │        │ Feature Extraction │
     │  ┌────────────┐  │ ─────▶ │  ┌──────────────┐ │
     │  │   Noise    │  │        │  │ Extract Shape,│ │
     │  │ Filtering  │  │        │  │    Colour    │ │
     │  └────────────┘  │        │  └──────────────┘ │
     └──────────────────┘        └──────────────────┘
```
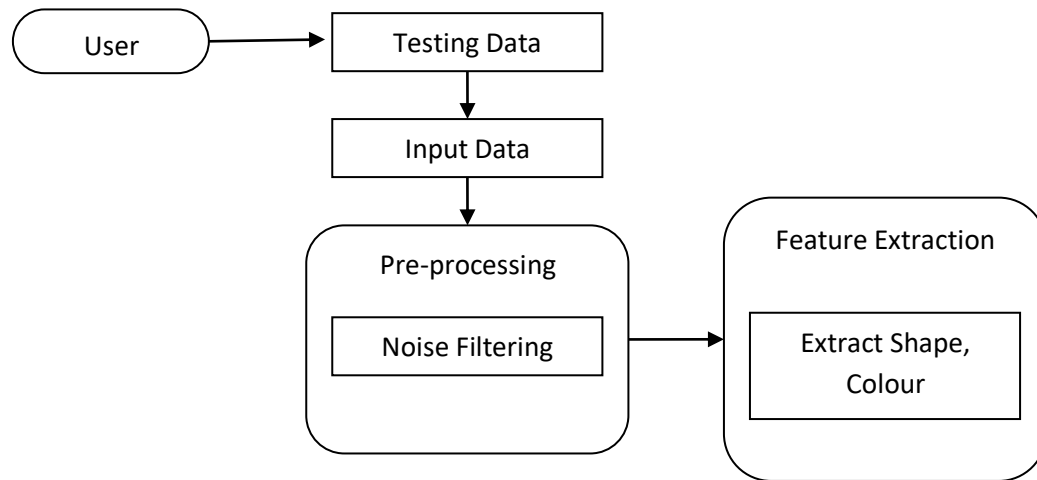
LEVEL 2

A Data Flow Diagram (DFD) tracks processes and their data paths within the business or system boundary under investigation. A DFD defines each domain boundary and illustrates the logical movement and transformation of data within the defined boundary. The diagram shows 'what' input data enters the domain, 'what' logical processes the domain applies to that data, and 'what' output data leaves the domain. Essentially, a DFD is a tool for process modelling and one of the oldest.

```
┌──────────────────┐        ┌──────────────────┐
│  Extracted Data  │ ─────▶ │   Matching with  │
└──────────────────┘        │ Training Data using│
                            │       CNN        │
                            └──────────────────┘
                                     │
                                     ▼
                            ┌──────────────────┐
                            │   Gives Result   │
                            └──────────────────┘
```
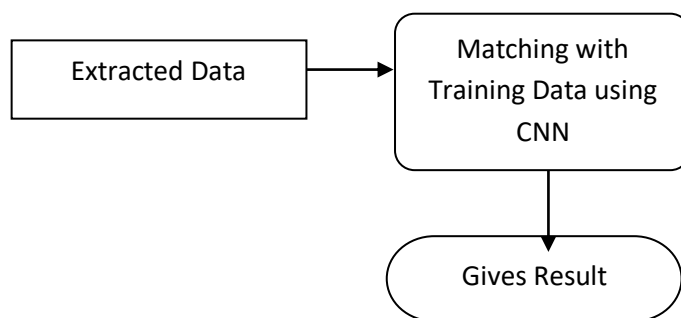
5.2 SOLUTION & TECHNICAL ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.
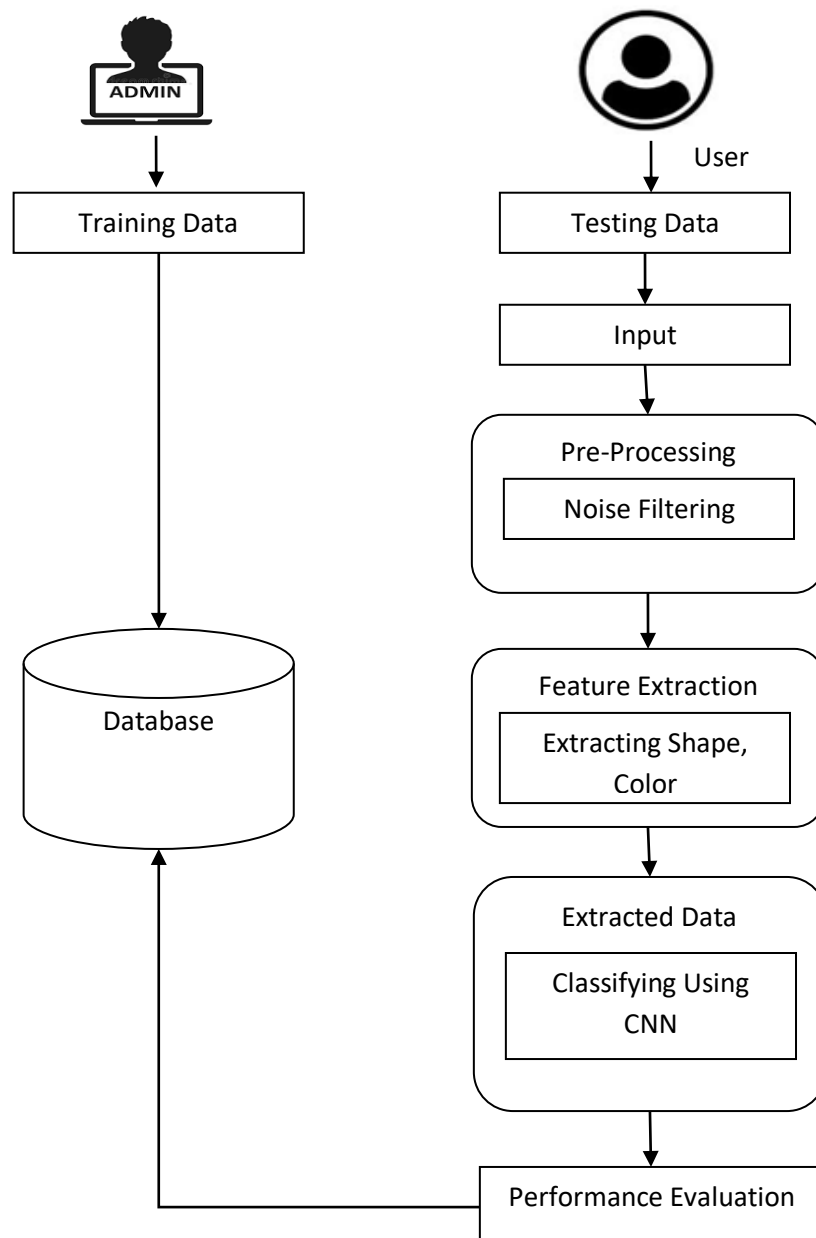
System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story No	User Story / Task	Story Points	Priority
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High
Sprint-1	Registration	USN-2	As a user, I will receive confirmation emailonce I have registered for the application	1	High
Sprint-2	Registration	USN-3	As a user, I can register for the application through phone number	2	Medium
Sprint-2	User interface	USN-4	Professional responsible for user requirements & needs	2	Medium
Sprint-3	Login	USN-5	As a user, I can log into the applicationby entering email & password	1	High
Sprint-3	Dashboard	USN-6	As a user, I must receive any updates orpop ups in my dashboard	2	High
Sprint-4	Details	USN-7	As a user, I should get notification about the progress and any updates via email orsms	1	Medium
Sprint-4	Privacy	USN-8	The developed application should be secure forthe users	2	High

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Sprint Release Date (Actual)
Sprint-1	6 Days	24 Oct 2022	29 Oct 2022	20	30 Oct 2022
Sprint-2	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	6 Days	07 Nov 2022	12 Nov 2022	20	13 Nov 2022
Sprint-4	6 Days	14 Nov 2022	19 Nov 2022	20	29 Nov 2022

6.3 REPORTS FROM JIRA
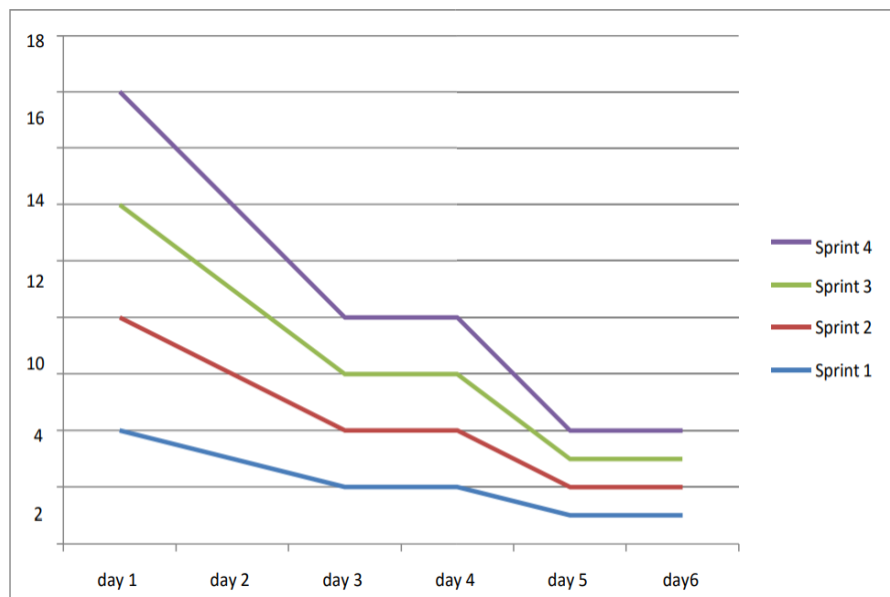
7. CODING & SOLUTIONING

7.1 FEATURE 1

Display.html

```html
<!DOCTYPE html>
<html>
<head>
  <title>Nutrition Assistant Application</title>
  <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700"
rel="stylesheet">
  <style>
  body{
   display: flex;
   font-family: Roboto, Arial, sans-serif;
   font-size: 15px;
   background-image: url('https://image.freepik.com/free-photo/diet-food-background-
concept-healthy-carbohydrates-carbs-products-fruits-vegetables-cereals-nuts-beans-
dark-blue-concrete-background-copy-space-frame_136595-2921.jpg');
   background-repeat:no-repeat;
   background-size:cover;

  }
  .styled {
   border-collapse: collapse;
   margin: 25px 0;
   font-size: 0.9em;
   font-family: sans-serif;
   min-width: 400px;
   background: #888888;
   box-shadow: 0 0 20px rgba(0, 0, 0, 0.15);
   }
  .styled thead tr {
   background-color: #8ebf42;
   color: #ffffff;
```

```html
        text-align: left;
        }
      .styled th,.styled td {
       padding: 12px 15px;
       }
       .main{
       border: 1px black;
       }
      th, td {
      padding: 15px;
      text-align: left;
      color: white;
       }
      th {
      background-color: #8ebf42;
      color: white;
      }
      tr:hover {
      background-color: #8ebf42;
      }
      .table-dis{
       justify-content: center;
       margin-top: 100px;
      }
      h2, h3{
     color: white;
      }
    </style>
</head>
<body>
<div class="heading1">
   <h2>Welcome, to Nutriooz</h2>
</div>
<div class="table-dis">
   <table class="styled">
     <th class="main" colspan=2>Nutritional Values for {{x}}:</th>
      </thead>
```

```html
        <tr>
         <th>Nutrients</th>
         <th>Nutritional Values</th>
        </tr>
       </thead>
       <tbody>
         {% for k, v in data.items() %}
        <tr class="active-row">
         <td>{{k}}</td>
         <td>{{v}}</td>
        </tr>
         {% endfor %}
       </tbody>

      </table>
     </div>
     </body>
</html>
```

Home.html

```html
<!DOCTYPE html>
<html>
<head>
     <meta charset="utf-8">
     <meta name="viewreport" content="width=device-width, initial-scale=1.0">
     <title>
              Nutrition Assistant Application
     </title>
     <link rel="stylesheet" type="text/css"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" />
     <link href="https://fonts.googleapis.com/css?family=Raleway&display=swap"
rel="stylesheet" />
     <style>

body{
     margin:0;
     padding:0;
```

25

```css
        background:#fff;
        font-family: 'Raleway',sans-serif;
        color: #fff;
}
.banner{
        height: 100vh;
        width: 100%;
        background-image: url('../static/background1.jpg');
        background-position: top;
        background-size:cover;
        /*background-attachment: fixed;*/
        background-repeat: no-repeat;
}
.banner .navbar{
        margin-top:4%;
}
.banner .navbar-brand{
        color: #fff;
        font-size:1.8em;
        font-weight: 700;
        margin-left: 10%;
}
.banner .nav{
        margin-right:10%;
}
.banner .nav li a{
        color:#aaa;
        font-size: 1.2em;
}
.banner .info{
        margin-top:15%;
        transform: translateY(-15%);
}
.banner .info h1{
        font-size: 2.5em;
        font-weight: 700;
        color: #fff;
```

```
        letter-spacing: 2px;
}
.banner .info p{
        font-size: 2em;
        font-weight: 500;
        color: white;
        letter-spacing: 2px;
}
.banner .info a{
        margin-left:50%;
        transform: translateX(-50%);
        color: #fff;
        background: #e91d43;
        padding:10px 20px;
        font-size: 2em;
        font-weight: 600;
}
.banner .info a:hover{
        background: #e91e63;
}
        </style>
</head>
<body>
        <div class="container-fluid banner">
                <div class="row">
                        <div class="col-md-12">
                                <nav class="navbar navbar-md">
                                        <div class="navbar-brand">NUTRITION
ASSISTANT APPLICATION</div>
                                </nav>
                        </div>
                        <div class="col-md-8 offset-md-2 info">
                                <h1 class="text-center">NUTRIOOZE</h1>
                                <p class="text-center">
                                        "To eat is a necessity, but to eat intelligently is an
art."
                                </p>
```
27

```
                                      <a href="/pythonlogin/" class="btn btn-md text-
center">GET STARTED</a>
                                </div>
                        </div>
                </div>
        </body>
</html>
```

Login.html

```
<!DOCTYPE html>
<html>
        <head>
                <meta charset="utf-8">
                <title>Login</title>
                <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
                <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
                <style>


* {
        box-sizing: border-box;
        font-family: -apple-system, BlinkMacSystemFont, "segoe ui", roboto, oxygen,
ubuntu, cantarell, "fira sans", "droid sans", "helvetica neue", Arial, sans-serif;
        font-size: 16px;
        -webkit-font-smoothing: antialiased;
        -moz-osx-font-smoothing: grayscale;
}
body {
        height: 100vh;
        width: 100%;
        background-image: url('../static/background2.jpg');
        background-size:cover;
        background-attachment: fixed;
        background-repeat: no-repeat;
        margin: 0;
}
```

28

```css
.login, .register {
        width: 400px;
        background-color: #ffffff;
        box-shadow: 0 0 9px 0 rgba(0, 0, 0, 0.3);
        margin: 100px auto;
}
.login h1, .register h1 {
        text-align: center;
        color: #5b6574;
        font-size: 24px;
        padding: 20px 0 20px 0;
        border-bottom: 1px solid #dee0e4;
}
.login .links, .register .links {
        display: flex;
        padding: 0 15px;
}
.login .links a, .register .links a {
        color: #adb2ba;
        text-decoration: none;
        display: inline-flex;
        padding: 0 10px 10px 10px;
        font-weight: bold;
}
.login .links a:hover, .register .links a:hover {
        color: #9da3ac;
}
.login .links a.active, .register .links a.active {
        border-bottom: 3px solid #3274d6;
        color: #3274d6;
}
.login form, .register form {
        display: flex;
        flex-wrap: wrap;
        justify-content: center;
        padding-top: 20px;
}
```

29

```css
.login form label, .register form label {
        display: flex;
        justify-content: center;
        align-items: center;
        width: 50px;
        height: 50px;
        background-color: #3274d6;
        color: #ffffff;
}
.login form input[type="password"], .login form input[type="text"], .login form
input[type="email"], .register form input[type="password"], .register form
input[type="text"], .register form input[type="email"] {
        width: 310px;
        height: 50px;
        border: 1px solid #dee0e4;
        margin-bottom: 20px;
        padding: 0 15px;
}
.login form input[type="submit"], .register form input[type="submit"] {
        width: 100%;
        padding: 15px;
        margin-top: 20px;
        background-color: #3274d6;
        border: 0;
        cursor: pointer;
        font-weight: bold;
        color: #ffffff;
        transition: background-color 0.2s;
}
.login form input[type="submit"]:hover, .register form input[type="submit"]:hover {
        background-color: #2868c7;
        transition: background-color 0.2s;
}
.navtop {
        background-color: #2f3947;
        height: 60px;
        width: 100%;
```

```css
        border: 0;
}
.navtop div {
        display: flex;
        margin: 0 auto;
        width: 1000px;
        height: 100%;
}
.navtop div h1, .navtop div a {
        display: inline-flex;
        align-items: center;
}
.navtop div h1 {
        flex: 1;
        font-size: 24px;
        padding: 0;
        margin: 0;
        color: #eaebed;
        font-weight: normal;
}
.navtop div a {
        padding: 0 20px;
        text-decoration: none;
        color: #c1c4c8;
        font-weight: bold;
}
.navtop div a i {
        padding: 2px 8px 0 0;
}
.navtop div a:hover {
        color: #eaebed;
}
body.loggedin {
        background-color: #f3f4f7;
}
.content {
        width: 1000px;
```

```css
        margin: 0 auto;
}
.content h2 {
        margin: 0;
        padding: 25px 0;
        font-size: 22px;
        border-bottom: 1px solid #e0e0e3;
        color: #4a536e;
}
.content > p, .content > div {
        box-shadow: 0 0 5px 0 rgba(0, 0, 0, 0.1);
        margin: 25px 0;
        padding: 25px;
        background-color: #fff;
}
.content > p table td, .content > div table td {
        padding: 5px;
}
.content > p table td:first-child, .content > div table td:first-child {
        font-weight: bold;
        color: #4a536e;
        padding-right: 15px;
}
.content > div p {
        padding: 5px;
        margin: 0 0 10px 0;
}
        </style>

        </head>
        <body>
                <div class="login">
                        <h1>Login</h1>
                        <div class="links">
                                <a href="{{ url_for('login') }}" class="active">Login</a>
                                <a href="{{ url_for('register') }}">Register</a>
                        </div>
```

```html
                              <form action="{{ url_for('login') }}" method="post">
                                  <label for="username">
                                      <i class="fas fa-user"></i>
                                  </label>
                                  <input type="text" name="username"
placeholder="Username" id="username" required>
                                  <label for="password">
                                      <i class="fas fa-lock"></i>
                                  </label>
                                  <input type="password" name="password"
placeholder="Password" id="password" required>
                                  <div class="msg">{{ msg }}</div>
                                  <input type="submit" value="Login">
                              </form>
                          </div>
                  </body>
</html>
```

7.2 FEATURE 2

Register.html

```html
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Register</title>
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
        <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
        <style>


* {
    box-sizing: border-box;
    font-family: -apple-system, BlinkMacSystemFont, "segoe ui", roboto, oxygen,
ubuntu, cantarell, "fira sans", "droid sans", "helvetica neue", Arial, sans-serif;
    font-size: 16px;
    -webkit-font-smoothing: antialiased;
```

```css
        -moz-osx-font-smoothing: grayscale;
}
body {
        height: 100vh;
        width: 100%;
        background-image: url('../static/background2.jpg');
        background-size:cover;
        background-attachment: fixed;
        background-repeat: no-repeat;
        margin: 0;
}
.login, .register {
        width: 400px;
        background-color: #ffffff;
        box-shadow: 0 0 9px 0 rgba(0, 0, 0, 0.3);
        margin: 100px auto;
}
.login h1, .register h1 {
        text-align: center;
        color: #5b6574;
        font-size: 24px;
        padding: 20px 0 20px 0;
        border-bottom: 1px solid #dee0e4;
}
.login .links, .register .links {
        display: flex;
        padding: 0 15px;
}
.login .links a, .register .links a {
        color: #adb2ba;
        text-decoration: none;
        display: inline-flex;
        padding: 0 10px 10px 10px;
        font-weight: bold;
}
.login .links a:hover, .register .links a:hover {
        color: #9da3ac;
```

```css
}
.login .links a.active, .register .links a.active {
        border-bottom: 3px solid #3274d6;
        color: #3274d6;
}
.login form, .register form {
        display: flex;
        flex-wrap: wrap;
        justify-content: center;
        padding-top: 20px;
}
.login form label, .register form label {
        display: flex;
        justify-content: center;
        align-items: center;
        width: 50px;
        height: 50px;
        background-color: #3274d6;
        color: #ffffff;
}
.login form input[type="password"], .login form input[type="text"], .login form
input[type="email"], .register form input[type="password"], .register form
input[type="text"], .register form input[type="email"] {
        width: 310px;
        height: 50px;
        border: 1px solid #dee0e4;
        margin-bottom: 20px;
        padding: 0 15px;
}
.login form input[type="submit"], .register form input[type="submit"] {
        width: 100%;
        padding: 15px;
        margin-top: 20px;
        background-color: #3274d6;
        border: 0;
        cursor: pointer;
        font-weight: bold;
```

35

```css
        color: #ffffff;
        transition: background-color 0.2s;
}
.login form input[type="submit"]:hover, .register form input[type="submit"]:hover {
        background-color: #2868c7;
        transition: background-color 0.2s;
}
.navtop {
        background-color: #2f3947;
        height: 60px;
        width: 100%;
        border: 0;
}
.navtop div {
        display: flex;
        margin: 0 auto;
        width: 1000px;
        height: 100%;
}
.navtop div h1, .navtop div a {
        display: inline-flex;
        align-items: center;
}
.navtop div h1 {
        flex: 1;
        font-size: 24px;
        padding: 0;
        margin: 0;
        color: #eaebed;
        font-weight: normal;
}
.navtop div a {
        padding: 0 20px;
        text-decoration: none;
        color: #c1c4c8;
        font-weight: bold;
}
```

```css
.navtop div a i {
        padding: 2px 8px 0 0;
}
.navtop div a:hover {
        color: #eaebed;
}
body.loggedin {
        background-color: #f3f4f7;
}
.content {
        width: 1000px;
        margin: 0 auto;
}
.content h2 {
        margin: 0;
        padding: 25px 0;
        font-size: 22px;
        border-bottom: 1px solid #e0e0e3;
        color: #4a536e;
}
.content > p, .content > div {
        box-shadow: 0 0 5px 0 rgba(0, 0, 0, 0.1);
        margin: 25px 0;
        padding: 25px;
        background-color: #fff;
}
.content > p table td, .content > div table td {
        padding: 5px;
}
.content > p table td:first-child, .content > div table td:first-child {
        font-weight: bold;
        color: #4a536e;
        padding-right: 15px;
}
.content > div p {
        padding: 5px;
        margin: 0 0 10px 0;
```

37

```html
        }

        </style>
        </head>
        <body>
                <div class="register">
                        <h1>Register</h1>
                        <div class="links">
                                <a href="{{ url_for('login') }}">Login</a>
                                <a href="{{ url_for('register') }}"
    class="active">Register</a>
                        </div>
                        <form action="{{ url_for('register') }}" method="post"
    autocomplete="off">
                                <label for="username">
                                        <i class="fas fa-user"></i>
                                </label>
                                <input type="text" name="username"
    placeholder="Username" id="username" required>
                                <label for="password">
                                        <i class="fas fa-lock"></i>
                                </label>
                                <input type="password" name="password"
    placeholder="Password" id="password" required>
                                <label for="email">
                                        <i class="fas fa-envelope"></i>
                                </label>
                                <input type="email" name="email"
    placeholder="Email" id="email" required>
                                <div class="msg">{{ msg }}</div>
                                <input type="submit" value="Register">
                        </form>
                </div>
        </body>
</html>
```

38

Register.html

```html
<!DOCTYPE html>
<html>
<head>
<link class="jsbin"
href="http://ajax.googleapis.com/ajax/libs/jqueryui/1/themes/base/jquery-ui.css"
rel="stylesheet" type="text/css" />
<link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700"
rel="stylesheet">
<script class="jsbin"
src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
<script class="jsbin" src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8.0/jquery-
ui.min.js"></script>
<meta charset=utf-8 />
<title>Nutrition Assitant Application</title>
<!--[if IE]>
  <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
<style>
  html, body{
    display: flex;
    font-family: Roboto, Arial, sans-serif;
    font-size: 15px;
     background-image: url('https://image.freepik.com/free-photo/diet-food-
background-concept-healthy-carbohydrates-carbs-products-fruits-vegetables-cereals-
nuts-beans-dark-blue-concrete-background-copy-space-frame_136595-2921.jpg');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
  }
  label{
    display: inline-block;
    background-color: #8ebf42;
    color: white;
    padding: 0.5rem;
    font-family: sans-serif;
    border-radius: 0.3rem;
```

```css
      cursor: grabbing;
      margin-top: 400px;
      margin-left: 150px;
     }
    fileChosen{
      margin-left: 0.3rem;
      font-family: sans-serif;
     }
    button{
      display: inline-block;
      background-color: #8ebf42;
      color: white;
      padding: 14px 0;
      margin: 10px 0;
      margin-left: 150px;
      border: none;
      cursor: grabbing;
      width: 50%;
      border-radius: 0.3rem;
     }
     span, h2{
     color: white;
     }
    img{
      margin-left: -400px;
      margin-bottom: 70%;
      margin-top: 50px;
      text-indent: -9999px;
     }
     .alt{
     color: transparent;
     }
</style>
</head>
<body>
<h2> Welcome, to Nutriooze </h2>
<form action="out" method="POST" enctype="multipart/form-data">
```

```html
        <label >{{data}}</label>



        <button type="submit">GetNutritionInfo</button>
        <p>{{error}}</p>
    </form>
    <img id="blah" src="#" alt="your image" />
    </body>
</html>
```

Submission.html

```html
<!DOCTYPE html>
<html>
<head>
<link class="jsbin"
href="http://ajax.googleapis.com/ajax/libs/jqueryui/1/themes/base/jquery-ui.css"
rel="stylesheet" type="text/css" />
<link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700"
rel="stylesheet">
<script class="jsbin"
src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
<script class="jsbin" src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8.0/jquery-
ui.min.js"></script>
<meta charset=utf-8 />
<title>Nutrition Assistant Application</title>
<!--[if IE]>
  <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
<style>
  html, body{
    display: flex;
    font-family: Roboto, Arial, sans-serif;
    font-size: 15px;
    background-image: url('../static/background3.jpg');
```

```css
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
label{
  display: inline-block;
  background-color: #8ebf42;
  color: white;
  padding: 0.5rem;
  font-family: sans-serif;
  border-radius: 0.3rem;
  cursor: grabbing;
  margin-top: 400px;
  margin-left: 150px;
}
fileChosen{
  margin-left: 0.3rem;
  font-family: sans-serif;
}
button{
  display: inline-block;
  background-color: #8ebf42;
  color: white;
  padding: 14px 0;
  margin: 10px 0;
  margin-left: 150px;
  border: none;
  cursor: grabbing;
  width: 50%;
  border-radius: 0.3rem;
}
span, h2{
color: white;
}
```

```css
        img{
         margin-left: -400px;
         margin-bottom: 70%;
         margin-top: 50px;
         text-indent: -9999px;
         }
         .alt{
         color: transparent;
         }
```
```html
</style>
</head>
<body>
<h2> Welcome, to Nutriooze </h2>
<form action="/pythonlogin/submission/display" method="POST"
enctype="multipart/form-data">
   <input type="file" id="fileInput" name="food" onClick="func1(this);" hidden
required/>
   <label for="fileInput">Choose File</label>
   <span id="fileChosen">No file chosen</span>
   <script>
     function func1(input){
       document.querySelector("#fileInput").onchange = function(){
         document.querySelector("#fileChosen").textContent = this.files[0].name;
         if (input.files && input.files[0]) {
           var reader = new FileReader();

           reader.onload = function (e) {
           $('#blah')
           .attr('src', e.target.result)
           .height(300)
           };

           reader.readAsDataURL(input.files[0]);
         }
       }
     }
```

```html
        </script>
        <button type="submit">Submit</button>
        <p>{{error}}</p>
    </form>
    <img id="blah" src="#" alt="your image" />
    </body>
</html>
```

8. TESTING

8.1 TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on "HOW" to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.

Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

S.NO	Scenario	Input	Excepted output	Actual output
1	User login	User name and password	Login	Login success.
2	Upload Image	Upload input image (fruits and vegetables)	Predicting calorie, fat and ingredients of given food image	Details are stored in a database.

8.2 USER ACCEPTANCE TESTING

This sort of testing is carried out by users, clients, or other authorised bodies to identify the requirements and operational procedures of an application or piece of software. The most crucial stage of testing is acceptance testing since it determines whether or not the customer will accept the application or programme. It could entail the application's U.I., performance, usability, and usefulness. It is also referred to as end-user testing, operational acceptance testing, and user acceptance testing (UAT).

9. RESULTS

9.1 PERFORMANCE METRICS
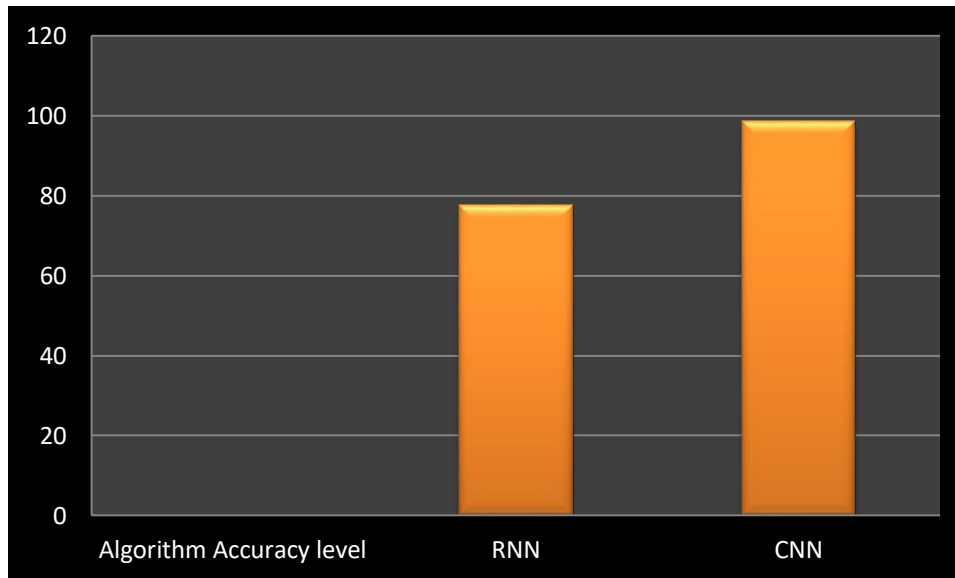
10. ADVANTAGES & DISADVANTAGES

ADVANTAGE

- Provide the nutrition content and ingredients of Multifoods
- Helps for fitness people to maintain and know the proteins and calories of the food
- Gives accurate results in real-time application

DISADVANTAGE

- Hard to know the details of nutrition and calories of food
- Doesn't ask to provide the users health condition
- Required more time to know the Multifoods

11. CONCLUSION

This project study suggests a technique for an automated food nutrition detection system that can estimate the quantity of nutrients in food. As of now, the machine can classify the meal into one of the numerous categories provided in the dataset. The categorization made use of the popular food dataset. Using a deep learning approach, the food photographs were categorised into the proper categories. It may be possible to improve the classification process by removing noise from the dataset. The same study can be conducted using a larger dataset, more classes, and more images in each class because doing so improves accuracy by teaching the algorithm new features and reduces loss rates. The weights of the model can be stored and used to develop designs for calorie extraction, food categorization, and image classification.

\

12. FUTURE SCOPE

Using a deep learning approach, the research study's food images are categorised into the relevant groupings. The classification task might be improved in the future by removing noise from the dataset. The same study may be conducted using a larger dataset, more classes, and more images in each class because doing so improves accuracy by teaching the algorithm new characteristics and reduces loss rates. A web or mobile application that categorises images and also extracts the calories from the food that has been identified may be created using the model's weights, which can be saved and used later.

13. APPENDIX

SOURCE CODE

```
import Flask, render_template, request, redirect, url_for, session

import requests, json, os
import ibm_db
import re

import cv2

app = Flask(__name__)

app.secret_key = 'a'


conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-
ba32-
21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30376;SEC
URITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=jqr24896;PWD
=XEJTmHY5JprWZngW","","")



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

@app.route('/pythonlogin/', methods=['GET', 'POST'])
def login():
    global userid
    msg=''
```

```python
        if request.method =='POST':
            username = request.form['username']
            password =request.form['password']
            sql ="SELECT * FROM users WHERE username =? AND password=?"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(stmt,1,username)
            ibm_db.bind_param(stmt,2,password)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            print (account)
            if account:
                session['loggedin']=True
                session['id'] = account ['USERNAME']
                userid = account['USERNAME']
                session['username'] = account['USERNAME']
                msg = 'logged in successfully !'


                return render_template('submission.html',msg = msg)


            else:
                msg ='Incorrect username / password !'
        return render_template('login.html',msg=msg)


@app.route('/pythonlogin/register', methods=['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM users WHERE username = ?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
```

```python
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg ='Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+',email):
            msg ='Invaild email address !'
        elif not re.match(r'[A-Za-z0-9]+',username):
            msg = 'Name must contain only characters and numbers!'
        else:
            insert_sql = "INSERT INTO users VALUES (?,?,?)"
            prep_stmt= ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1 , username)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, password)
            ibm_db.execute(prep_stmt)

            sendmsg(email,' you have successfully registered !')
            msg = ' you have successfully registered !'
    elif request.method == 'POST':
        # Form is empty... (no POST data)
        msg = 'Please fill out the form!'
    # Show registration form with message (if any)
    return render_template('register.html', msg=msg)
@app.route('/submission')
def submission():

    return render_template('submission.html')


@app.route('/pythonlogin/submission/display', methods = ["POST", "GET"])
def display():
    if request.method == "POST":
```

```python
image = request.files["food"]
image.save('static/Out/Test.jpg')


import tensorflow as tf
classifierLoad = tf.keras.models.load_model('model.h5')


import numpy as np
from keras.preprocessing import image


test_image = image.load_img('static/Out/Test.jpg', target_size=(200, 200))
img1 = cv2.imread('static/Out/Test.jpg')
# test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis=0)
result = classifierLoad.predict(test_image)


print(result)


out = ''
fer = ''
if result[0][0] == 1:
    out = "APPLES"
elif result[0][1] == 1:
    out = "Badam"
elif result[0][2] == 1:
    out = "Badam Drink"
elif result[0][3] == 1:
    out = "Banana"
elif result[0][4] == 1:
    out = "Beef Steak"
elif result[0][5] == 1:
    out = "BeetrootFry"


elif result[0][6] == 1:
    out = "Biriyani"
```

54

```python
        elif result[0][7] == 1:
            out = "Biscuits"
        elif result[0][8] == 1:
            out = "BitterGuardFry"
        elif result[0][9] == 1:
            out = "Boiledegg"
        elif result[0][10] == 1:
            out = "Bread with Peanutbutter"

        elif result[0][11] == 1:
            out = "BreadandJam"
        elif result[0][12] == 1:
            out = "Badam"
        elif result[0][13] == 1:
            out = "Burger"
        elif result[0][14] == 1:
            out = "CapsicumCurry"
        elif result[0][15] == 1:
            out = "Cashew"
        elif result[0][16] == 1:
            out = "Chappathi"

        elif result[0][17] == 1:
            out = "Cheeseballs"
        elif result[0][18] == 1:
            out = "ChilliBeef"
        elif result[0][19] == 1:
            out = "Chocolate"
        elif result[0][20] == 1:
            out = "ChocolateIcecream"
        elif result[0][21] == 1:
            out = "ChoolapooriwithChanna"
        elif result[0][22] == 1:
            out = "CoffeeorLatte"
```

```python
        elif result[0][23] == 1:
            out = "CrabMasala"
        elif result[0][24] == 1:
            out = "Cucumber"
        elif result[0][25] == 1:
            out = "Curdrice"
        elif result[0][26] == 1:
            out = "Dosa"


        foodName= out

        session["out"]=foodName

        return render_template('Result.html', data=session["out"])

    else:
        return render_template('submission.html')

@app.route("/pythonlogin/submission/out", methods=['GET', 'POST'])
def out():
    if request.method == "POST":
        nutrients = {}
        USDAapiKey = 'flYmBbhNL4RXqkJ7M80adL5dNKvn3THtM0eQ48Er'

        foodName = session["out"]
        response = requests.get(

'https://api.nal.usda.gov/fdc/v1/foods/search?api_key={}&query={}&requireAllWord
s={}'.format(USDAapiKey,

                                                        foodName,
                                                        True))
```

```python
        data = json.loads(response.text)

        concepts = data['foods'][0]['foodNutrients']

        arr = ["Sugars", "Energy", "Vitamin A", "Vitamin D", "Vitamin B", "Vitamin
C", "Protein", "Fiber", "Iron",
            "Magnesium",
            "Phosphorus", "Cholestrol", "Carbohydrate", "Total lipid (fat)", "Sodium",
"Calcium", ]
        for x in concepts:
            if x['nutrientName'].split(',')[0] in arr:
                if (x['nutrientName'].split(',')[0] == "Total lipid (fat)"):
                    nutrients['Fat'] = str(x['value']) + " " + x['unitName']
                else:
                    nutrients[x['nutrientName'].split(',')[0]] = str(x['value']) + " " +
x['unitName']


        return render_template('display.html', x=foodName, data=nutrients,
account=session['username'])



def sendmsg(Mailid,message):
    import smtplib
    from email.mime.multipart import MIMEMultipart
    from email.mime.text import MIMEText
    from email.mime.base import MIMEBase
    from email import encoders

    fromaddr = "sampletest685@gmail.com"
    toaddr = Mailid


    # instance of MIMEMultipart
    msg = MIMEMultipart()


    # storing the senders email address
    msg['From'] = fromaddr
```

```python
        # storing the receivers email address
        msg['To'] = toaddr

        # storing the subject
        msg['Subject'] = "Alert"

        # string to store the body of the mail
        body = message

        # attach the body with the msg instance
        msg.attach(MIMEText(body, 'plain'))

        # creates SMTP session
        s = smtplib.SMTP('smtp.gmail.com', 587)

        # start TLS for security
        s.starttls()

        # Authentication
        s.login(fromaddr, "hneucvnontsuwgpj")

        # Converts the Multipart msg into a string
        text = msg.as_string()

        # sending the mail
        s.sendmail(fromaddr, toaddr, text)

        # terminating the session
    if __name__=='__main__':
app.run(host='0.0.0.0',debug = True, port = 5000)
```
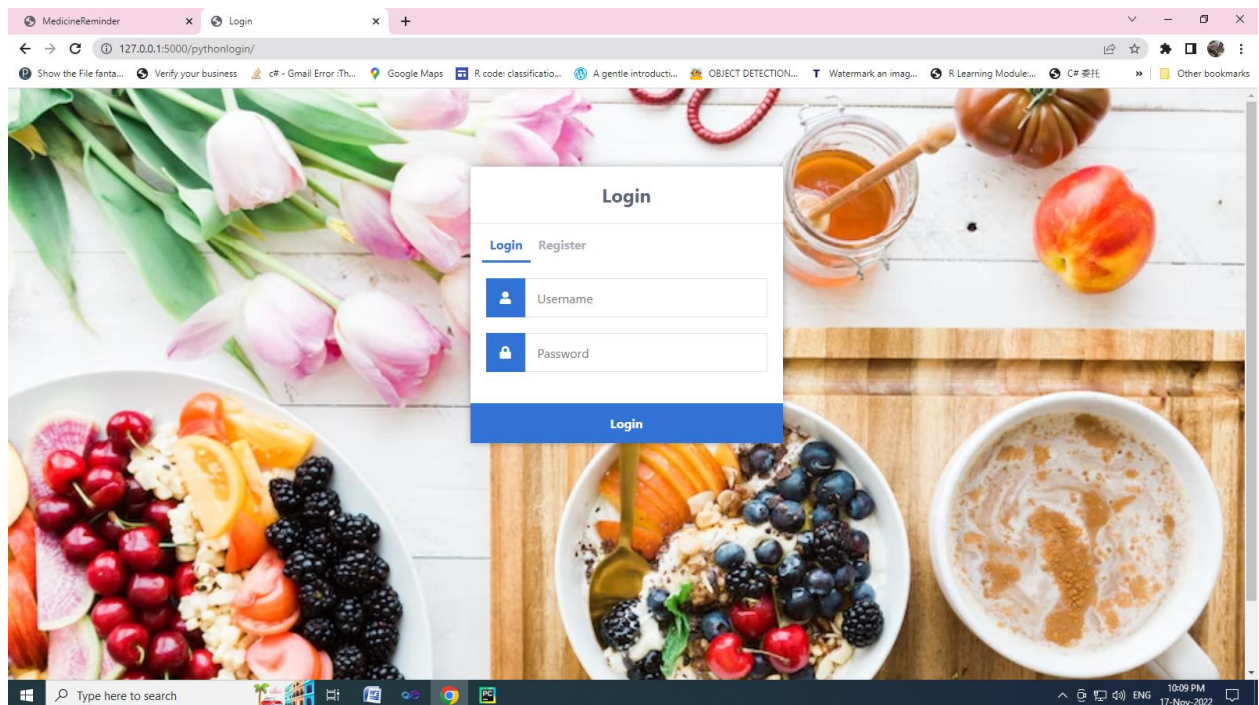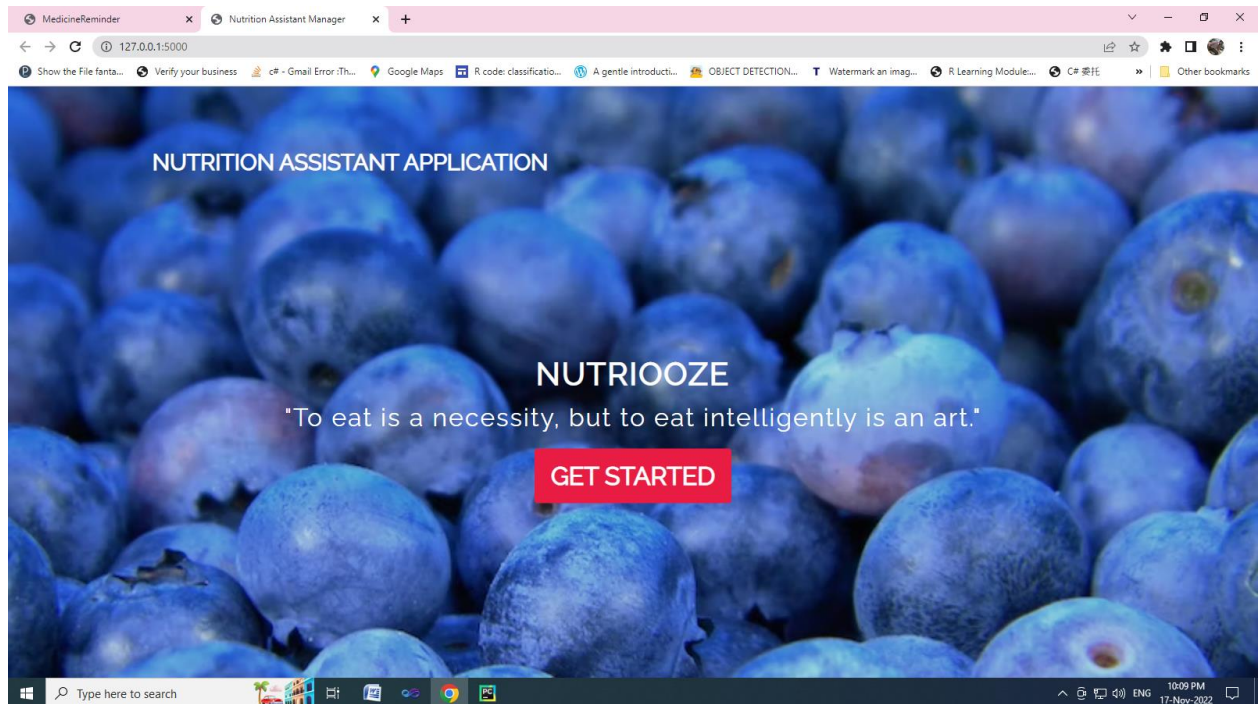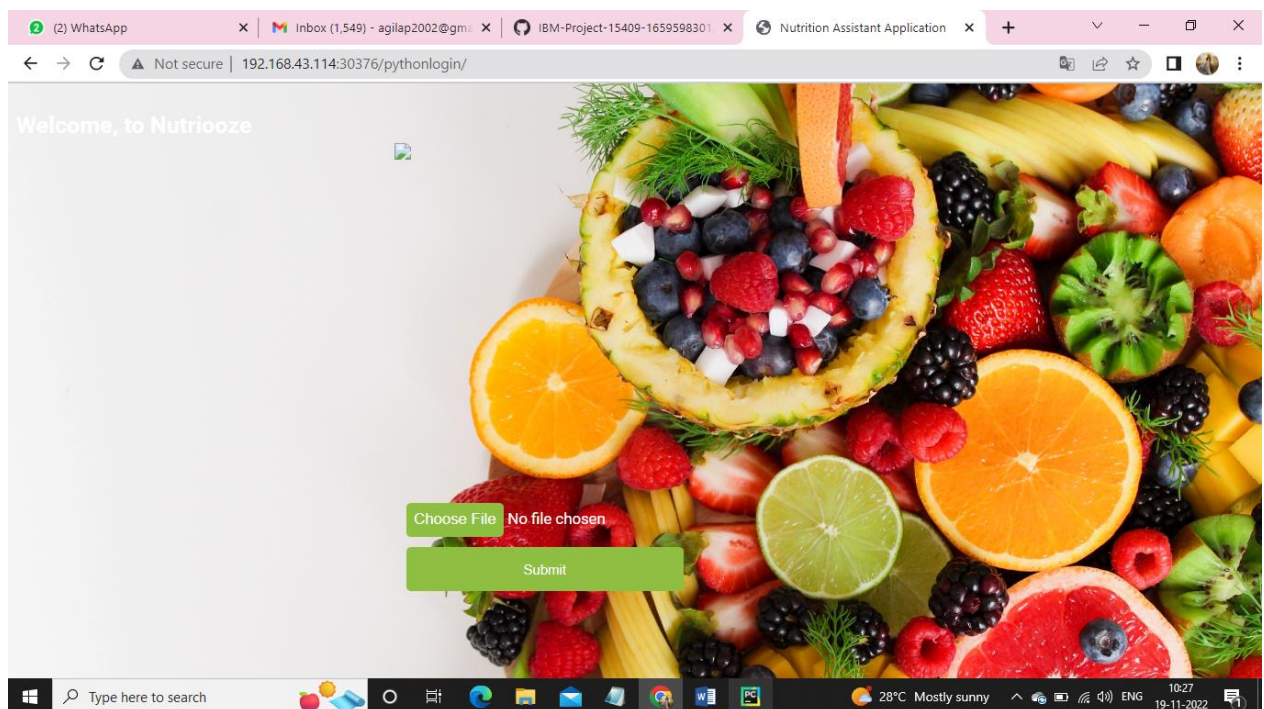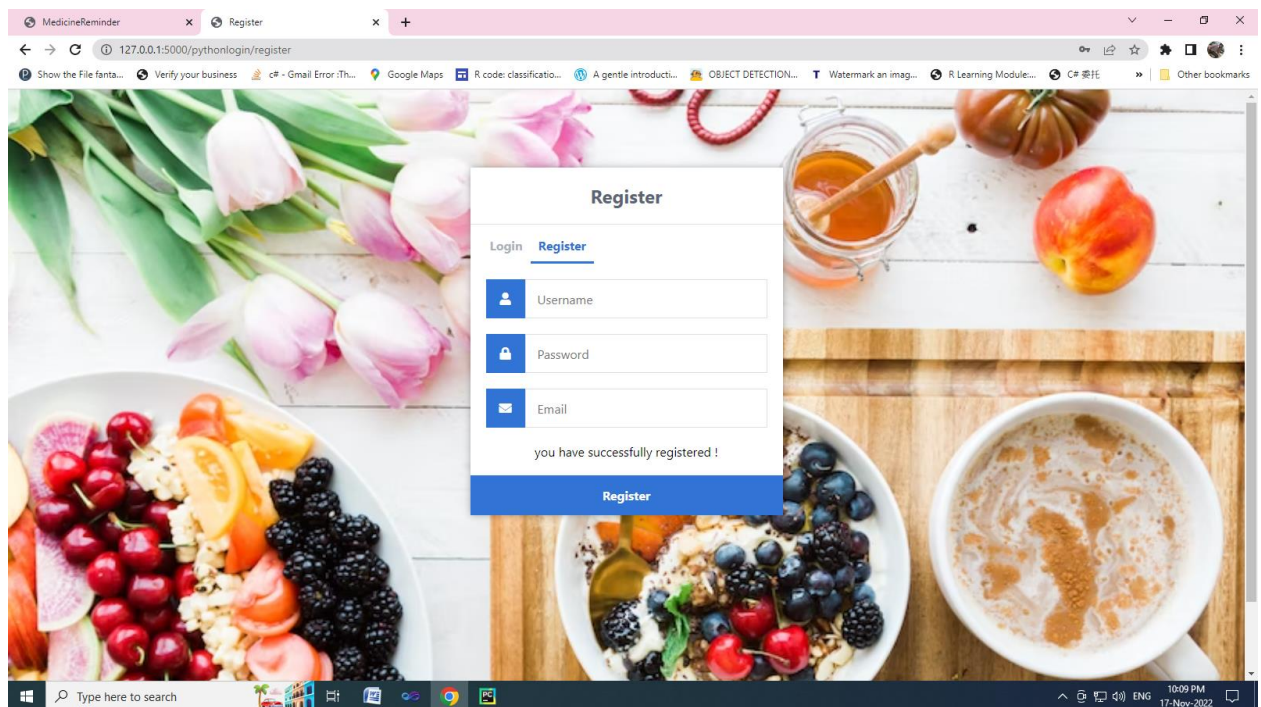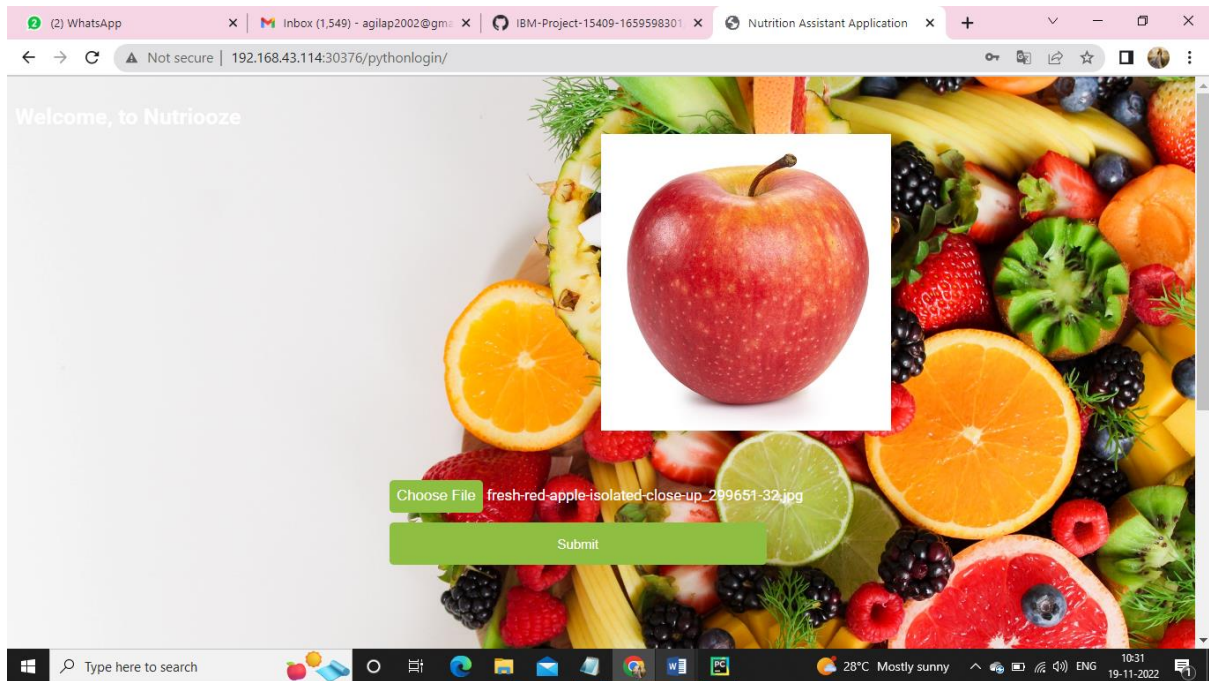
SCREENSHOTS:

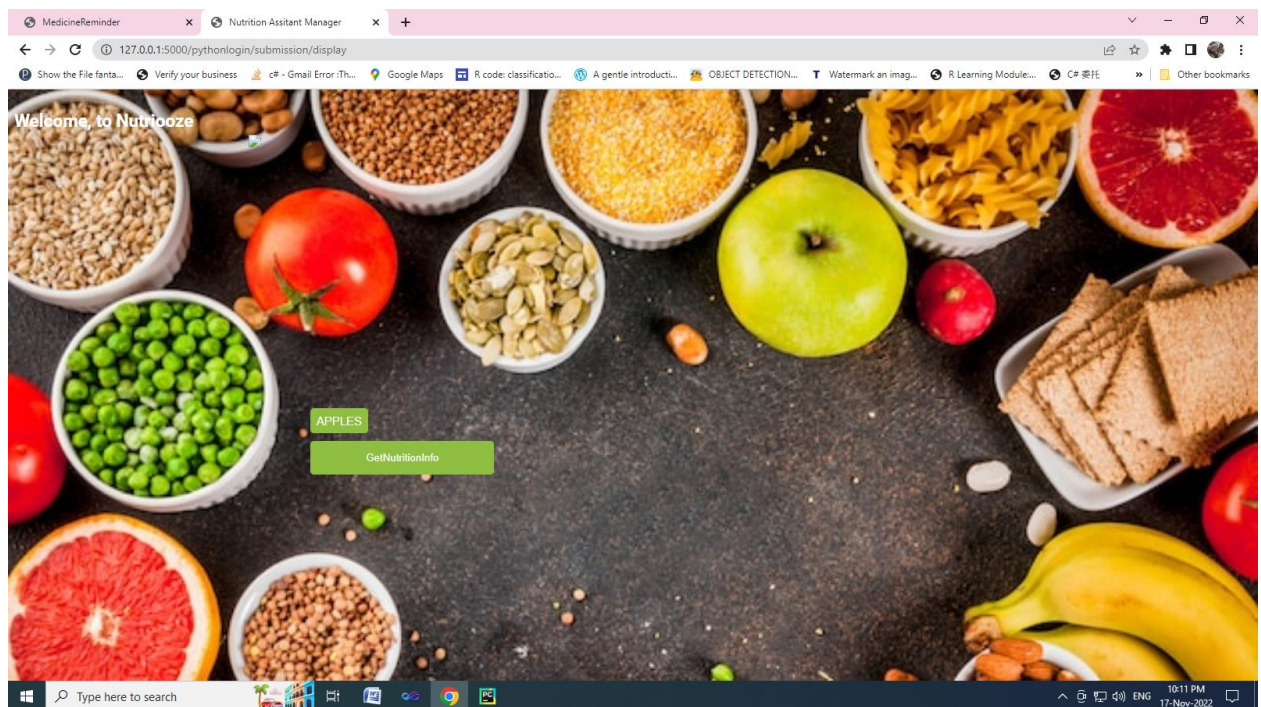Nutritional Values for APPLES:

Nutrients	Nutritional Values
Fat	0.0 G
Calcium	0.0 MG
Iron	0.23 MG
Sodium	0.0 MG
Vitamin A	65.0 IU
Vitamin C	3.1 MG
Protein	0.0 G
Carbohydrate	14.3 G
Energy	52.0 KCAL
Sugars	10.4 G
Fiber	3.2 G

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-15409-1659598301.git

PROJECT DEMO LINK:

https://youtu.be/NB5FA54IpQk