AI-powered Nutrition Analyzer for Fitness Enthusiasts

Industry Mentor(s) Name: Sri Tulasi
Faculty Mentor(s) Name: SUGANYA VISWANATHAN

Team Membors(s) Name:

Jino Rohit

Pranav RR

Sharath N

Siddharth S

 # IBM-Project-10868-1659240779

1. Introduction

1.1 Overview

Food is a necessity for human life and has been addressed in numerous medical conventions. Modern dietary evaluation and nutrition analysis technologies give consumers more possibilities to explore nutrition patterns, comprehend their daily eating habits, and keep up a balanced diet.

The biggest challenge for fitness lovers is keeping track of their daily nutrition intake, which is crucial for staying in shape. But with today's busy world and the abundance of internet fitness resources, keeping track of your nutrition will become increasingly difficult and inaccurate.

Fitness fanatics typically stick to their diet programmes, but they have trouble keeping track of the food's nutritional value. Fruits are easily digestible since they are high in vitamins, fibre, and minerals, but eating too much of them can cause weight gain and even diabetes because fruit contains natural sugar.

Fitness aficionados eat a diet high in fruits, vegetables, foods high in protein, and low in carbohydrates. However, it is difficult to identify and keep track of the nutritional components of unknown foods, such as fibre, protein, and nutrition.

1.2 Purpose

The main aim of the project is to building a model which is used for classifying the fruit depends on the different characteristics like colour, shape, texture etc.

Here the user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).

2. Literature Review:

2.1 Existing Problem

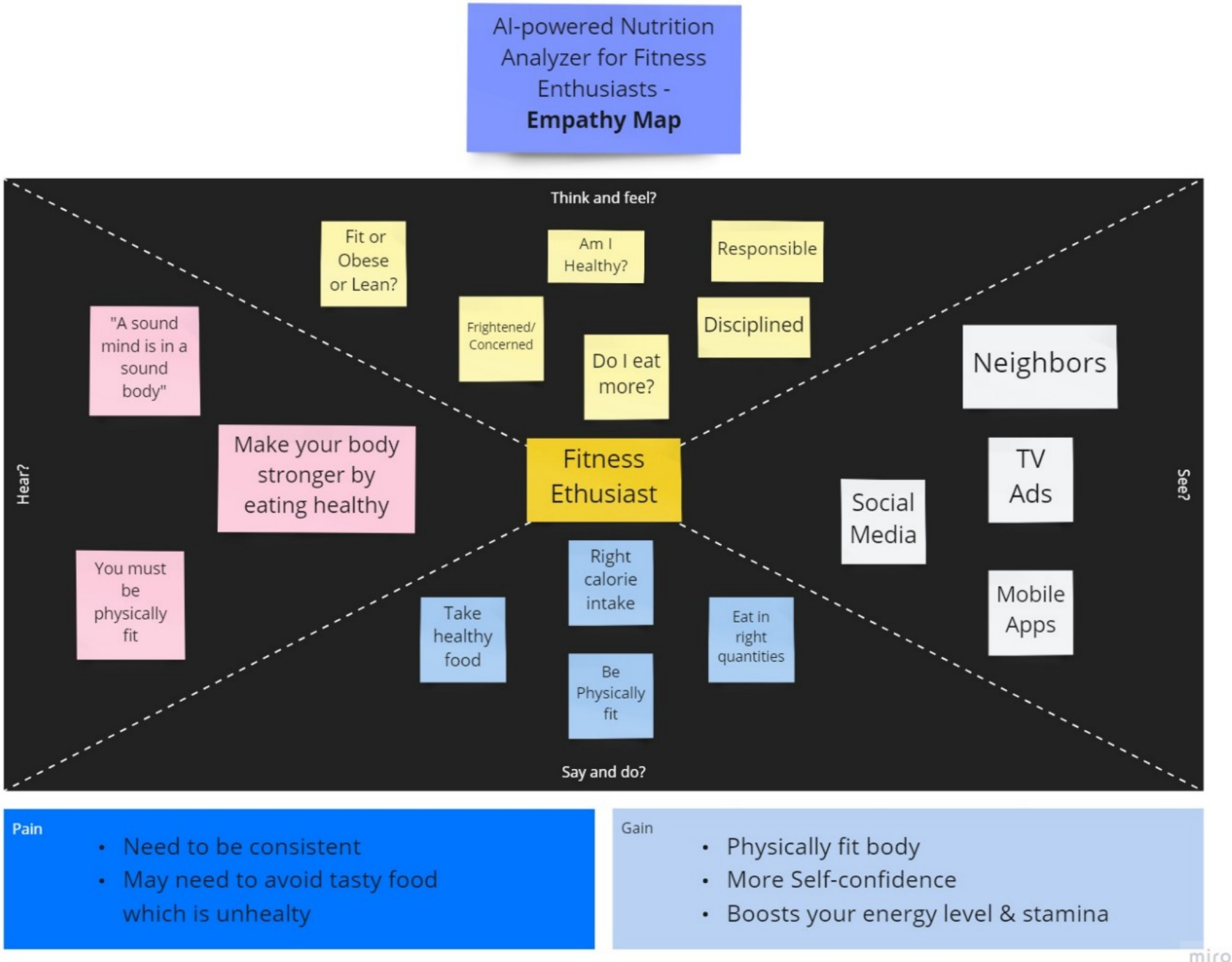
Nutrition is very important for the growth of a human body. Nutritional Analysis ensures that the food has optimal requirement of vitamins and minerals wherein the examining of nutrition in food helps in understanding about the fat proportion, carbohydrates dilution, proteins, fiber, sugar, etc. Another thing we need to take care of is not to exceed daily calorie needs. If exceeded, we maybe end up being obese.

2.2 References

Paper / Title	Author	Year	Journal	Objective	Proposed Technique	Limitations/ Improvements
DeepFood: Deep Learning-Based Food Image Recognition for Computer-Aided Dietary Assessment	Chang Liu, Yu Cao, Yan Luo, Guanling Chen, Vinod Vokkarane, and Yunsheng Ma	2016	Springer International Publishing Switzerland	To propose a new CNN architecture for food image recognition and apply benchmark on UEC-256 and Food-101	A new architecture was proposed based on the backbones of LeNet, AlexNet and GoogleNet. After convolutions, it was followed by sub-sampling to reduce dimensions and FC layers.	The inference time is extremely long for even a single image and hence not feasible to deploy in real time
AN IMAGE ANALYSIS SYSTEM FOR DIETARY ASSESSMENT AND EVALUATION	Fengqing Zhu, Marc Bosch, Carol J. Boushey and Edward J. Delp	2011	NCBI	To use a mobile device with a built-in camera, network connectivity, integrated image analysis and visualization tools, and a nutrient database, to allow a user to easily record foods eaten. Images acquired before and after foods are eaten can be used to estimate the amount of food consumed.	1) Image Segmentation 2) Classification using SVM 3) Volume Estimation with the help of Camera Calibration	Not be able to recognize every food or differentiate between similar looking foods.
EVIDENCE-BASED DEVELOPMENT OF A MOBILE TELEPHONE FOOD RECORD	Bethany L Six, TusaRebecca E Schap, Anand Mariappan,	2011	NCBI	(1) to test whether participants' proficiency with the mpFR improved after training and repeated use, and (2) to measure changes in perceptions regarding use of the mpFR after training and repeated use.	1) Image Segmentation 2) Volume Estimation 3) FNDDS Indexing Nutrient Info	Needs to accommodate the lifestyles of its users to ensure useful images and continuous use throughout the day or multiple days.
AUTOMATIC FRUIT RECOGNITION: A SURVEY AND NEW RESULTS USING RANGE/ATTENUATION IMAGES	Jimenez A, Jain A, Ceres R, Pons J.	1999	Science Direct	To recognize spherical fruits in different situations such as shadows, bright areas, occlusions and overlapping fruits.	Two images represent the azimuth and elevation angles the attenuation is in *ATTE*(*x, y*) and the reflectance image *REFL*(*x, y*). The image analysis process uses the images obtained from the scanner to detect the position of the fruits by thresholding and clustering.	Cannot work with low resolution images.
FOOD IMAGE ANALYSIS AND DIETARY ASSESSMENT VIA DEEP MODEL	Landu Jiang	2020	Research Gate	To design and implement a system for food image analysis - output the amount of nutritional ingredients of each food items from daily captured images. A thorough dietary assessment report will be generated based on what you have during the meal.	Extract the regions of interests (ROIs) by ap-plying the Region Proposal Network derived from the Faster R-CNN model. Apply Convolutional Neural Network (CNN) on selected RoIs and classify them into different food item categories. A regression module is also used to locate the food coordinates in the image.	To provide a healthy diet, an automatic diet calculator.
DEEP-LEARNING-ASSISTED MULTI-DISH FOOD RECOGNITION APPLICATION FOR DIETARY INTAKE REPORTING	Ying-Chieh Liu	2022	Research Gate	To integrate ML innovations of a realistic mobile health application using mobile ICT and AI technology to allow people to report their dietary intake easily and accurately under real conditions.	Adopted EfficientDet-D1 with EfficientNet-B1 as the backbone. EfficientDet detector architecture with EfficientNet was selected	Yet to be integrated with a mobile app or web application.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Empathy Map Canvas

Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article

Need some inspiration?
See a finished version of this template to kickstart your work.

Open example

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
AI-powered Nutrition Analyzer for Fitness Enthusiasts

Key rules of brainstorming
To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

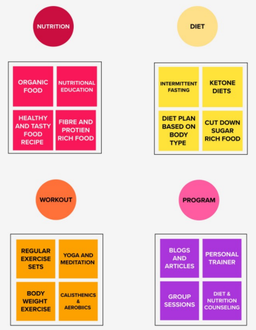
10 minutes

1) JINO ROHIT		2) SHARAT N		3) PRANAV RR		4) SIDDHARTH S	
FOOD RECIPE AND CUSHINE	FIBRE AND PROTIEN RICH FOOD	PROPER SLEEP	PERSONAL TRAINERS	REGULAR EXERCISE SETS	BODY WEIGHT EXERCISE	HYDRATION TRACKER	INTERMITTENT FASTING
CALORIE COUNTER	ORGANIC FOOD	DIET PLAN BASED ON BODY TYPE	KETONE DIETS	YOGA AND MEDITATION SESSIONS	BALANCED DIET PLANS	DAILY & WEEKLY GOALS	AEROBICS

3 Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

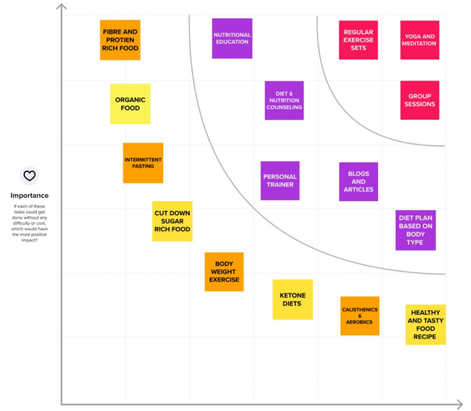
20 minutes



4 Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template

Share template feedback

3.3 Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Filling the gap of not knowing the nutrients intake and helping to maintain a proper balanced consumption for the customers/users.
2.	Idea / Solution description	A web based application that allows user to retrieve the nutritional facts of a food item by adding an image of it.
3.	Novelty / Uniqueness	• Web based easy to use interface. • Provides an image based facts retrieval of food. • Suggestion based on the facts. • Basic details and guidance based on the user profile.
4.	Social Impact / Customer Satisfaction	With the assistance of the data obtained, users get to maintain a healthy body, which makes them feel confident. In addition, it boosts their energy and stamina.
5.	Business Model (Revenue Model)	• Connecting to hospitals and centres that provide support for the user's. • Subscription based plan for users to unlock all features. • Incorporating advertisements of all categories. • Providing online resources for customers.
6.	Scalability of the Solution	• Making a user-friendly interface and making it all platform compatible. • Providing rewards for usage and sharing of application. • Using client data and feedback to improvise. • Collaborating with other parties for larger scale usage.

3.4 Proposed Solution Fit:

Problem-Solution fit canvas 2.0

Purpose / Vision AI-powered Nutrition Analyzer for Fitness Enthusiasts

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

This application can be used by people of age 18 and above who are interested to have a balance diet.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Lacking in-depth knowledge about necessary nutrients of items consumed.
Not sure of the individual components in the food.
Lethargic about their health
Expensive diet to follow.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

People have been using nutrients chart available in the market and also in the back of the food products, which are mostly not accurate.
Some applications such as Neutrino, FitGenie, etc. are available which are helpful to a certain extent

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

The customer are devoid of the details regarding the nutrients facts of the food they consume, which leads to imbalance in the intake of nutrients.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

being overweight or obese
tooth decay
high blood pressure
high cholesterol
heart disease and stroke
type-2 diabetes
osteoporosis
Cancer
depression
eating disorders.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits;

Firstly,
people use the nutrition facts on the back of the food products and physically calculate their consumption amount.
Secondly, people take the assistance if the online application for calculating the nutrients intake.

Identify strong TR & EM

3. TRIGGERS TR

What triggers custome… alling solar panels, reading ab… s.

Weight Gain, Nutrient Deficiencies, High Blood Pressure, High Cholesterol Levels, etc. all are triggering factors telecasted in various mediums.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Trepid -> intrepid
Stressed -> Calm
Lackadaisical -> enthusiastic

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a p…

existing solution needs accurate input of the food item, whereas, our solution identifies the food and displays the nutrition facts.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers tak… ?7

Browse the web for the calories of food they intake

8.2 OFFLINE
What kind of actions do customers … and use them for customer develo…

Try to intake appropriate calories.
Continent themselves from talking oily and fatty foods

Extract online & offline CH of BE

4. Requirement Analysis

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Facebook
FR-2	User Confirmation	Confirmation via Email
FR-3	Scan/Upload Food Items	Provide options to scan the fruits or upload the photo of the fruits the user takes.
FR-4	Show Nutritional Analysis	Use the AI Model to predict the fruits and an External API to know the calories and nutrition content of the corresponding fruit.
FR-5	Calorie Consumption Alert	Enable the user to set a daily calorie limit and alert the user if his food consumption exceeds the limit
FR-6	View calorie consumption of earlier days	Provide visualizing options for the user to compare the calorie intake of day, afternoon, and night of last month.

4.2 Non-Functional Requirements:

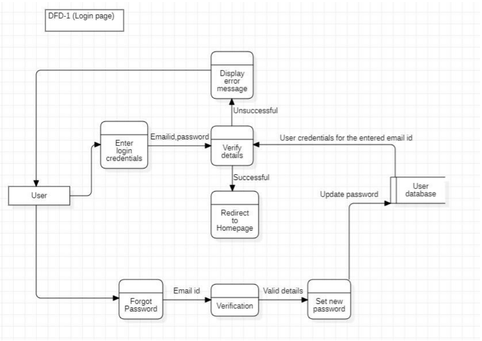
Following are the non-functional requirements of the proposed solution.

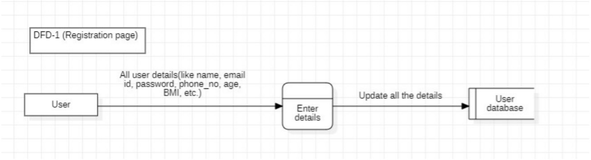
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	❖ The website's interface is user-friendly and easy to use. ❖ Provide scanning options for mobile users and uploading options for laptop users.
NFR-2	**Security**	❖ Users have to first signup with email and password. ❖ Only authenticated users can login and make use of the website. ❖ The user credentials are stored securely in the database with hashing techniques. ❖ Authorized admins only can make changes to the data. ❖ Managing and Enabling CORS, preventing SQL and other kinds of injections.
NFR-3	**Reliability**	The system must perform without failure in 95 percent of use cases during a month.
NFR-4	**Performance**	The AI Model API supports 500 users per hour must provide 6 second or less response time in a Chrome desktop browser.
NFR-5	**Availability**	The website must be available to the Indian users, 95 percent of the time every month during business hours IST.
NFR-6	**Scalability**	The system must be scalable enough to support 1000 visits at the same time while maintaining optimal performance.

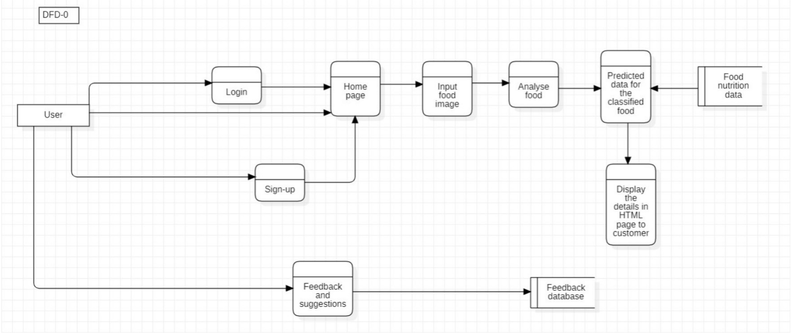
5. Project Design

5.1 Functional Requirements:

Data Flow Diagrams:







5.2 Solution & Technical Architecture:

Architecture:

Solution Architecture Diagram:

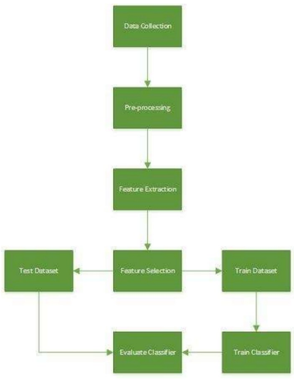


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python, Flask
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	Cloud based Storage
8.	External API-1	Used to get the Nutritional content of the fruits	Fruityvice
9.	External API-2	Fetches random food images	Foodish
10.	Machine Learning Model	Purpose of Machine Learning Model	Pytorch
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Docker images

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Pytorch, Sklearn, Seaborn
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	IAM user, SSL certs
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Web Server – HTML, CSS Application Server – Python Flask Database Server – IBM Cloud
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	IBM Cloud hosting
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	IBM Load Balance

5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
User	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
User	Registration	USN-3	As a user, I can register for the application through Gmail	I can login with my Gmail account	Low	Sprint-3
User	Login	USN-4	As a user, I can log into the application by entering email & password	I can login the Application by entering password	High	Sprint-1
User	Logout	USN-5	As a user, I can logout of the application	I can logout	High	Sprint-1
User	Profile	USN-6	As a user, I can view and update my profile	I can view and change my details and save them	Medium	Sprint-3
User	Home page	USN-7	As a user I can view the nutritional facts of the food by either uploading or selecting the food image	I can access the nutrition information's of fruit by uploading image from gallery or camera	High	Sprint-3
User	Suggest page	USN-8	As a user, I can view recommended nutritional intake based on my details	I can get fruit suggestion based on my details	Low	Sprint-4
User	Dashboard	USN-9	As a user, I can view my daily intake nutrition facts	I can view proper information about my nutrition and the calorie intake	Low	Sprint-4
User	Feedback page	USN-10	As a user, I can give feedback about the pages and details	I can give feedback on Sprint-4 any content or event	Low	Sprint-4
Administrator	Dash Board	USN-11	As an administrator, I can view and manages users, contents and everything	I can give feedback on any content or event I can manage users and contents in the application	Medium	Sprint-4

6. Project Planning and Scheduling

6.1 Sprint Planning and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Pre-requisites for Model Building	USN-1	As a developer I have to collect different type of data possible and other data supporting the model	3	High	Jino Rohit Pranav RR
Sprint-1	Model Building	USN-2	Development of the model with the prepared data set	4	High	Jino Rohit Sharat.N
Sprint-2	Home page	USN-3	As a user I can land into the main page of the website	3	High	Pranav RR Sharat.N
Sprint-2	Image page	USN-4	As a user I can upload the image of the food item	3	High	Pranav RR Siddharth S
Sprint-2	Image prediction page	USN-5	As a user I can view the nutritional facts of the food image uploaded.	3	High	Jino Rohit Sharat N
Sprint-3	Registration	USN-6	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Pranav RR Sharat.N
Sprint-3	Login	USN-7	As a user, I can log into the application by entering email & password	1	High	Pranav RR Sharat.N
Sprint-3	Logout	USN-8	As a user, I can logout of the application	1	High	Jino Rohit Sharat.N
Sprint-4	Dashboard	USN-9	As a user, I can view my daily intake nutrition facts	2	Medium	Jino Rohit Siddharth S
Sprint-4	Profile	USN-10	As a user, I can view and update my Profile	2	Medium	Pranav RR Sharat.N
Sprint-4	Dashboard	USN-11	As an administrator, I can view and mange users, contents and everything	1	Medium	Pranav RR Siddharth S
Sprint-4	Feedback page	USN-12	As a user, I can give feedback about the pages and details	1	Low	Jino Rohit Sharat.N
Sprint-4	Registration	USN-13	As a user, I can register for the application through Gmail	1	Medium	Pranav RR Siddharth S

Project Tracker, Velocity & Burndown Chart: (4 Marks)

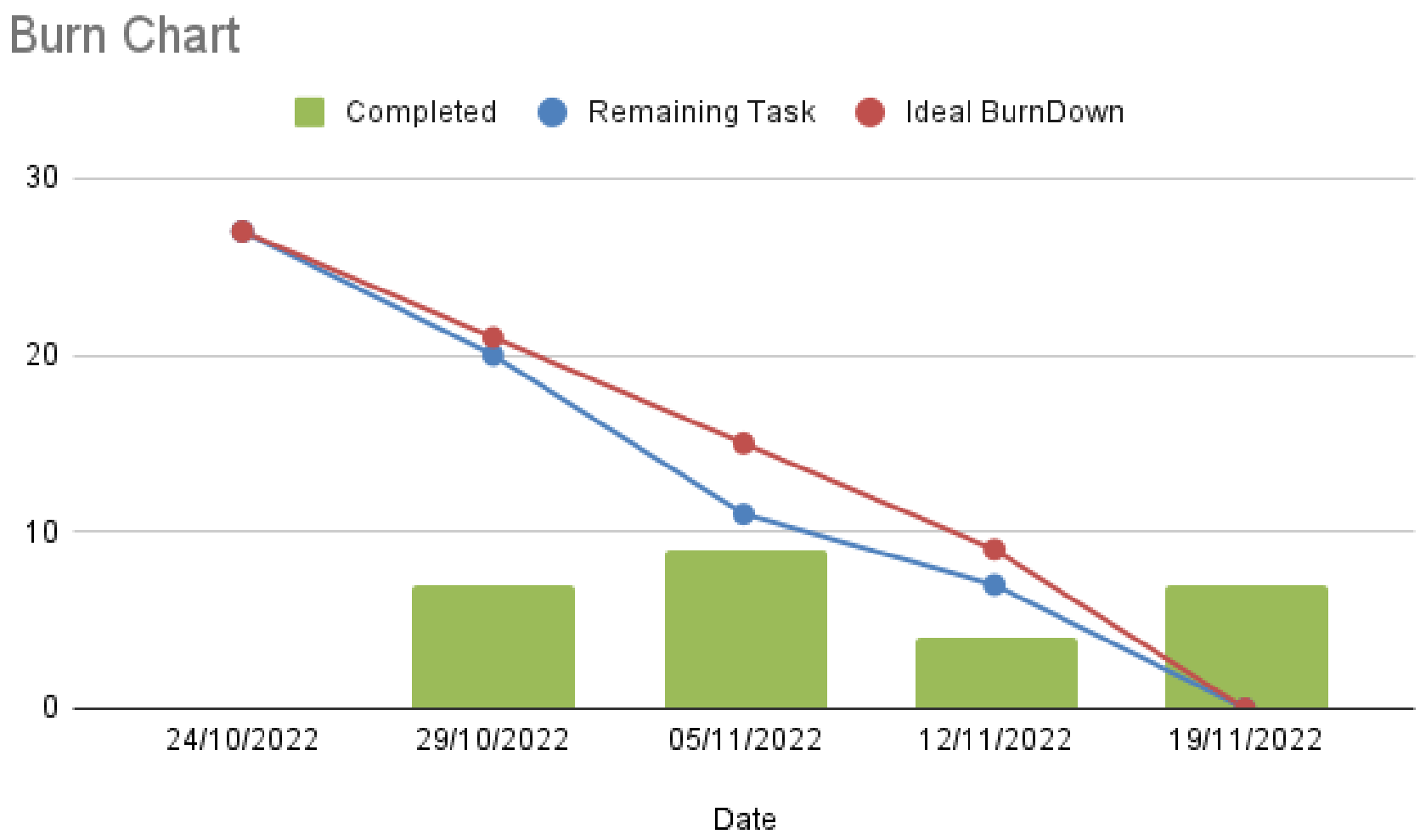
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	23	6 Days	24 Oct 2022	29 Oct 2022	7	30 Oct 2022
Sprint-2	23	6 Days	31 Oct 2022	05 Nov 2022	16	10 Nov 2022
Sprint-3	23	6 Days	07 Nov 2022	12 Nov 2022	20	15 Nov 2022
Sprint-4	23	6 Days	14 Nov 2022	19 Nov 2022	27	20 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

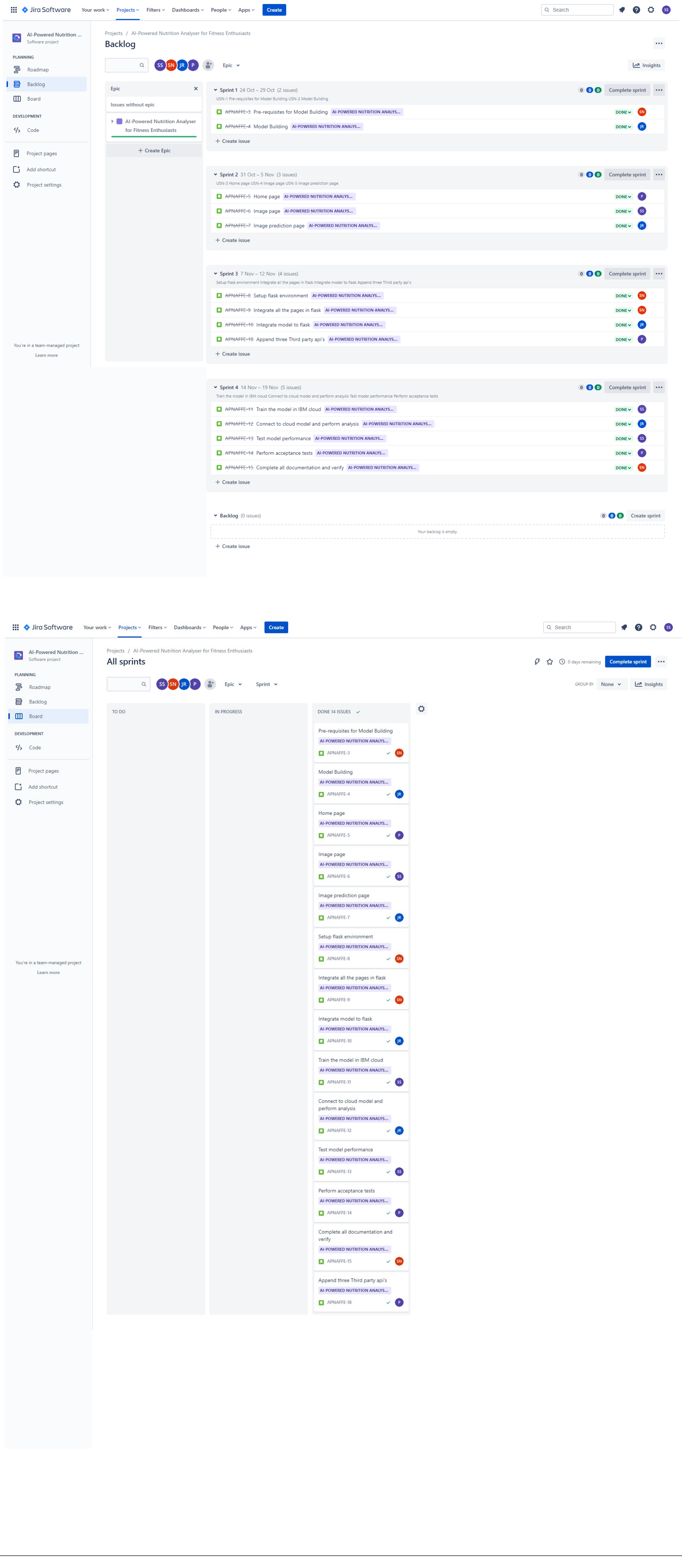
Burndown Chart:



6.2 Sprint Delivery Schedule

S.NO	MILESTONES	ACTIVITIES	DATE
1.	**Preparation Phase**	Pre-requisites	24 Aug 2022
		Prior Knowledge	25 Aug 2022
		Project Structure	23 Aug 2022
		Project Flow	23 Aug 2022
		Project Objectives	22 Aug 2022
		Registrations	26 Aug 2022
		Environment Set-up	27 Aug 2022
2.	**Ideation Phase**	Literature Survey	29 Aug 2022 – 03 Sept 2022
		Empathy Map	5 Sept 2022 - 7 Sept 2022
		Problem Statement	8 Sept 2022 - 10 Sept 2022
		Ideation	12 Sept 2022 – 16 Sept 2022
3.	**Project Design Phase - 1**	Proposed Solution	19 Sept 2022 – 23 Sept 2022
		Problem Solution Fit	24 Sept 2022 – 26 Sept 2022
		Solution Architecture	27 Sept 2022 – 30 Sept 2022
4.	**Project Design Phase - 2**	Customer Journey Map	03 Oct 2022 – 08 Oct 2022
		Requirement Analysis	09 Oct 2022 – 11 Oct 2022
		Data Flow Diagrams	11 Oct 2022 – 14 Oct 2022
		Technology Architecture	15 Oct 2022 - 16 Oct 2022
5.	**Project Planning Phase**	Milestones & Tasks	17 Oct 2022 – 18 Oct 2022
		Sprint Schedules	19 Oct 2022 – 22 Oct 2022
6.	**Project Development Phase**	Sprint - 1	26 Oct 2022 – 31 Oct 2022
		Sprint – 2	01 Nov 2022 – 07 Nov 2022
		Sprint – 3	08 Nov 2022 – 13 Nov 2022
		Sprint – 4	15 Nov 2022 – 20 Nov 2022

6.3 Reports from Jira



7. Coding & Solutioning

7.1 Models and Training:

The Configuration Class - This class contains a list of all the parameters and hyperparameters of the data and the model

Config

```
# ====================================================
# CFG
# ====================================================
class CFG: #the configuration where all parameters/hyperparameters are stored
    debug=False
    apex=False
    print_freq=100
    num_workers=4
    model_name='tf_efficientnet_b0_ns'
    size=256
    scheduler='CosineAnnealingWarmRestarts'
    criterion='CrossEntropyLoss'
    epochs=2
    T_0=10
    lr=1e-4
    min_lr=1e-6
    batch_size=32
    weight_dec=1e-6
    gradient_accumulation_steps=1
    max_grad_norm=1000
    seed=42
    target_size = 5
    target_col = 'label'
    n_fold=5
    trn_fold = [0, 1, 2, 3, 4]
    train=True
    smoothing=0.05

if CFG.debug:
    CFG.epochs = 1
    train_data = train_data.sample(n=1000, random_state=CFG.seed).reset_index(drop=True)
```

Python

The Model Architecture Class – This is the class where the architecture of the model is defined. The pretrained weights are used from the timm library and a custom head is attached to the model to fit the number of classes.

```
MODEL
# ====================================================
class CustomEfficientNet(nn.Module):
    """ using the efficientnet arch and modifying the final classification head"""
    def __init__(self, model_name=CFG.model_name, pretrained=False):
        super().__init__()
        self.model = timm.create_model(CFG.model_name, pretrained=pretrained)
        n_features = self.model.classifier.in_features
        self.model.classifier = nn.Linear(n_features, CFG.target_size)

    def forward(self, x):
        x = self.model(x)
        return x
```

Python

```
## for debug only
model = CustomEfficientNet(model_name=CFG.model_name, pretrained=False)
train_dataset = TrainDataset(train_data, transform=get_transforms(data='train'))
train_loader = DataLoader(train_dataset, batch_size=4, shuffle=True,
                          num_workers=4, pin_memory=True, drop_last=True)

for image, label in train_loader:
    output = model(image)
    print(output)
    break
```

Python

```
tensor([[ 3.6885e-02,  8.5828e-02,  9.2741e-02,  2.1368e-01, -4.4491e-01],
        [-8.7685e-02, -1.0966e-01,  6.5145e-02,  9.0393e-02,  1.0179e-02],
        [-1.1302e-01, -1.6268e-02,  5.9678e-03,  2.7062e-02,  2.2634e-04],
        [-1.2348e-01, -1.0647e-01,  5.4219e-02, -2.9790e-02, -2.3000e-02]],
       grad_fn=<AddmmBackward0>)
```

Train script – This block of code contains how the data is passed to the model for training

```
def train_fn(train_loader, model, criterion, optimizer, epoch, scheduler, device):
    """the main train function is defined here, gradient clipping is done and final loss average is returned"""
    batch_time = AverageMeter()
    data_time = AverageMeter()
    losses = AverageMeter()
    scores = AverageMeter()
    # switch to train mode
    model.train()
    start = end = time.time()
    global_step = 0
    for step, (images, labels) in enumerate(train_loader):
        # measure data loading time
        data_time.update(time.time() - end)
        images = images.to(device)
        labels = labels.to(device)
        batch_size = labels.size(0)
        y_preds = model(images)
        loss = criterion(y_preds, labels)
        # record loss
        losses.update(loss.item(), batch_size)
        if CFG.gradient_accumulation_steps > 1:
            loss = loss / CFG.gradient_accumulation_steps
        if CFG.apex:
            with amp.scale_loss(loss, optimizer) as scaled_loss:
                scaled_loss.backward()
        else:
            loss.backward()
        grad_norm = torch.nn.utils.clip_grad_norm_(model.parameters(), CFG.max_grad_norm)
        if (step + 1) % CFG.gradient_accumulation_steps == 0:
            optimizer.step()
            optimizer.zero_grad()
            global_step += 1
        # measure elapsed time
        batch_time.update(time.time() - end)
        end = time.time()
```

Valid script – This block of code freezes the model weights and tests the performance of the model on the current validation set

```
def valid_fn(valid_loader, model, criterion, device):
    """ the validation function that returns the average validation loss and the prediction probability for each class(through softmax)"""
    batch_time = AverageMeter()
    data_time = AverageMeter()
    losses = AverageMeter()
    scores = AverageMeter()
    # switch to evaluation mode
    model.eval()
    preds = []
    start = end = time.time()
    for step, (images, labels) in enumerate(valid_loader):
        # measure data loading time
        data_time.update(time.time() - end)
        images = images.to(device)
        labels = labels.to(device)
        batch_size = labels.size(0)
        # compute loss
        with torch.no_grad():
            y_preds = model(images)
        loss = criterion(y_preds, labels)
        losses.update(loss.item(), batch_size)
        # record accuracy
        preds.append(y_preds.softmax(1).to('cpu').numpy())
        if CFG.gradient_accumulation_steps > 1:
            loss = loss / CFG.gradient_accumulation_steps
        # measure elapsed time
        batch_time.update(time.time() - end)
        end = time.time()
        if step % CFG.print_freq == 0 or step == (len(valid_loader)-1):
            print('EVAL: [{0}/{1}] '
                  'Data {data_time.val:.3f} ({data_time.avg:.3f}) '
                  'Elapsed {remain:s} '
                  'Loss: {loss.val:.4f}({loss.avg:.4f}) '
                  .format(
                   step, len(valid_loader), batch_time=batch_time,
```

Training log – Here is a log of the model training

```
if __name__ == '__main__': #the entry point for the program to run
    main()
```

Python

```
========== fold: 0 training ==========
INFO:__main__:========== fold: 0 training ==========
Downloading: "https://github.com/rwightman/pytorch-image-models/releases/download/v0.1-weights/tf_efficientnet_b0_ns-c0e6a31c.pth" to
/root/.cache/torch/hub/checkpoints/tf_efficientnet_b0_ns-c0e6a31c.pth
Criterion: CrossEntropyLoss()
INFO:__main__:Criterion: CrossEntropyLoss()

Epoch: [1][0/65] Data 14.182 (14.182) Elapsed 0m 21s (remain 23m 20s) Loss: 1.5587(1.5587) Grad: 2.1206
Epoch: [1][64/65] Data 13.430 (3.503) Elapsed 4m 10s (remain 0m 0s) Loss: 0.1789(0.7073) Grad: 1.0743
EVAL: [0/17] Data 13.731 (13.731) Elapsed 0m 13s (remain 3m 40s) Loss: 0.0570(0.0570)

Epoch 1 - avg_train_loss: 0.7073  avg_val_loss: 0.0500  time: 315s
INFO:__main__:Epoch 1 - avg_train_loss: 0.7073  avg_val_loss: 0.0500  time: 315s
Epoch 1 - Accuracy: 1.0
INFO:__main__:Epoch 1 - Accuracy: 1.0
Epoch 1 - Save Best Score: 1.0000 Model
INFO:__main__:Epoch 1 - Save Best Score: 1.0000 Model

EVAL: [16/17] Data 4.179 (5.711) Elapsed 1m 4s (remain 0m 0s) Loss: 0.0311(0.0500)
Epoch: [2][0/65] Data 0.647 (0.647) Elapsed 0m 0s (remain 1m 1s) Loss: 0.0023(0.0023) Grad: 0.5457
Epoch: [2][64/65] Data 0.000 (0.011) Elapsed 0m 15s (remain 0m 0s) Loss: 0.0200(0.0048) Grad: 0.2402
EVAL: [0/17] Data 0.611 (0.611) Elapsed 0m 0s (remain 0m 12s) Loss: 0.0118(0.0118)

Epoch 2 - avg_train_loss: 0.0048  avg_val_loss: 0.0109  time: 18s
INFO:__main__:Epoch 2 - avg_train_loss: 0.0048  avg_val_loss: 0.0109  time: 18s
Epoch 2 - Accuracy: 1.0
INFO:__main__:Epoch 2 - Accuracy: 1.0
```

Infer Scripts – This block of code contains loading the trained weights to run inference on the test set

```
# ====================================================
# Helper functions
# ====================================================
def load_state(model_path):
    """loading the state of model dictionary"""
    model = CustomEfficientNet(CFG.model_name, pretrained=False)
    model.load_state_dict(torch.load(model_path)['model'], strict=True)
    state_dict = torch.load(model_path)['model']
    return state_dict

def inference(model, states, test_loader, device):
    """making predictions on the test set"""
    model.to(device)
    tk0 = tqdm(enumerate(test_loader), total=len(test_loader))
    probs = []
    for i, (images) in tk0:
        images = images.to(device)
        avg_preds = []
        for state in states:
            model.load_state_dict(state)
            model.eval()
            with torch.no_grad():
                y_preds = model(images)
                #print(f'y_preds: {y_preds.shape}')
            avg_preds.append(y_preds.softmax(1).to('cpu').numpy())
            #print(f'avg_preds: {avg_preds}')
        avg_preds = np.mean(avg_preds, axis=0)
        probs.append(avg_preds)
    probs = np.concatenate(probs)
    return probs
```

Python

```
# ====================================================
# inference
# ====================================================
model = CustomEfficientNet(CFG.model_name, pretrained=False)
MODEL_DIR = '/content/'
states = [load_state(MODEL_DIR+f'{CFG.model_name}_fold{fold}_best.pth') for fold in CFG.trn_fold]
test_dataset = TestDataset(test_data, transform=get_transforms(data='valid'))
test_loader = DataLoader(test_dataset, batch_size=CFG.batch_size, shuffle=False,
                         num_workers=CFG.num_workers, pin_memory=True)
predictions = inference(model, states, test_loader, device)

test_data['model_preds'] = predictions.argmax(1) #argmax returns the highest index(class label)
test_data.head()
```

Python

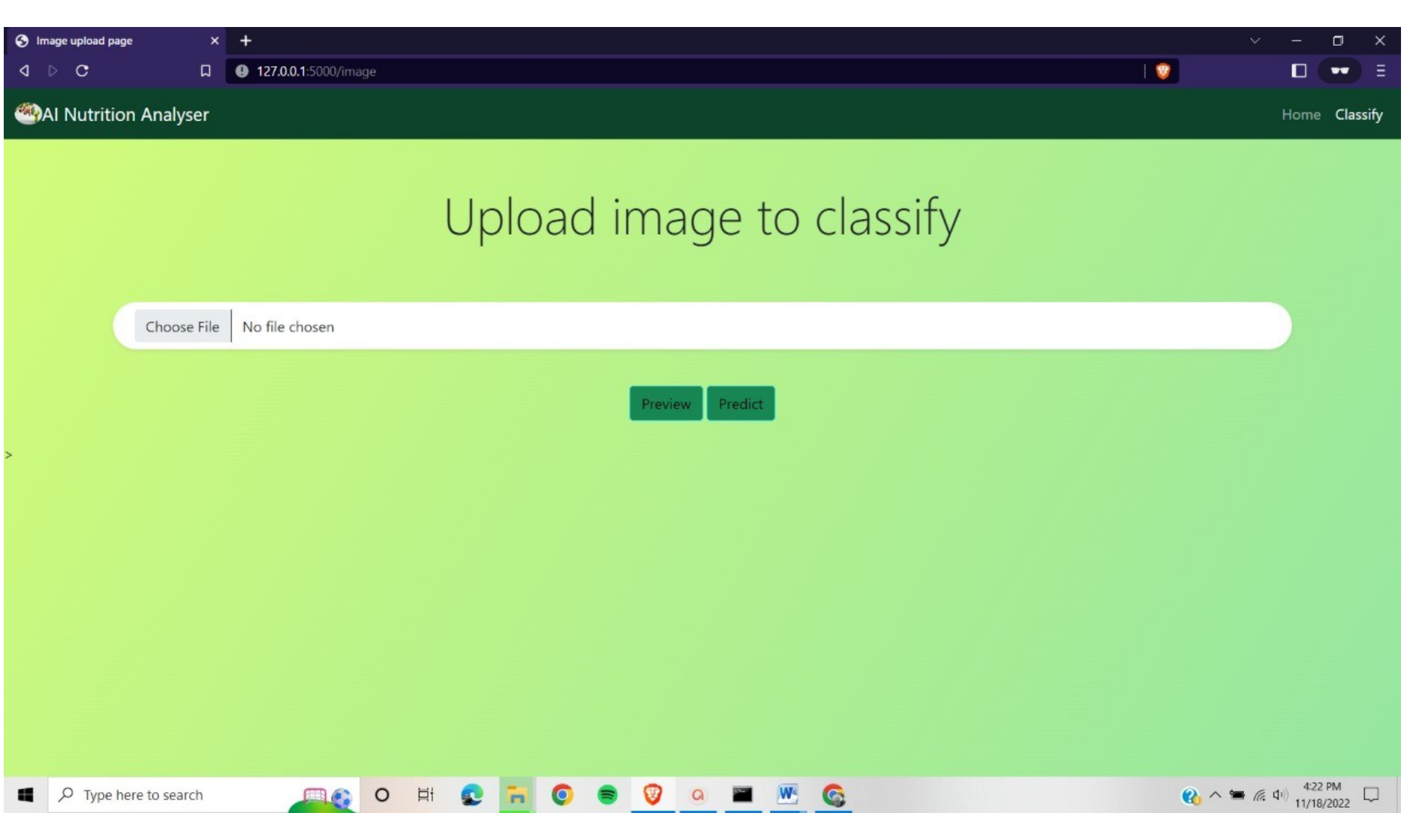
	fruit	path	model_preds
0	APPLES	/content/drive/MyDrive/ibm_data/TEST_SET/APPLE...	0
1	APPLES	/content/drive/MyDrive/ibm_data/TEST_SET/APPLE...	0
2	APPLES	/content/drive/MyDrive/ibm_data/TEST_SET/APPLE...	0
3	APPLES	/content/drive/MyDrive/ibm_data/TEST_SET/APPLE...	0
4	APPLES	/content/drive/MyDrive/ibm_data/TEST_SET/APPLE...	0

7.2 User Interface:

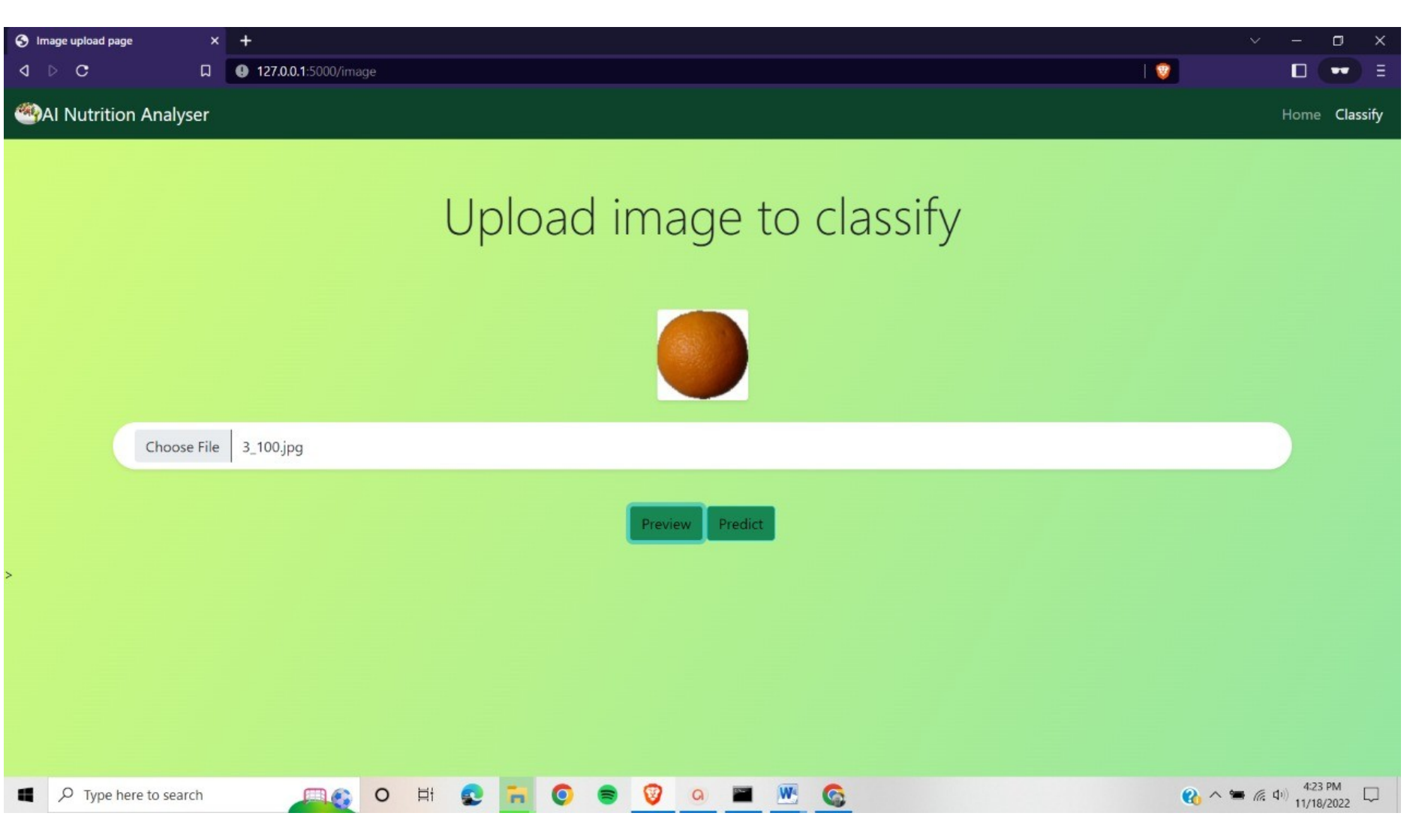
Image Upload section – Here the user is given two options

1. You can upload the image and then preview it
2. You can upload the image and then predict for the fruit and nutritional content

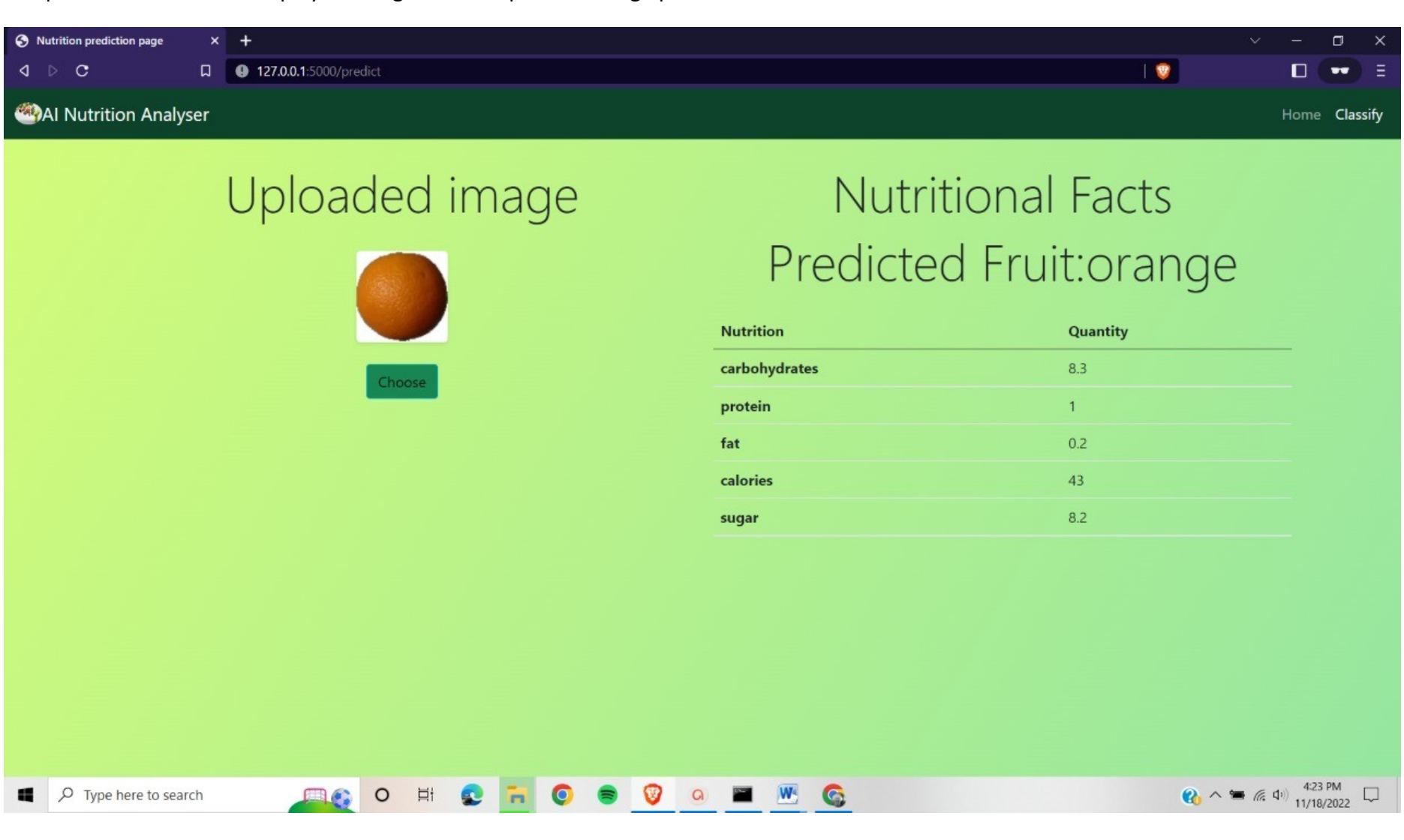
```html
<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

  <!-- CSS Link -->
  <link rel="stylesheet" href="/static/style.css">
  <style>
    html,body{
      background-image: linear-gradient(120deg, #d4fc79 0%, #96e6a1 100%);
    }
  </style>
  <title>Nutrition prediction page</title>
</head>

<body>

  <!-- navbar -->
  <nav class="navbg navbar navbar-expand-lg navbar-dark bg-success">
    <div class="container-fluid">

      <!-- Brand -->
      <img src='/static/images/logo.webp' alt="logo" width="30" height="24"
        class="d-inline-block align-text-top brandimg">
      <a class="navbar-brand" href="#">AI Nutrition Analyser</a>

      <!-- Toggle button -->
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup"
        aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <!-- Navbar options -->
      <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
        <div class="navbar-nav ms-auto">
          <a class="nav-link" href="/home">Home</a>
          <a class="nav-link active" aria-current="page" href="/image">Classify</a>
        </div>
      </div>
    </div>
  </nav>

  <!-- upload section -->
  <div class="container">
    <div class="row my-4">
      <div class="col">
        <header class="text-center">
          <h1 class="display-4">Uploaded image</h1>
        </header>
        <div class="container">
          <div class="row">
            <div class="col-lg-6 mx-auto">

              <!-- image preview area -->
              <div class="image-area mt-4">
                <img src="/preview/{{filename}}" alt="image preview area"
                  class="img-fluid rounded shadow-sm mx-auto d-block">
              </div>
            </div>

            <div class="text-center my-4">
              <button type="button" class="btn btn-info bg-success"><a href="/image"
                  style="text-decoration:none; color:black">Choose</a></button>
            </div>
          </div>
        </div>
      </div>
      <!-- Nutritonal facts display -->
      <div class="col">
        <div>
          <header class="text-center">
            <h1 class="display-4">Nutritional Facts</h1>
            <h1 class="display-4">Predicted Fruit:{{fruit}}</h1>
          </header>
          <table class="table table-responsive table-hover my-4">
            <thead class="thead-dark">
              <tr>
                <th scope="col">Nutrition</th>
                <th scope="col">Quantity</th>
              </tr>
            </thead>
            <tbody>
              {%for key in nutrition%}
              <tr>
                <th scope="row">{{key}}</th>
                <td>{{nutrition[key]}}</td>
              </tr>
              {%endfor%}
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
    integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF"
    crossorigin="anonymous"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>

</body>
</html>
```



Here's an example of previewed image



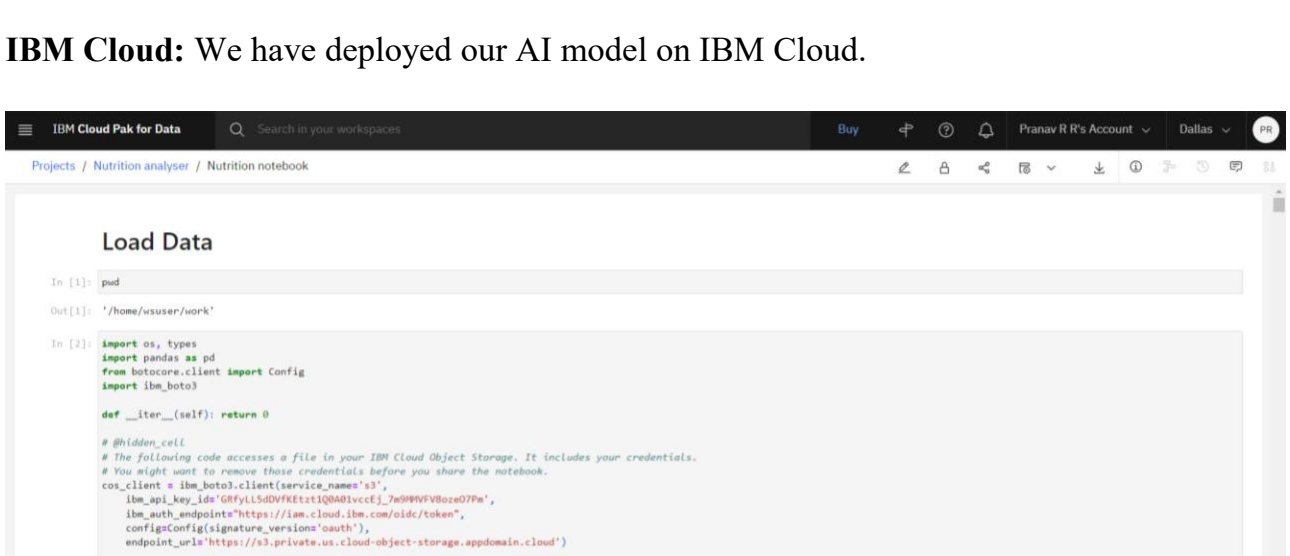
The predicted results are displayed along with the uploaded image preview




```html
<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

  <!-- CSS Link -->
  <link rel="stylesheet" href="/static/style.css">
  <style>
    html,body{
      background-image: linear-gradient(120deg, #d4fc79 0%, #96e6a1 100%);
    }
  </style>
  <title>Nutrition prediction page</title>
</head>

<body>

  <!-- navbar -->
  <nav class="navbg navbar navbar-expand-lg navbar-dark bg-success">
    <div class="container-fluid">

      <!-- Brand -->
      <img src='/static/images/logo.webp' alt="logo" width="30" height="24"
        class="d-inline-block align-text-top brandimg">
      <a class="navbar-brand" href="#">AI Nutrition Analyser</a>

      <!-- Toggle button -->
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup"
        aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <!-- Navbar options -->
      <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
        <div class="navbar-nav ms-auto">
          <a class="nav-link" href="/home">Home</a>
          <a class="nav-link active" aria-current="page" href="/image">Classify</a>
        </div>
      </div>
    </div>
  </nav>

  <!-- upload section -->
  <div class="container">
    <div class="row my-4">
      <div class="col">
        <header class="text-center">
          <h1 class="display-4">Uploaded image</h1>
        </header>
        <div class="container">
          <div class="row">
            <div class="col-lg-6 mx-auto">

              <!-- image preview area -->
              <div class="image-area mt-4">
                <img src="/preview/{{filename}}" alt="image preview area"
                  class="img-fluid rounded shadow-sm mx-auto d-block">
              </div>
            </div>

            <div class="text-center my-4">
              <button type="button" class="btn btn-info bg-success"><a href="/image"
                  style="text-decoration:none; color:black">Choose</a></button>
            </div>
          </div>
        </div>
      </div>
      <!-- Nutritonal facts display -->
      <div class="col">
        <div>
          <header class="text-center">
            <h1 class="display-4">Nutritional Facts</h1>
            <h1 class="display-4">Predicted Fruit:{{fruit}}</h1>
          </header>
          <table class="table table-responsive table-hover my-4">
            <thead class="thead-dark">
              <tr>
                <th scope="col">Nutrition</th>
                <th scope="col">Quantity</th>
              </tr>
            </thead>
            <tbody>
              {%for key in nutrition%}
              <tr>
                <th scope="row">{{key}}</th>
                <td>{{nutrition[key]}}</td>
              </tr>
              {%endfor%}
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
    integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF"
    crossorigin="anonymous"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>

</body>

</html>
```

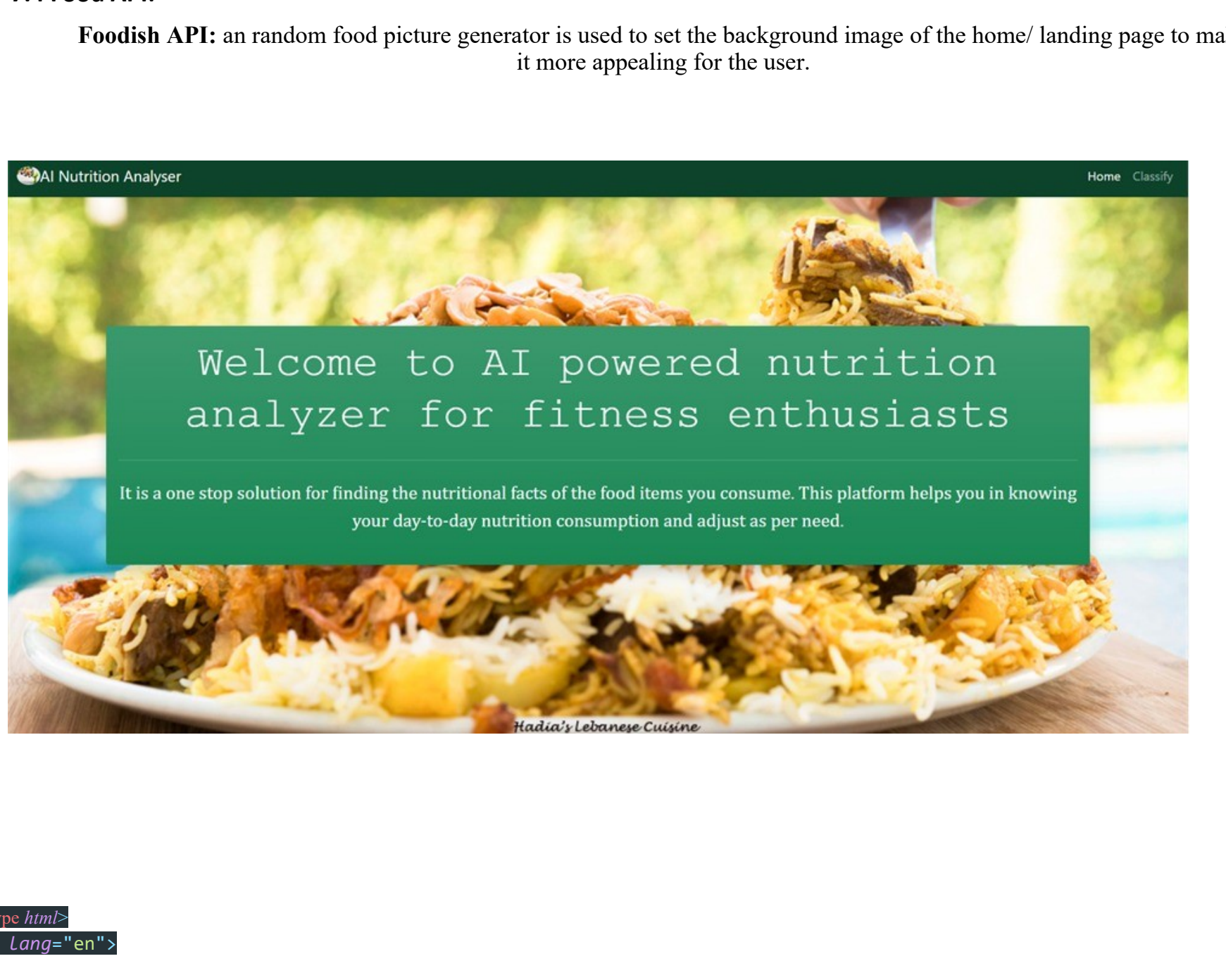

7.3 deployment:

IBM Cloud: We have deployed our AI model on IBM Cloud.



7.4 Food API:

Foodish API: an random food picture generator is used to set the background image of the home/ landing page to make it more appealing for the user.




```html
<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
  <!-- CSS Link -->
  <link rel="stylesheet" href="/static/style.css">
  <script src="/static/js/background-image.js"></script>
  <title>Home page</title>
</head>

<body>

  <!-- navbar -->
  <nav class="navbg navbar navbar-expand-lg navbar-dark bg-success">
    <div class="container-fluid">

      <!-- Brand -->
      <img src='/static/images/logo.webp' alt="logo" width="30" height="24"
        class="d-inline-block align-text-top brandimg">
      <a class="navbar-brand" href="#">AI Nutrition Analyser</a>

      <!-- Toggle button -->
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup"
        aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <!-- Navbar options -->
      <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
        <div class="navbar-nav ms-auto">
          <a class="nav-link active" aria-current="page" href="/home">Home</a>
          <a class="nav-link" href="/image">Classify</a>
        </div>
      </div>
    </div>
  </nav>

  <!-- main banner -->
  <section id="home">
    <div class="container-fluid">
      <div class="row">
        <div class="col-lg-10 col-md-10 col-sm-10 col-xs-10 hero-text shadow p-3 mb-5 rounded bg-success bg-gradient p-2
text-white bg-opacity-65 ">
          <h2 class="hero_title">Welcome to AI powered nutrition analyzer for fitness enthusiasts</h2>
          <hr class="my-4" />
          <p class="hero_desc my-4">
            It is a one stop solution for finding the nutritional facts of the food items you consume.
            This platform helps you in knowing your day-to-day nutrition consumption and adjust as per need.
          </p>
        </div>
      </div>
    </div>
  </section>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
  integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF"
  crossorigin="anonymous"></script>
</body>

</html>
```

8. TESTING

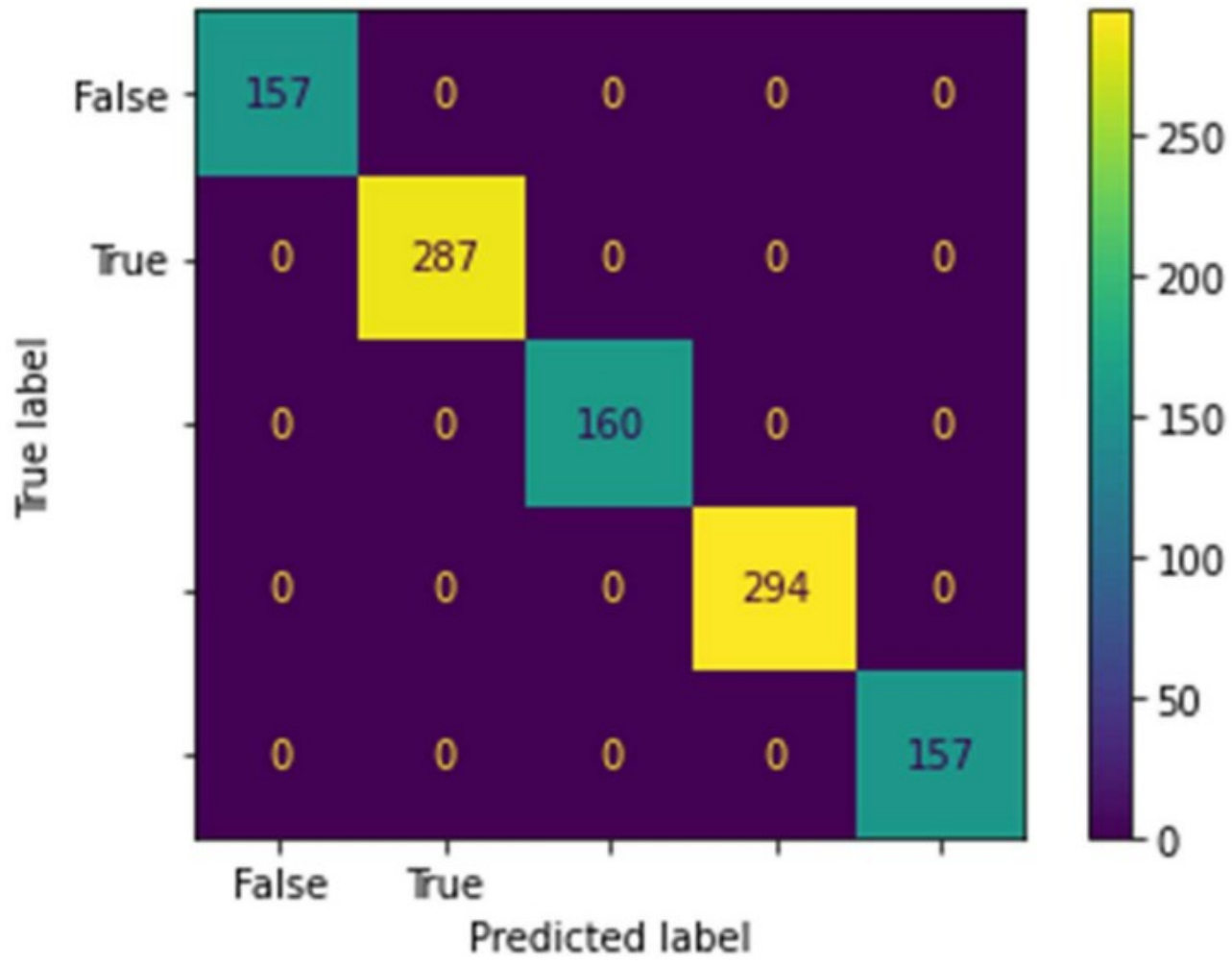
8.1 Test Cases & 8.2 User Acceptance Testing

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
LandingPage_TC_OO1	UI	Home Page	Verify the Random food APIworks properly		1.Enter URL and click go	http://127.0.0.1:5000	Any random food image should appear as background	Working as expected	Pass
LandingPage_TC_OO2	UI	Home Page	Verify the UI renders the "Predict" button in the navbar		1.Enter URL and click go	http://127.0.0.1:5000/	The "Predict" button that takes users to the prediction page renders properly.	Working as expected	Pass
ImagePage_TC_OO1	Functional	Image page	Verify user is able to upload images		1.Enter URL and click go 2.Click on Predict button in the navbar. 3. Click "Choose"	http://127.0.0.1:5000/	User should be able to choose the image from his machine through File Explorer.	Working as expected	Pass
ImagePage_TC_OO2	Functional	Image page	Verify user has an option only to select images from his device		1. Enter URL and click go 2.Click on Predict button in the navbar. 3. Click "Choose"	http://127.0.0.1:5000/	The File Manager should display only images for the user to select from.	Working as expected	Pass
ImagePage_TC_OO3	Functional	Image page	Verify user is able to preview the uploaded image		1.Enter URL(http://127.0.0.1:5000/) and click go Click on Predict button in the navbar. 3. Click "Choose" 4. Upload any image of a fruit. 5. Click "Preview".	Any image of a fruit	The page should display the image of the fruit the user uploaded for prediction.	Working as expected	Pass
ImagePredictionPage_TC_OO1	Functional	Image Prediction page	Verify the model is able to make predictions with the image uploaded by the user	Verify the libraries for prediction are available in the test environment.	1.Enter URL(http://127.0.0.1:5000/) and click go Click on Predict button in the navbar. 3. Click "Choose" 4. Upload any image of a fruit. 5. Click "Predict".	Any image of a fruit	The application takes the user to the Image Prediction page and displays the name of the predicted page	Working as expected	Pass
ImagePredictionPage_TC_OO2	Functional	Image Prediction Page	Verify the Nutrition API is able to fetch the nutritional content for the predicted fruit	Verify the libraries for prediction are available in the test environment.	1.Enter URL(http://127.0.0.1:5000/) and click go Click on Predict button in the navbar. 3. Click "Choose" 4. Upload any image of a fruit. 5. Click "Predict".	Any image of a fruit	The page should display the nutritional content of the predicted fruit in a nicely formatted tabular manner	Working as expected	Pass

9. Project Planning and Scheduling

9.1 Performance Metrics

1. Confusion Matrix




```
print(metrics.classification_report(test_data['label'].values, test_data['model_preds'].values))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	157
1	1.00	1.00	1.00	287
2	1.00	1.00	1.00	160
3	1.00	1.00	1.00	294
4	1.00	1.00	1.00	157
accuracy			1.00	1055
macro avg	1.00	1.00	1.00	1055
weighted avg	1.00	1.00	1.00	1055

2. Accuracy – 100 %

```
[8]  print(f"the accuracy is {metrics.accuracy_score(test_data['label'].values, test_data['model_preds'].values)}")
```

```
the accuracy is 1.0
```

3. Precision – 100 %

```
[11] print(f"the precision is {metrics.precision_score(test_data['label'].values, test_data['model_preds'].values, average = 'weighted')}")
```

```
the precision is 1.0
```

4. Recall – 100 %

```
[12] print(f"the recall is {metrics.recall_score(test_data['label'].values, test_data['model_preds'].values, average = 'weighted')}")
```

```
the recall is 1.0
```

5. Specificity – 100 %

```
print(f"the specificity is {metrics.recall_score(test_data['label'].values, test_data['model_preds'].values, pos_label=0,average = 'weighted')}")
```

```
the specificity is 1.0
```

6. F1-Score – 100 %

```
[13] print(f"the f1 score is {metrics.f1_score(test_data['label'].values, test_data['model_preds'].values,average = 'weighted')}")
```

```
the f1 score is 1.0
```

10. Advantages and Disadvantages:

10.1 Advantages

- Web based easy to use interface.
- Provides an image based facts retrieval of food.
- Suggestion based on the facts.
- Basic details and guidance based on the user profile.
- Making a user-friendly interface and making it all platform compatible.

10.2 Challenges

- It is web based, using it offline is tricky.
- Have to train more models to make it future ready.

11. Conclusion:

To summarise, we have developed an online solution that predicts the food item and returns the nutritional facts of the predicted food. Currently, only five fruits: orange, pineapple, apple, banana, and watermelon are predicted, but we can train the model to predict any food item from natural to processed by training it with the relevant dataset and making it future-ready. We can add primarily many modules for user-based login, which can unlock ways to store user profiles, consumption data, and user-based suggestions.

The primitive aim of the project is to assist health-conscious people to keep track of their nutritional intake and help them boost their health.

12. Future Scope:

- Making a user-friendly interface and making it all platform compatible.
- Providing rewards for usage and sharing of application.
- Using client data and feedback to improvise.
- Collaborating with other parties for larger scale usage
- Subscription based plan for users to unlock all features.
- Providing online resources for customers.

13. Appendix:

13.1 Source Code - https://github.com/IBM-EPBL/IBM-Project-10868-1659240779/tree/main/Final%20Deliverables/Final%20code

13.2 Github - https://github.com/IBM-EPBL/IBM-Project-10868-1659240779

13.3 Demo - https://drive.google.com/file/d/1RxRrEhPs89SjtE3qGBqIejYH7XFX523U/view