AI-powered Nutrition Analyzer forFitness Enthusiasts

TEAM ID : PNT2022TMID27062

NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READINESS FOR INNOVATION, EMPLOYMENT AND ENTREPRENEURSHIP

A PROJECT REPORT
BY

INFANT RICARDO A(310819106035)
IMMANUAL J (310819106033)
KAMESH P (310819106037)
KISHORE S (310819106041)

TEAM ID:

INDUSTRY MENTOR: Sri Tulasi
FACULTY MENTOR: Balachandran G

DEPARTMENT OF ECE

JEPPIAAR ENGINEERING COLLEGE, CHENNAI-119

INDEX

1. INTRODUCTION

Project Overview

Food is essential for human life and has been the concern of many healthcare conventions. Nowadays new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintaining a healthy diet. Nutritional analysis is the process of determining the nutritional content of food. It is a vital part of analytical chemistry that provides information about the chemical composition, processing, quality control and contamination of food.

The main aim of the project is to build a model which is used for classifying the fruit depending on the different characteristics like color, shape, texture etc. Here the user can capture the images of different fruits and then the image will be sent to the trained model. The model analyzes the image and detects the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).

Purpose

As the world is growing more fitness-conscious with time, there is an increasing demand for advanced technological solutions to cater to it. Lately, many applications worldwide are using predictive analytics artificial intelligence as well as natural language processing to help scores of fitness enthusiasts to monitor their nutrition and calorie intake. Artificial Intelligence and its subsets have been leveraged by these platforms to identify the calorie intake and then make food recommendations for a healthy diet.

2. LITERATURE SURVEY

[1] Deep Food : Food Image Analysis and Dietary Assessment via Deep Model. This system will analyze the nutritional ingredients based on the recognition results and generate a dietary assessment report by calculating the amount of calories, fat, carbohydrate and protein.

ALGORITHMS USED:

- Region-based Convolutional Neural Network
- Convolutional Neural Network
- Non-maximum suppression
- Bounding Box Regression
- Deep learning techniques

CHALLENGES:

Three main challenges in real food image recognition and analysis are addressed as follows:
1. Region of Interest
2. The Delay of Food Recognition
3. Insufficient Information of Nutrition Content for dietary assessment

[2]A New Deep Learning-based Food Recognition System for Dietary Assessment on An Edge Computing Service Infrastructure

A design of food recognition system employing edge computing-based service computing paradigm to overcome some inherent problems of traditional mobile cloud computing paradigm, such as unacceptable system latency and low battery life of mobile devices.

ALGORITHMS USED:
- K-means clustering algorithms
- Convolutional Neural Network
- Bounding Box Regression
- Deep learning

CHALLENGES:
Using this simple cropping-based approach will not work well if the food is scattered on different parts of the image

[3] Precision Nutrient Management Using Artificial Intelligence Based on Digital Data Collection Framework

Nutritional intake is fundamental to human growth and health, and the intake of different types of nutrients and micronutrients can affect health. The content of the diet affects the occurrence of disease, with the incidence of many diseases increasing each year while the age group at which they occur is gradually decreasing.

ALGORITHM USED:
- Okapi BM25
- TF-IDF
- Levenshtein
- Jaccard
- Synonyms

CHALLENGES:
This model has very little error and can significantly improve the efficiency of the analysis.

[4]Calculating Nutrition Facts with Computer Vision

People are becoming more health-conscious than before. However, there is a lack of knowledge about different fitness and wellness aspects of food. Thus, I come up with Foodify.ai—a deep

learning-based application that detects food from the image and provides information of food such as protein, vitamins, calories, minerals, carbs, etc.

ALGORITHM USED:

- Deep learning
- Machine learning
- Image Processing

CHALLENGES:

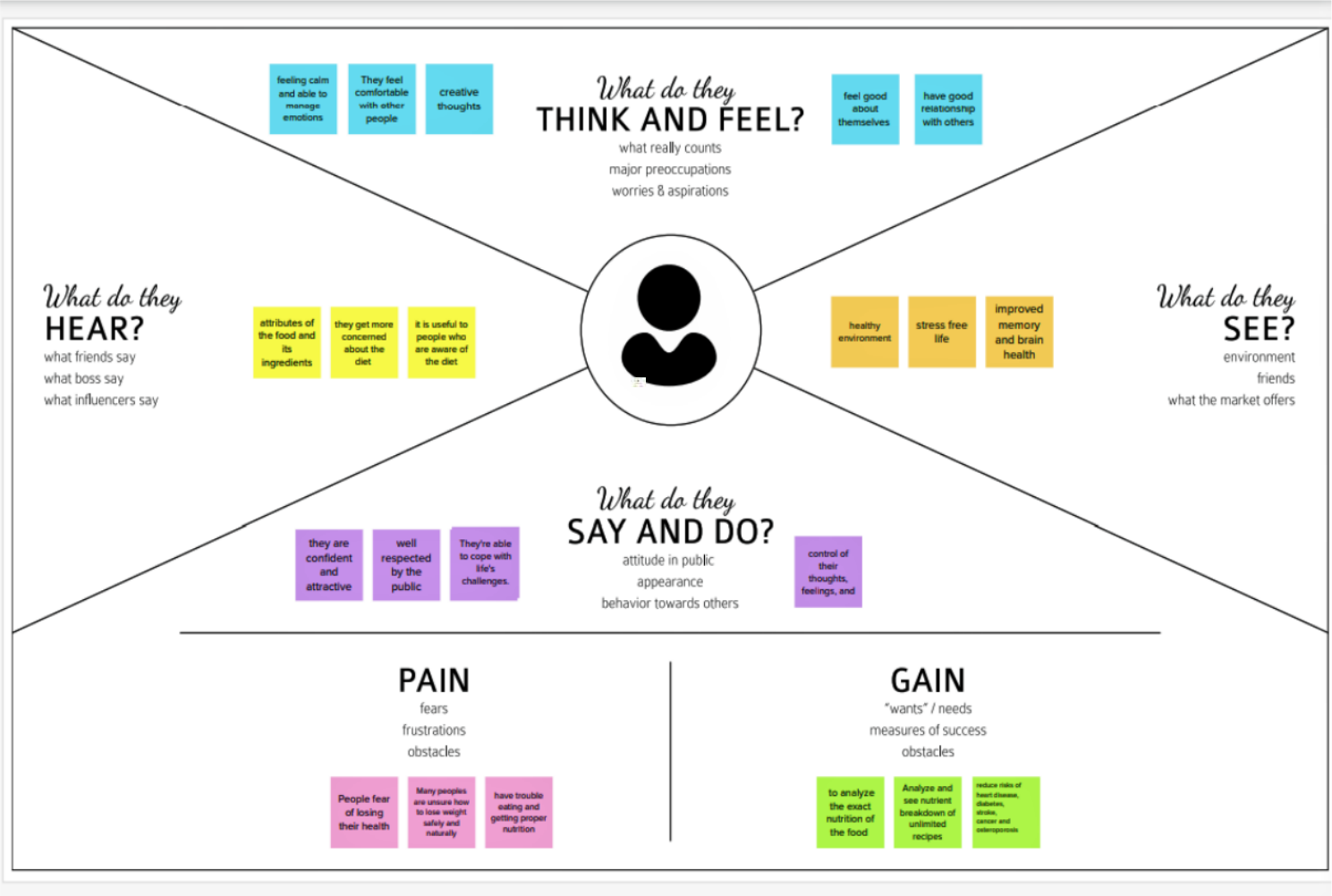
1. This is to collect images to create a huge dataset.
2. This is related to training the deep learning model. It is an extremely computationally expensive and time-consuming task to train the model again and again. This can be solved by using cloud-based service

3. IDEATION & PROPOSED SOLUTION

Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes.

It is a useful tool to help teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



Ideation & Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement

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

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

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

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

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

Open article →

TEAM LEADER : AVINASH C - 310619205016

TEAM MEMBER: LAKSMAN S - 3106920505 2
HARISH D - 310619205034
KAILASH V - 310619205043

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

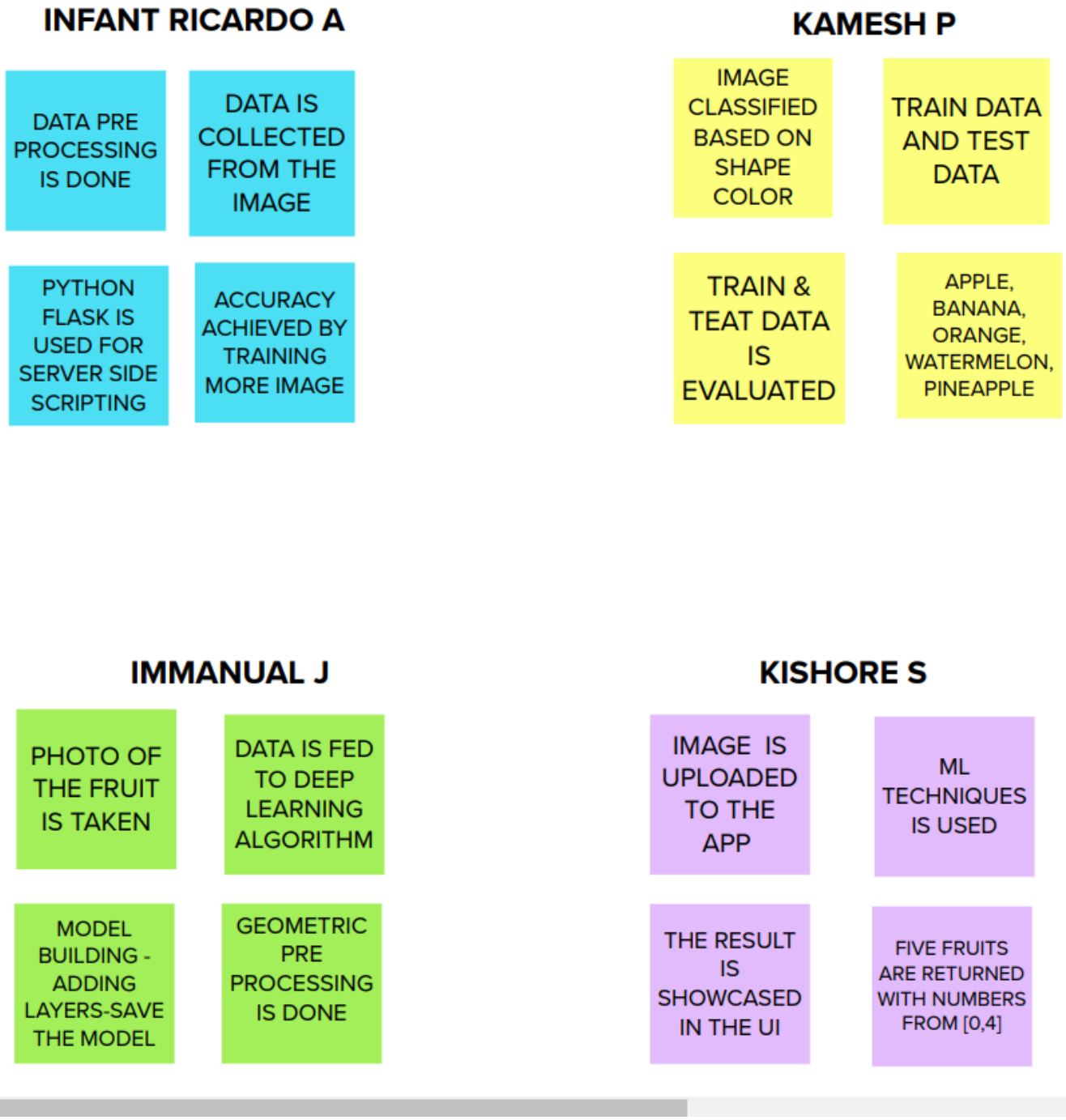
PROBLEM
How might we [your problem statement]?

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.

Step-2 : Brainstorm,Idea Listing and Grouping



Step-3: Idea Prioritization

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.

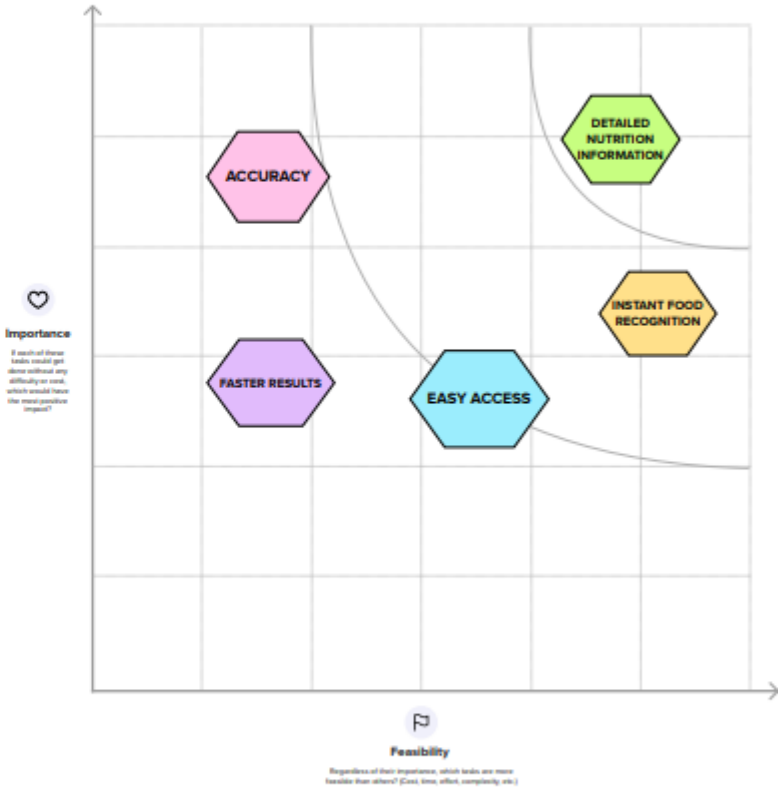
20 minutes

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.

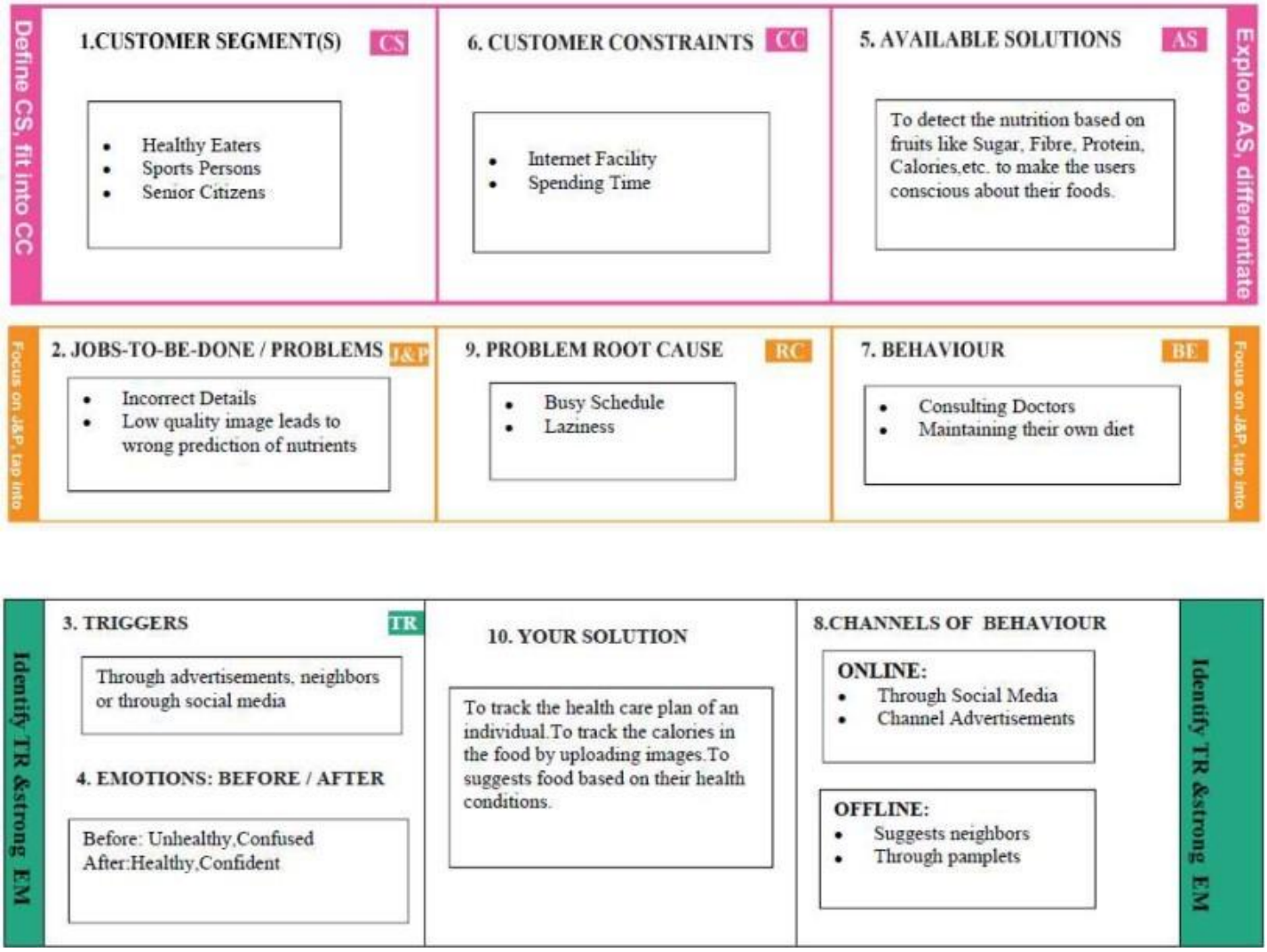
20 minutes



Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Food is essential for human life and has been the concern of many healthcare conventions. Nowadays new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintaining a healthy diet. Nutritional analysis is the process of determining the nutritional content of food. It is a vital part of analytical chemistry that provides information about the chemical composition, processing, quality control and contamination of food.
2.	Idea / Solution description	The idea of the project is to build a model which is used for classifying the fruit depending on the different characteristics like color, shape, texture etc.
3.	Novelty / Uniqueness	Here the user can capture the images of different fruits and then the image will be sent to the trained model. The model analyzes the image and detects the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.)
4.	Social Impact / Customer Satisfaction	This project is very helpful to People. Everyone Maintaining their own diet, to manage the time.
5.	Business Model (Revenue Model)	By using this system, the users can predict and analyze the picture of the fruits and foods. In which it results in visualizing the description of the foods taken as input.
6.	Scalability of the Solution	By implementing this system, the people can efficiently and effectively gain knowledge about the fitness.They want and they wish to use at any time. This system can also be integrated with the future technologies

Problem Solution fit



4. REQUIREMENT ANALYSIS

Functional requirement

Functional requirement:

Following are the functional requirements of the proposed solution.

Fr.no	Functional requirement	Sub requirement (story/subtask)
Fr-1	User registration	Registration through form Registration through Gmail
Fr-2	User confirmation	Confirmation via OTP Confirmation via Email
Fr-3	Capturing image	Capture the image of the leaf And check the parameter of the captured image .
Fr-4	Image processing	Upload the image for the prediction of the disease in the leaf.
Fr-5	Leaf identification	Identify the leaf and predict the disease in leaf.
Fr-6	Image description	Suggesting the best fertilizer for the disease.

Non-Functional requirements

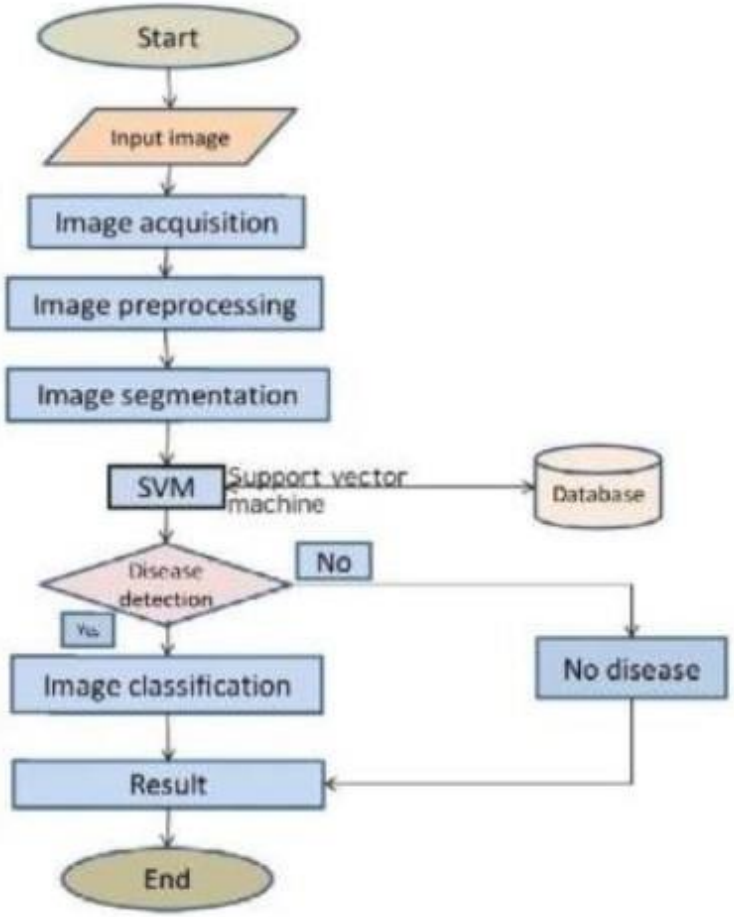
Non-functional requirement:

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

NFr.no	Non-functional requirement	Description
Nfr-1	Usability	Datasets of all the leaf is used to detecting the disease that present in the leaf.
Nfr-2	Security	The information belongs to the user and leaf are secured highly.
Nfr-3	Reliability	The leaf quality is important for the predicting the disease in leaf.
Nfr-4	Performance	The performance is based on the quality of the leaf used for disease prediction
Nfr-5	Availability	It is available for all user to predict the disease in the plant
Nfr-6	Scalability	Increasing the prediction of the disease in the leaf

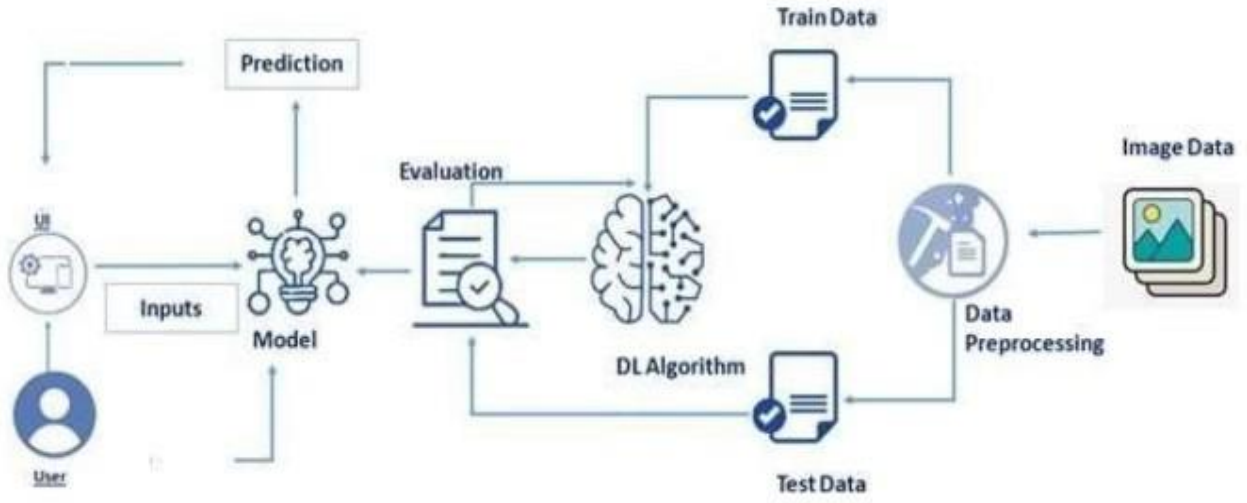
5. PROJECT DESIGN

Data Flow Diagrams

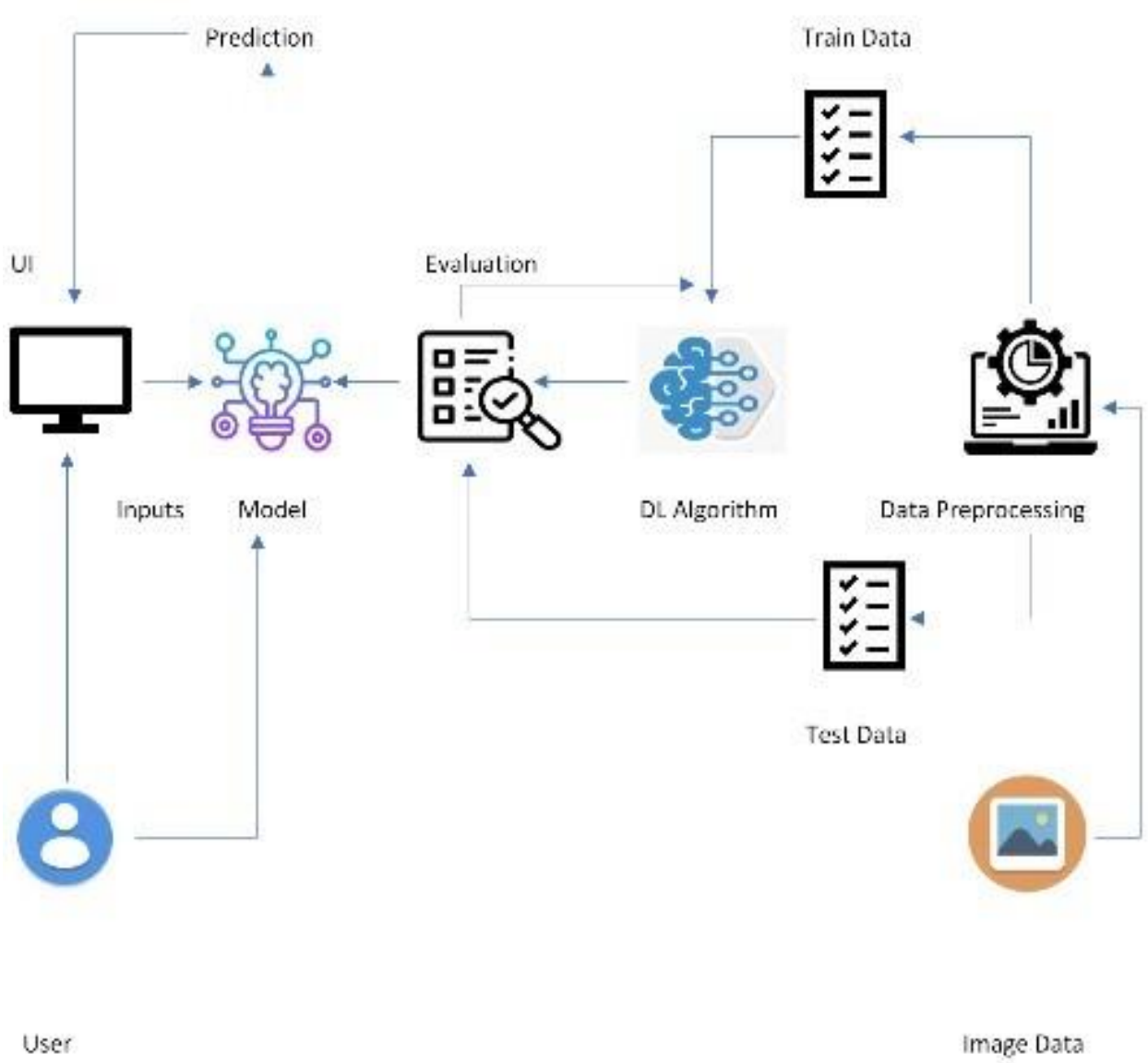


Solution & Technical Architecture

Technical Stack:



Solution Architecture:



6. PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, First I have to register for IBM cloud	2	High	Infant Ricardo ,kishore, kamesh
Sprint-1	Collecting the datasets	USN-2	As a user, I have to collect and download the datasets	2	High	Imman, kamesh
Sprint-1	Image Preprocessing	USN-3	After collecting the datasets,Image Preprocessing has to be done	2	Medium	Kishore, Imman, Ricardo
Sprint-1	Model building	USN-4	After image preprocessing, user has to build the model	2	High	Imman,kamesh,Infant Ricardo
Sprint-2	-	USN-5	As a user, I have to develop a code for this model building and I have to build a model	2	High	Imman,kishore
Sprint-2	Application building	USN-6	After model building,I have to create an application for the end users	2	High	Kishore, Kamesh, imman

Sprint-3	-	USN-7	As a user, I have to Create a folder which contains all the necessary html, css,js and python coding files	2	Medium	Kamesh, Ricardo,k ishore
Sprint-3	-	USN-8	I have to create a folder name flask,where I have to paste all the above mentioned coding files in that folder	1	High	Kamesh, Ricardo,I mman
Sprint-4	Outputs	USN-9	Link the flask file with html files and I have to share the screenshots of the output webpage	2	High	Ricardo, Kamesh, Kishore, imman
Sprint-4	-	USN-10	As a user, I have to deploy the model on IBM	2	High	Ricardo, Kamesh, Kishore, imman

Project Tracker, Velocity & Burndown Chart:

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Sprint Delivery Schedule:

The delivery plan of project deliverables is a strategic element for every Project Manager. The goal of every project is, in fact, to produce a result that serves a specific purpose. With the word "purpose ", we can mean the most disparate goals: a software program, a chair, a building, a translation, etc.

In Project Spirit Delivery Planning is one of the processes of completing the project and Showcasing the TimeLine of the Project Planning.

This Delivery plan helps to understand the process and WorkFlow of the Project working by the Team Mates.

Every Single Module are assigned to the teammates to showcase their work and contribution of developing the Project.

Reports from JIRA:

JIRA has categorized reports in four levels, which are –

1. Agile
2. Detection of Items
3. Details Generation
4. Others

VELOCITY: SPRINT - 1

Sprint duration = 5 days

Velocity of team = 20 points

Average Velocity (AV) = $\frac{\text{Velocity}}{\text{Sprint duration}}$

AV = 20/5 = 4

Average Velocity = 4

VELOCITY: Sprint 1 - 4

Sprint duration = 20 days

Velocity of team = 80 points

Average Velocity (AV) = $\frac{\text{Velocity}}{\text{Sprint duration}}$

AV = 80/20 = 4

Total Average Velocity = 4

7. CODING & SOLUTIONING

Feature 1 - PREDICTION USING CNN

In [73]:
```
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout
from keras.preprocessing.image import ImageDataGenerator
```

DATA AUGMENTATION

TESTING AND TRAINING

In [74]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='5Lj0H6wwDuTqeL545tX4VeIBqI9_mpPA6hiunt9ltdjb',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'imageclassification-donotdelete-pr-2slksiubpgncif'
object_key = 'Dataset.zip'

streaming_body_2 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']
```

In [75]:
```
from io import BytesIO
import zipfile
```

In [76]:
```
pwd
```

Out[76]:
```
'/home/wsuser/work'
```

In [77]:
```
import os
filesnames = os.listdir("/home/wsuser/work/Dataset/TRAIN_SET")
```

In [78]:
```
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [79]:
```
x_train = train_datagen.flow_from_directory(
    r'/home/wsuser/work/Dataset/TRAIN_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
x_test = test_datagen.flow_from_directory(
    r'/home/wsuser/work/Dataset/TEST_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```

```
Found 2626 images belonging to 5 classes.
Found 1055 images belonging to 5 classes.
```

In [80]:
```
print(x_train.class_indices)
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

In [81]:
```
print(x_test.class_indices)
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

In [82]:
```
from collections import Counter as c
c(x_train .labels)
```

Out[82]:
```
Counter({0: 606, 1: 445, 2: 479, 3: 621, 4: 475})
```

INITIALIZATION OF CNN

PERFORMING THE FOLLOWING:

1. Adding the convolution layer
2. Adding maxpooling layer
3. Second Maxpooling snf convolution layers
4. Flattening of layers
5. Adding Dense layer

In [83]:
```
classifier = Sequential()
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Conv2D(32, (3, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Flatten())
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=5, activation='softmax'))
```

In [84]:
```
classifier.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 62, 62, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 31, 31, 32)       0
 2D)

 conv2d_3 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_3 (MaxPooling  (None, 14, 14, 32)       0
 2D)
```

```
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning
-client) (1.21.41)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->b
oto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>=
1.21.21->boto3->watson-machine-learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-lea
rning-client) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machi
ne-learning-client) (2.11.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client)
(3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learn
ing-client) (2.0.4)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (20
21.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client)
(1.20.3)
```

In [35]:
```
from ibm_watson_machine_learning import APIClient
wml_credentials= {"url":"https://us-south.ml.cloud.ibm.com","apikey":"sFYJM2RQbIt39atcqI8BbnhEEJsBl9nBHWLeyeKC0vdq"}
client=APIClient(wml_credentials)
```

In [36]:
```
def guid_from_space_name(client,space_name):
    space=client.spaces.get_details()
    return(next(item for item in space['resources']if item['entity']["name"]==space_name)['metadata']['id'])
```

In [50]:
```
space_uid=guid_from_space_name(client,'imageclassification')
print("Space UID = "+ space_uid)
```

```
Space UID = c69b5cda-6240-47d0-9324-3f683ca61ce2
```

In [51]:
```
client.set.default_space(space_uid)
```

Out[51]:
```
'SUCCESS'
```

```
pytorch-onnx_1.1-py3.6-edt     32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37           36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10        396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6             39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt     40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10     40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9         4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                 42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                   493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6          49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6             4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6         50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8            52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11     55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                 5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1              5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                     5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
runtime-22.2-py3.10-xc         5e8cddff-db4a-5a6a-b8aa-2d4af9864dab  base
autoai-kb_3.1-py3.7            632d4b22-10aa-5180-88f0-f52dfb6444d7  base
-----------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

In [53]:
```
software_spec_uid=client.software_specifications.get_uid_by_name("tensorflow_1.15-py3.6")
software_spec_uid
```

Out[53]:
```
'2b73a275-7cbf-420b-a912-eae7f436e0bc'
```

In [54]:
```
software_spec_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
software_spec_uid
```

Out[54]:
```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

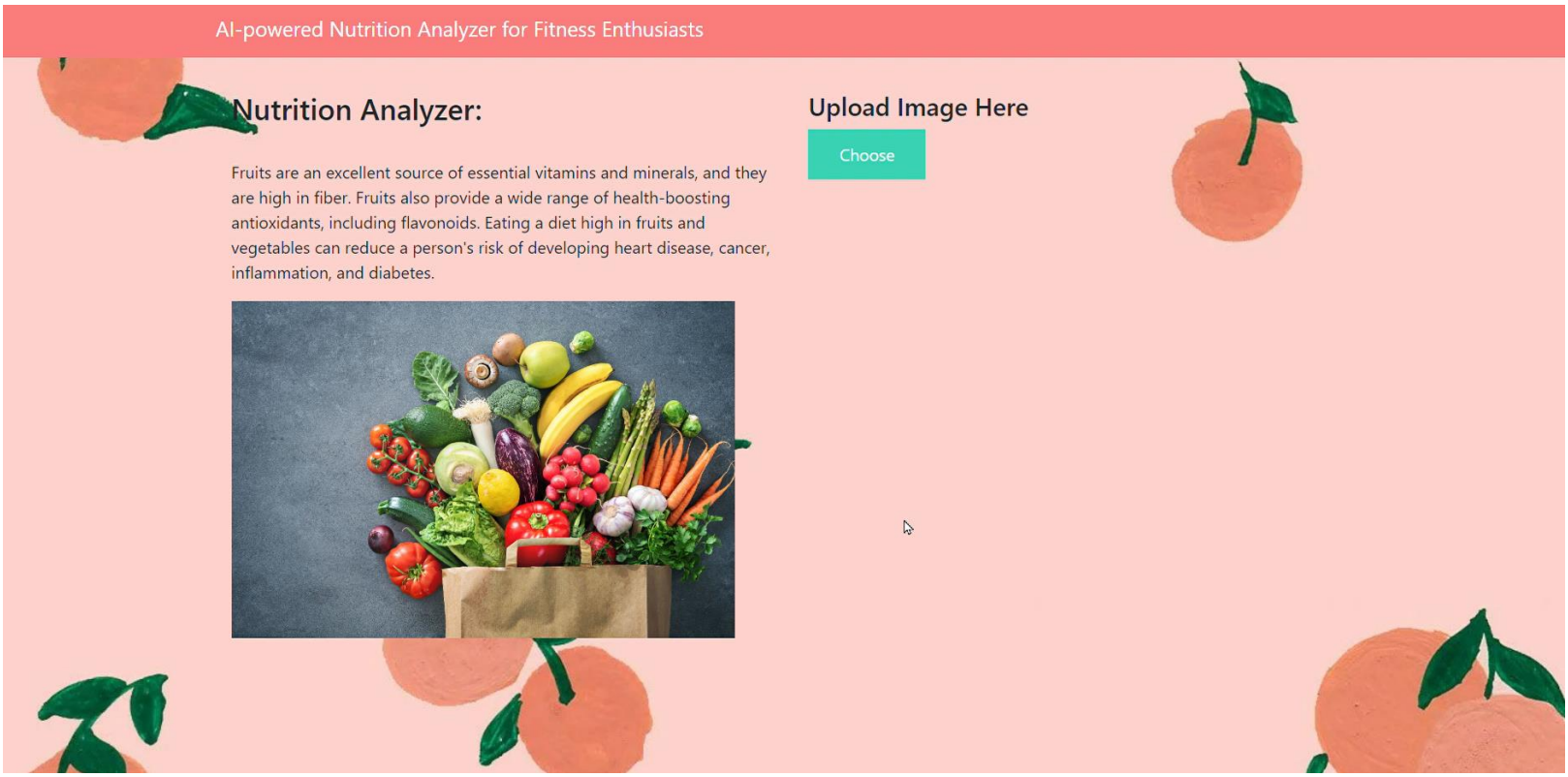
Feature 2- DETAILS GENERATION

```
import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask,render_template,request

app=Flask(__name__)

model=load_model('C:/Users/Malan/Desktop/Flask/nutrition.h5')

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

@app.route('/predict',methods=['GET','POST'])
def upload():
    text=""
    if request.method=='POST':
        f=request.files['image']
        basepath=os.path.dirname(__file_)
        filepath=os.path.join(basepath,'uploads',f.filename)
                                                    f.save(filepath)
        img=image.load_img(filepath,target_size=(64,64))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        pred=np.argmax(model.predict(x),axis=1)
        #index=['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']

        if pred==0:
            text="""APPLE===>
                *Calories 96
                *Protein - 0.59g
                *Carbohydrate 25g
                *Fats -0.39g
                *Dietary Fiber 4.4g
                *Sugar 14 g
                *Sodium 18mg
                *Potassium 194.7mg"""
            print(text)

        elif pred==1:
            text="""BANANA===>
```

```
            *Calories 105
            *Protein 1.39 g
            *carbohydrate 279g
            *Fats 0.49g
            *Dietary fiber 6.14g
            *Sodium 1.2 mg
            *Potassium 422 mg"""
        print(text)

    elif pred==2:
        text="""ORANGE===>
                *Calories 105
                *Protein 0.9g
                *Fats 0.1g
                *Carbohydrate 18g
                *Dietary fiber 2.39
                *Sugar 9g
                *Sodium 0mg
                *Potassium 173.8mg"""
        print(text)

    elif pred==3:
        text="""PINEAPPLE===>
                *Calories 452"
                *Protein-4.99g
                *Fats 11g
                *Carbohydrates -199g
                *Dietary Fiber 139g
                *Sugar 89g
                *Sodium 9.1 mg
                *Potassium 986.5mg"""
        print(text)

    elif pred==4:
        text="""WATERMELON===>
                *Calories 1371
                *Protein 26g
                *Fats-7g
                *Carbohydrate 341g
                *Dietary Fiber 18g
                *Sugar 280g
                *Sodium 45.2 mg
                *Potassium 5060.2 mg"""
        print(text)
```

```
    return text
if __name__=='__main__':
    app.run(debug=False)
```

FRONT END:



8. TESTING

Test Cases

Test Case ID	Purpose	TestCases	Result
TC 1	Validation	Image in PDF format	Image should be in JPG, JPEG or PNG
TC 2	Validation	Image in DOCX format	Image should be in JPG, JPEG or PNG
TC 3	Validation	Image in BMP format	Image should be in JPG, JPEG or PNG

9. RESULTS

Performance Metrics

- Tracking nutrients intake: Monitoring the diet plan and tracking all the nutrients intake.
- Validating outcome: Capture and find the nutrients present in the given data sample.
- Reports: The tracking app generates and sends reports to give a detailed insight about the diet plan,amount of calorie intake and nutritional value of the given sample.

10. ADVANTAGES

- **Achieve your fitness goals** with a tailored web app that perfectly fits your diet.
- Deliver an **outstanding** user experience through additional control over the app.
- Control the **security** of your user data.
- Increase **efficiency** and **user satisfaction** with an app aligned to their needs.

DISADVANTAGES

- Some nutritional software packages are of poor quality, and the technical support provided to users is sometimes **inadequate**.
- Although many excellent software packages and databases are available, they are open to **misuse** by users who do not understand or appreciate the limitations of such systems.
- This review examines some of the **sources of error** associated with the use of nutritional analysis software.

11. CONCLUSION

This work provided an overview of existing AI nutrition recommender systems, a field that has experienced substantial growth in the last few years. A categorization of such systems into task specific components was presented, along with approaches concerned with each component and relevant data-sets. An assessment of the feasibility of implementing an ideal AI nutrition recommender system using current methods was also provided, with the general conclusion being that some of the required components have not reached a mature state yet.

12. FUTURE

The project assists well to record the income and expenses in general. However, this project has some limitations:

- The application is unable to maintain the backup of data once it is uninstalled.
- This application does not provide higher decision capability.

To further enhance the capability of this application, we recommend the following features to be incorporated into the system:

- Multiple language interfaces.
- Provide backup and recovery of data.
- Provide a better user interface for users.
- Mobile apps advantage.

13. APPENDIX

Source Code :

INDEX.HTML:

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Nutrition Analyzer</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
    <style>
        #result {
            color: #000000;
        }

        body {
            background-color: #a0e4cb;
            background-image: url("https://www.transparenttextures.com/patterns/mirrored-squares.png");
            /* This is mostly intended for prototyping; please download the pattern and re-host for production
environments. Thank you! */
        }
    </style>
</head>

<body>

    <nav class="navbar navbar-dark bg-dark">
        <div class="container">
            <a class="navbar-brand" href="#">AI-Powered Nutrition Analyzer for Fitness Enthusiasts</a>
        </div>
    </nav>
    <div class="container">
        <div id="content" style="margin-top:2em">
            <div class="container">
                <div class="row" style="height:60%">
                    <div class="col-sm-6 bd">
                        <h3>NUTRITION ANALYZER: </h3>
                        <br>
                        <p>Nutritional analysis is the process of determining the nutritional content of food. It is a vital
part of analytical chemistry that provides information about the chemical composition, processing, quality control
and contamination
                            of food.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.</p>

                </div>
                <div class="col-sm-6">
                  <img style="height: 70%"
src="https://www.mlchc.org/sites/default/files/styles/max_650x650/public/2022-
03/nutrition_image2.jpg?itok=fUi0J40D" height="20%" ,width="5%">

                </div>

            </div>
            <div style="display: flex;justify-content:center;">
                <div>

                    <h4>Upload Image Here</h4>
                    <form action="http://localhost:5000/" id="upload-file" method="post" enctype="multipart/form-
data">
                        <label for="imageUpload" class="upload-label">
                        Choose
                    </label>
                        <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">
                    </form>


                    <div class="image-section" style="display:none;">
                        <div class="img-preview">
                            <div id="imagePreview">
                            </div>
                        </div>
                        <div>
                            <button type="button" class="btn btn-info btn-lg " id="btn-predict">Analyze!</button>
                        </div>
                    </div>

                    <div class="loader" style="display:none;"></div>

                    <h3>
                        <span id="result"> </span>
                    </h3>

                </div>
            </div>
        </div>

    </div>
  </div>
</body>

<footer>
   <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>

</html>
```

Python code:

```
import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask,render_template,request

app=Flask(_name_)

model=load_model('C:/Users/Malan/Desktop/Flask/nutrition.h5')

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

@app.route('/predict',methods=['GET','POST'])
def upload():
    text=""
    if request.method=='POST':
        f=request.files['image']
        basepath=os.path.dirname(_file_)
        filepath=os.path.join(basepath,'uploads',f.filename)
        f.save(filepath)
        img=image.load_img(filepath,target_size=(64,64))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        pred=np.argmax(model.predict(x),axis=1)
        #index=['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']

        if pred==0:
            text="""APPLE===>
                *Calories 96
                *Protein - 0.59g
                *Carbohydrate 25g
                *Fats -0.39g
                *Dietary Fiber 4.4g
                *Sugar 14 g
                *Sodium 18mg
                *Potassium 194.7mg"""
            print(text)

        elif pred==1:
            text="""BANANA===>
                *Calories 105
                *Protein 1.39 g
                *carbohydrate 279g
                *Fats 0.49g
                *Dietary fibre 6.14g
                *Sodium 1.2 mg
                *Potassium 422 mg"""
            print(text)

        elif pred==2:
            text="""ORANGE===>
```

```
                *Calories 105
                *Protein 0.9g
                *Fats 0.1g
                *Carbohydrate 18g
                *Dietary fiber 2.39
                *Sugar 9g
                *Sodium 0mg
                *Potassium 173.8mg"""
            print(text)

        elif pred==3:
            text="""PINEAPPLE===>
                *Calories 452"
                *Protein-4.99g
                *Fats 11g
                *Carbohydrates -199g
                *Dietary Fiber 139g
                *Sugar 89g
                *Sodium 9.1 mg
                *Potassium 986.5mg"""
            print(text)

        elif pred==4:
            text="""WATERMELON===>
                *Calories 1371
                *Protein 26g
                *Fats-7g
                *Carbohydrate 341g
                *Dietary Fiber 18g
                *Sugar 280g
                *Sodium 45.2 mg
                *Potassium 5060.2 mg"""
            print(text)

    return text
if__name_=='_main_':
    app.run(debug=False)
```

Github Link :

https://github.com/IBM-EPBL/IBM-Project-1071-1658340096

Project Demo Link:

https://drive.google.com/drive/folders/1-28fTFHKQZ8VGle0x6x7IJgBaXmMM0D6