NAALAIYA THIRAN PROJECT - 2022
19ECI01-PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

AI-powered Nutrition Analyzer for Fitness Enthusiasts

A PROJECT REPORT

Submitted by

PRAKASH R	**411519104067**
PRIYADHARSAN V	**411519104070**
RISHI ASHOK A	**411519104075**
SILAMBARASAN K	**411519104084**
SIVANANDHAN R	**411519104085**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PERI INSTITUTE OF TECHNOLOGY, MANNIVAKKAM– 600 048

Approved by AICTE, New Delhi. Accredited with B+ Grade by NAAC.

(Affiliated to Anna University)

ANNA UNIVERSITY: CHENNAI 600025

NOVEMBER 2022

PERI INSTITUTE OF TECHNOLOGY, MANNIVAKKAM – 600048
Approved by AICTE, New Delhi Accredited with B+ Grade by NAAC.
(Affiliated to Anna University)

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this report **"AI-powered Nutrition Analyzer for Fitness Enthusiasts"** is the bonafide work of **PRAKASH R (411519104067), PRIYADHARSAN V(411519104070), RISHIASHOK A (411519104075), SILAMBARASAN K(411519104084), AND SIVANANDHAN R (411519104085)** who carried out **19ECI01 Professional Readiness for Innovation, Employability and Entrepreneurship** project offered by IBM and Anna University, Chennai.

SIGNATURE
R.S.Abbirami
FACULTY MENTOR
Associate Professor
Department of
Computer Science and
Engineering

SIGNATURE
M.Divya
FACULTY EVALUATOR
Associate Professor
Department of
Computer Science and
Engineering

SIGNATURE
K VARALAKSHMI B.E.,M.E
HEAD OF DEPARTMENT
Professor
Department of
Computer Science and
Engineering

PROJECT CALENDER

Phase	Phase Description	Week	Dates	Activity Details
1	Preparation Phase (Pre-requisites, Registrations, Environment Set-up, etc.)	1	22 - 27 Aug 2022	Creation GitHub account & collaborate withProject repositoryin project workspace
2	Ideation Phase (Literature Survey, Empathize, DefiningProblem Statement,Ideation)	2	29 Aug – 3rd Sept 2022	Literature survey (Aim, objective, problem statement and need for the project)
		3	5 - 10th Sept 2022	Preparing Empathy Map Canvas tocapture theuser Pains & Gains
		4	12 - 17 Sept 2022	Listing of the ideas using brainstormingsession
3	Project Design Phase -I (ProposedSolution, Problem- Solution Fit, Solution Architecture)	5	19 - 24 Sept 2022	Preparin the proposed solution g docume nt
		6	26 Sept - 01 Oct 2022	Preparing problem - solution fit document &Solution Architecture
4	Project Design Phase -II (Requirement Analysis, CustomerJourney, Data FlowDiagrams,Technology Architecture)	7	3 - 8 Oct 2022	Preparing the customer journey maps
		8	10 - 15 Oct 2022	Preparing the Functional RequirementDocument & Data- Flow Diagrams and TechnologyArchitecture
5	Project Planning Phase (Milestones & Tasks, Sprint Schedules)	9	17 - 22 Oct 2022	Preparing Milestone & Activity List, SprintDelivery Plan
6	Project Development Phase (Coding & Solutioning, acceptance Testing, Performance Testing)	10	24 - 29 Oct 2022	Preparing Project Development - Delivery of Sprint-1
		11	31 Oct – 5 Nov 2022	Preparing Project Development - Delivery of Sprint-2
		12	7 - 12 Nov 2022	Preparing Project Development - Delivery ofSprint-3
		13	14 - 19 Nov 2022	Preparing Project Development - Delivery ofSprint-4

TABLE OF CONTENTS

SOURCE CODE

GITHUB AND DEMO LINK

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

Nutritional intake is the basis for human growth and health. The intake of different types of nutrients and micronutrients can affect health. Most diseases get linked with diet. Diabetes, cardiovascular diseases, gout, peptic ulcers, and gastroenteritis are all diet-related diseases. These are increasing in prevalence every year.The age group of above suffering from these diseases is gradually decreasing. The development of Internet has made it possible to conduct online nutrition surveys through large-scale food,nutrition databases.Which linked to automated dietary records. There are now a growing number of AI-POWERED NUTRITION ANALYZER FOR FITNESS ENTHUSIASTS software, platforms, and applications for nutrition surveys.This study develops an artificial intelligence model for a precision nutrient analysis, which allows users to enter the name of a dish and serving size to assess a total of 24 nutrients. The recipes get modified by the user, which allows the model to use in all countries and all contexts.

1.2 PURPOSE

Fitness trackers don’t just track your workouts. Nutrition is just as important as working out to achieve an active lifestyle A fitness tracker lets you watch and record your heart rate, daily burned calories and step counts. Self - tracking allows you to stick to a healthier diet, exercise more and sleep better.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING SOLUTION

A variety of medical problems can affect your appetite. Your illness , medicines or surgery can cause these problems. Many people get frustrated when they know they need to get well but they are not hungry. When they gain weight they get fatigued and unable to exact exercise. Each of the following sections describes a nutritional problem and suggests possible solutions.

2.2 PROBLEM STATEMENT DEFINITION

Nutritional problem may lead to decrease the appetite. Lack of appetite, or decreased hunger, is one of the most troublesome nutrition problems you can experience. Although it is a common problem, its cause is often unknown. Appetite-stimulating medicines are available.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a collaborative visualization used to express clearly what one knows about a particular type of user. It externalizes knowledge about users in order to create a shared understanding of user needs, and aid in decision making.

Empathy maps are split into 4 quadrants (Says, Thinks, Does, and Feels), with the user in the middle. Empathy maps provide a glance into who a user is as a whole. The ***Says*** quadrant contains what the user says or what he needs. The ***Thinks*** quadrant captures what the user is thinking throughout the experience. The ***Does*** quadrant encloses the actions the user takes. The ***Feels*** quadrant is the user's emotional state.

The empathy map for AI-powered Nutrition Analyzer for Fitness Enthusiasts for retailers is shown in Fig 3.1

Fig 3.1 Empathy map

3.2 IDEATION AND BRAINSTORMING

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. Brainstorming is usually conducted by getting a group of people together to come up with either general new ideas or ideas for solving a specific problem or dealing with a specific situation. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity. Both brainstorming and ideation are processes invented to create new valuable ideas, perspectives, concepts and insights, and both are methods for envisioning new frameworks and systemic problem solving.

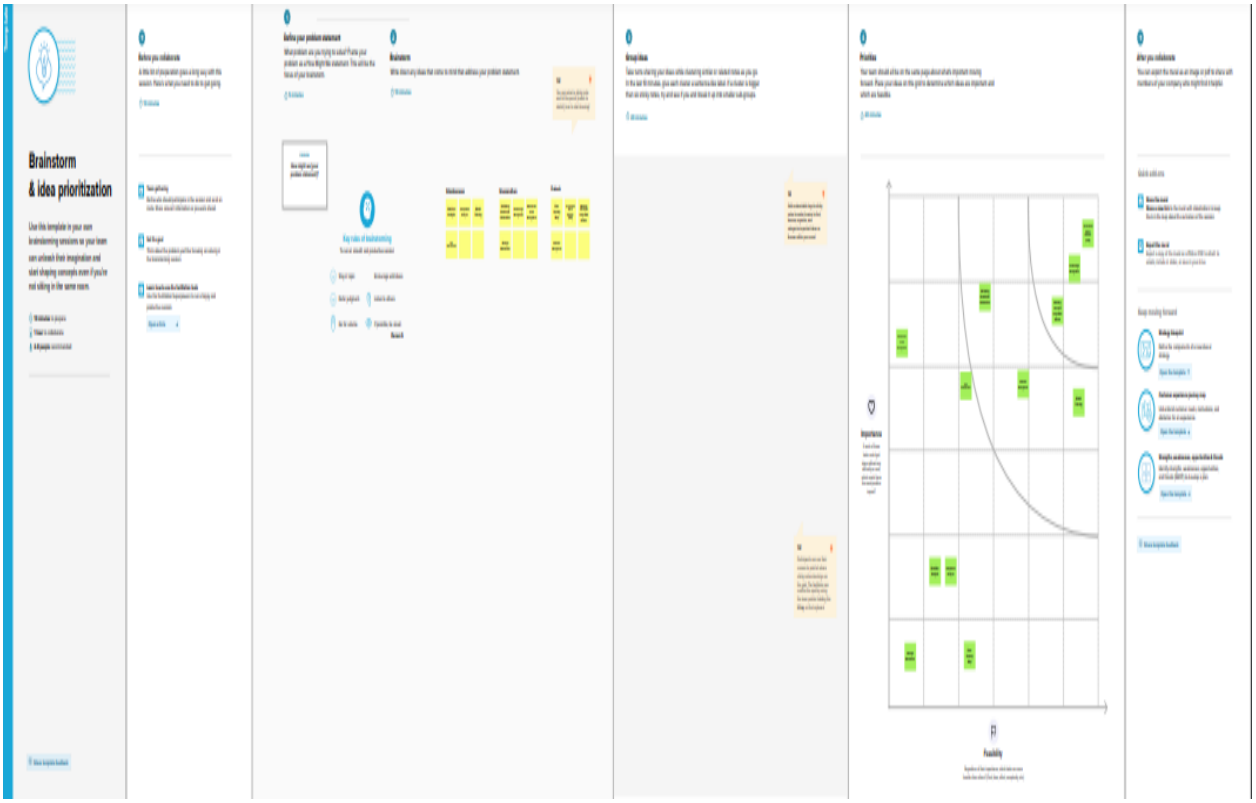

Table 3.2 Ideation and Brainstorming

3.3 PROPOSED SOLUTION

The proposed solution for AI-powered Nutrition Analyzer for Fitness Enthusiasts for retailers is shown in table 3.3

S.No.	Parameter	Description
1.	**Problem Statement (Problem to besolved)**	The main aim of the project is to build a model which is used for identifying the fruit depends on the different characteristics like colour, shape, texture etc using image processing. Here the user can capture the images of different fruits and then the image will be analysed with the trained model.
2.	**Idea / Solution description**	The image will be sent to the trained model. The model analyses the image and detects the nutrition based on the fruits like (Sugar, Fibre, Protein, Calorie intake, etc.). The above idea is achieved by using the Convolution Neural Network (CNN) .
3.	**Novelty / Uniqueness**	We aim to design an AI-powered Nutrition Analyzer for Fitness Enthusiastswhich is used to manage the inventory details and aims to save for the future investments.The Application will notify the user when a stock is about to complete.
4.	**Social Impact / Customer Satisfaction**	This will acquire knowledge and provide information about nutrition. Now a days, no one follows the diet plan. Providing this information, they come to know about the nutrition present in each food item to schedule a diet plan by taking the image of a food item and if we send it, we can get information about each food nutrition like carbohydrates, fat, proteins, vitamins, minerals and sugar.
5.	**Business Model (Revenue Model)**	Social media is the best way to spread the word about our application and with the help of influencers we can attract normal people.

		Clustering and targeting the fitness people with the help of local gyms.If the products sold through advertisements, then it is even better.
6.	**Scalability of the Solution**	Artificial intelligence (AI) can be used to predict investment outcomes quickly and effectively, as well as to devise strategies or establish long-term goals. Scalable AI pertains to how data models, infrastructures, and algorithms can increase or decrease their complexity, speed, or size at scale in order to best handle the requirements of the situation at hand.

Table 3.3 Proposed Solution

3.4 PROBLEM SOLUTION FIT

The Problem solution fit simply means that one have found a problem with the customer and that the solution one have realised for it actually solves the customers problem. The problem solution fit is an important step towards the Product-Market Fit. The structure of problem solution fit is given below.

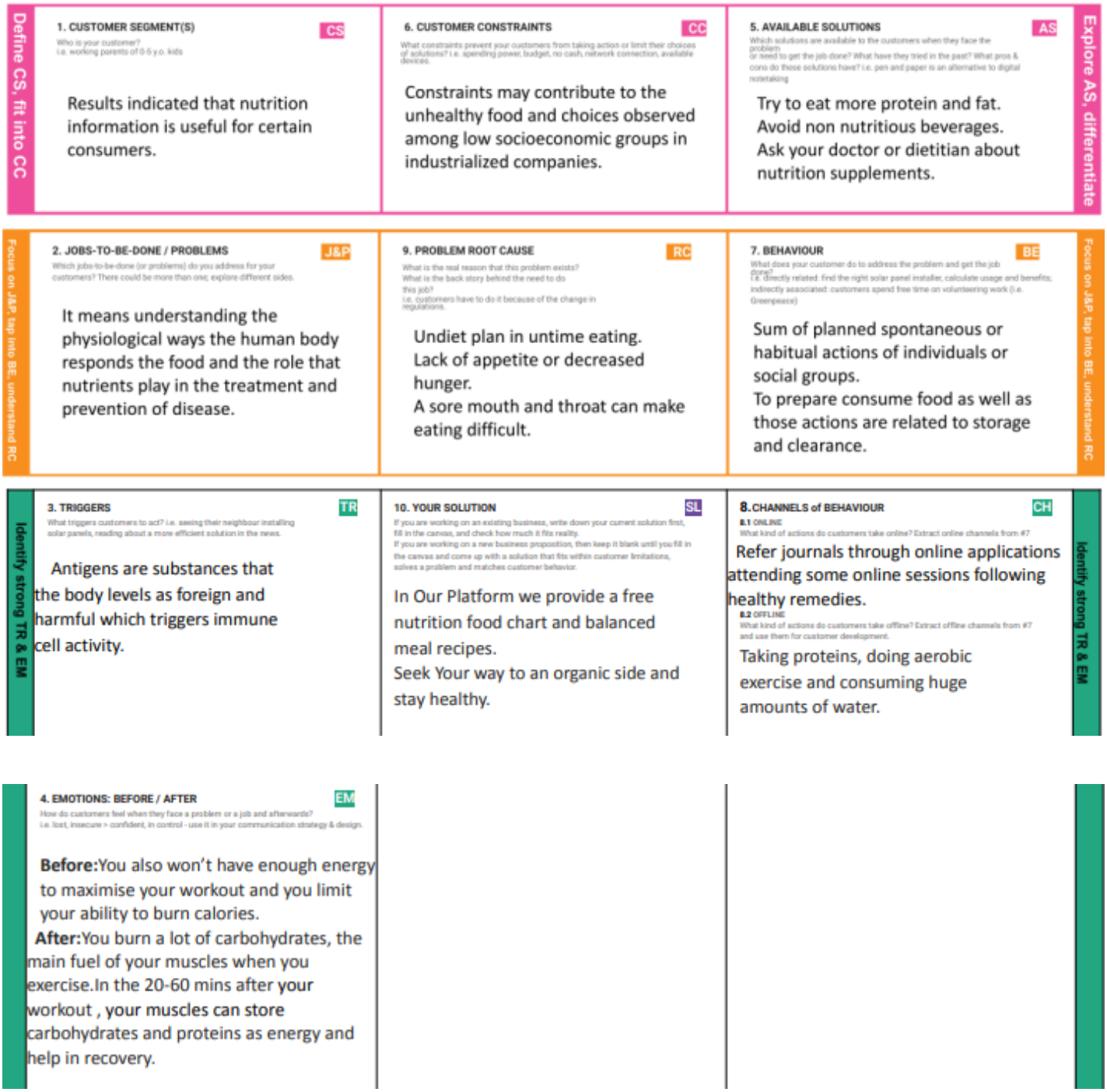

Fig 3.4 Problem Solution fit

CHAPTER 4

REQUIREMENT ANALYSIS

Requirements analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and Non-functional requirements.

4.1 FUNCTIONAL REQUIREMENTS

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements. The below table 4.1 shows the Functional Requirements for the cloud Based AI-powered Nutrition Analyzer for Fitness Enthusiasts

FR No.	**Functional Requirement (EPIC)**	**Sub Requirement (Story/ Sub Task)**
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation Via Email Confirmation Via OTP
FR-3	User login	Login through Google Login through Email
FR-4	Choose package	Selection of desired package
FR-5	Generate the daily plan	Daily plans will be generated by dietician
FR-6	Query	The user can ask for changes in plan

Table 4.2 Functional Requirements of cloud-based AI-powered Nutrition Analyzer for Fitness Enthusiasts

4.2 NON-FUNCTIONAL REQUIREMENTS

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements. The below table 4.2 shows the Non-Functional Requirements for the cloud Based AI-powered Nutrition Analyzer for Fitness Enthusiasts

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Easy to use with interactive User Interface
NFR- 2	**Security**	User can access only their personal information
NFR- 3	**Reliability**	The average time of failure shall be 7 days or 1 month
NFR- 4	**Performance**	The result hasto be shown within 5 sec
NFR-5	**Availability**	The dietician shall be available to users 24 hours a day or 7 days of a week
NFR-6	**Scalability**	Supports various food items

Table 4.2 Non-Functional Requirements of cloud-based AI-powered Nutrition Analyzer for Fitness Enthusiasts

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

There are four main elements of a DFD — external entity, process, data store, and data flow.

- **External entity**

An external entity, which are also known as terminators, sources, sinks, or actors, are an outside system or process that sends or receives data to and from the diagrammed system. They're either the sources or destinations of information, so they're usually placed on the diagram's edges. External entity symbols are similar across models except for Unified, which uses a stick-figure drawing instead of a rectangle, circle, or square.

- **Process**

Process is a procedure that manipulates the data and its flow by taking incoming data, changing it, and producing an output with it. A process can do this by performing computations and using logic to sort the data, or change its flow of direction. Processes usually start from the top left of the DFD and finish on the bottom right of the diagram.

- **Data store**

Data stores hold information for later use, like a file of documents that's waiting to be processed. Data inputs flow through a process and then through a data store while data outputs flow out of a data store and then through a process.

- **Data flow**

Data flow is the path the system's information takes from external entities through processes and data stores. With arrows and succinct labels, the DFD can show the direction of the data flow. The below Fig 5.1 shows the Data Flow Diagram for AI-powered Nutrition Analyzer for Fitness Enthusiasts for retailers

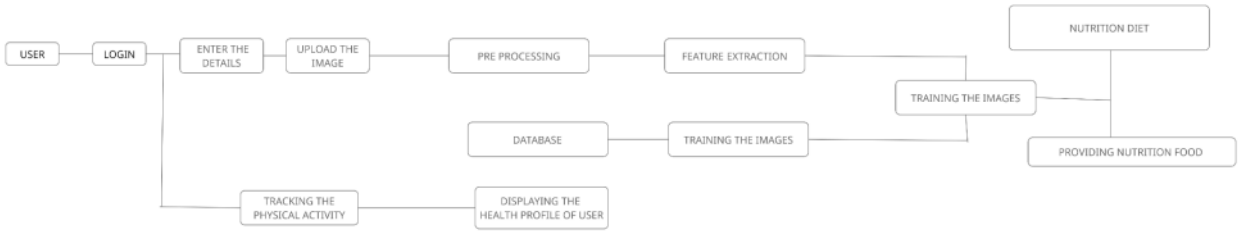

Fig 5.1: Data Flow Diagram for AI-powered Nutrition Analyzer for Fitness Enthusiasts for retailers

5.2 SOLUTION AND TECHNICAL ARCHITECHTURE

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

• Find the best tech solution to solve existing business problems.

• Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.

• Define features, development phases, and solution requirements.

• Provide specifications according to which the solution is defined, managed, and delivered.

The below figure 5.2 shows the Solution architecture of Nutrition Analyzer for retailers

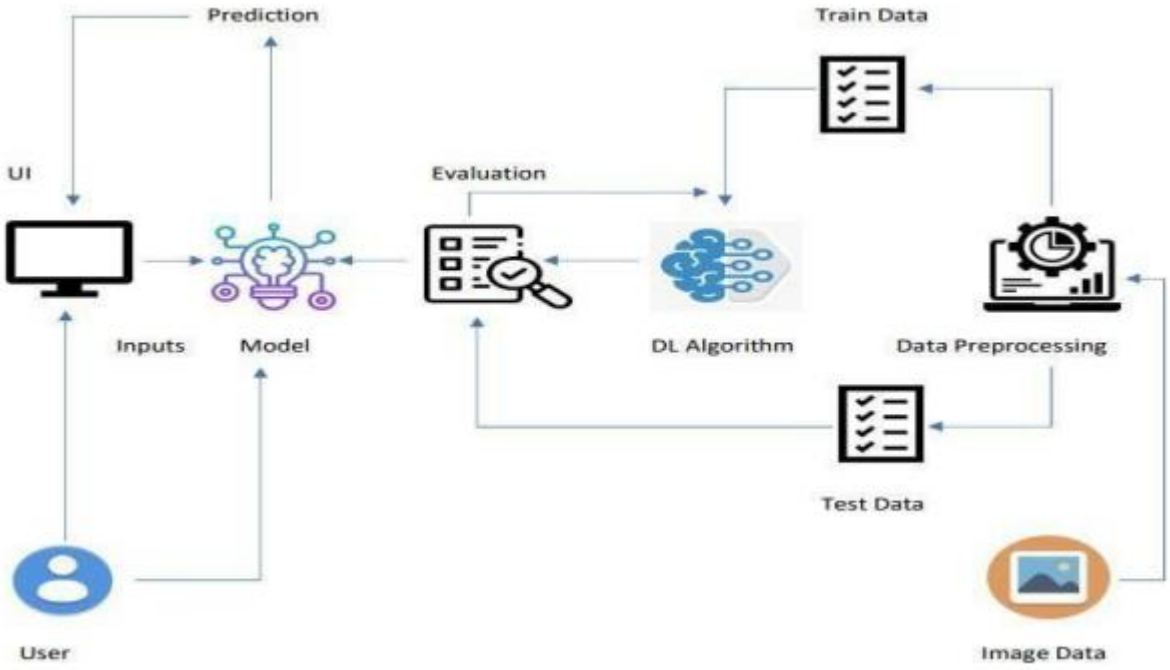

Fig 5.2: Solution architecture of Nutrition Analyzer for retailer

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team.It needs to decide on how long the time box is going to be, the sprint goal, and where it is going to start. The sprint planning session kicks off the sprint by setting the agenda and focus. If done correctly, it also creates an environment where the team is motivated, challenged, and can be successful. The below table 6.1 shows the Sprint Planning and estimation for AI-powered Nutrition Analyzer for Fitness Enthusiasts for Retailers

Sprint	Functional Requirement	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Data Collection	USN-1	Dataset - Collecting images of food items apples , banana, orange, pineapple, watermelon for analysis	5	High
Sprint-1	Image Preprocessing	USN-2	Image data augmentation - Increasing the amount of data by generating new data points from existing data	4	Medium
Sprint-1		USN-3	Image Data Generator Class - Used for getting the input of the original data	6	Medium
Sprint-1		USN-4	Applying image data generator functionality to train set and test set	5	High
Sprint-2	Modeling Phase	USN-5	Defining the model architecture - Building the model using deep learning approach and adding CNN layers	3	High
Sprint-2		USN-6	Training , saving, testing and predicting the model	6	High

Sprint-2	Development phase	USN-7	User database creation - It contains the details of users, Home page creation - It shows options of the application	5	Medium
Sprint-2		USN-8	Login and registration page creation - User can register and login through gmail with Id and password	6	Low
Sprint-3		USN-9	Dashboard creation — Dashboard contains the information of user profile and features of the application	4	Low
Sprint-3		USN-10	User Input Page Creation - It is for the user to feed the input images, Analysis and prediction page creation - It shows the prediction of given user input	6	Medium
Sprint-3		USN-11	Creation of about us , feedback and rating page — It shows application history and feedback page to users	5	Medium
Sprint-3	Application Phase	USN-12	Building the python code and importing the flask module into the Project, Create the Flask application and loading the model	5	Medium
Sprint-4		USN-13	API integration - Connecting front end and back end and perform routing and run the application	4	Low
Sprint-4	Deployment Phase	USN-14	Deployment of application by using IBM cloud	4	High
Sprint-4	Testing Phase	USN-15	Functional testing — Checking usability and accessibility	5	High
Sprint-4		USN-16	Non Functional testing — Checking scalability and performance of the application	3	Medium

6.2

6.3

6.4

Table 6.1: Sprint Planning and estimation for AI-powered Nutrition Analyzer for Fitness Enthusiasts

6.2 SPRINT DELIVERY SCHEDULE

The sprint delivery plan is scheduled accordingly as shown in the below table 6.2 which consists of the sprints with respective to their duration, sprint start and end date and the releasing date.

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	5 Days	29 Oct 2022	2 Nov 2022	20	3 Nov 2022
Sprint-2	20	5 Days	3 Nov 2022	07 Nov 2022	20	08 Nov 2022
Sprint-3	20	5 Days	08 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	16	5 Days	13 Nov 2022	17 Nov 2022	16	16 Nov 2022

Table 6.2: Sprint Planning done for AI-powered Nutrition Analyzer for Fitness Enthusiasts

CHAPTER 7
CODING & SOLUTIONING

7.1 IBM Cloud

The IBM Cloud platform combines platform as a service (PaaS) with infrastructure as a service (IaaS) to provide an integrated experience. The platform scales and supports both small development teams and organizations, and large enterprise businesses. Globally deployed across data centers around the world, the solution you build on IBM Cloud spins up fast and performs reliably in a tested and supported environment you can trust!

IBM Cloud provides solutions that enable higher levels of compliance, security, and management, with proven architecture patterns and methods for rapid delivery for running mission-critical workloads.

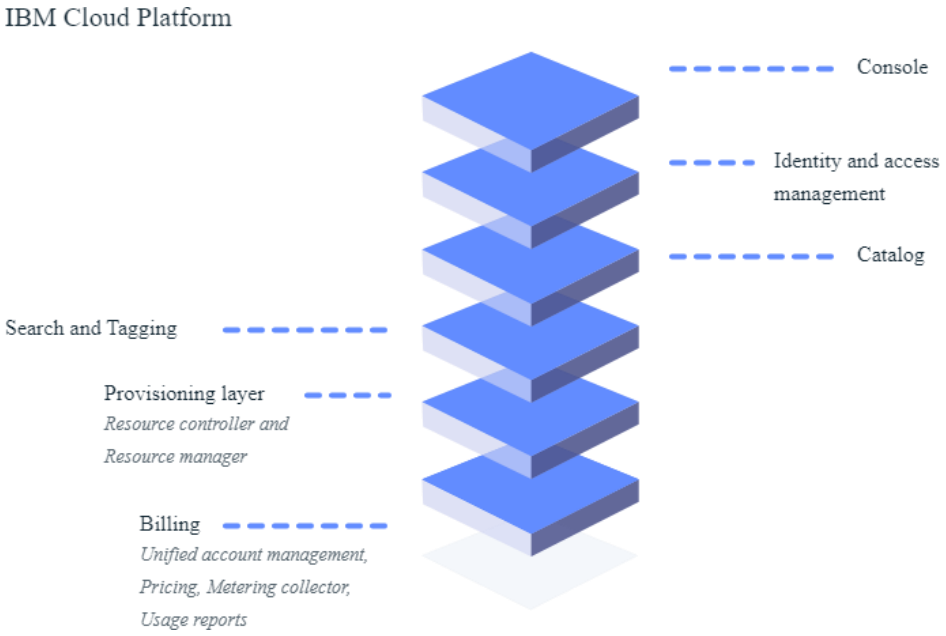

Fig 7.1: IBM cloud platform

7.2 Flask framework

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist

for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

7.3 IBM DB2 Module

Module features allow you to

- Extend schema support by allowing you to group together, in a named set, a collection of related data type definitions, database object definitions and other logic elements including:
    - SQL procedures
    - A module initialization procedure for implicit execution upon module initialization
    - User-defined data type definitions including: distinct type, array type, associative array type, row type, and cursor type
- Define a namespace such that objects defined within the module can refer to other objects defined in the module without providing an explicit qualifier.
- Add object definitions that are private to the module. These objects can only be referenced by other objects within the module.
- Add object definitions that are published. Published objects can be referenced from within the module or from outside of the module.
- Define published prototypes of routines without routine-bodies in modules and later implement the routine-bodies using the routine prototype.
- Initialize the module by executing the module initialization procedure for the module. This procedure can include SQL statements, SQL PL statements, and can be used to set default values for global variables or to open cursors.
- Reference objects defined in the module from within the module and from outside of the module by using the module name as a qualifier (2-part name support) or a combination of the module name and schema name as qualifiers (3-part name support).
- Drop objects defined within the module.
- Drop the module.
- Manage who can reference objects in a module by allowing you to grant and revoke the EXECUTE privilege for the module.

7.4 Docker CLI

The Docker client enables users to interact with Docker. The Docker client can reside on the same host as the daemon or connect to a daemon on a remote host. A docker client can communicate with more than one daemon. The Docker client provides a command line interface (CLI) that allows you to issue build, run, and stop application commands to a Docker daemon. The main purpose of the Docker Client is to provide a means to direct the pull of images from a registry and to have it run on a Docker host. Common commands issued by a client are:

- docker build
- docker pull
- docker run

7.5 IBM cloud CLI

IBM Cloud CLI provides full management of your IBM Cloud account via command line. Some installation steps described along this guide may need the IBM Cloud Command Line Interface (CLI) available to be performed.

7.6 SendGrid API

SendGrid's web API allows users to pull information about their email program without having to actually log on to SendGrid.com. Users can pull lists, statistics, and even email reports. In addition to this, users can send email via the web API without using traditional SMTP.

7.7 Kubernetes

Kubernetes is an open-source Container Management tool which automates container deployment, container scaling, and descaling and container load balancing (also called as container orchestration tool). It is written in Golang and has a huge community because it was first developed by Google and later donated to CNCF (Cloud Native Computing Foundation). Kubernetes can group 'n' number of containers into one logical unit for managing and deploying them easily. It works brilliantly with all cloud vendors i.e. public, hybrid and on-premises. Kubernetes is an open-source platform that manages Docker containers in the form of a cluster. Along with the automated deployment and scaling of containers, it provides healing by automatically restarting failed containers and rescheduling them when their hosts die. This capability improves the application's availability.

CHAPTER 8
TESTING AND RESULTS

8.1 TEST CASES:

The infusion of AI technology has done wonders for software testing inrecent times. AI and machine learning methods implement problem solving and reasoning algorithms to improve and automate software testing.

8.2 USER ACCEPTANCE TESTING:

• Full analysis of food recipes in real time - entity extraction, measure and quantity extraction with computation of the applicable nutrition for the recipe, applicable health and diet labels, and recipe classification for cuisine, meal, and dish types. Finally, it adjusts quantity for certain ingredients to account for the cooking process. For example, it calculates oil absorption for fried recipes, excludes solids from stock and broth recipes, calculates marinate absorption for marinates and much more.

• Extraction of food entities with measures and quantities from unstructured text.

• Usage in chatbots transcribing natural speech to text.

CHAPTER 9

PERFORMANCE RESULTS

9.1 PERFORMANCE METRICES

Fig 9.1 shows the performance metrics of the flask application using Google Developer Tools.

Fig 9.1: Performance metrices

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

- They can be used to determine the nutrient density of foods.
- They make it possible to quickly estimate the amount of calories in foods that we eat.
- They allow us to see how closely intake matches dietary standards.
- Analysis of food not only provides information about composition, appearance, texture, flavour , shelf-life, safety, processibility, and microstructure, but also guarantees product quality.
- Knowledge of the chemical and biochemical composition of foods is important to the health, well-being, and safety of the consumers.

10.2 DISADVANTAGES

- Variability in the composition of foods.
- Limited coverage of food items.
- Coverage of nutrients.
- Inappropriate database or food composition values.

CHAPTER 11

CONCLUSION

Dietary assessment is a necessary component of nutritional status assessment of Individuals, and also is useful for other purposes. It can be done using a variety of methods, each of which has advantages and limitations.

CHAPTER 12
FUTURE SCOPE

The employment opportunities in this field can range from being a consultant, food inspector, sports nutritionist, nutrition trainer to a food show host. In the private sector, the job roles can also involve that of a quality control officer, production manager, or supervisor.

CHAPTER 13
APPENDIX

A "composite" system is used to determine the expected nutrient content of each food group.Mixed foods are disaggregated into their ingredients,and similar ingredients are aggregated into item. The proportional intake of each item c luster within each food group or subgroup is calculated, and a nutrient-dense form of the food is selected as the representative food for each cluster. For example, the red - orange vegetable subgroup has 12 Item clusters, including cooked carrots, raw carrots, cooked tomatoes, and raw tomatoes. While cooked carrots may be consumed in many forms, plain cooked carrots are selected as the representative food for this cluster. Using the nutrients in each representative food and the item cluster's proportional intake, a nutrient profile is calculated for each food group or subgroup. Nutrient profiles are also calculated for oils and solid fats using food supply data to determine proportional intakes.

SOURCE CODE

APP.py

```
from flask import Flask,render_template,request
import os
import numpy as np
import requests
from tensorflow.keras.models import load_model #to load our trained model
from tensorflow.keras.preprocessing import image
import requests

app = Flask(__name__,template_folder='template') #initializing a flask app
# Loading the model
model=load_model('nutrition.h5')
print("Loaded model from disk")

@ app.route('/')# route to display the home page
def home():
    print("Loaded model from disk")
    return render_template('Home.html') #rendering the home page

@ app.route('/image1', methods=['GET', 'POST']) # routes to the index html
def image1():
    return render_template("Image.html")


@ app.route('/predict' ,methods=['GET','POST']) # route to show the predictions in a Web UI
```

```
def lanuch():
    if request.method=='POST':
        f=request.files['file'] # requesting the file
        f
        basepath=os.path.dirname('__file__') #storing the file directory
        print(basepath)
        filepath=os.path.join(basepath,"test",f.filename)

         #storing the file in uploads folder
        f.save(filepath) #saving the file

        img=image.load_img(filepath,target_size=(32,32)) #load and reshaping the image
        x=image.img_to_array(img) #converting image to an array
        x=np.expand_dims(x,axis=0) #changing the dimensions of the image

        pred=np.argmax(model.predict(x), axis=1)
        print("prediction",pred) #printing the prediction

index=['APPLE','BANANA','ORANGE','BANANA','WATERMELON','WATERMELON','WATE
RMELON','APPLE','BANANA','WATERMELON']

        result=str(index[pred[0]])
        print(result)
        x=result
        result=nutrition(result)
```

```
        print(result)

        return render_template("0.html",showcase=(result),showcase1=(x))
def nutrition(index):

    import requests

    url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"

    querystring = {"query":index}

    headers = {
        "X-RapidAPI-Key": "85887549f4msh51e7315b280a87ep1f43e0jsn585c940f2ea6",
        "X-RapidAPI-Host": "calorieninjas.p.rapidapi.com"
     }

    response = requests.request("GET", url, headers=headers, params=querystring)

    print(response.text)
    return response.json()['items']

if __name__ == "__main__":
    # running the app
    app.run(debug=False)
```

main.yml

```
name: Build and deploy Python app to Azure Web App - foood

  on:
    push:
      branches:
        - main
    workflow_dispatch:


jobs:
  build:
    runs-on: ubuntu-latest

steps:
- uses: actions/checkout@v2


- name: Set up Python version



uses: actions/setup-python@v1
with:
  python-version: '3.8'

- name: Create and start virtual environment
run:
   python -m venv venv source
   venv/bin/activate

- name: Install dependencies
run: pip install -rrequirements.txt


# Optional: Add step to run tests here (PyTest, Django test suites, etc.)



- name: Upload artifact for deployment jobs
uses: actions/upload-artifact@v2
with:
name: python-app
path: |
.
```

```
!venv/


deploy:
runs-on: ubuntu-latest
needs: build
environment:name: 'Production'
url: ${{ steps.deploy-to-webapp.outputs.webapp-url }}


steps:
- name: Download artifact from build job
uses: actions/download-artifact@v2 with:
name: python-app
path: .

- name: 'Deploy to Azure Web App'
uses: azure/webapps-deploy@v2 id:
deploy-to-webapp




    with:
    app-name: 'foood'
    slot-name:'Production'
    publish-profile: ${{
secrets.AZUREAPPSERVICE_PUBLISHPROFILE_F6FCF510CE004208B6D1C454B08695A7
}}
```

Demo.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<!-- Bootstrap CSS --> <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css
" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFA
W/dAiS6JXm" crossorigin="anonymous">
<title>Demo</title>
```

```
</head>
<body>

<style>

@charset "UTF-8";
 .image  {
   width:
   250px;float:
   left; margin:
   20px;
}

body {
   font-size:
   small;line-
   height: 1.4;
}

p {
   margin: 0;
}
 .performance-facts {
   border: 1px solid
   black;margin: 20px;
   float: left;
   width:
   320px;
   padding: 0.5rem;
}
 .performance-facts table
   { border-collapse:
   collapse;
}
 .performance-facts
   title {font-weight:
   bold;
   font-size: 2rem;
```

```
  margin: 0 0 0.25rem
  0;
}
 .performance-facts header {
  border-bottom: 10px solid
  black;padding: 0 0 0.25rem 0;
  margin: 0 0 0.5rem 0;
}
 .performance-facts header p
  {margin: 0;
}

 .performance-facts table, .performance-facts table--small, .performance-facts table--grid
  {width: 100%;
}
.performance-facts__table thead tr th, .performance-facts__table--small thead tr th,
.performance-facts_table--grid thead tr th,
.performance-facts_table thead tr td,
.performance-facts_table--small thead tr td,
 .performance-facts table--grid thead tr
  td {border: 0;
}
.performance-facts__table th, .performance-facts__table--small th, .performance-facts__table-grid
th,
.performance-facts_table td,
.performance-facts_table--small td,
 .performance-facts table--grid
  td {font-weight: normal;
  text-align: left;
  padding: 0.25rem
  0;
  border-top: 1px solid
  black;white-space:
  nowrap;
}
 .performance-facts table td:last-child, .performance-facts table--small td:last-child,
 .performance-facts table--grid td:last-child {
  text-align: right;
}
```

```
 .performance-facts table .blank-cell, .performance-facts table--small .blank-cell,
 .performance-facts table--grid .blank-cell {
  width: 1rem;
  border-top:
  0;
}
.performance-facts__table .thick-row th, .performance-facts__table--small .thick-row th,
.performance-facts__table--grid .thick-row th,
.performance-facts__table .thick-row td,
.performance-facts__table--small .thick-row td,
 .performance-facts table--grid .thick-row
  td {border-top-width: 5px;
}

.small-info {
  font-size: 0.7rem;
}
 .performance-facts table--
  small { border-bottom: 1px
  solid #999; margin: 0 0
  0.5rem 0;
}
 .performance-facts table--small thead
  tr {border-bottom: 1px solid black;
}
 .performance-facts table--small td:last-
  child {text-align: left;
}
.performance-facts__table--small th,
 .performance-facts table--small
  td {border: 0;
  padding: 0;
}
 .performance-facts table--
  grid {margin: 0 0 0.5rem 0;
}
 .performance-facts table--grid td:last-
  child {text-align: left;
}
```

```
 .performance-facts table--grid td:last-
  child::before {content: "•";
  font-weight: bold;
  margin: 0 0.25rem 0
  0;
}

.text-center {
  text-align: center;
}

.thick-end {
  border-bottom: 10px solid black;
}

.thin-end {
  border-bottom: 1px solid black;
}
</style>
<section class="performance-facts">
    <img src="/uploads/{{fname}}" alt={{fname}} class="performance-facts
 table"> <headerclass="performance-facts header">
    <h1 class="performance-facts title">{{data["foodName"].title()}}</h1>
     <p>Portion Size: {{data["serving_size"]}}</p>
  </header>
  <table class="performance-facts table">
   <thead>
    <tr>
     <th colspan="3" class="small-
      info">Nutrition Information
     </th>
    </tr>
   </thead>
   <tbody>
    <tr>
     <th colspan="2">
      <b>Calories</b>
      {{data["nutritional_info"]["calories"]}}
     </th>
    </tr>
```

```
    <tr class="thick-row">
      <td colspan="3" class="small-info">
        <b>% Daily Value*</b>
      </td>
</tr>
    {% for i in data["nutritional_info"]["dailyIntakeReference"] %}
    <tr>
      <th colspan="2">
        <b>{{data["nutritional_info"]["dailyIntakeReference"][i]["label"]}}</b>
        ({{data["nutritional_info"]["dailyIntakeReference"][i]["level"]}})
</th>       <td>
        <b>{{(data["nutritional_info"]["dailyIntakeReference"][i]["percent"])|round}}%</b>
</td>
    </tr>
    {% endfor %}
   </tbody>
 </table>

 <p class="small-info">* Percent Daily Values are based on a 2,000 calorie diet. Your daily
values may behigher or lower depending on your calorie needs:</p>

 <table class="performance-facts  table--small small-info">
   <thead>
     <tr>
       <td colspan="2"></td>
       <th>Calories:</th>
       <th>{{data["nutritional_info"]["calories"]}}</th>
     </tr>
   </thead>
   <tbody>
   {% for i in data["nutritional_info"]["totalNutrients"] %}
     <tr>
       <th colspan="2">{{data["nutritional_info"]["totalNutrients"][i]["label"]}}</th>
<td>Less than</td>

<td>{{data["nutritional_info"]["totalNutrients"][i]["quantity"]}}{{data["nutritional_info"]["tota
lNutrient s"][i]["unit"]}}</td>          </tr>
   {% endfor %}
   </tbody>
 </table>
```

```
   <p class="small-
    info">Calories per
    gram:
   </p>
   <p class="small-info text-center">
    Fat 9
    &bul
    l;
    Carbohydrate
    4&bull;
    Protein 4
   </p>

</section>
</section>
<script                    src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF
93hXpG5KkN" crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0
b4Q" crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>


</body>
</html>
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

```
 <!-- Bootstrap CSS --> <link rel="stylesheet"
 href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.cs
 s" integrity="sha384-
 Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
 crossorigin="anonymous">
<title>Food Calorie Estimation</title>
<style>
    body {
    background-color: #f2f7fb
}
.mt-100 {
    margin-top: 10px
}

.card {
    border-radius: 5px;
    -webkit-box-shadow: 0 0 5px 0 rgba(43, 43, 43, .1), 0 11px 6px -7px rgba(43, 43, 43, .1);
                                                                                box-
 shadow: 0 0 5px 0 rgba(43, 43, 43, .1), 0 11px 6px -7px rgba(43, 43, 43,
    .1);border: none;
    margin-bottom: 30px;
    -webkit-transition: all .3s ease-in-
    out;transition: all .3s ease-in-out
}
.card .card-header {
    background-color:
    transparent;border-bottom:
    none; padding: 20px;
    position: relative
}
 .card .card-header h5:after
    {content: "";
    background-color:
    #d2d2d2;width: 101px;
    height: 1px;
    position:
    absolute;bottom:
    6px;
    left: 20px
}
```

```
 .card .card-block {
    padding: 1.25rem
}
 .dropzone.dz-
    clickable {cursor:
    pointer
}
.dropzone {
    min-height: 150px;
    border: 1px solid rgba(42, 42, 42, 0.05);
    background: rgba(204, 204, 204,
    0.15);padding: 20px;
    border-radius: 5px;
    -webkit-box-shadow: inset 0 0 5px 0 rgba(43, 43, 43, 0.1);
    box-shadow: inset 0 0 5px 0 rgba(43, 43, 43, 0.1)
}

.m-t-20 {
    margin-top: 20px
}
.btn-primary,
.sweet-alert button.confirm,
 .wizard>.actions a {
    background-color:
    #4099ff;border-color:
    #4099ff; color: #fff;
    cursor: pointer;
    -webkit-transition: all ease-in
    .3s;transition: all ease-in .3s
}

.btn {
    border-radius: 2px;
    text-transform:
    capitalize;font-size:
    15px;
    padding: 10px 19px;

    cursor: pointer
```

```
}
</style>
</head>
<body>
<div class="row d-flex justify-content-center mt-100">
    <div class="col-md-
                8">
     <div class="card">
        <div class="card-header">
            <h3>Food Calorie Estimation</h4>
            <h6>Estimate live food calories & nutrition information from a single food image</h6>
 <a href="https://cal.virajman3.repl.co/demo/2">Example Click Here</a>
          </div>
          <div class="card-block">
            <form action="/result" method = "POST" class="dropzone dz-clickable"
 enctype ="multipart/form-data">
                  <input type = "file" name = "file" />
            <div class="text-center m-t-20">
              <input class="btn btn-primary" type = "submit"/>
            </div>
            </form>
<br>
<h4>Instructions:</h4>
<dl>
  <dt>Limitations</dt>
  <dd>- The image size must be under 1024KB.</dd>
  <dd>- The image format must be in JPEG, JPG or PNG.</dd>
  <dt>Do's</dt>
  <dd>- Take pictures from an eye-level perspective.</dd>
  <dd>- Take a picture for each food item separately.</dd>
  <dd>- Center the food on the picture.</dd>
  <dd>- Upload squared images, meaning that height and width are the same.</dd>
  <dt>Dont's</dt>
  <dd>- Occlusions of other items.</dd>
  <dd>- Top or side view images.</dd>
  <dd>- Include only a part of the food.</dd>
  <dd>- Blurry images.</dd>
  <dd>- Images taken on screens or display monitors.</dd>
  <dd>- Images that include multiple food items.</dd>
</dl>
<br>
<h4>Example:</h4>
```

```
<div class="card-deck">
  <div class="card">
<embed type="text/html" src="https://cal.virajman3.repl.co/demo/2" height="1300">  </div>
  <div class="card">
<embed type="text/html" src="https://cal.virajman3.repl.co/demo/1" height="1300"> </div>
</div>
        </div>
      </div>
    </div>
</div>

<script                    src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF
93hXpG5KkN" crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b
4Q" crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>

</body>
</html>
```

indexold.html

```
      <form class="file-upload-wrapper" action = "/result" method =
      "POST"enctype = "multipart/form-data">
      <input type = "file" name = "file" />
      <input type = "submit"/>
    </form>
```

Result.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags -->
```

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<!-- Bootstrap CSS --> <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css
" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFA
W/dAiS6JXm" crossorigin="anonymous">
<title>Result</title>

</head>
<body>

<style>

@charset "UTF-8";
 .image  {
   width:
   250px;float:
   left; margin:
   20px;
}

body {
   font-size:
   small;line-
   height: 1.4
 solid    #999;    margin:    0    0
   0.5rem 0;
```

```
}

p {  margin:
  0;
}
  .performance-facts {
    border: 1px solid
    black;margin: 20px;
    float: left;
    width:
    320px;
    padding: 0.5rem;
}
  .performance-facts table
    { border-collapse:
    collapse;
}
  .performance-facts title
    {font-weight: bold;
    font-size: 2rem;
    margin: 0 0 0.25rem
    0;
}
  .performance-facts header {
    border-bottom: 10px solid
    black;padding: 0 0 0.25rem 0;
    margin: 0 0 0.5rem 0;
}
  .performance-facts header p
    {margin: 0;
}

  .performance-facts table, .performance-facts table--small, .performance-facts table--grid
    {width: 100%;
}
.performance-facts__table thead tr th, .performance-facts__table--small thead tr th,
.performance-facts_table--grid thead tr th,
.performance-facts_table thead tr td,
```

```
.performance-facts_table--small thead tr td,
 .performance-facts table--grid thead tr
  td {border: 0;
}
.performance-facts__table th, .performance-facts__table--small th, .performance-facts__table-grid
th,
.performance-facts_table td,
.performance-facts_table--small td,
 .performance-facts table--grid
  td {font-weight: normal;
  text-align: left;
  padding: 0.25rem
  0;
  border-top: 1px solid
  black;white-space:
  nowrap;
}
 .performance-facts table td:last-child, .performance-facts table--small td:last-child,
 .performance-facts table--grid td:last-child {
  text-align: right;
}
.performance-facts__table .blank-cell, .performance-facts__table--small .blank-cell,
 .performance-facts table--grid .blank-cell { width:
  1rem;border-top: 0;
}
.performance-facts__table .thick-row th, .performance-facts__table--small .thick-row th,
.performance-facts_table--grid .thick-row th,
.performance-facts_table .thick-row td,
.performance-facts_table--small .thick-row td,
 .performance-facts table--grid .thick-row
  td {border-top-width: 5px;
}

.small-info {
  font-size: 0.7rem;
}
 .performance-facts        table--
  small {border-bottom: 1px
```

```
}
 .performance-facts table--small thead
  tr {border-bottom: 1px solid black;
}
 .performance-facts table--small td:last-
  child {text-align: left;
}
.performance-facts_table--small th,
 .performance-facts table--small
  td {border: 0;
  padding: 0;
}
 .performance-facts table--
  grid {margin: 0 0 0.5rem 0;
}
 .performance-facts table--grid td:last-
  child {text-align: left;
}
 .performance-facts table--grid td:last-
  child::before {content: "•";
  font-weight: bold;
  margin: 0 0.25rem 0
  0;
}

.text-center {
  text-align: center;
}

.thick-end {
  border-bottom: 10px solid black;
}

.thin-end {
  border-bottom: 1px solid black;
}
```

```
</style>
<section class="performance-facts">
    <img src="/uploads/{{fname}}" alt={{fname}}  class="performance-facts  table">
   <header class="performance-facts  header">
    <h1 class="performance-facts  title">{{data["foodName"].title()}}</h1>
     <p>Portion Size: {{data["serving_size"]}}</p>
   </header>
   <table class="performance-facts  table">
    <thead>
     <tr>
      <th colspan="3" class="small-
       info">Nutrition Information
      </th>
     </tr>
    </thead>
    <tbody>
     <tr>
      <th colspan="2">
       <b>Calories</b>
       {{data["nutritional_info"]["calories"]}}
      </th>
     </tr>
     <tr class="thick-row">
      <td colspan="3" class="small-info">
       <b>% Daily Value*</b>
      </td>
 </tr>
     {% for i in data["nutritional_info"]["dailyIntakeReference"] %}
     <tr>
      <th colspan="2">
       <b>{{data["nutritional_info"]["dailyIntakeReference"][i]["label"]}}</b>
       ({{data["nutritional_info"]["dailyIntakeReference"][i]["level"]}})
 </th>      <td>
       <b>{{(data["nutritional_info"]["dailyIntakeReference"][i]["percent"])|round}}%</b>
 </td>
```

```
    </tr>
    {% endfor %}
   </tbody>
  </table>

  <p class="small-info">* Percent Daily Values are based on a 2,000 calorie diet. Your daily values may behigher or lower depending on your calorie needs:</p>

  <table class="performance-facts  table--small small-info">
   <thead>
    <tr>
     <td colspan="2"></td>
     <th>Calories:</th>
     <th>{{data["nutritional_info"]["calories"]}}</th>
    </tr>
   </thead>
   <tbody>
   {% for i in data["nutritional_info"]["totalNutrients"] %}
    <tr>
     <th colspan="2">{{data["nutritional_info"]["totalNutrients"][i]["label"]}}</th>
<td>Less than</td>

<td>{{data["nutritional_info"]["totalNutrients"][i]["quantity"]}}{{data["nutritional_info"]["tota
lNutrient s"][i]["unit"]}}</td>          </tr>
   {% endfor %}
   </tbody>
  </table>

  <p class="small-
   info">Calories per
   gram:
  </p>
  <p class="small-info text-center">
   Fat 9
   &bull
   ;
   Carbohydrate
   4&bull;
```

```
    Protein 4
   </p>

</section>
<script      src="https://code.jquery.com/jquery-3.2.1.slim.min.js"      integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5K
kN" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>

</body>
</html>
```

home.html

```
        @app. route( '/' )# route to display the home page
def home ( ) :
        return render_template( 'home. html ')#rendering the home page


@app. route( '/image1 ' , methods=[ 'GET', 'POST'])# routes to the index html
def image1() :
        return render_template("image. html")


@app. route('/predict', methods=[ 'GET', 'POST'])# route to show the predictions in a web UI

def launch() :

        if request. Method ==  “POST”:

        f=request. files[ 'file'] #requesting the file

        basepath=os. path. dirname( ' file_')#storing the file directory

        pathos. path. join(basepath, "uploads", f. filename)#storing the file in uploads folder

        f. save(filepath)#saving the file



        img=image . load_ing(filepath, target_size=(64, 64)) #load and reshaping the image
```

```
x-image. ing_to_array(ing)#convert converting image to an array
x=np. expand_dims (x, axis=0)#changing the dimensions of the image

result=str(index[pred[0]])
x=result
print(x)
result-nutrition(result)
print(result)
render_template( "0. html", showcase=(result), showcase1=(x))

pred=np. argmax(model. predict(x), axis=1)
print ("prediction", pred)#printing the prediction
index=[ 'APPLES' , 'BANANA' , "ORANGE' , 'PINEAPPLE' , 'WATERMELON' ]
def nutrition ( index) :

        url = "https://calorieninjas. p. rapidapi.com/v1/nutrition"
        querystring = {"query": index}

        headers = {
                'x-rapidapi-key' : "5d797ab107mshe668f26bd044e64piffd34jsnf47bfa9aBee4" ,
                'x-rapidapi-host': "calorieninjas.p. rapidapi.com"
                }
        response = requests . request("GET", url, headers=headers, params=querystring)
        print (response . text)
        return response . json ( ) [ 'items']

if __name__  == “__main__”:
        #running the app
        app.run(debug==False
```

GitHub: -https://github.com/IBM-EPBL/IBM-Project-7730-16588967941658896794

Demo link: https://drive.google.com/file/d/1tFfF22_nyf1vG29PXij-PGfTviBvC6R0/view?usp=share_link

REFERENCES

1. Snap Meal Meal Snap for iPhone: Magical Meal Logging: https://apps.apple.com/us/app/mealsnap-photofooddiary/id1431522193
2. Neutrino- Artificial Intelligence Nutrition App : https://www.nutrinohealth.com/