AI-POWERED NUTRITION FOR FITNESS ENTHUSIAST

Serial number	Register number	name
1	913319104018	Francis huban K
2	913319104019	Ganapathy Ram SK
3	913319104001	Abbas noor mohammad.S
4	913319104301	Gowshigan E.R

BE-CSE

1. INTRODUCTION

Physical fitness is a general state of health and well-being and, more specifically, the ability to perform aspects of sports, occupations and daily activities. Physical fitness is generally achieved through proper nutrition, moderate-vigorous physical exercise, physical activity, and sufficient rest. Before the industrial revolution, fitness was defined as the capacity to carry out the day's activities without undue fatigue. However, with automation and changes in lifestyles physical fitness is now considered a measure of the body's ability to function efficiently and effectively. In today's lifestyle, people are moving towards achieving a fit and healthy body. This shift has changed the way of living in almost every household. Now everyone craves for healthy and nutritious food to be placed on their plates. Hence, healthy and nutritious food have become an essential part of everyone's lifestyle to achieve a balanced and healthy life in such a busy and hectic environment. Hence, to make their fitness path a bit smoother and to enhance their experience, we have created an Web application to provide a broader approach in providing a better living through nutritious and fit diet plan to the users.

1.1 PROJECT OVERVIEW

The platform provides nutrition-based data services, analytics, and technologies to its consumers and wants to turn itself into a leading source of nutrition-related insight platform. This application is created using AI and image classification technology to identify the food correctly and accurately and also calculated the amount of calories obtained using classification of image by integrating with deep learning concepts. The platform also further breaks down the nutrition information calories, macro and micronutrients as well as ingredients. Further, by connecting the AI with diverse data sets, the food which is automatically recognised by the platform is paired with detailed nutrition information.

1.2 PURPOSE

To create positive attitudes toward good nutrition and physical activity and provide motivation for improved nutrition and lifestyle practices conducive to promoting and maintaining the best attainable level of wellness for an individual. To provide adequate knowledge and skills necessary for critical thinking regarding diet and health so the individual can make healthy food choices from an increasingly complex food supply.Information on the roles of nutrition and physical activity in maintaining health and independence, and preventing or managing chronic diseases such as diabetes, heart disease, high blood pressure, osteoporosis, and arthritis. To assist the individual to identify resources for continuing access to sound food and nutrition information.

2. LITERATURE SURVEY

ABSTRACT

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 maintain 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 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.).

2.1 EXISTING PROBLEM

SURVEY:1

Artificial intelligence in nutrients

Author: *Jaroslaw sak andmagdalena suchodolska.*

AI modeling has been identified with research organization of the production of several nutrients. Huang et al. define an artificial natural network shown by the production of a retinol derivative named acting laureate. It leads to the chair of the department of humanities and social medicine university of Lubin,20-093, Poland. It is the form of biomolecular resources research infrastructure Poland of medicine, Lubin of the medical university the address should be corresponded by the users.

SURVEY: 2

Artificial intelligence application in nutrition diabetics.

Author: *Izzetulker.*

It may be discussed among the disciplines most of them are extensively important to the field function with scientific evidence it is good preparation and different apps will play a important role in health promotion. is a comprehensively nutritional assessment that explore the nutrition status it includes the dietary history (food recall food frequency which are questionaries in digital image technology to monitor of food like 38 physical examination. It is that the presence of a skilled nutritional specialist to interpret information obtained from the assessment is imperative.

SURVEY : 3

AI in research in production of nutrients.

Author : *Anahtar kelimeler – beslenme ve diyetetik;yapayzeka.*

Artificial intelligence is increasing rapidly due to the medical field of diagnosis risk estimation and medical diagnosis. The research may face many challenges when decided which method is the best. issues of participants' burden motivation and willingness to accurately report diet, and participation literacy and memory should be considered. Also the time to enter and analyze diet data, and therefore the availability of resources to accurately analyze dietary recalls, must be taken into an account before starting all the method has some dietary assessment have limitations.

SURVEY: 4

AI in research on the production of nutrients.

Author: *Kumar satanic, D; Yadav.D*

Compiling the nonlinear nutrients among 16 nutrients as GA – FUZZY – a evolutionary algorithm is comprised of the genetic algorithm of and genetic algorithm. Where that the use of technology indicate the vitamin d on functioning. he used as the evolutionary of algorithm among genetic algorithm (GA). And the fuzzy logic methodology(FLM) for the optimation of the production of phycobiliproteins(PBS) From cyanobacteria. Most of the function is to prevent it biomedically by research on vitamins

SURVEY: 5

DietaryAssignment nutrition,

Author : *Rozga m, Latulippeme,*

Researchers have major challenges about assigning dietary intake the different methods such as 3 days record 24 – H recaller food frequency requirement. It is specified the special individual Information found in science to promote dietary changes that positively affect health outcomes is the technology of the development function like nutrigenomics, nutrigenetics, metabolisms, and also foodmics it invites the new sight of the molecular level of data to improve some special invention of nutrition

2.2 REFERENCES

1. Oka, R.; Nomura, A.; Kometani, M.; Gondoh, Y.; Yoshimura, K.; Yoneda, T.Study protocol for the effects of artificial intelligence { AI } – supported automated nutritional intervention on glycemic control in patients with 2 diabetes mellitus.

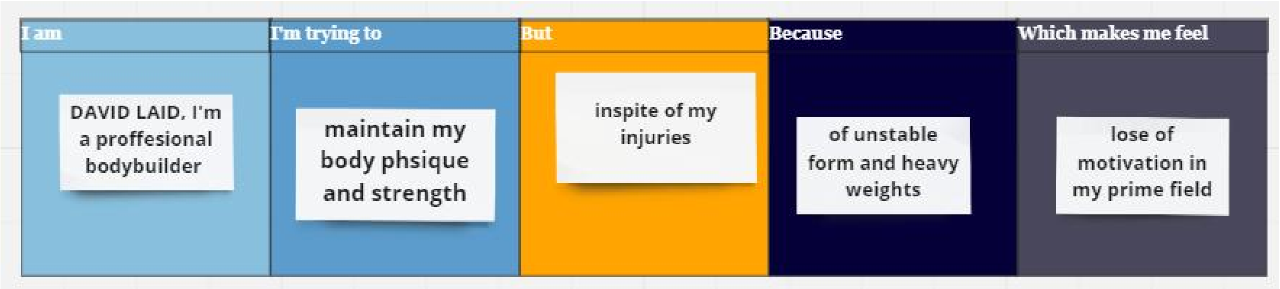
2. Kumar Saini, D.; rabbi, S.; Chhabra, D.; Shukla, P.Phycobiliproteins from Anabaena variabilis CCC421 and its production enhancement strategies using combinatory evolutionary algorithm approach.Bioresource.techno;.2020,309,123347.

3. Rozga M, Latulippe ME, Steiber A. "Advancements in personalized nutrition technologies: guiding principles for registered dietitian nutritionists'' Jornal of the acdemy of nutrition and dietetics.

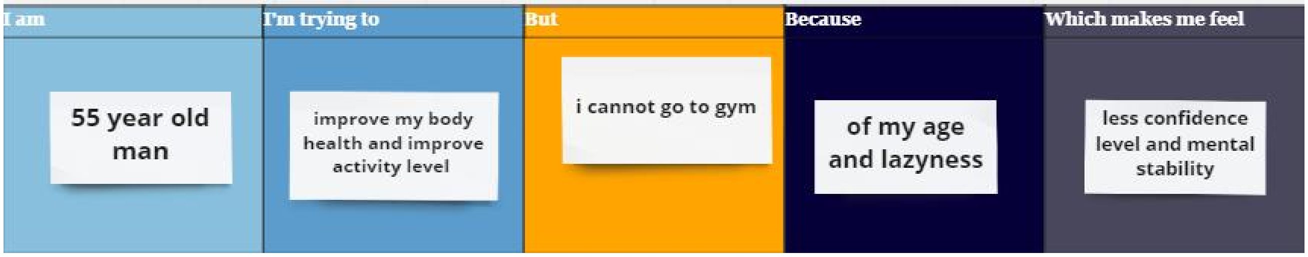
4. Haung, S.- M.; Li, H.-J.; Liu,Y.-C.; Kuo, C.;H.; Shieh,C.J An efficient approach for lipase – caralyzed synthesis of retinl laurate nutraceutical by combining ultrasound assistance anf artificial neural network ----- Genatic algorithm to optimize wheat germ fermentation condition: application to the production of two anti – tumor benzoquinones.

2.3 PROBLEM STATEMENT DEFINITION

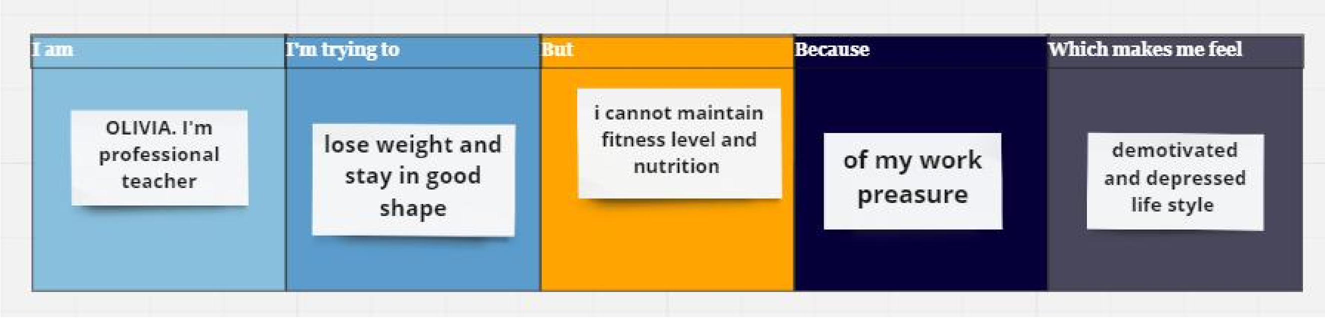
Create a problem statement to understand the customer's point of view. The Customer Problem Statement template helps us to focus on what matters to create experiences people will love. A well-articulated customer problem statement allows us to find the ideal solution for the challenges customers face. Throughout the process, we’ll also be able to empathize with our customers, which helps us better to understand how they perceive our product or service.



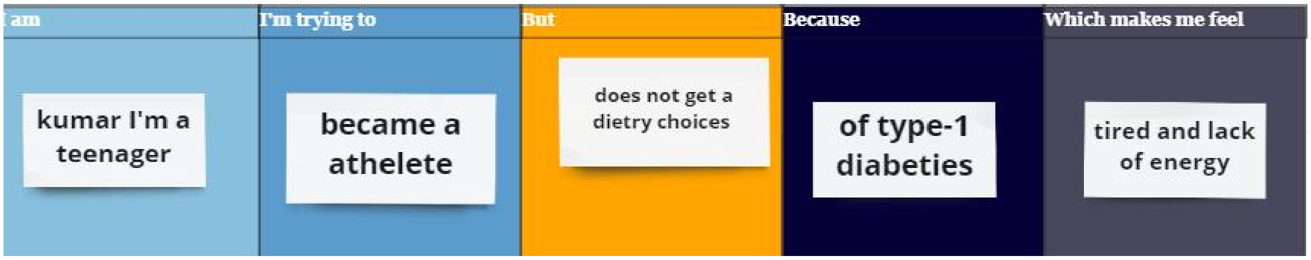
I am	I'm trying to	But	Because	Which makes me feel
DAVID LAID, I'm a proffesional bodybuilder	maintain my body phsique and strength	inspite of my injuries	of unstable form and heavy weights	lose of motivation in my prime field



I am	I'm trying to	But	Because	Which makes me feel
55 year old man	improve my body health and improve activity level	i cannot go to gym	of my age and lazyness	less confidence level and mental stability



I am	I'm trying to	But	Because	Which makes me feel
OLIVIA. I'm professional teacher	lose weight and stay in good shape	i cannot maintain fitness level and nutrition	of my work preasure	demotivated and depressed life style



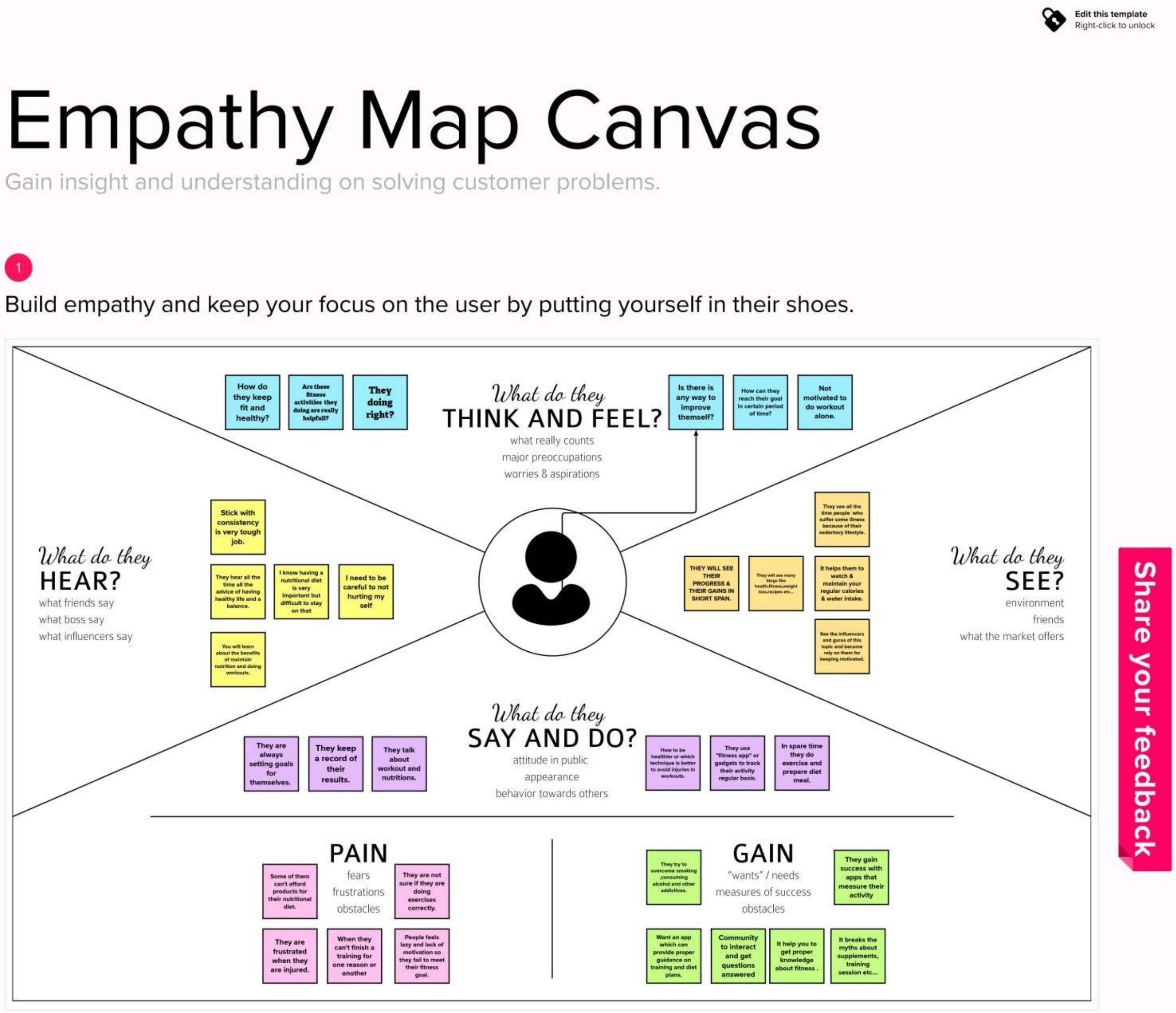
I am	I'm trying to	But	Because	Which makes me feel
kumar I'm a teenager	became a athelete	does not get a dietry choices	of type-1 diabeties	tired and lack of energy

PROBLEM STATEMENT (PS)	I AM CUSTOMER)	I'M TRYING TO	BUT	BECAUSE	WHICH MAKES ME FEEL
PS-1	College student	Build strong muscles and bones	I'm suffering from diabetes	Lactose intolerance in my body	Less motivated and unstable mindset
PS-2	Professional body builder	Maintain my body physique and strength	Inspite of my injuries	Of unstable form and heavy weight	Less motivation in my prime field
PS-3	Professional teacher	Lose weight and stay in good shape	I cannot maintain my nutrition and fitness level	Of my work pleasure	Demotivated and depressed life style
PS-4	55 year old man	Improve my body's health and my activity level	I cannot go to the gym	Of my age and laziness	Less confidence level and mental stability
PS-5	Teenager	Became an athlete	Do not get dietary choices	Of type-1 diabetes	Tired and lack of energy

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes. It is a useful tool to helps 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.



Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they THINK AND FEEL?

what really counts

major preoccupations

worries & aspirations

- How do they keep fit and healthy?
- Are these fitness activities they doing are really helpfull?
- They doing right?
- Is there is any way to improve themself?
- How can they reach their goal in certain period of time?
- Not motivated to do workout alone.

What do they HEAR?

what friends say

what boss say

what influencers say

- Stick with consistency is very tough job.
- They hear all the time all the advice of having healthy life and a balance.
- I know having a nutritional diet is very important but difficult to stay on that
- I need to be careful to not hurting my self
- You will learn about the benefits of maintain nutrition and doing workouts.

What do they SEE?

environment

friends

what the market offers

- They see all the time people who suffer some illness because of their sedentary lifestyle.
- THEY WILL SEE THEIR PROGRESS & THEIR GAINS IN SHORT SPAN.
- They will see many blogs like health,fitness,weight loss,recipes etc...
- It helps them to watch & maintain your regular calories & water intake.
- See the influencers and gurus of this topic and become rely on them for keeping motivated.

What do they SAY AND DO?

attitude in public

appearance

behavior towards others

- They are always setting goals for themselves.
- They keep a record of their results.
- They talk about workout and nutritions.
- How to be healthier or which technique is better to avoid injuries in workouts.
- They use "fitness app" or gadgets to track their activity regular basis.
- In spare time they do exercise and prepare diet meal.

PAIN

fears

frustrations

obstacles

- Some of them can't afford products for their nutritional diet.
- They are not sure if they are doing exercises correctly.
- They are frustrated when they are injured.
- When they can't finish a training for one reason or another
- People feels lazy and lack of motivation so they fail to meet their fitness goal.

GAIN

“wants” / needs

measures of success

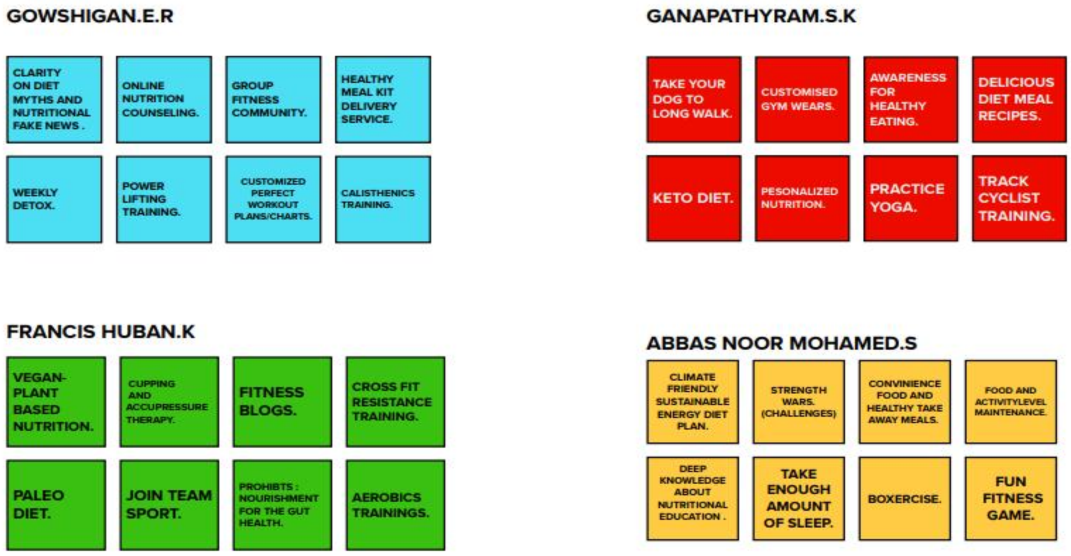
obstacles

- They try to overcome smoking ,consuming alcohol and other addictives.
- They gain success with apps that measure their activity
- Want an app which can provide proper guidance on training and diet plans.
- Community to interact and get questions answered
- It help you to get proper knowledge about fitness .
- It breaks the myths about supplements, training session etc...

3.2 IDEATION & BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich number of creative solutions. 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.

BRAINSTORM



GOWSHIGAN.E.R

CLARITY ON DIET MYTHS AND NUTRITIONAL FAKE NEWS .

ONLINE NUTRITION COUNSELING.

GROUP FITNESS COMMUNITY.

HEALTHY MEAL KIT DELIVERY SERVICE.

WEEKLY DETOX.

POWER LIFTING TRAINING.

CUSTOMIZED PERFECT WORKOUT PLANS/CHARTS.

CALISTHENICS TRAINING.

GANAPATHYRAM.S.K

TAKE YOUR DOG TO LONG WALK.

CUSTOMISED GYM WEARS.

AWARENESS FOR HEALTHY EATING.

DELICIOUS DIET MEAL RECIPES.

KETO DIET.

PESONALIZED NUTRITION.

PRACTICE YOGA.

TRACK CYCLIST TRAINING.

FRANCIS HUBAN.K

VEGAN-PLANT BASED NUTRITION.

CUPPING AND ACCUPRESSURE THERAPY.

FITNESS BLOGS.

CROSS FIT RESISTANCE TRAINING.

PALEO DIET.

JOIN TEAM SPORT.

PROHIBTS : NOURISHMENT FOR THE GUT HEALTH.

AEROBICS TRAININGS.

ABBAS NOOR MOHAMED.S

CLIMATE FRIENDLY SUSTAINABLE ENERGY DIET PLAN.

STRENGTH WARS. (CHALLENGES)

CONVINIENCE FOOD AND HEALTHY TAKE AWAY MEALS.

FOOD AND ACTIVITYLEVEL MAINTENANCE.

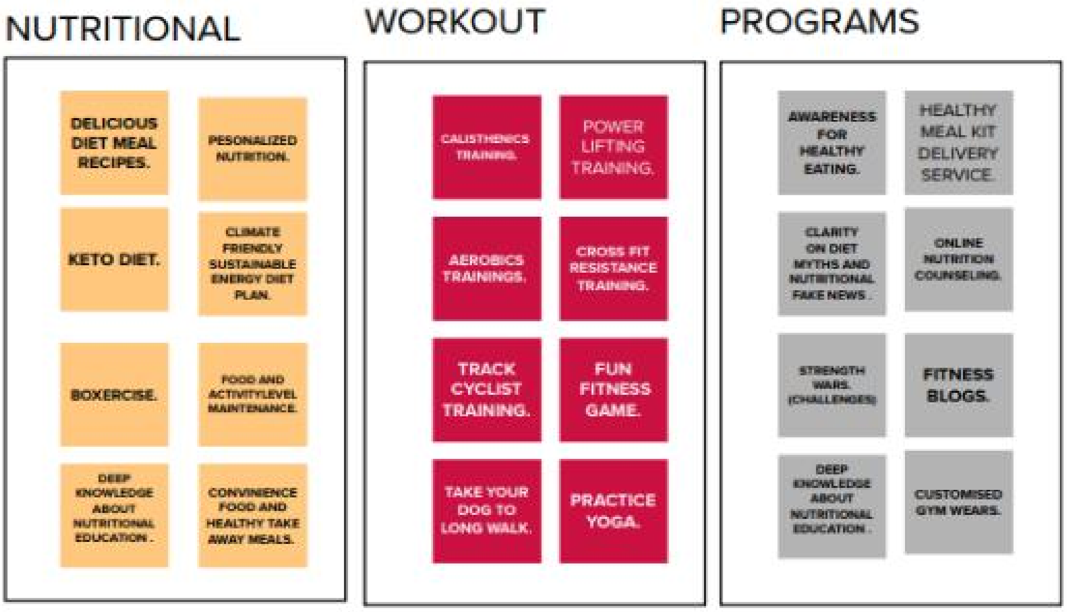
DEEP KNOWLEDGE ABOUT NUTRITIONAL EDUCATION .

TAKE ENOUGH AMOUNT OF SLEEP.

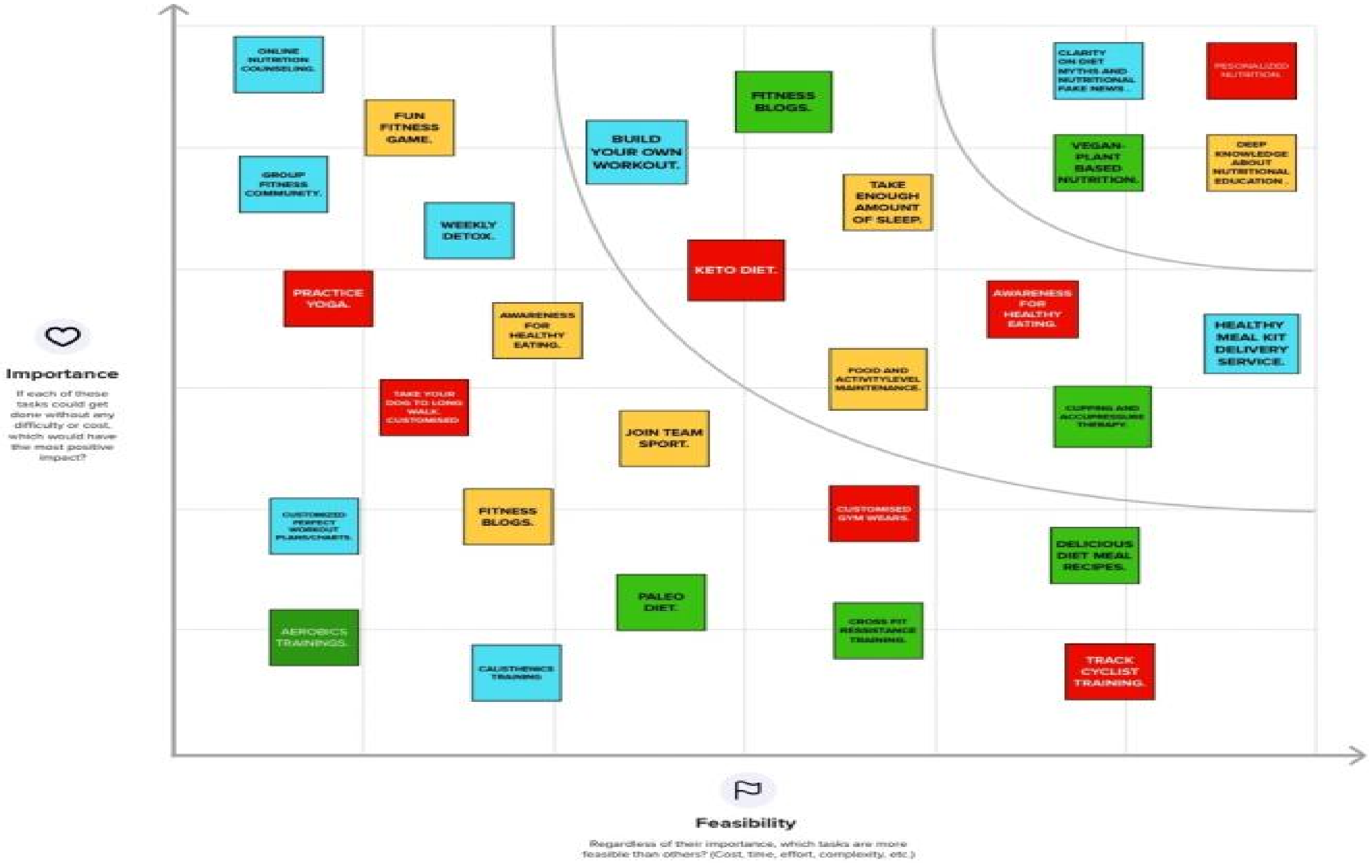
BOXERCISE.

FUN FITNESS GAME.

GROUP IDEAS


NUTRITIONAL
DELICIOUS DIET MEAL RECIPES.
PESONALIZED NUTRITION.
KETO DIET.
CLIMATE FRIENDLY SUSTAINABLE ENERGY DIET PLAN.
BOXERCISE.
FOOD AND ACTIVITYLEVEL MAINTENANCE.
DEEP KNOWLEDGE ABOUT NUTRITIONAL EDUCATION .
CONVINIENCE FOOD AND HEALTHY TAKE AWAY MEALS.
WORKOUT
CALISTHENICS TRAINING.
POWER LIFTING TRAINING.
AEROBICS TRAININGS.
CROSS FIT RESISTANCE TRAINING.
TRACK CYCLIST TRAINING.
FUN FITNESS GAME.
TAKE YOUR DOG TO LONG WALK.
PRACTICE YOGA.
PROGRAMS
AWARENESS FOR HEALTHY EATING.
HEALTHY MEAL KIT DELIVERY SERVICE.
CLARITY ON DIET MYTHS AND NUTRITIONAL FAKE NEWS .
ONLINE NUTRITION COUNSELING.
STRENGTH WARS. (CHALLENGES)
FITNESS BLOGS.
DEEP KNOWLEDGE ABOUT NUTRITIONAL EDUCATION .
CUSTOMISED GYM WEARS.

IDEA PRIORITIZATION



3.3 PROPOSED SOLUTION

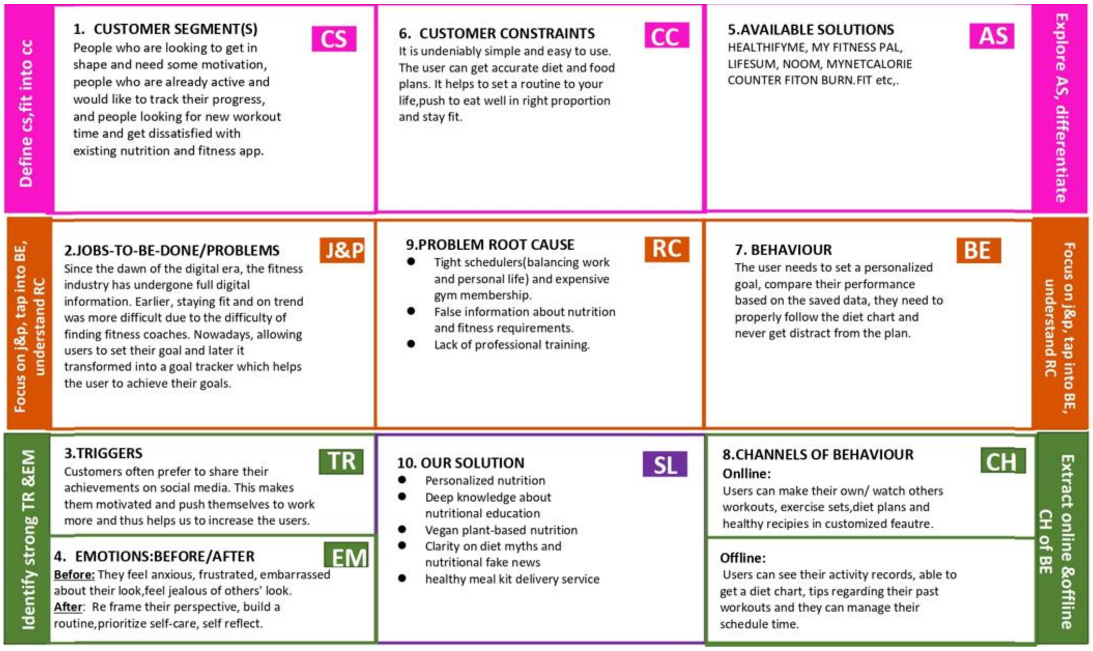
S.NO.	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	Eating foods that are Nutrient dense rather than energy dense choose food that are low added Sugar Saturated fats, Sodium. Pick whole grain& lean Sources of protein and dairy Products.Practice all four types of exercises like Endurance, strength, balance & flexibility.
2.	Idea / Solution description	Nutritional & Fitness ideas educating activities like Presentations, Nutritional ratiodemonstrations, panel discussion, Self evolution, Set goals, myth buster Session, Q&A, workout videos, tasty nutritional diet recipes & So-on.
3.	Novelty / Uniqueness	Research your potential customers. The key to the success of fitness app is optimizing users' engagement and motivation. It's also great to keep looking at newer technology trends and change in users' habits .
4.	Social Impact / Customer Satisfaction	Nutritional dietary eating well can help you feel better physically & mentally. It can make you more likely to seek and enjoy Social activities. Increased confidence, peer acceptance, leadership skills & empathy this kind ofQualities are manifests automatically by following Nutrition diet & Fitness workouts
5.	Business Model (Revenue Model)	Allowing third-party vendors to sell their products through your app via advertisements is way to generate money. If the products sold through advertisements, then it is even better.Limited Service And Premium Service Offering .
6.	Scalability of the Solution	The solution should have the best features such as personal account, setting goals, actual activity tracking, and social elements & offer a custom workout plans creation feature. It's an excellent idea to collaborate with as many wearable producers as possible (and cost-efficient)

3.4 PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer’s problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why

Purpose:

- Solve complex problems in a way that fits the state of your customers.
- Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- Sharpen your communication and marketing strategy with the right triggers and messaging.
- Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.
- Understand the existing situation in order to improve it for your target group



Define cs,fit into cc

1. CUSTOMER SEGMENT(S) CS

People who are looking to get in shape and need some motivation, people who are already active and would like to track their progress, and people looking for new workout time and get dissatisfied with existing nutrition and fitness app.

6. CUSTOMER CONSTRAINTS CC

It is undeniably simple and easy to use. The user can get accurate diet and food plans. It helps to set a routine to your life,push to eat well in right proportion and stay fit.

5.AVAILABLE SOLUTIONS AS

HEALTHIFYME, MY FITNESS PAL, LIFESUM, NOOM, MYNETCALORIE COUNTER FITON BURN.FIT etc,.

Explore AS, differentiate

Focus on j&p, tap into BE, understand RC

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

Since the dawn of the digital era, the fitness industry has undergone full digital information. Earlier, staying fit and on trend was more difficult due to the difficulty of finding fitness coaches. Nowadays, allowing users to set their goal and later it transformed into a goal tracker which helps the user to achieve their goals.

9.PROBLEM ROOT CAUSE RC

- Tight schedulers(balancing work and personal life) and expensive gym membership.
- False information about nutrition and fitness requirements.
- Lack of professional training.

7. BEHAVIOUR BE

The user needs to set a personalized goal, compare their performance based on the saved data, they need to properly follow the diet chart and never get distract from the plan.

Focus on j&p, tap into BE, understand RC

Identify strong TR &EM

3.TRIGGERS TR

Customers often prefer to share their achievements on social media. This makes them motivated and push themselves to work more and thus helps us to increase the users.

4. EMOTIONS:BEFORE/AFTER EM

Before: They feel anxious, frustrated, embarrassed about their look,feel jealous of others' look.

After: Re frame their perspective, build a routine,prioritize self-care, self reflect.

10. OUR SOLUTION SL

- Personalized nutrition
- Deep knowledge about nutritional education
- Vegan plant-based nutrition
- Clarity on diet myths and nutritional fake news
- healthy meal kit delivery service

8.CHANNELS OF BEHAVIOUR CH

Onlline:

Users can make their own/ watch others workouts, exercise sets,diet plans and healthy recipies in customized feautre.

Offline:

Users can see their activity records, able to get a diet chart, tips regarding their past workouts and they can manage their schedule time.

Extract online &offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR NO.	FUNCTIONAL REQUIREMENT (EPIC)	SUB REQUIREMENT (STORY / SUB-TASK)
FR-1	**User Registration**	Registration through Web application Form Registration through Gmail
FR-2	**User Confirmation**	Confirmation via EmailConfirmation via OTP

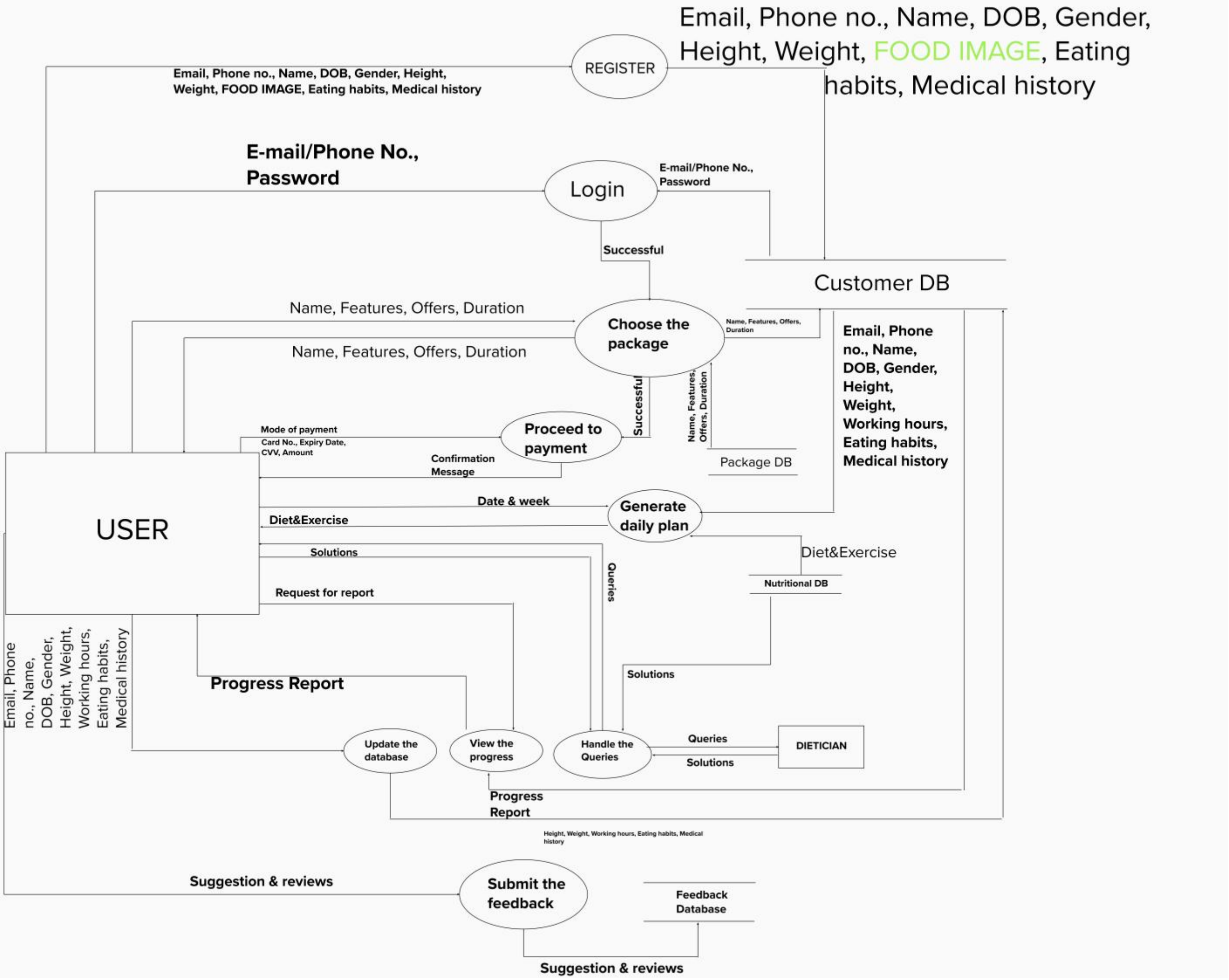
4.2 NON-FUNCTIONAL REQUIREMENTS

FR NO.	NON-FUNCTIONAL REQUIREMENT	DESCRIPTION
NFR-1	**Usability**	Efficient for frequent users, it can be easy to understand what the application does and how it feels.
NFR-2	**Security**	Assures all the data inside the system will be protected against malware attacks or unauthorized access. This application must remain resilient in the face of attacks. The behaviour of the application must be correct and predictable.
NFR-3	**Reliability**	This application must perform without failure in 95 percent of use cases during a month.
NFR-4	**Performance**	This application supporting 1,000 users per hour must provide 6 seconds or less response time in a desktop browser, including the rendering of text and images, over an LTE connection.
NFR-5	**Availability**	The web dashboard must be available to user's 99.9 percent of the time every month during business hours EST. Users can access any time.
NFR-6	**Scalability**	The application must be scalable enough to support 10,000 visits at the same time while maintaining optimal performance.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

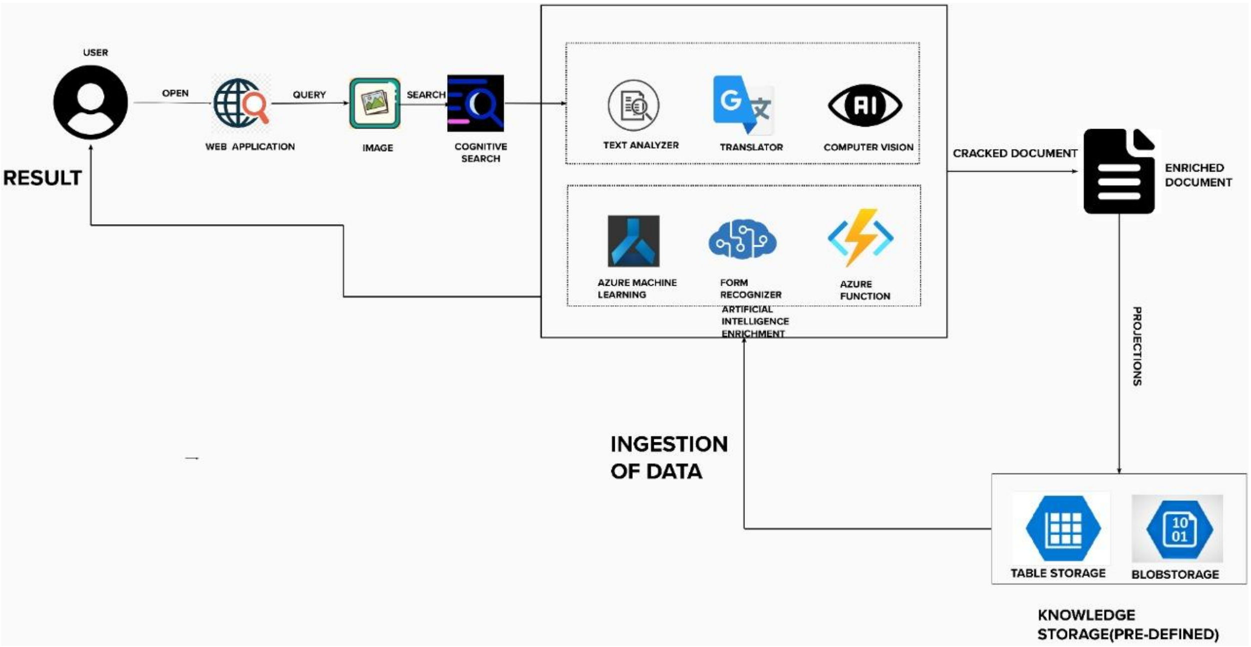

Email, Phone no., Name, DOB, Gender, Height, Weight, FOOD IMAGE, Eating habits, Medical history
REGISTER
Email, Phone no., Name, DOB, Gender, Height, Weight, FOOD IMAGE, Eating habits, Medical history
E-mail/Phone No., Password
Login
E-mail/Phone No., Password
Successful
Customer DB
Name, Features, Offers, Duration
Choose the package
Name, Features, Offers, Duration
Name, Features, Offers, Duration
Email, Phone no., Name, DOB, Gender, Height, Weight, Working hours, Eating habits, Medical history
Successful
Name, Features, Offers, Duration
Mode of payment
Card No., Expiry Date, CVV, Amount
Proceed to payment
Confirmation Message
Package DB
USER
Date & week
Generate daily plan
Diet&Exercise
Solutions
Diet&Exercise
Queries
Nutritional DB
Request for report
Email, Phone no., Name, DOB, Gender, Height, Weight, Working hours, Eating habits, Medical history
Progress Report
Solutions
Update the database
View the progress
Handle the Queries
Queries
Solutions
DIETICIAN
Progress Report
Height, Weight, Working hours, Eating habits, Medical history
Suggestion & reviews
Submit the feedback
Feedback Database
Suggestion & reviews

5.2 SOLUTION & TECHNICAL ARCHITECTURE

SOLUTION ARCHITECTURE

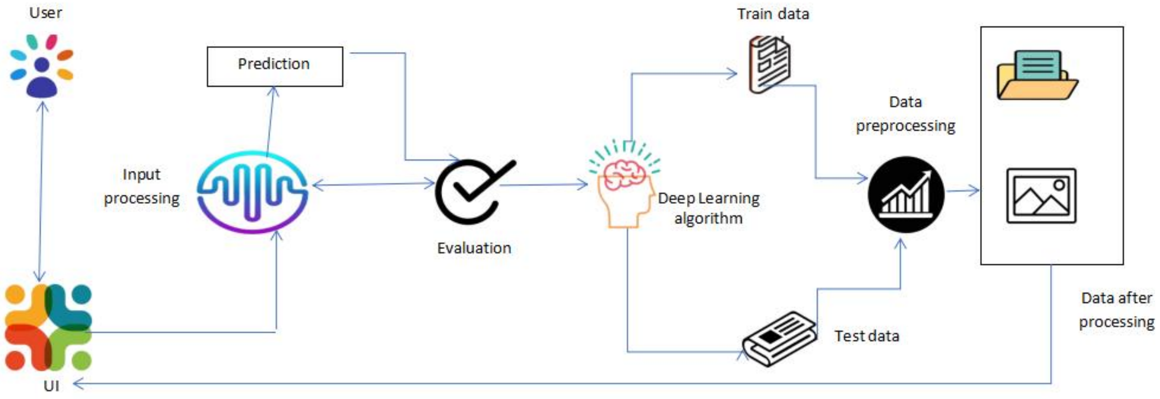
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, 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.



TECHNICAL ARCHITECTURE:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



Components & technologies:

S.NO	COMPONENT	DESCRIPTION	TECHNOLOGY
1.	Application	The user interacts with application for the prediction of nutrition evaluation using images or data.	HTML, CSS, JavaScript
2.	Image processing/data processing	Users upload or process the data in our application	Python
3.	Dataase	User data, configuration, dataset will be stored.	SQL
4.	Cloud database	Database services on the cloud	IBM Watson cloud
5.	File storage	User requirements will be processed through the file	Cloud-> drive
6.	Machine learning model	Image processing, data visualization and evaluation can be done.	ANN, CNN, RNN
7.	Specifying Alert	Notifying the users on their daily plan	SendGrid
8.	Infrastructure	Cloud based web applications.	Cloud application

Application characteristics:

S.NO	CHARACTERISTICS	DESCRIPTION	TECHNOLOGY
1.	Open-Source Frameworks	It is made freely available code for possible notification and redistribution	SendGrid , js , jupiter (python)
2.	Security Implementations	Request for authentication using encryption	Encryption, SSL certs
3.	Scalable Architecture	This application must remain resilient in the face of attacks. The behaviour of the application must be correct and predictable	HTML, CSS, JS, PYTHON, FLASK, IBM CLOUD.
4.	Availability	The web dashboard must be available to user's 99.9 percent of the time every month during business hours	IBM Cloud hosting
5.	Performance	The application must be scalable enough to support 10,000 visits at the same time while maintaining optimal performance	IBM Load blance

5.3 USER STORIES

User type	Functional requirement (epic)	user story number	user story / task	acceptance criteria	priority	release
Customer (Mobile user)	Access daily plan	USN-1	As a user, I want to access the daily plan so that I can get regular workout plans	I can workout by accessing the daily plan	Medium	Sprint-2
	Registration credentials	USN-2	As a user,I can register for my application by entering my account details.	My account can only be signed in with my authority	High	Sprint-1
	Email Registration	USN-3	As a user, I can register for the application through EMAIL	I can register & access the dashboard with EMAIL	Low	Sprint-2
	Daily activities	USN-4	As a user, I can increase my core strength with the activities mentioned in the plan	To maintain workouts in the plans which is used to develop core strength	High	Sprint-1
Customer (Web user)	Membership plan	USN-5	As a user, I can perform workout and additional diet plans by purchasing the membership plans.	Customized plans provide me with specified and consistent growth.	Medium	Sprint-1
Customer Care Executive	Diet plan	USN-6	As an executive, I must give the right nutrition with proper diet facts.	A complete guide for nutrition is provided to the user.	High	Sprint-1
Administrator	Stores the user data	USN-7	As an admin, I want to store the previous records of users.	Users can get a proper diet plans by comparing them with previous result	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 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 can register for the application by entering their name, email, password, and confirmed password.	3	High	Gowshigan.E.R
Sprint-1		USN-2	As a user will receive a confirmation email once they have registered in the application	1	low	Abbas NoorMohamed.S
Sprint-1	Login	USN-3	As a user can log into the application by entering an email & password	2	Medium	Gowshigan.E.R
Sprint-2	Dashboard	USN-4	As a user can use a dashboard for getting their personalized nutrition plan.and some nutritional education.	2	High	GanapathyRam.S.K
Sprint-2		USN-5	As a user can use a dashboard for getting their workout plans based on their levels as mentioned.	3	Medium	Francis Huban.K

Sprint-2		USN-6	As a user, can use the dashboard to clarify the diet myths and get a healthy meal plans.	3	Low	Abbas NoorMohamed.S
Sprint-2	Health insights	USN-7	As a user can monitor their daily plan and check their activity status	2	High	GanapathyRam.S.K
Sprint-2		USN-8	As a user can compare their activity based on previous day and they are able to set a reminder for daily activities.	3	Medium	Francis Huban.K
Sprint-3	Chat bot	USN-9	As a user can use a chat bot to get nutritional counseling.	2	Low	GanapathyRam.S.K
Sprint-3	Membership plan	USN-10	As a user can perform workout and additional diet plans by purchasing the membership plans.	3	Medium	Gowshigan.E.R
Sprint-3	Model building using flask	USN-11	Flask provides tools, libraries and technologies that allow to build a web application.	5	High	Francis Huban.K
Sprint-4	Generation of diet plan and workout details	USN-12	As a user can get the correct diet plan chart and workout details.	3	High	GanapathyRam.S.K
Sprint-4	Deployment in cloud	USN-12	To ensure that the system successfully reached to the users.	8	Medium	Abbas NoorMohamed.S

6.2 SPRINT DELIVERY SCHEDULE

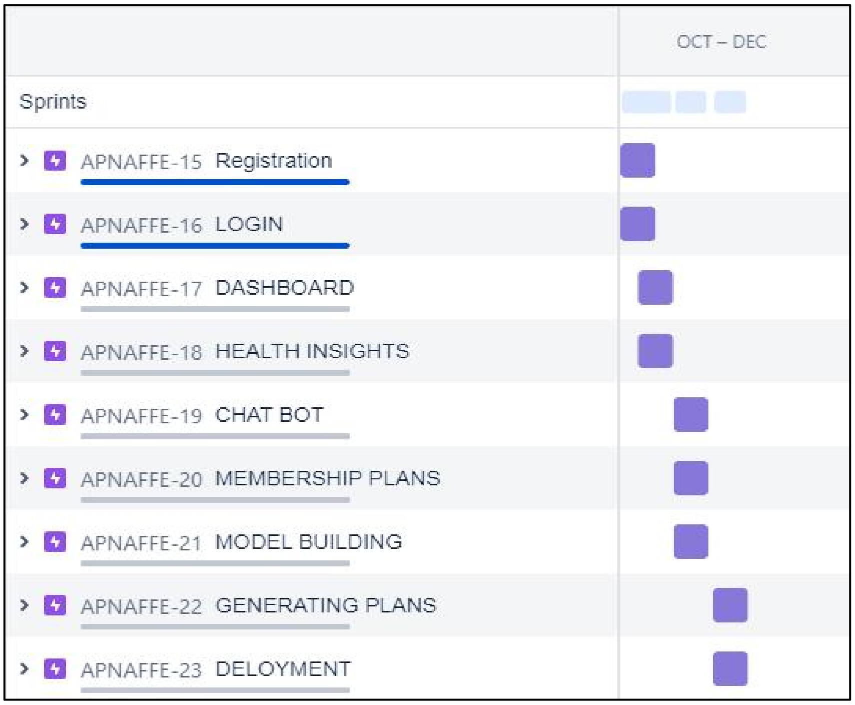
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	6	6 Days	24 Oct 2022	29 Oct 2022	6	29 Oct 2022
Sprint-2	13	6 Days	31 Oct 2022	05 Nov 2022	13	05 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	11	6 Days	14 Nov 2022	19 Nov 2022	11	19 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)

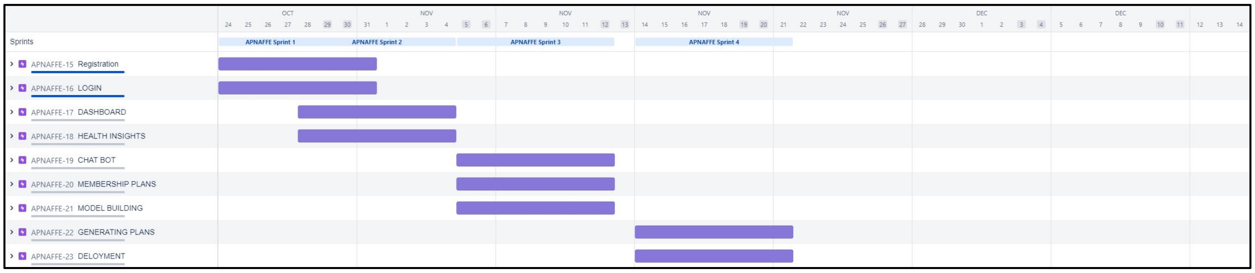
SPRINT	TOTAL STORY POINTS	DURATION	AVG VELOCITY
SPRINT-1	6	6 Days	1
SPRINT-2	13	6 Days	2.16
SPRINT-3	10	6 Days	1.66
SPRINT-4	11	6 Days	1.83
TOTAL	42	24	1.75

6.3 REPORTS FROM JIRA



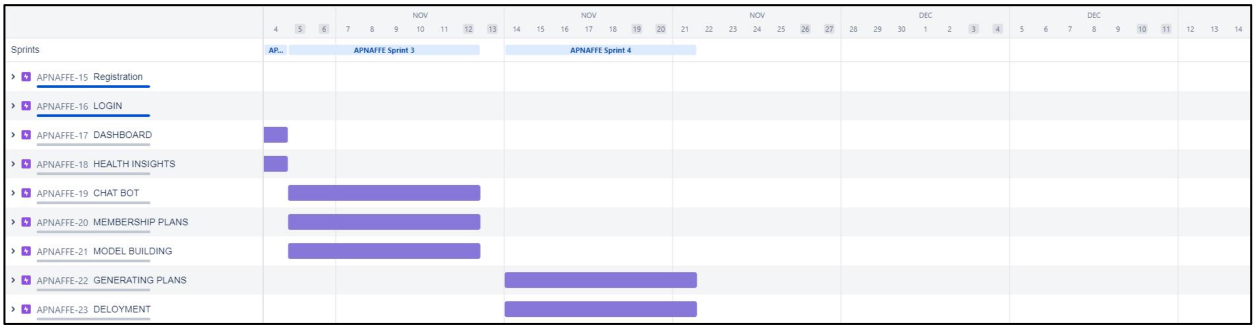
Sprints

- APNAFFE-15 Registration
- APNAFFE-16 LOGIN
- APNAFFE-17 DASHBOARD
- APNAFFE-18 HEALTH INSIGHTS
- APNAFFE-19 CHAT BOT
- APNAFFE-20 MEMBERSHIP PLANS
- APNAFFE-21 MODEL BUILDING
- APNAFFE-22 GENERATING PLANS
- APNAFFE-23 DELOYMENT



Sprints: APNAFFE Sprint 1, APNAFFE Sprint 2, APNAFFE Sprint 3, APNAFFE Sprint 4

- APNAFFE-15 Registration
- APNAFFE-16 LOGIN
- APNAFFE-17 DASHBOARD
- APNAFFE-18 HEALTH INSIGHTS
- APNAFFE-19 CHAT BOT
- APNAFFE-20 MEMBERSHIP PLANS
- APNAFFE-21 MODEL BUILDING
- APNAFFE-22 GENERATING PLANS
- APNAFFE-23 DELOYMENT



Sprints: AP..., APNAFFE Sprint 3, APNAFFE Sprint 4

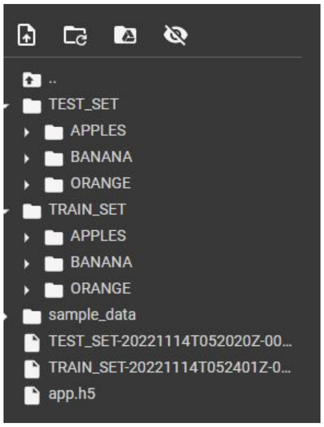
- APNAFFE-15 Registration
- APNAFFE-16 LOGIN
- APNAFFE-17 DASHBOARD
- APNAFFE-18 HEALTH INSIGHTS
- APNAFFE-19 CHAT BOT
- APNAFFE-20 MEMBERSHIP PLANS
- APNAFFE-21 MODEL BUILDING
- APNAFFE-22 GENERATING PLANS
- APNAFFE-23 DELOYMENT

7. CODING & SOLUTIONING

7.1 IMAGE PROCESSING

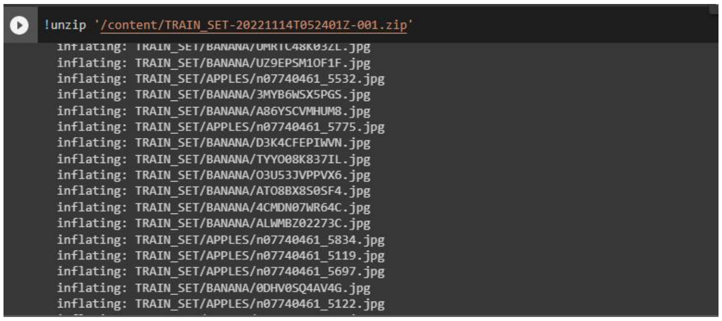
COLECT THE DATASET

In this project, we have collected images of 3 types of food items: apples, 'banana' and 'orange'. They are saved in the respective sub directories with their respective names. For more accurate results, we can collect images at high resolution and feed the model with more images



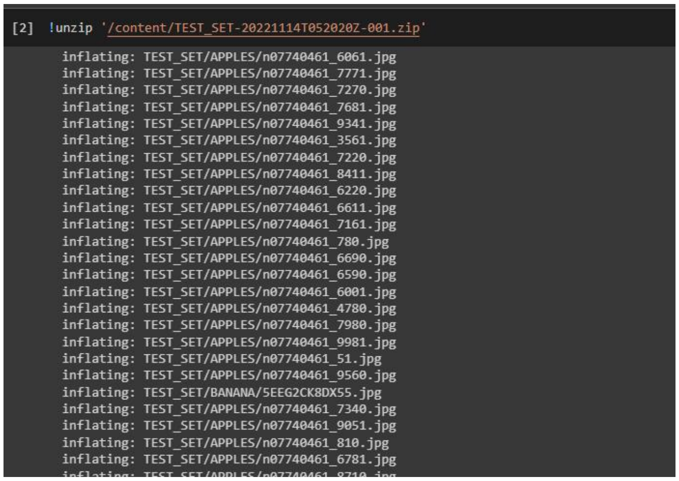
```
..
TEST_SET
    APPLES
    BANANA
    ORANGE
TRAIN_SET
    APPLES
    BANANA
    ORANGE
sample_data
TEST_SET-20221114T052020Z-00...
TRAIN_SET-20221114T052401Z-0...
app.h5
```

>> !unzip '/content/TRAIN_SET-20221114T052401Z-001.zip



```
!unzip '/content/TRAIN_SET-20221114T052401Z-001.zip'
   inflating: TRAIN_SET/BANANA/UMRTC48K03ZL.jpg
   inflating: TRAIN_SET/BANANA/UZ9EPSM1OF1F.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5532.jpg
   inflating: TRAIN_SET/BANANA/3MYB6WSX5PGS.jpg
   inflating: TRAIN_SET/BANANA/A86YSCVMHUM8.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5775.jpg
   inflating: TRAIN_SET/BANANA/D3K4CFEPIWVN.jpg
   inflating: TRAIN_SET/BANANA/TYYO08K837IL.jpg
   inflating: TRAIN_SET/BANANA/O3U53JVPPVX6.jpg
   inflating: TRAIN_SET/BANANA/ATO8BX8S0SF4.jpg
   inflating: TRAIN_SET/BANANA/4CMDN07WR64C.jpg
   inflating: TRAIN_SET/BANANA/ALWMBZ02273C.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5834.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5119.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5697.jpg
   inflating: TRAIN_SET/BANANA/0DHV0SQ4AV4G.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5122.jpg
```

>>!unzip '/content/TEST_SET-20221114T052020Z-001.zip'



```
[2]  !unzip '/content/TEST_SET-20221114T052020Z-001.zip'
        inflating: TEST_SET/APPLES/n07740461_6061.jpg
        inflating: TEST_SET/APPLES/n07740461_7771.jpg
        inflating: TEST_SET/APPLES/n07740461_7270.jpg
        inflating: TEST_SET/APPLES/n07740461_7681.jpg
        inflating: TEST_SET/APPLES/n07740461_9341.jpg
        inflating: TEST_SET/APPLES/n07740461_3561.jpg
        inflating: TEST_SET/APPLES/n07740461_7220.jpg
        inflating: TEST_SET/APPLES/n07740461_8411.jpg
        inflating: TEST_SET/APPLES/n07740461_6220.jpg
        inflating: TEST_SET/APPLES/n07740461_6611.jpg
        inflating: TEST_SET/APPLES/n07740461_7161.jpg
        inflating: TEST_SET/APPLES/n07740461_780.jpg
        inflating: TEST_SET/APPLES/n07740461_6690.jpg
        inflating: TEST_SET/APPLES/n07740461_6590.jpg
        inflating: TEST_SET/APPLES/n07740461_6001.jpg
        inflating: TEST_SET/APPLES/n07740461_4780.jpg
        inflating: TEST_SET/APPLES/n07740461_7980.jpg
        inflating: TEST_SET/APPLES/n07740461_9981.jpg
        inflating: TEST_SET/APPLES/n07740461_51.jpg
        inflating: TEST_SET/APPLES/n07740461_9560.jpg
        inflating: TEST_SET/BANANA/5EEG2CK8DX55.jpg
        inflating: TEST_SET/APPLES/n07740461_7340.jpg
        inflating: TEST_SET/APPLES/n07740461_9051.jpg
        inflating: TEST_SET/APPLES/n07740461_810.jpg
        inflating: TEST_SET/APPLES/n07740461_6781.jpg
        inflating: TEST_SET/APPLES/n07740461_8710.jpg
```

IMPORT THE DATA GENERATOR LIBRARY ,CONFIGURE IMAGE DATA GENERATOR CLASS AND APPLY TO THE TRAIN AND TEST DATA SET

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class. Let us import the ImageDataGenerator class from Keras

i. IMPORT THE REQUIRED LIBRARIES

import numpy as np#used for numerical analysis

import tensorflow #open source used for both ML and DL for computation

from tensorflow.keras.models import Sequential #it is a plain stack of layers

from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function

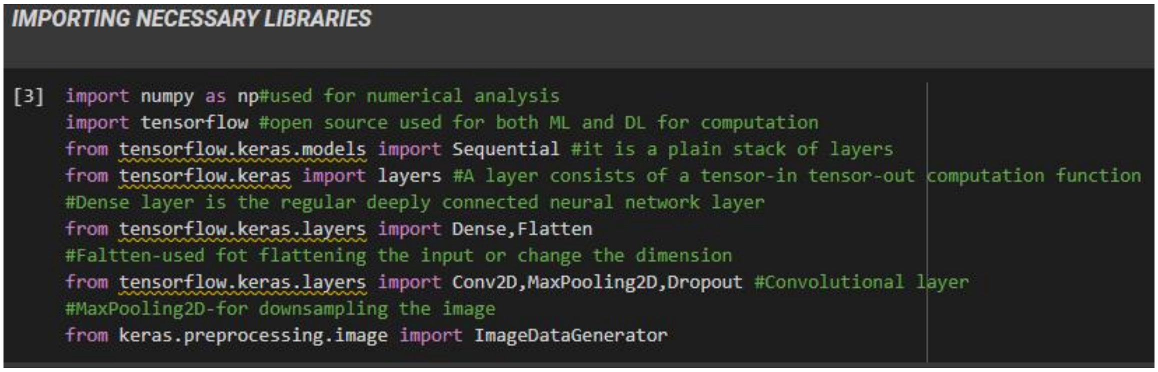
#Dense layer is the regular deeply connected neural network layer from tensorflow.keras.layers import Dense,Flatten

#Faltten-used fot flattening the input or change the dimension

from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional layer

#MaxPooling2D-for downsampling the image

from keras.preprocessing.image import ImageDataGenerator



IMPORTING NECESSARY LIBRARIES

```
[3]  import numpy as np#used for numerical analysis
     import tensorflow #open source used for both ML and DL for computation
     from tensorflow.keras.models import Sequential #it is a plain stack of layers
     from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
     #Dense layer is the regular deeply connected neural network layer
     from tensorflow.keras.layers import Dense,Flatten
     #Faltten-used fot flattening the input or change the dimension
     from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional layer
     #MaxPooling2D-for downsampling the image
     from keras.preprocessing.image import ImageDataGenerator
```

ii. IMAGE AUGUMENTATION

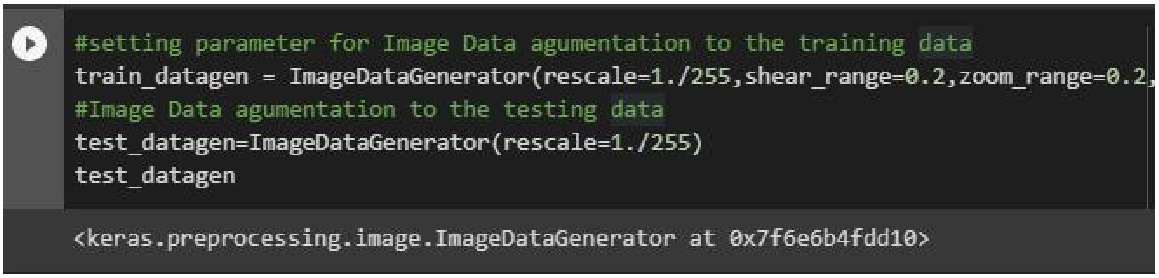
#setting parameter for Image Data agumentation to the training data

train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)

#Image Data agumentation to the testing data

test_datagen=ImageDataGenerator(rescale=1./255)

test_datagen

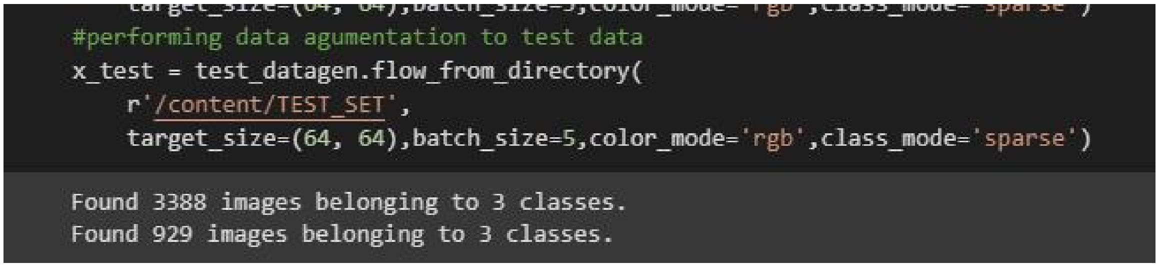


```
#setting parameter for Image Data agumentation to the training data
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
test_datagen
```

```
<keras.preprocessing.image.ImageDataGenerator at 0x7f6e6b4fdd10>
```

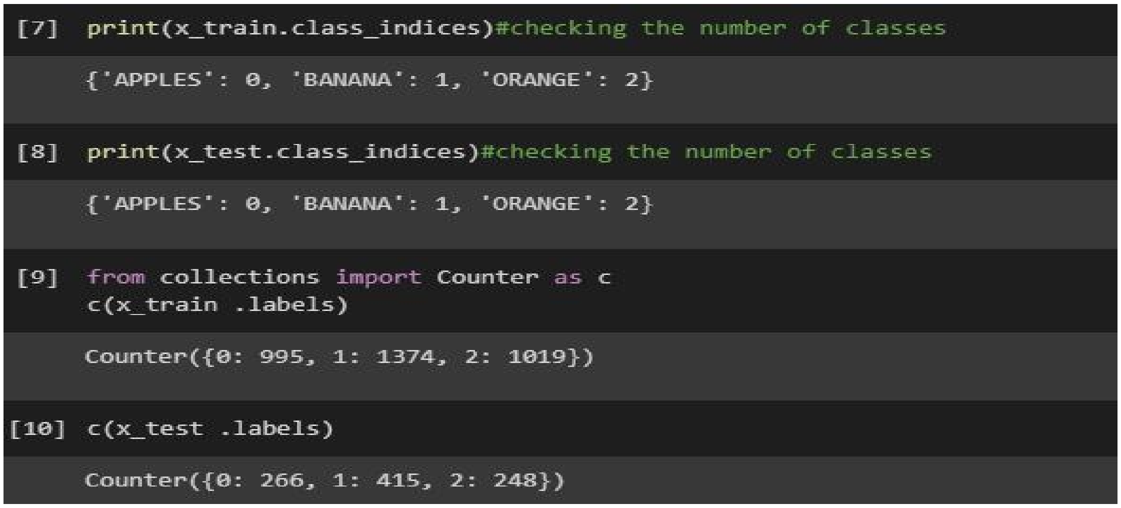
iii. LOADING DATA

```
#performing data agumentation to train data

x_train = train_datagen.flow_from_directory(

        r'/content/TRAIN_SET',

        target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')

#performing data agumentation to test data

x_test = test_datagen.flow_from_directory(

        r'/content/TEST_SET',

   target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```



```
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory(
    r'/content/TEST_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```

```
Found 3388 images belonging to 3 classes.
Found 929 images belonging to 3 classes.
```



```
[7]  print(x_train.class_indices)#checking the number of classes

     {'APPLES': 0, 'BANANA': 1, 'ORANGE': 2}

[8]  print(x_test.class_indices)#checking the number of classes

     {'APPLES': 0, 'BANANA': 1, 'ORANGE': 2}

[9]  from collections import Counter as c
     c(x_train .labels)

     Counter({0: 995, 1: 1374, 2: 1019})

[10] c(x_test .labels)

     Counter({0: 266, 1: 415, 2: 248})
```

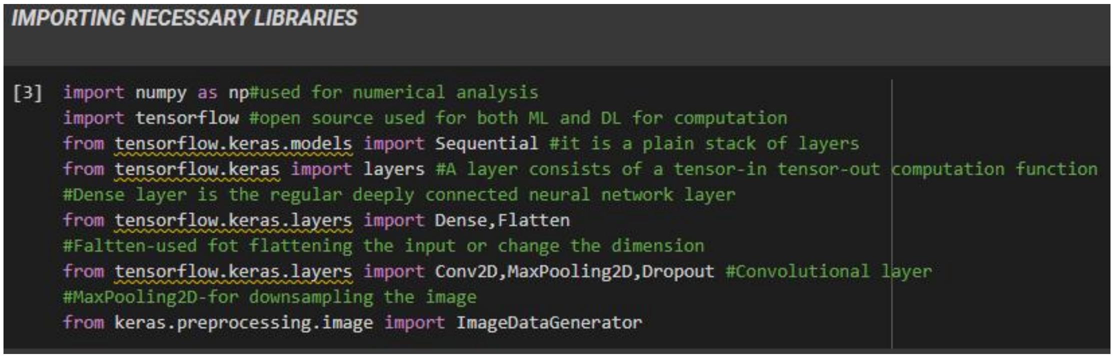
7.2 MODEL BUILDING

Steps to Build a Deep Learning Model

1. Defining model architecture

This is a very crucial step in our deep learning model building process. We have to define how our model will look and that requires

- **Importing the libraries**



IMPORTING NECESSARY LIBRARIES

```
[3] import numpy as np#used for numerical analysis
    import tensorflow #open source used for both ML and DL for computation
    from tensorflow.keras.models import Sequential #it is a plain stack of layers
    from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
    #Dense layer is the regular deeply connected neural network layer
    from tensorflow.keras.layers import Dense,Flatten
    #Faltten-used fot flattening the input or change the dimension
    from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional layer
    #MaxPooling2D-for downsampling the image
    from keras.preprocessing.image import ImageDataGenerator
```

- **Adding CNN (Convolution Neural Network) Layers**

Keras has 2 ways to define a neural network:

- Sequential

- Function API

The Sequential class is used to define a linear initialization of network layers which then, collectively, constitute a model. In our example below, we will use the Sequential constructor to create a model, which will then have layers added to it using the add() method

- **Adding Dense layers**

We will be adding three layers for CNN

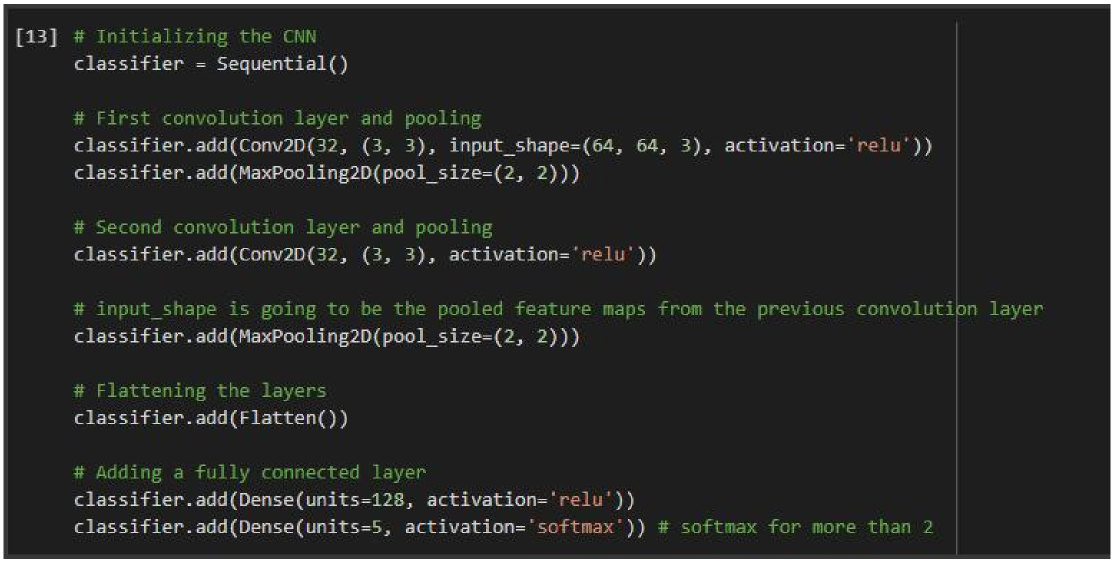
- Convolution layer
- Pooling layer
- Flattening layer

- **Initializing the model**

```
# Initializing the CNN

classifier = Sequential()

# First convolution layer and pooling

classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))

classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Second convolution layer and pooling

classifier.add(Conv2D(32, (3, 3), activation='relu'))

# input_shape is going to be the pooled feature maps from the previous convolution layer

classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Flattening the layers

classifier.add(Flatten())

# Adding a fully connected layer
```

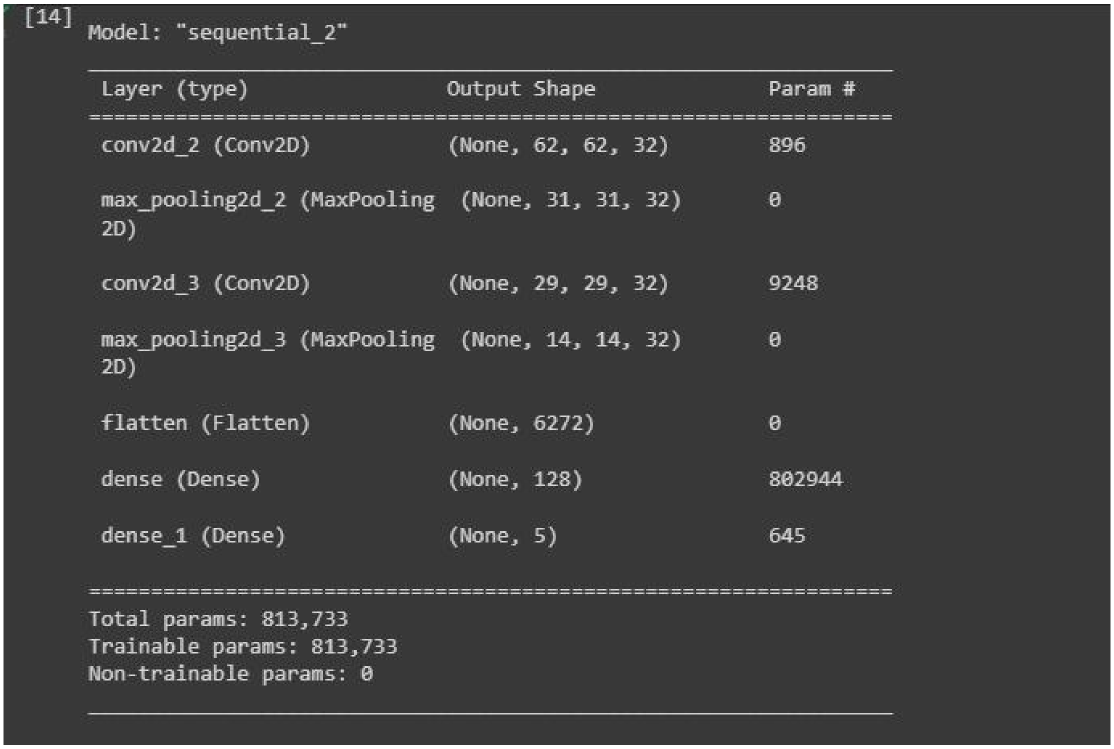
classifier.add(Dense(units=128, activation='relu'))

classifier.add(Dense(units=5, activation='softmax')) # softmax for more than 2



```
[13] # Initializing the CNN
     classifier = Sequential()

     # First convolution layer and pooling
     classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
     classifier.add(MaxPooling2D(pool_size=(2, 2)))

     # Second convolution layer and pooling
     classifier.add(Conv2D(32, (3, 3), activation='relu'))

     # input_shape is going to be the pooled feature maps from the previous convolution layer
     classifier.add(MaxPooling2D(pool_size=(2, 2)))

     # Flattening the layers
     classifier.add(Flatten())

     # Adding a fully connected layer
     classifier.add(Dense(units=128, activation='relu'))
     classifier.add(Dense(units=5, activation='softmax')) # softmax for more than 2
```

classifier.summary()#summary of our model



```
[14]
 Model: "sequential_2"
 _________________________________________________________________
  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)

  flatten (Flatten)           (None, 6272)              0

  dense (Dense)               (None, 128)               802944

  dense_1 (Dense)             (None, 5)                 645

 =================================================================
 Total params: 813,733
 Trainable params: 813,733
 Non-trainable params: 0
 _________________________________________________________________
```

2. Configure the learning process

With both the training data defined and model defined, it's time configure the learning process. This is accomplished with a call to the compile() method of the Sequential model class. Compilation requires 3 arguments: an optimizer, a loss function, and a list of metrics.

In our example, set up as a multi-class classification problem, we will use the Adam optimizer, the categorical cross entropy loss function, and include solely the accuracy metric.

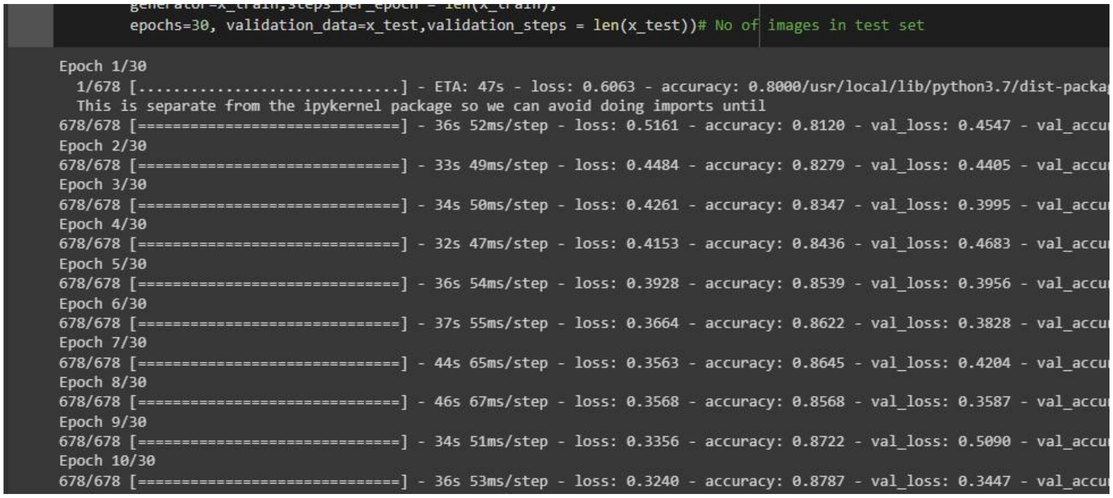
```
    # Compiling the CNN

    # categorical_crossentropy for more than 2

    classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=
['accuracy'])
```

3. Train The Model

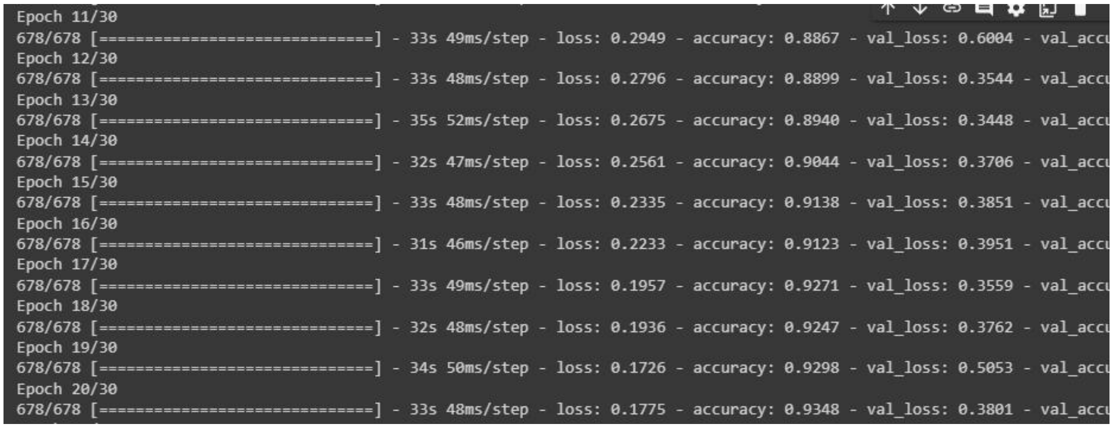
At this point we have training data and a fully configured neural network to train with said data. All that is left is to pass the data to the model for the training process to commence, a process which is completed by iterating on the training data. Training begins by calling the fit() method.

```
>>>classifier.fit_generator(

  generator=x_train,steps_per_epoch = len(x_train),
```

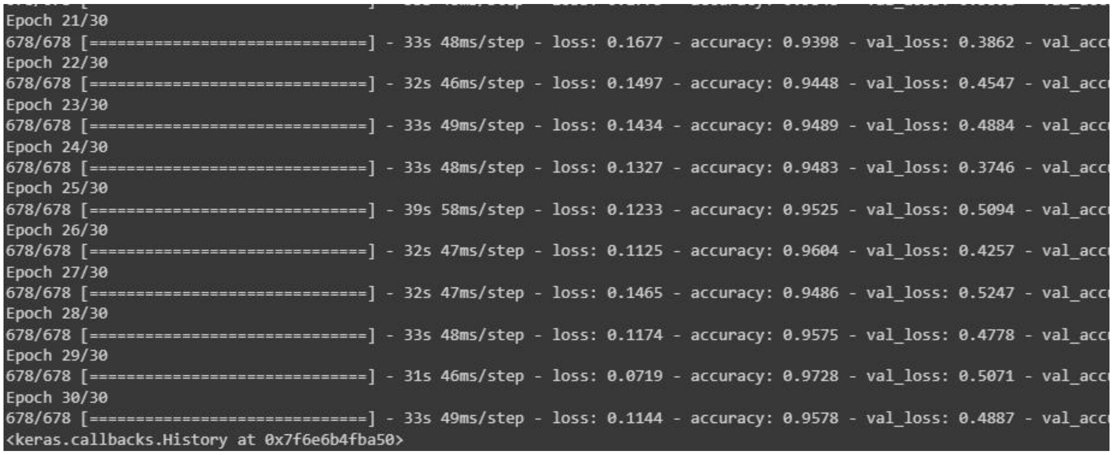
epochs=30,validation_data=x_test,validation_steps=len(x_test))



```
        generator=x_train,steps_per_epoch = len(x_train),
        epochs=30, validation_data=x_test,validation_steps = len(x_test))# No of images in test set

Epoch 1/30
  1/678 [..............................] - ETA: 47s - loss: 0.6063 - accuracy: 0.8000/usr/local/lib/python3.7/dist-packa
  This is separate from the ipykernel package so we can avoid doing imports until
678/678 [==============================] - 36s 52ms/step - loss: 0.5161 - accuracy: 0.8120 - val_loss: 0.4547 - val_accu
Epoch 2/30
678/678 [==============================] - 33s 49ms/step - loss: 0.4484 - accuracy: 0.8279 - val_loss: 0.4405 - val_accu
Epoch 3/30
678/678 [==============================] - 34s 50ms/step - loss: 0.4261 - accuracy: 0.8347 - val_loss: 0.3995 - val_accu
Epoch 4/30
678/678 [==============================] - 32s 47ms/step - loss: 0.4153 - accuracy: 0.8436 - val_loss: 0.4683 - val_accu
Epoch 5/30
678/678 [==============================] - 36s 54ms/step - loss: 0.3928 - accuracy: 0.8539 - val_loss: 0.3956 - val_accu
Epoch 6/30
678/678 [==============================] - 37s 55ms/step - loss: 0.3664 - accuracy: 0.8622 - val_loss: 0.3828 - val_accu
Epoch 7/30
678/678 [==============================] - 44s 65ms/step - loss: 0.3563 - accuracy: 0.8645 - val_loss: 0.4204 - val_accu
Epoch 8/30
678/678 [==============================] - 46s 67ms/step - loss: 0.3568 - accuracy: 0.8568 - val_loss: 0.3587 - val_accu
Epoch 9/30
678/678 [==============================] - 34s 51ms/step - loss: 0.3356 - accuracy: 0.8722 - val_loss: 0.5090 - val_accu
Epoch 10/30
678/678 [==============================] - 36s 53ms/step - loss: 0.3240 - accuracy: 0.8787 - val_loss: 0.3447 - val_accu
```



```
Epoch 11/30
678/678 [==============================] - 33s 49ms/step - loss: 0.2949 - accuracy: 0.8867 - val_loss: 0.6004 - val_accu
Epoch 12/30
678/678 [==============================] - 33s 48ms/step - loss: 0.2796 - accuracy: 0.8899 - val_loss: 0.3544 - val_accu
Epoch 13/30
678/678 [==============================] - 35s 52ms/step - loss: 0.2675 - accuracy: 0.8940 - val_loss: 0.3448 - val_accu
Epoch 14/30
678/678 [==============================] - 32s 47ms/step - loss: 0.2561 - accuracy: 0.9044 - val_loss: 0.3706 - val_accu
Epoch 15/30
678/678 [==============================] - 33s 48ms/step - loss: 0.2335 - accuracy: 0.9138 - val_loss: 0.3851 - val_accu
Epoch 16/30
678/678 [==============================] - 31s 46ms/step - loss: 0.2233 - accuracy: 0.9123 - val_loss: 0.3951 - val_accu
Epoch 17/30
678/678 [==============================] - 33s 49ms/step - loss: 0.1957 - accuracy: 0.9271 - val_loss: 0.3559 - val_accu
Epoch 18/30
678/678 [==============================] - 32s 48ms/step - loss: 0.1936 - accuracy: 0.9247 - val_loss: 0.3762 - val_accu
Epoch 19/30
678/678 [==============================] - 34s 50ms/step - loss: 0.1726 - accuracy: 0.9298 - val_loss: 0.5053 - val_accu
Epoch 20/30
678/678 [==============================] - 33s 48ms/step - loss: 0.1775 - accuracy: 0.9348 - val_loss: 0.3801 - val_accu
```

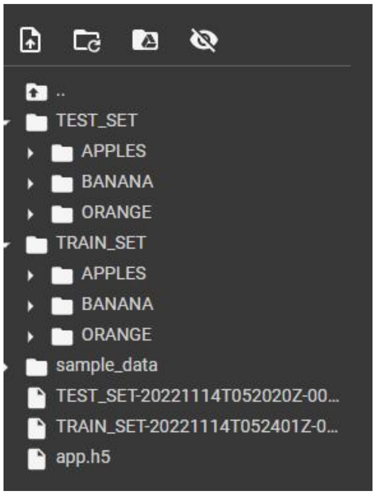


```
Epoch 21/30
678/678 [==============================] - 33s 48ms/step - loss: 0.1677 - accuracy: 0.9398 - val_loss: 0.3862 - val_acc
Epoch 22/30
678/678 [==============================] - 32s 46ms/step - loss: 0.1497 - accuracy: 0.9448 - val_loss: 0.4547 - val_acc
Epoch 23/30
678/678 [==============================] - 33s 49ms/step - loss: 0.1434 - accuracy: 0.9489 - val_loss: 0.4884 - val_acc
Epoch 24/30
678/678 [==============================] - 33s 48ms/step - loss: 0.1327 - accuracy: 0.9483 - val_loss: 0.3746 - val_acc
Epoch 25/30
678/678 [==============================] - 39s 58ms/step - loss: 0.1233 - accuracy: 0.9525 - val_loss: 0.5094 - val_acc
Epoch 26/30
678/678 [==============================] - 32s 47ms/step - loss: 0.1125 - accuracy: 0.9604 - val_loss: 0.4257 - val_acc
Epoch 27/30
678/678 [==============================] - 32s 47ms/step - loss: 0.1465 - accuracy: 0.9486 - val_loss: 0.5247 - val_acc
Epoch 28/30
678/678 [==============================] - 33s 48ms/step - loss: 0.1174 - accuracy: 0.9575 - val_loss: 0.4778 - val_acc
Epoch 29/30
678/678 [==============================] - 31s 46ms/step - loss: 0.0719 - accuracy: 0.9728 - val_loss: 0.5071 - val_acc
Epoch 30/30
678/678 [==============================] - 33s 49ms/step - loss: 0.1144 - accuracy: 0.9578 - val_loss: 0.4887 - val_acc
<keras.callbacks.History at 0x7f6e6b4fba50>
```

4. Save the Model

Your model is to be saved for the future purpose. This saved model ac also be integrated with an android application or web application in order to predict something

```
>># Save the model

classifier.save('app.h5')
```



```
..
TEST_SET
    APPLES
    BANANA
    ORANGE
TRAIN_SET
    APPLES
    BANANA
    ORANGE
sample_data
TEST_SET-20221114T052020Z-00...
TRAIN_SET-20221114T052401Z-0...
app.h5
```

5. Predictions

The last and final step is to make use of the Saved model to make predictions. We use load model class to load the model. We use imread() class from opencv library to read an image and give it to the model to predict the result. Before giving the original image to predict the class, we have to pre-process that image and apply predictions to get accurate result.

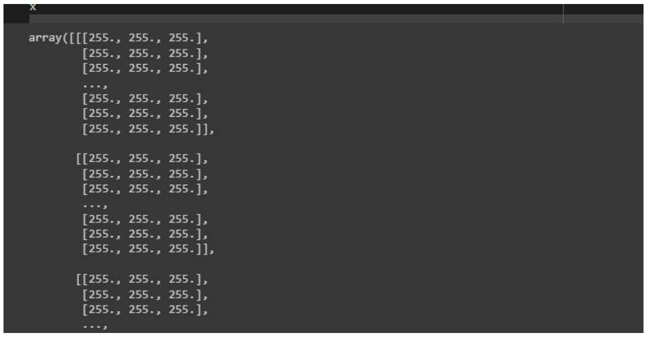
```
>>>

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image
```

model = load_model("app.h5") #loading the model for testing

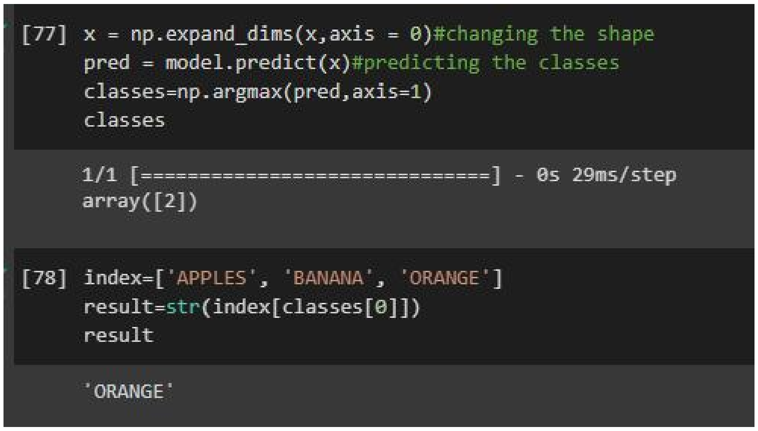
img=tensorflow.keras.utils.load_img("/content/TEST_SET/ORANGE/n07749192_1081.jpg",grayscale=False,target_size= (64,64))#loading of the image

x = image.img_to_array(img)#image to array

x



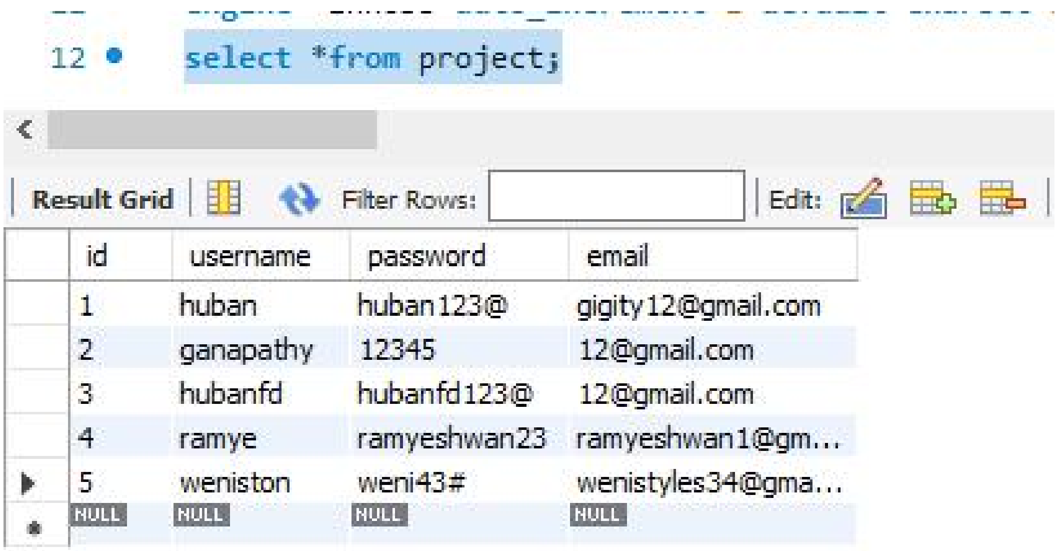
```
x
array([[[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.]],

       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.]],

       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
```



```
[77] x = np.expand_dims(x,axis = 0)#changing the shape
     pred = model.predict(x)#predicting the classes
     classes=np.argmax(pred,axis=1)
     classes

     1/1 [==============================] - 0s 29ms/step
     array([2])

[78] index=['APPLES', 'BANANA', 'ORANGE']
     result=str(index[classes[0]])
     result

     'ORANGE'
```

7.3 DATABASE SCHEMA

```
create schema `hubidb`;
use hubidb;
create table if not exists `project`(
`id` int not null auto_increment,
`username` varchar(50) not null,
`password` varchar(100) not null,
`email` varchar(225) not null,
primary key (`id`)
)
engine= InnoDB auto_increment=2 default charset=utf8;
```

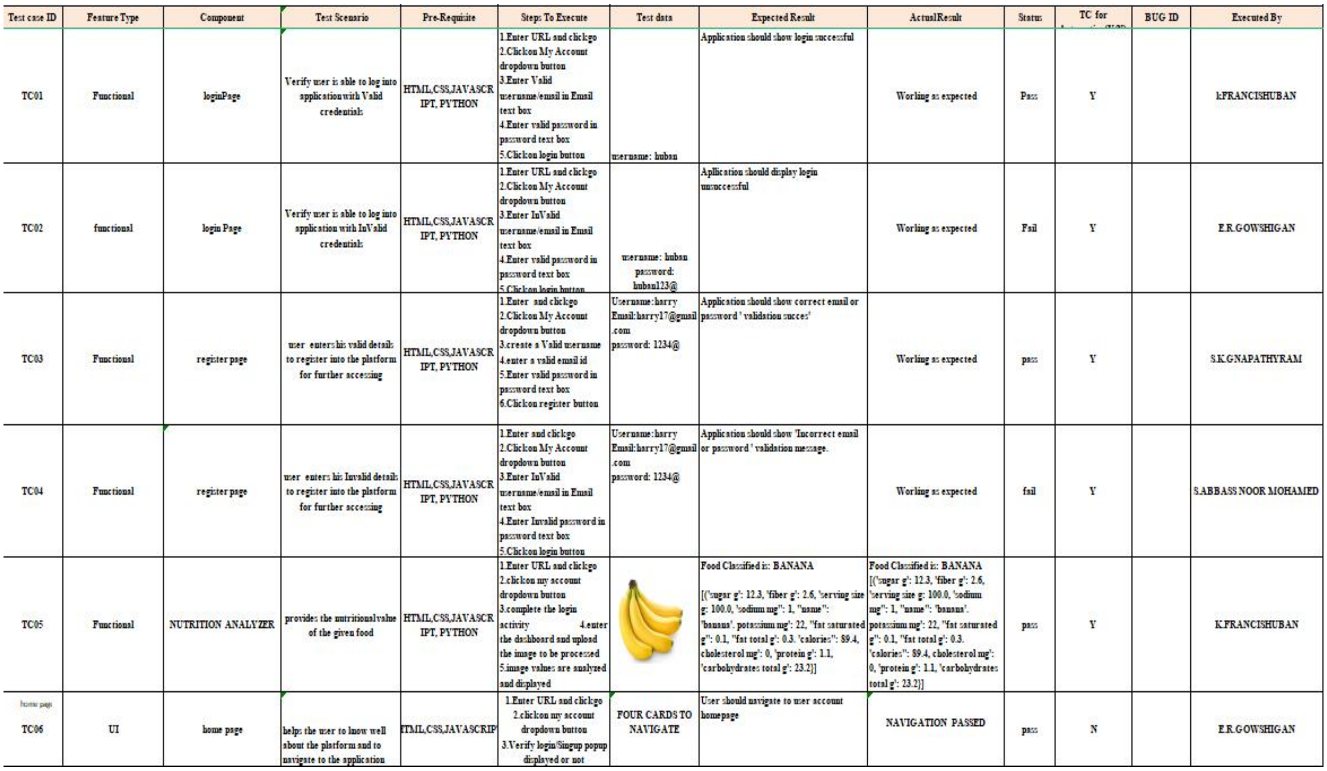


```
select *from project;
```

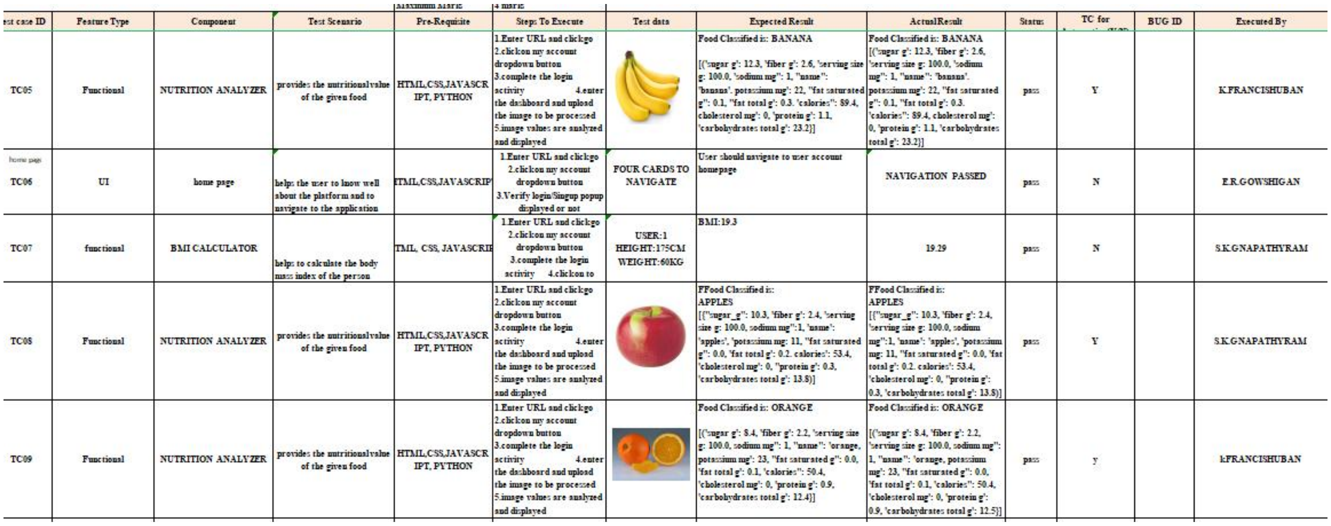
	id	username	password	email
	1	huban	huban123@	gigity12@gmail.com
	2	ganapathy	12345	12@gmail.com
	3	hubanfd	hubanfd123@	12@gmail.com
	4	ramye	ramyeshwan23	ramyeshwan1@gm...
▶	5	weniston	weni43#	wenistyles34@gma...
✱	NULL	NULL	NULL	NULL

8. TESTING

8.1 TEST CASES



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test data	Expected Result	ActualResult	Status	TC for	BUG ID	Executed By
TC01	Functional	loginPage	Verify user is able to log into application with Valid credentials	HTML,CSS,JAVASCRIPT, PYTHON	1.Enter URL and clickgo 2.Clickon My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Clickon login button	username: huban	Application should show login successful	Worling as expected	Pass	Y		k.FRANCISHUBAN
TC02	functional	login Page	Verify user is able to log into application with InValid credentials	HTML,CSS,JAVASCRIPT, PYTHON	1.Enter URL and clickgo 2.Clickon My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Clickon login button	username: huban password: huban123@	Apllication should display login unsuccessful	Worling as expected	Fail	Y		E.R.GOWSHIGAN
TC03	Functional	register page	user enters his valid details to register into the platform for further accessing	HTML,CSS,JAVASCRIPT, PYTHON	1.Enter and clickgo 2.Clickon My Account dropdown button 3.create a Valid username 4.enter a valid email id 5.Enter valid password in password text box 6.Clickon register button	Username:harry Email:harry17@gmail.com password: 1234@	Application should show correct email or password ' validation succes'	Worling as expected	pass	Y		S.K.GNAPATHYRAM
TC04	Functional	register page	user enters his Invalid details to register into the platform for further accessing	HTML,CSS,JAVASCRIPT, PYTHON	1.Enter and clickgo 2.Clickon My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Clickon login button	Username:harry Email:harry17@gmail.com password: 1234@	Application should show 'Incorrect email or password ' validation message.	Worling as expected	fail	Y		S.ABBASS NOOR MOHAMED
TC05	Functional	NUTRITION ANALYZER	provides the nutritional value of the given food	HTML,CSS,JAVASCRIPT, PYTHON	1.Enter URL and clickgo 2.clickon my account dropdown button 3.complete the login activity 4.enter the dashboard and upload the image to be processed 5.image values are analyzed and displayed		Food Classified is: BANANA [('sugar g': 12.3, 'fiber g': 2.6, 'serving size g: 100.0, 'sodium mg": 1, "name": 'banana'. potassium mg': 22, "fat saturated g": 0.1, "fat total g': 0.3. 'calories": 89.4, cholesterol mg': 0, 'protein g': 1.1, 'carbohydrates total g': 23.2}]	Food Classified is: BANANA [('sugar g': 12.3, 'fiber g': 2.6, 'serving size g: 100.0, 'sodium mg": 1, "name": 'banana'. potassium mg': 22, "fat saturated g": 0.1, "fat total g': 0.3. 'calories": 89.4, cholesterol mg': 0, 'protein g': 1.1, 'carbohydrates total g': 23.2}]	pass	Y		K.FRANCISHUBAN
home page TC06	UI	home page	helps the user to know well about the platform and to navigate to the application	HTML,CSS,JAVASCRIPT	1.Enter URL and clickgo 2.clickon my account dropdown button 3.Verify login/Singup popup displayed or not	FOUR CARDS TO NAVIGATE	User should navigate to user account homepage	NAVIGATION PASSED	pass	N		E.R.GOWSHIGAN



				Maximum Marks	4 marks							
est case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test data	Expected Result	ActualResult	Status	TC for	BUG ID	Executed By
TC05	Functional	NUTRITION ANALYZER	provides the nutritional value of the given food	HTML,CSS,JAVASCRIPT, PYTHON	1.Enter URL and clickgo 2.clickon my account dropdown button 3.complete the login activity 4.enter the dashboard and upload the image to be processed 5.image values are analyzed and displayed		Food Classified is: BANANA [('sugar g': 12.3, 'fiber g': 2.6, 'serving size g: 100.0, 'sodium mg": 1, "name": 'banana'. potassium mg': 22, "fat saturated g": 0.1, "fat total g': 0.3. 'calories": 89.4, cholesterol mg': 0, 'protein g': 1.1, 'carbohydrates total g': 23.2}]	Food Classified is: BANANA [('sugar g': 12.3, 'fiber g': 2.6, 'serving size g: 100.0, 'sodium mg": 1, "name": 'banana'. potassium mg': 22, "fat saturated g": 0.1, "fat total g': 0.3. 'calories": 89.4, cholesterol mg': 0, 'protein g': 1.1, 'carbohydrates total g': 23.2}]	pass	Y		K.FRANCISHUBAN
home page TC06	UI	home page	helps the user to know well about the platform and to navigate to the application	HTML,CSS,JAVASCRIPT	1.Enter URL and clickgo 2.clickon my account dropdown button 3.Verify login/Singup popup displayed or not	FOUR CARDS TO NAVIGATE	User should navigate to user account homepage	NAVIGATION PASSED	pass	N		E.R.GOWSHIGAN
TC07	functional	BMI CALCULATOR	helps to calculate the body mass index of the person	TML, CSS, JAVASCRIP	1.Enter URL and clickgo 2.clickon my account dropdown button 3.complete the login activity 4.clickon to	USER:1 HEIGHT:175CM WEIGHT:60KG	BMI:19.3	19.29	pass	N		S.K.GNAPATHYRAM
TC08	Functional	NUTRITION ANALYZER	provides the nutritional value of the given food	HTML,CSS,JAVASCRIPT, PYTHON	1.Enter URL and clickgo 2.clickon my account dropdown button 3.complete the login activity 4.enter the dashboard and upload the image to be processed 5.image values are analyzed and displayed		FFood Classified is: APPLES [{"sugar_g": 10.3, 'fiber g': 2.4, 'serving size g: 100.0, sodium mg":1, 'name': 'apples', 'potassium mg: 11, "fat saturated g": 0.0, 'fat total g': 0.2. calories': 53.4, 'cholesterol mg': 0, "protein g': 0.3, 'carbohydrates total g': 13.8)]	FFood Classified is: APPLES [{"sugar_g": 10.3, 'fiber g': 2.4, 'serving size g: 100.0, sodium mg":1, 'name': 'apples', 'potassium mg: 11, "fat saturated g": 0.0, 'fat total g': 0.2. calories': 53.4, 'cholesterol mg': 0, "protein g': 0.3, 'carbohydrates total g': 13.8)]	pass	Y		S.K.GNAPATHYRAM
TC09	Functional	NUTRITION ANALYZER	provides the nutritional value of the given food	HTML,CSS,JAVASCRIPT, PYTHON	1.Enter URL and clickgo 2.clickon my account dropdown button 3.complete the login activity 4.enter the dashboard and upload the image to be processed 5.image values are analyzed and displayed		Food Classified is: ORANGE [('sugar g': 8.4, 'fiber g': 2.2, 'serving size g: 100.0, sodium mg": 1, "name": 'orange, potassium mg': 23, "fat saturated g": 0.0, 'fat total g': 0.1, 'calories": 50.4, 'cholesterol mg': 0, 'protein g': 0.9, 'carbohydrates total g': 12.4}]	Food Classified is: ORANGE [('sugar g': 8.4, 'fiber g': 2.2, 'serving size g: 100.0, sodium mg": 1, "name": 'orange, potassium mg': 23, "fat saturated g": 0.0, 'fat total g': 0.1, 'calories": 50.4, 'cholesterol mg': 0, 'protein g': 0.9, 'carbohydrates total g': 12.5}]	pass	y		k.FRANCISHUBAN

8.2 USER ACCEPTANCE TESTING

1.PURPOSE OF THE DOCUMENT

The purpose of this document is to briefly explain the test coverage and open issues of the B-FIT project at the time of the release to User Acceptance Testing (UAT).

2. DEFINE ANALYSIS

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	6	3	1	18
Duplicate	1	0	0	2	3
External	5	3	6	1	15
Fixed	11	9	12	14	46
Not Reproduced	2	0	1	0	3
Skipped	0	3	0	2	5
Won't Fix	0	0	3	5	8
Totals	27	21	25	25	98

3. TEST CASE ANALYSIS

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	9	0	0	9
Client Application	10	0	0	10
Security	1	0	0	1
Outsource Shipping	2	0	0	2
Exception Reporting	11	0	0	11
Final Report Output	8	0	0	8
Version Control	8	0	0	8

9. RESULTS

9.1 PERFORMANCE METRIC

Model Performance Testing: NFT – Detailed Test Plan

S.no	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/Signoff
1	Display nutrient content	Stress	App Crash/Developer team/ Site Down	Approved
2	Display nutrient content	Load	Server Crash/ Developer team	Approved

END OF TEST REPORT

NFR-Met	Test Outcome	GO/NOGO decision	Recommendations	Idetified Defects (Detected/Closed/Open)
Performance	CPU - 01	GO	High Performance	Closed
Database Information	Storage	NO-GO	Mysql	Closed

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES

Yet people consume foods, not nutrients, so it is helpful to view food or a meal as more than just a set of nutrients that impact our health. Some weight-loss diets have assigned a negative connotation to certain nutrients, such as low-fat or low-carbohydrate diets. This can create a view that a specific nutrient is bad, regardless of the role it plays when foods containing that nutrient are consumed as part of a healthy, balanced diet. This model helps in analysing the nutrition in the food.

DISADVANTAGES:

Like anything, there are always drawbacks. In some cases, the predicting algorithm may give the wrong output.

11.CONCLUSION

Nowadays, people becoming conscious about their diets and fitness goals, there is a wide scope of diet and fitness apps thriving in the app world. This application be used in the field of nutrition and fitness. The further apps will help both in health promotion and monitoring and evaluation of dietary assessment.Hence, to make their fitness path a bit smoother and to enhance their experience, we have created this Web application to provide a broader approach to providing a better life through a nutritious and fit diet plan to the users.

12.FUTURE SCOPE

In the future, this application could enhance its functionality by adding image recognition, which can be used to analyze the food image and produce the result with the nutrition values contained in that particular food item. A Google map can be added to track the distance covered by the user using the Activity Tracker to provide a more visual representation of the activity to the user. The activity tracker can also be updated using the Google fit API for a more accurate result

13.APPENDIX

SOURCE CODE

DESIGNING THE GET STARTED PAGE

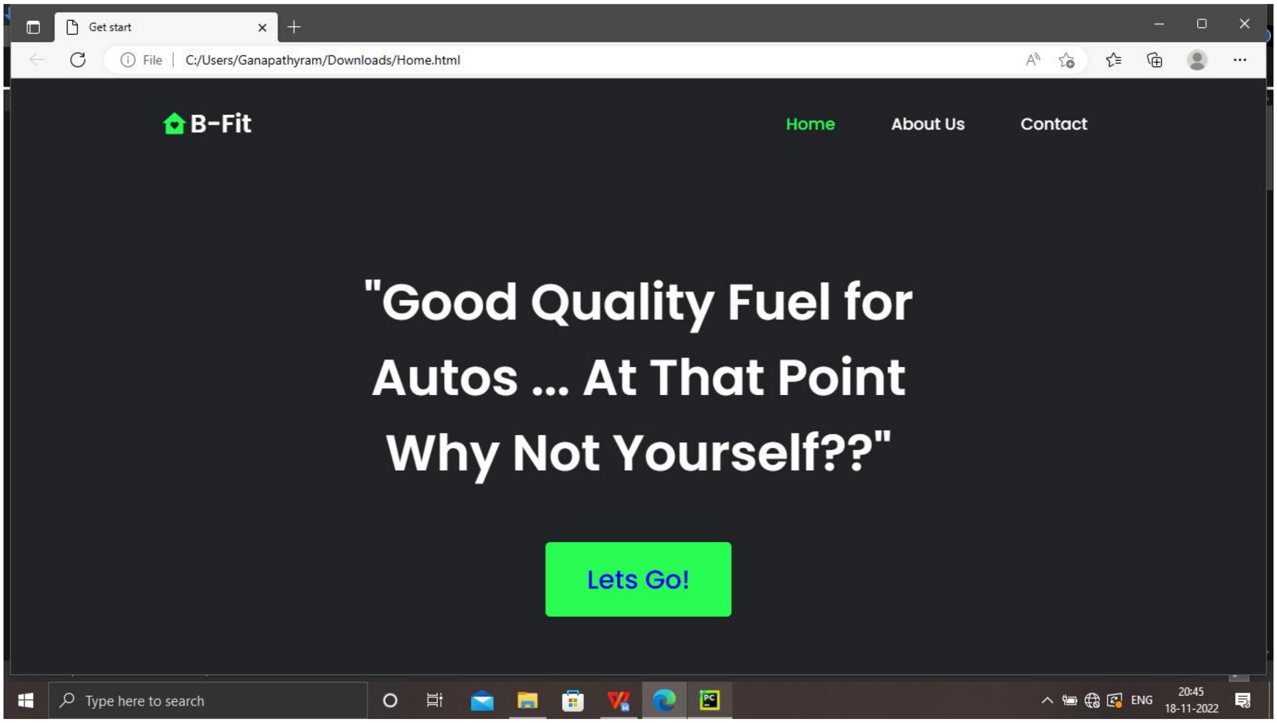
```
  <!DOCTYPE html>

  <html>

  <head>

      <meta charset="utf-8">

      <meta name="viewport" content="width=device-width, initial-scale=1">

      <title> Get start </title>

      <link href="https://cdn.jsdelivr.net/npm/remixicon@2.5.0/fonts/remixicon.css"
rel="stylesheet">

      <linkrel="stylesheet"
href="https://unpkg.com/boxicons@latest/css/boxicons.min.css">link

     <link rel="preconnect" href="https://fonts.googleapis.com">

     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

  <href="https://fonts.googleapis.com/css2?family=Poppins:wght@500;600&display=s
wap" rel="stylesheet">

</head>

<style>

  *{
      padding: 0;
      margin: 0;
      box-sizing: border-box;
      font-family: 'Poppins', sans-serif;
      text-decoration: none;
      list-style: none;
  }
  :root{
      --bg-color: #222327;
      --text-color: #fff;
      --main-color: #29fd53;
  }
  body{
      min-height: 100vh;
      background: var(--bg-color);
      color: var(--text-color);
      text-align:center;
  }
  header{
      position: fixed;

      width: 100%;

      top: 0;

      right: 0;

      z-index: 1000;

      display: flex;

      align-items: center;

      justify-content: space-between;

      background: transparent;

      padding: 28px 12%;

      transition: all .50s ease;

  }

  .logo{

      display: flex;

      align-items: center;

  }

  .logo i{

      color: var(--main-color);

      font-size: 28px;
      margin-right: 3px;

  }

  .logo span{

      color: var(--text-color);

      font-size: 1.7rem;

      font-weight: 600;

  }

  .navbar{

      display: flex;

  }

  .navbar a{

      color: var(--text-color);

      font-size: 1.1rem;

      font-weight: 500;

      padding: 5px 0;

      margin: 0px 30px;

      transition: all .50s ease;

  }

  .navbar a:hover{
      color: var(--main-color);
  }
  .navbar a.active{
      color: var(--main-color);
  }
  .main{
      display: flex;
      align-items: center;
  }
  .main a{
      margin-right: 25px;
      margin-left: 10px;
      color: var(--text-color);
      font-size: 1.1rem;
      font-weight: 500;
      transition: all .50s ease;
  }
  .user{
      display: flex;
      align-items: center;
  }
  .user i{
      color: var(--main-color);
      font-size: 28px;
      margin-right: 7px;
  }
  .main a:hover{
      color: var(--main-color);
  }
  #menu-icon{
      font-size: 35px;
      color: var(--text-color);
      cursor: pointer;
      z-index: 10001;
      display: none;
  }
  @media (max-width: 1280px){
      header{
              padding: 14px 2%;

              transition: .2s;

      }

      .navbar a{

              padding: 5px 0;

              margin: 0px 20px;

      }

  }

  @media (max-width: 1090px){

      #menu-icon{

              display: block;

      }

      .navbar{

              position: absolute;

              top: 100%;

              right: -100%;

              width: 270px;

              height: 29vh;

              background: var(--main-color);
```

```
        display: flex;
        flex-direction: column;
        justify-content: flex-start;
        border-radius: 10px;
        transition: all .50s ease;
}
.navbar a{
        display: block;
        margin: 12px 0;
        padding: 0px 25px;
        transition: all .50s ease;
}
.navbar a:hover{
        color: var(--text-color);
        transform: translateY(5px);
}
.navbar a.active{
        color: var(--text-color);
}
```

```
     .navbar.open{
            right: 2%;
     }
}
.btn {
    display: inline-block;
    border-radius: 5px;
    background-color: var(--main-color);
    border: none;
    text-align: center;
    font-size: 28px;
    padding: 20px;
    width: 200px;
    height:80px;
    transition: all 0.5s;
    cursor: pointer;
    margin: 5px;
    bottom: 75%;
    color:var(--bg-color);
```

```
}
.btn span {
    cursor: pointer;
    display: inline-block;
    position: relative;
    transition: 0.5s;
}
.btn span:after {
    content: '\00bb';
    position: absolute;
    opacity: 0;
    top: 0;
    right: -20px;
    transition: 0.5s;
}
.btn:hover span {
    padding-right: 25px;
}
.btn:hover span:after {
   opacity: 1;
   right: 0;
}
        .content1{
    padding-top:200px;
    text-align:center;
    padding-bottom:50px;
    font-weight: bold;
    font-size:4vw;
    font-family: poppins;
}
button{
    color:#D9E4F5;
    border:none;
    transition: .4 ease-in;
    z-index:1;
    }
    button::before,
    button::after{
    position: absolute;

    content:"";

    z-index:-1;

    }

    .btn:hover{

    background: var(--main-color);

    box-shadow: 0 0 5px var(--main-color), 0 0 25px var(--main-color), 0 0 50px var(--main-color), 0 0 200px var(--main-color);

    }

</style

<body>

    <header>

            <a    href="#"    class="logo"><i    class="ri-home-heart-fill"></i><span>B-Fit</span></a>

            <ul class="navbar">

                    <li><a href="#" class="active">Home</a></li>

                    <li><a href="#">About Us</a></li>

                    <li><a href="#">Contact</a></li>

            </ul>

            </header>
            <div class="content1">
            <p> "Good Quality Fuel for<br>
                  Autos ... At That Point<br>
                    Why Not Yourself??"
                </p>
           </div>
         <div>
                      <a href="login.html"><button class="btn" ><span>Lets Go!
</span></a></button>
            </div>
    <!--js link--->
    <script>
    let menu = document.querySelector('#menu-icon');
let navbar = document.querySelector('.navbar');
menu.onclick = () => {
    menu.classList.toggle('bx-x');
    navbar.classList.toggle('open');
}
    </script>
```

```
</body>

</html>
```



"Good Quality Fuel for Autos ... At That Point Why Not Yourself??"

CREATING HTML PAGES FOR REGISTRATION AND LOGIN ,ROUTING INTO FLASK AND CONFRIMATION MESSAGE

LOGIN PAGE

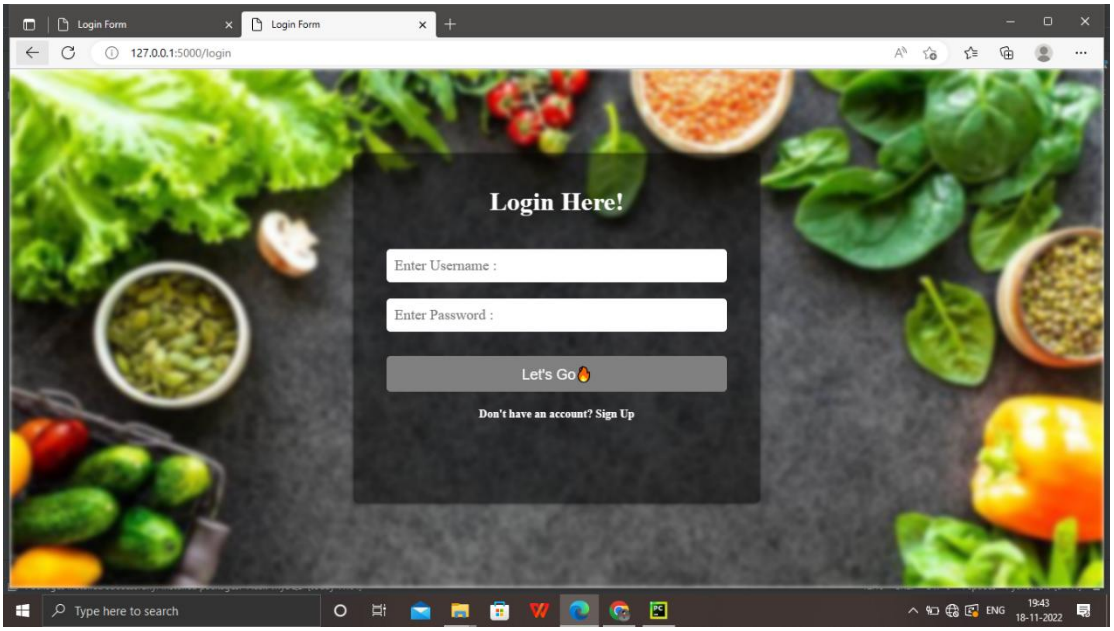
Login page which gets a user name and password and verifies with a user

```
<!DOCTYPE html>

<html>

    <head>

    <meta charset="UTF-8">
    <title>Login Form</title>

    </head>

<style>

* {

    box-sizing: border-box;

}

body {

    font-family: poppins;

    font-size: 16px;

    color: #fff;

}

.form-box {

    background-color: rgba(0, 0, 0, 0.5);

    margin: auto auto;

    padding: 40px;

    border-radius: 5px;

    box-shadow: 0 0 5px #000;

    position: absolute;

    top: 0;
    bottom: 0;

    left: 0;

    right: 0;

    width: 500px;

    height: 430px;

}

.form-box:before {

    background-image: url("https://i.postimg.cc/Fs7qM1YH/images.jpg");

    width: 100%;

    height: 100%;

    background-size: cover;

    content: "";

    position: fixed;

    left: 0;

    right: 0;

    top: 0;

    bottom: 0;

    z-index: -1;

    filter: blur(3px);
```

```
}

.form-box .header-text {

    font-size: 32px;

    font-weight: 600;

    padding-bottom: 30px;

    text-align: center;

}

.form-box input {

    margin: 10px 0px;

    border: none;

    padding: 10px;

    border-radius: 5px;

    width: 100%;

    font-size: 18px;

    font-family: poppins;

}

.form-box input[type=checkbox] {

    display: none;

}
```

```
.form-box label {
    position: relative;
    margin-left: 5px;
    margin-right: 10px;
    top: 5px;
    display: inline-block;
    width: 20px;
    height: 20px;
    cursor: pointer;
}
.form-box label:before {
    content: "";
    display: inline-block;
    width: 20px;
    height: 20px;
    border-radius: 5px;
    position: absolute;
    left: 0;
    bottom: 1px;
    background-color: #ddd;
}
.form-box span {
    font-size: 14px;
}
.form-box button {
    background-color: #808080;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    width: 100%;
    font-size: 18px;
    padding: 10px;
    margin: 20px 0px;
}
span a {
    color: #BBB;
}
```

```
    h4{

    margin:0px 0px 0px 0px;

    text-align:center;

    }

    </style>

        <body>

               <form action="{{ url_for('login') }}" method="post">

               <div class="form-box">

        <div class="header-text">

    Login Here!

                </div>

                    <div class="msg">{{ msg }}</div>

                        <input id="username" name="username" type="text" placeholder="Enter Username :">

                        <input id="password" name="password" type="password" placeholder="Enter Password :" >

                        <button><a href="index.html" style="color:#fff; text-decoration:none;">Let's Go&#128293</a></button>

                         <span><h4><b>Don't have an account? </b><a href="{{url_for('register')}}" style="color:#fff; text-decoration:none;"><b>Sign Up</b></h4></a></span>
    </div>

    </form>

    </body>

    </html>
```



Login Here!

Enter Username :

Enter Password :

Let's Go🔥

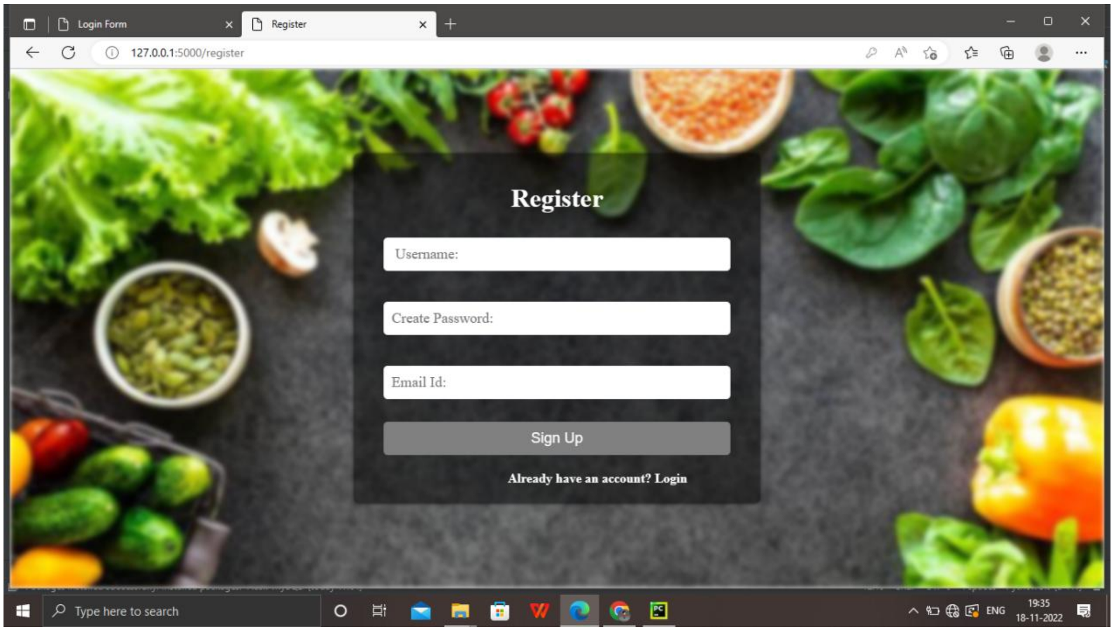
Don't have an account? Sign Up

<u>Register page</u>

```
<!DOCTYPE html>

    <html>

        <head>

            <meta charset="utf-8">

            <title> Register </title>

            <style>
            * {
        box-sizing: border-box;
    }
    body {
        font-family: poppins;
        font-size: 16px;
        color: #fff;
    }
    .form-box {
        background-color: rgba(0, 0, 0, 0.5);
        margin: auto auto;
        padding: 36px;
        border-radius: 5px;
        box-shadow: 0 0 5px #000;
        position: absolute;
        top: 0;
        bottom: 0;
        left: 0;
        right: 0;
        width: 500px;

        height: 430px;

    }

    .form-box:before {

        background-image: url("https://i.postimg.cc/Fs7qM1YH/images.jpg");

        width: 100%;

        height: 100%;

        background-size: cover;

        content: "";

        position: fixed;

        left: 0;

        right: 0;

        top: 0;

        bottom: 0;

        z-index: -1;

        filter: blur(3px);

    }

    .form-box .header-text {

        font-size: 32px;
        font-weight: 600;

        padding-bottom: 20px;

        text-align: center;

        padding-top:0px;

    }

    .form-box input {

        margin: 10px 0px;

        border: none;

        padding: 10px;

        border-radius: 5px;

        width: 100%;

        font-size: 18px;

        font-family: poppins;

    }

    .form-box input[type=checkbox] {

        display: none;

    }

    .form-box label {

        position: relative;
        margin-left: 5px;

        margin-right: 10px;

        top: 5px;

        display: inline-block;

        width: 20px;

        height: 20px;

        cursor: pointer;

    }

    .form-box label:before {

        content: "";

        display: inline-block;

        width: 20px;

        height: 20px;

        border-radius: 5px;

        position: absolute;

        left: 0;

        bottom: 1px;

        background-color: #ddd;

    }
```

```
.form-box span {

     font-size: 14px;

}

.btn {

     background-color: #808080;

     color: #fff;

     border: none;

     border-radius: 5px;

     cursor: pointer;

     width: 100%;

     font-size: 18px;

     padding: 10px;

     margin: 20px 0px;

     margin-top:0px;

}

span a {

     color: #bbb;

}

h4{
  margin:0px 0px 10px 100px;

  text-align:center;

  }


            </style>

       </head>

       <body></br></br></br>

            <div class="form-box">

       <div class="header-text">Register

                 </div>

                      <form action="{{ url_for('register') }}" method="post">

                           <div class="msg">{{ msg }}</div>

                           <input id="username" name="username" type="text"
placeholder=" Username:" class="textbox"/></br></br>

                           <input id="password" name="password" type="password"
placeholder="Create Password:" class="textbox"/></br></br>

                           <input id="email" name="email" type="text" placeholder="Email
Id:" class="textbox"/></br></br>

                           <button class="btn"><a href="{{url_for('login')}}"
style="color:#fff; text-decoration:none;" >Sign Up</button></form>
```

```
            <span><b><h4>Already have an account?</b><a href="{{url_for('login')}}" style="color:#fff; text-decoration:none;"><b>Login</h4></b></a></span>

          </div>

      </body>

  </html>
```

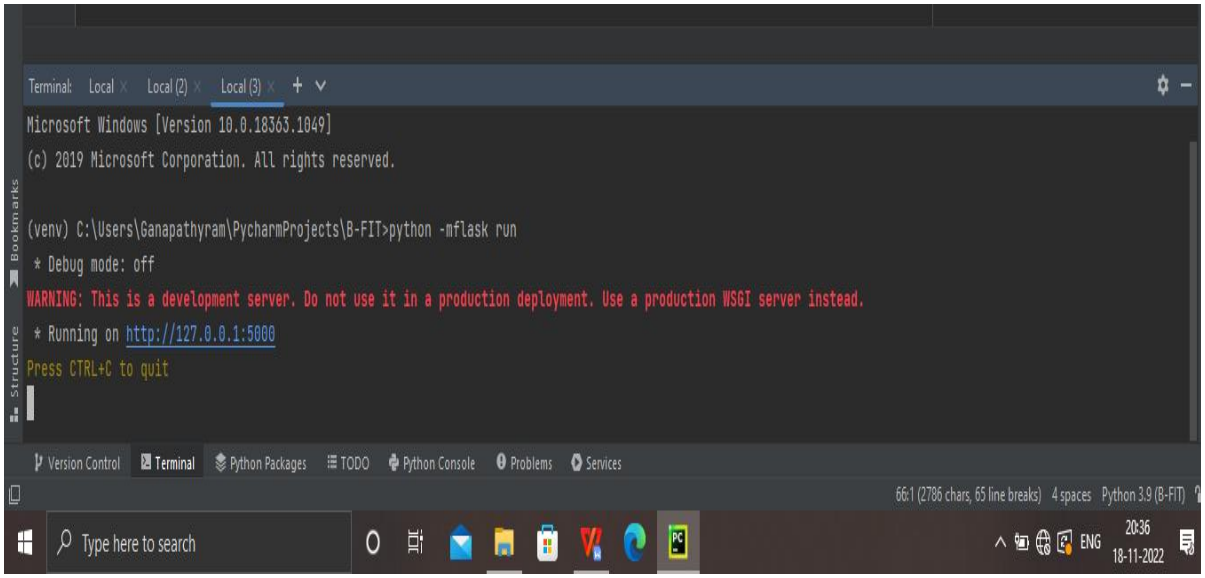


Routing to flask

```
import MySQLdb

from flask import Flask, render_template, request, redirect, url_for, session

from flask_mysqldb import MySQL

import MySQLdb.cursors
import re
app = Flask(__name__, template_folder='template')
app.secret_key = 'This is my secret key'
app.config['MYSQL_server'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config["MYSQL_PASSWORD"] = '12345qwert@@'
app.config['MYSQL_DB'] = 'hubidb'
mysql = MySQL(app)
@app.route('/')
@app.route('/login', methods=['GET', 'POST'])
def login():
    msg = ''
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM project WHERE username = % s AND password = % s', (username, password, ))
        account = cursor.fetchone()
        if account:
            session['loggedin'] = True
            session['id'] = account['id']
            session['username'] = account['username']
            msg = 'Logged in successfully !'
            return render_template('index.html', msg=msg)
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg=msg)
@app.route('/')
@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect(url_for('login'))
@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
```

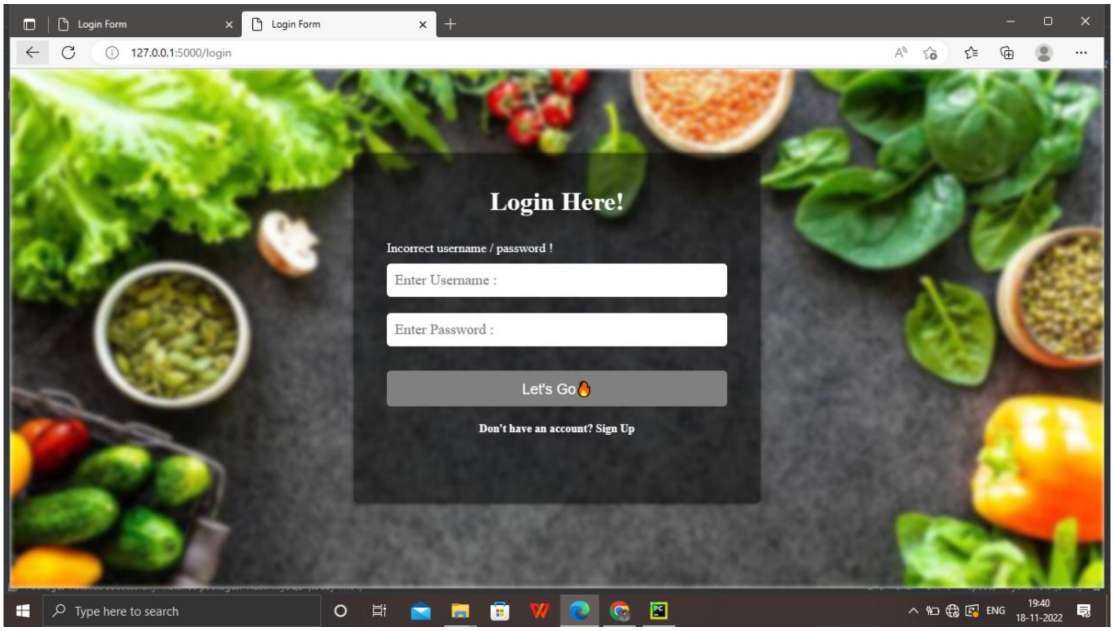
```
if request.method == 'POST' and 'username' in request.form and 'password' in request.form and 'email' in request.form:
    username = request.form['username']
    password = request.form['password']
    email = request.form['email']
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute('SELECT *FROM project WHERE username = % s', (username, ))
    account = cursor.fetchone()
    if account:
        msg = 'Account already exists !'
    elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
        msg = 'Invalid email address !'
    elif not re.match(r'[A-Za-z0-9]+', username):
        msg = 'Username must contain only characters and numbers !'
    elif not username or not password or not email:
        msg = 'Please fill out the form !'
    else:
        cursor.execute('INSERT INTO project VALUES (NULL, % s, % s, % s)', (username, password, email, ))
```

```
            mysql.connection.commit()

            msg = 'You have successfully registered !'

    elif request.method == 'POST':

        msg = 'Please fill out the form !'

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

if __name__ == '__main__':

        app.run(debug=True)
```



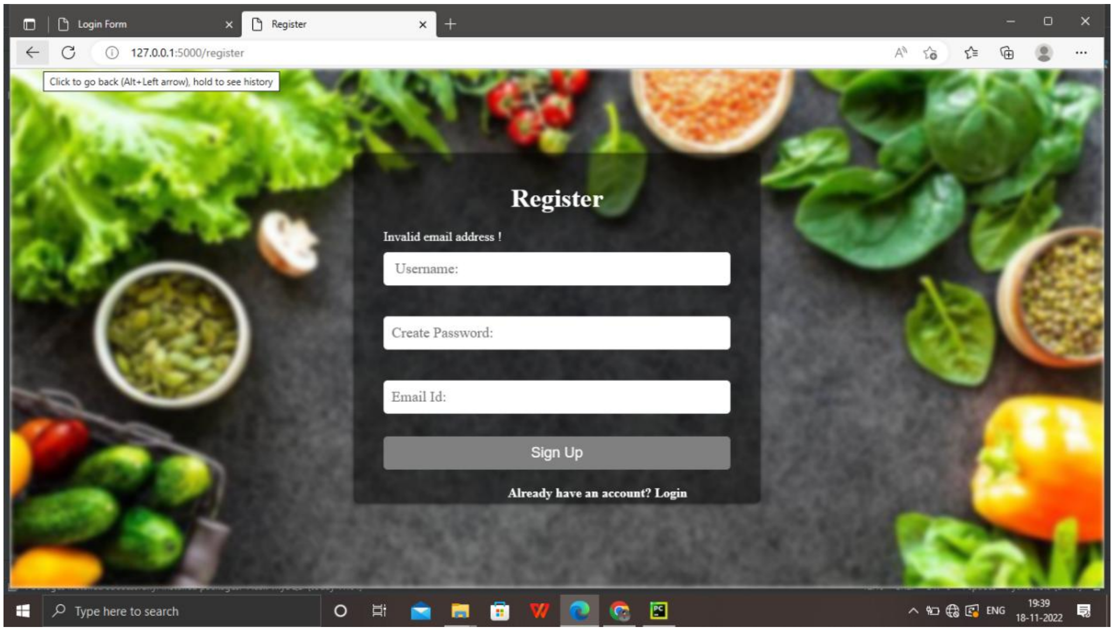
```
Microsoft Windows [Version 10.0.18363.1049]
(c) 2019 Microsoft Corporation. All rights reserved.

(venv) C:\Users\Ganapathyram\PycharmProjects\B-FIT>python -mflask run
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Invalid login



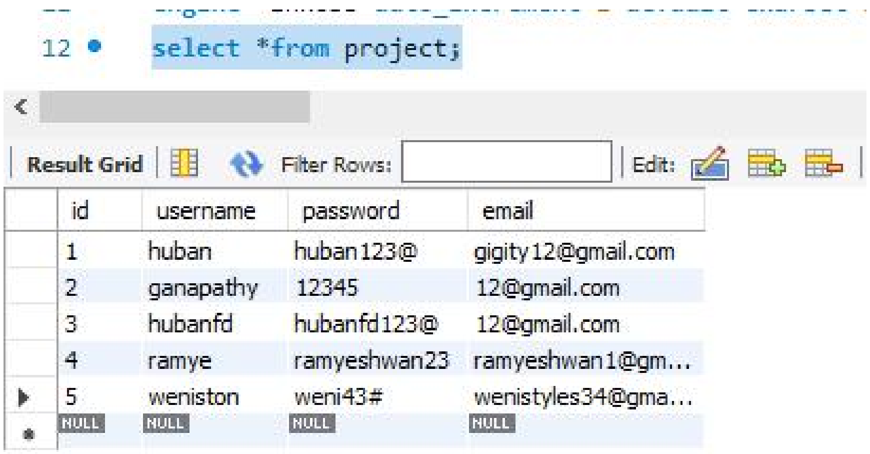
Incorrect username / password !

Invalid register



Invalid email address !

Database:



```
12 ●     select *from project;
```

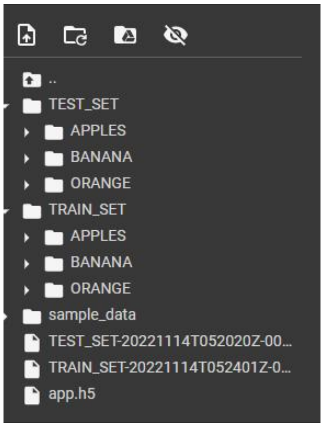
id	username	password	email
1	huban	huban123@	gigity12@gmail.com
2	ganapathy	12345	12@gmail.com
3	hubanfd	hubanfd123@	12@gmail.com
4	ramye	ramyeshwan23	ramyeshwan1@gm...
5	weniston	weni43#	wenistyles34@gma...
NULL	NULL	NULL	NULL

SPRINT-2

<u>Image processing</u>

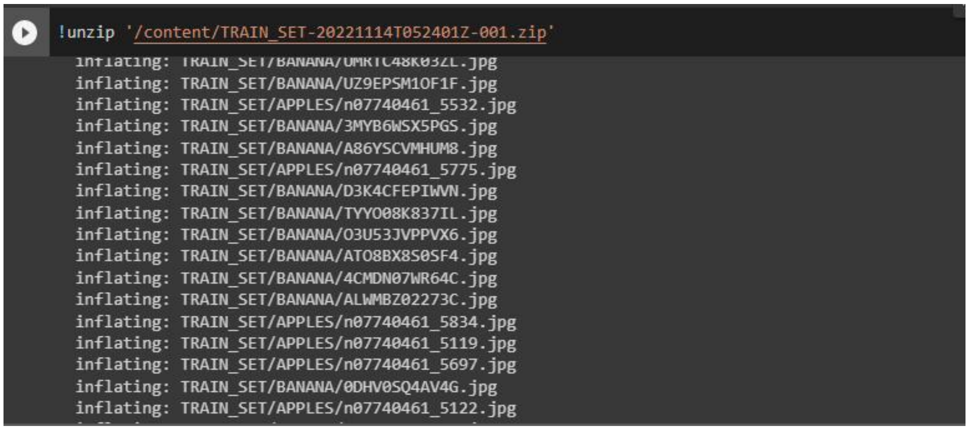
COLlECT THE DATASET

In this project, we have collected images of 3 types of food items: apples, 'banana' and 'orange'. They are saved in the respective sub directories with their respective names. For more accurate results, we can collect images at high resolution and feed the model with more images



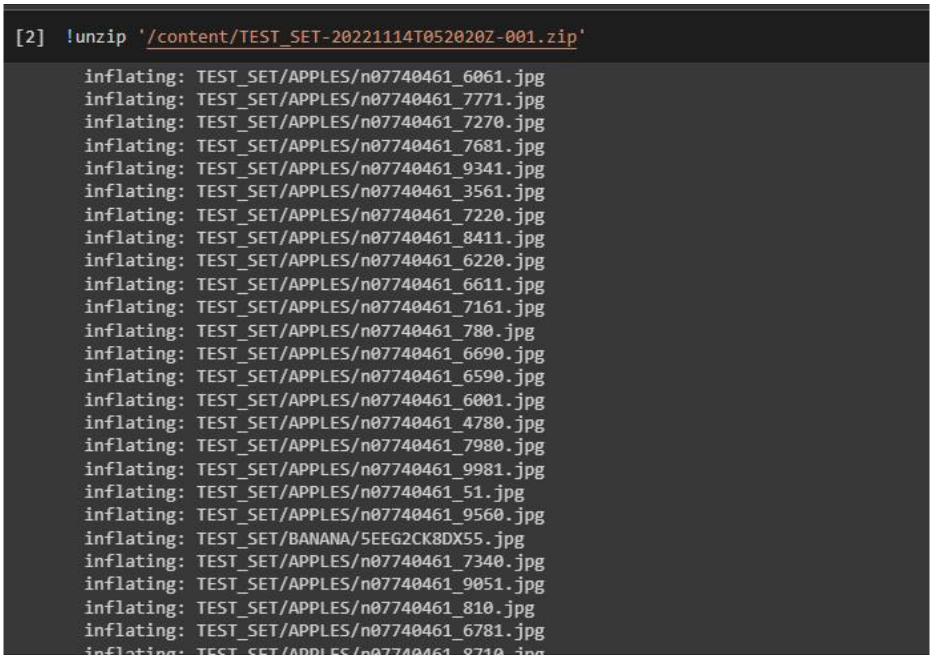
```
..
TEST_SET
    APPLES
    BANANA
    ORANGE
TRAIN_SET
    APPLES
    BANANA
    ORANGE
sample_data
TEST_SET-20221114T052020Z-00...
TRAIN_SET-20221114T052401Z-0...
app.h5
```

>> !unzip '/content/TRAIN_SET-20221114T052401Z-001.zip



```
!unzip '/content/TRAIN_SET-20221114T052401Z-001.zip'
   inflating: TRAIN_SET/BANANA/UMKTC48K03ZL.jpg
   inflating: TRAIN_SET/BANANA/UZ9EPSM1OF1F.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5532.jpg
   inflating: TRAIN_SET/BANANA/3MYB6WSX5PGS.jpg
   inflating: TRAIN_SET/BANANA/A86YSCVMHUM8.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5775.jpg
   inflating: TRAIN_SET/BANANA/D3K4CFEPIWVN.jpg
   inflating: TRAIN_SET/BANANA/TYYO08K837IL.jpg
   inflating: TRAIN_SET/BANANA/O3U53JVPPVX6.jpg
   inflating: TRAIN_SET/BANANA/ATO8BX8S0SF4.jpg
   inflating: TRAIN_SET/BANANA/4CMDN07WR64C.jpg
   inflating: TRAIN_SET/BANANA/ALWMBZ02273C.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5834.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5119.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5697.jpg
   inflating: TRAIN_SET/BANANA/0DHV0SQ4AV4G.jpg
   inflating: TRAIN_SET/APPLES/n07740461_5122.jpg
```

>>!unzip '/content/TEST_SET-20221114T052020Z-001.zip'



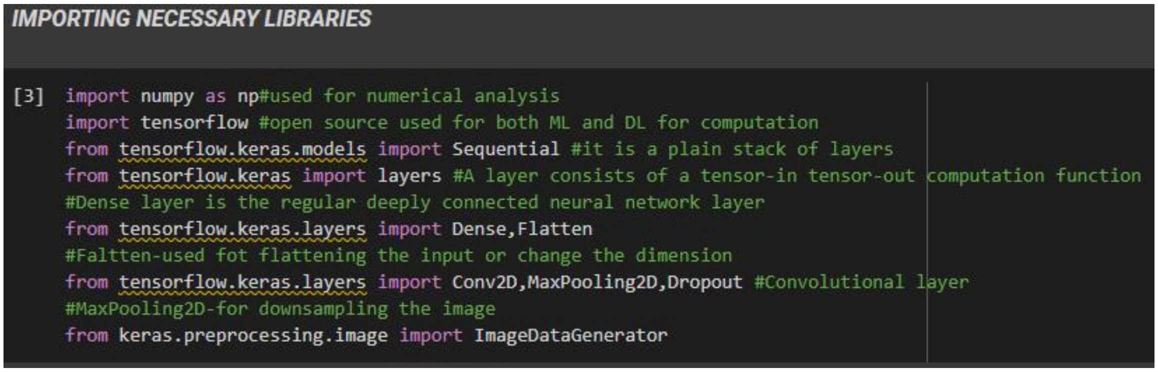
```
[2]  !unzip '/content/TEST_SET-20221114T052020Z-001.zip'
       inflating: TEST_SET/APPLES/n07740461_6061.jpg
       inflating: TEST_SET/APPLES/n07740461_7771.jpg
       inflating: TEST_SET/APPLES/n07740461_7270.jpg
       inflating: TEST_SET/APPLES/n07740461_7681.jpg
       inflating: TEST_SET/APPLES/n07740461_9341.jpg
       inflating: TEST_SET/APPLES/n07740461_3561.jpg
       inflating: TEST_SET/APPLES/n07740461_7220.jpg
       inflating: TEST_SET/APPLES/n07740461_8411.jpg
       inflating: TEST_SET/APPLES/n07740461_6220.jpg
       inflating: TEST_SET/APPLES/n07740461_6611.jpg
       inflating: TEST_SET/APPLES/n07740461_7161.jpg
       inflating: TEST_SET/APPLES/n07740461_780.jpg
       inflating: TEST_SET/APPLES/n07740461_6690.jpg
       inflating: TEST_SET/APPLES/n07740461_6590.jpg
       inflating: TEST_SET/APPLES/n07740461_6001.jpg
       inflating: TEST_SET/APPLES/n07740461_4780.jpg
       inflating: TEST_SET/APPLES/n07740461_7980.jpg
       inflating: TEST_SET/APPLES/n07740461_9981.jpg
       inflating: TEST_SET/APPLES/n07740461_51.jpg
       inflating: TEST_SET/APPLES/n07740461_9560.jpg
       inflating: TEST_SET/BANANA/5EEG2CK8DX55.jpg
       inflating: TEST_SET/APPLES/n07740461_7340.jpg
       inflating: TEST_SET/APPLES/n07740461_9051.jpg
       inflating: TEST_SET/APPLES/n07740461_810.jpg
       inflating: TEST_SET/APPLES/n07740461_6781.jpg
       inflating: TEST_SET/APPLES/n07740461_8710.jpg
```

<u>IMPORT THE DATA GENERATOR LIBRARY ,CONFIGURE IMAGE DATA GENERATOR CLASS AND APPLY TO THE TRAIN AND TEST DATA SET</u>

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class. Let us import the ImageDataGenerator class from Keras

iv. IMPORT THE REQUIRED LIBRARIES

```
import numpy as np#used for numerical analysis

import tensorflow #open source used for both ML and DL for computation

from tensorflow.keras.models import Sequential #it is a plain stack of layers

from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function

#Dense layer is the regular deeply connected neural network layer

from tensorflow.keras.layers import Dense,Flatten

#Faltten-used fot flattening the input or change the dimension

from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional layer

#MaxPooling2D-for downsampling the image

from keras.preprocessing.image import ImageDataGenerator
```



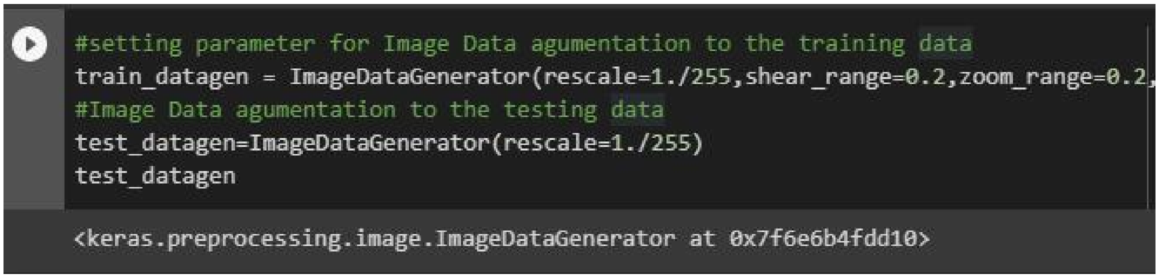
IMPORTING NECESSARY LIBRARIES

```
[3]  import numpy as np#used for numerical analysis
     import tensorflow #open source used for both ML and DL for computation
     from tensorflow.keras.models import Sequential #it is a plain stack of layers
     from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
     #Dense layer is the regular deeply connected neural network layer
     from tensorflow.keras.layers import Dense,Flatten
     #Faltten-used fot flattening the input or change the dimension
     from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional layer
     #MaxPooling2D-for downsampling the image
     from keras.preprocessing.image import ImageDataGenerator
```

v. IMAGE AUGUMENTATION

```
#setting parameter for Image Data agumentation to the training data

train_datagen =
ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)

#Image Data agumentation to the testing data

test_datagen=ImageDataGenerator(rescale=1./255)
```

```
test_datagen
```

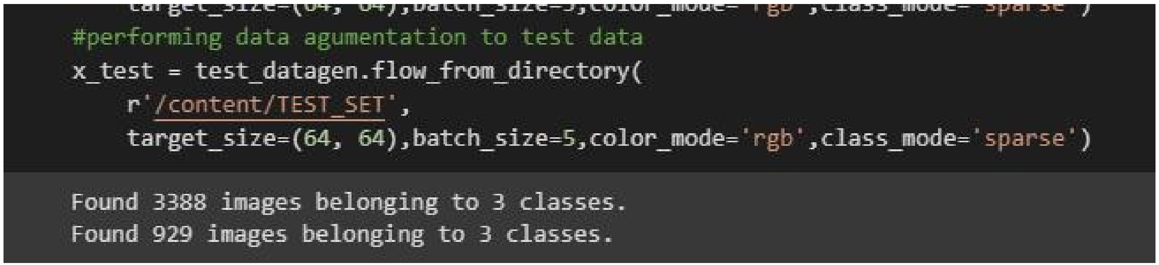


```
#setting parameter for Image Data agumentation to the training data
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
test_datagen
```

```
<keras.preprocessing.image.ImageDataGenerator at 0x7f6e6b4fdd10>
```

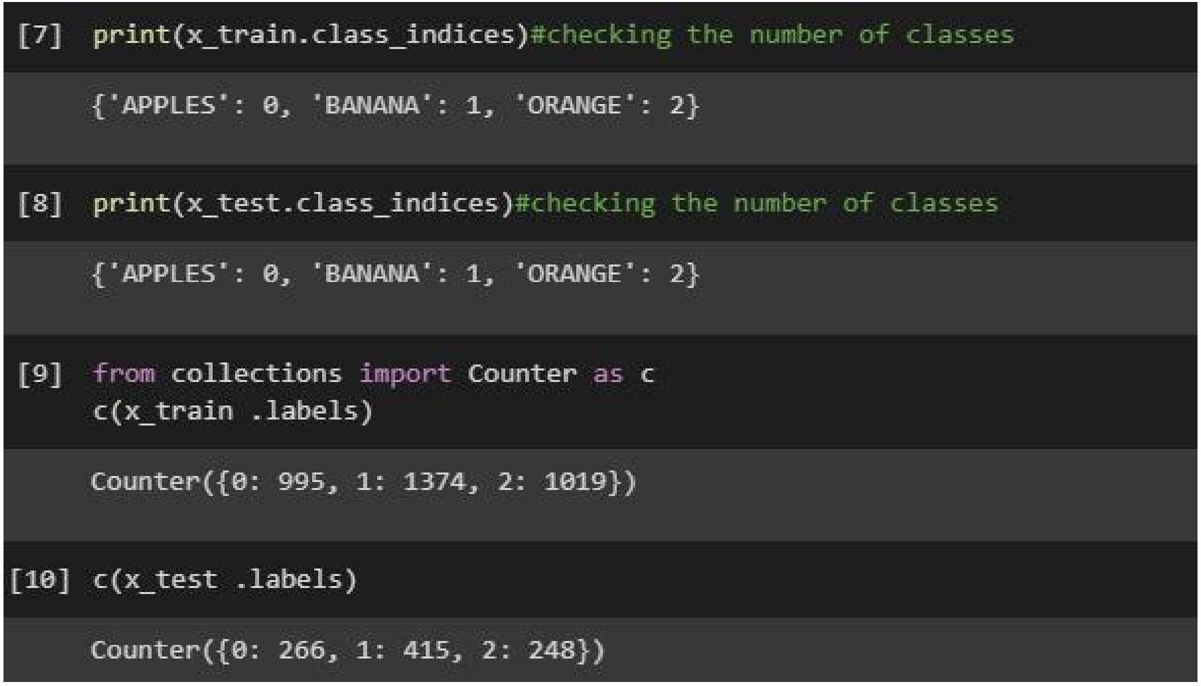
vi. LOADING DATA

```
#performing data agumentation to train data

x_train = train_datagen.flow_from_directory(

        r'/content/TRAIN_SET',

        target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')

#performing data agumentation to test data

x_test = test_datagen.flow_from_directory(

        r'/content/TEST_SET',

    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```



```
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
#performing data agumentation to test data
x_test = test_datagen.flow_from_directory(
    r'/content/TEST_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```

```
Found 3388 images belonging to 3 classes.
Found 929 images belonging to 3 classes.
```



```
[7]  print(x_train.class_indices)#checking the number of classes
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2}
```

```
[8]  print(x_test.class_indices)#checking the number of classes
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2}
```

```
[9]  from collections import Counter as c
     c(x_train .labels)
```

```
Counter({0: 995, 1: 1374, 2: 1019})
```

```
[10] c(x_test .labels)
```

```
Counter({0: 266, 1: 415, 2: 248})
```

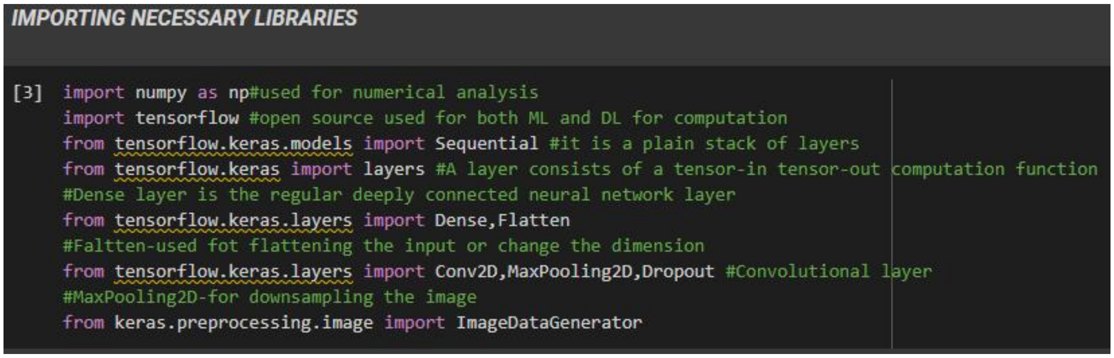
MODEL BUILDING

Steps to Build a Deep Learning Model

6. Defining model architecture

This is a very crucial step in our deep learning model building process. We have to define how our model will look and that requires

- **Importing the libraries**



IMPORTING NECESSARY LIBRARIES

```
[3] import numpy as np#used for numerical analysis
    import tensorflow #open source used for both ML and DL for computation
    from tensorflow.keras.models import Sequential #it is a plain stack of layers
    from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
    #Dense layer is the regular deeply connected neural network layer
    from tensorflow.keras.layers import Dense,Flatten
    #Faltten-used fot flattening the input or change the dimension
    from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional layer
    #MaxPooling2D-for downsampling the image
    from keras.preprocessing.image import ImageDataGenerator
```

- **Adding CNN (Convolution Neural Network) Layers**

Keras has 2 ways to define a neural network:

- Sequential
- Function API

The Sequential class is used to define a linear initialization of network layers which then, collectively, constitute a model. In our example below, we will use the Sequential constructor to create a model, which will then have layers added to it using the add() method

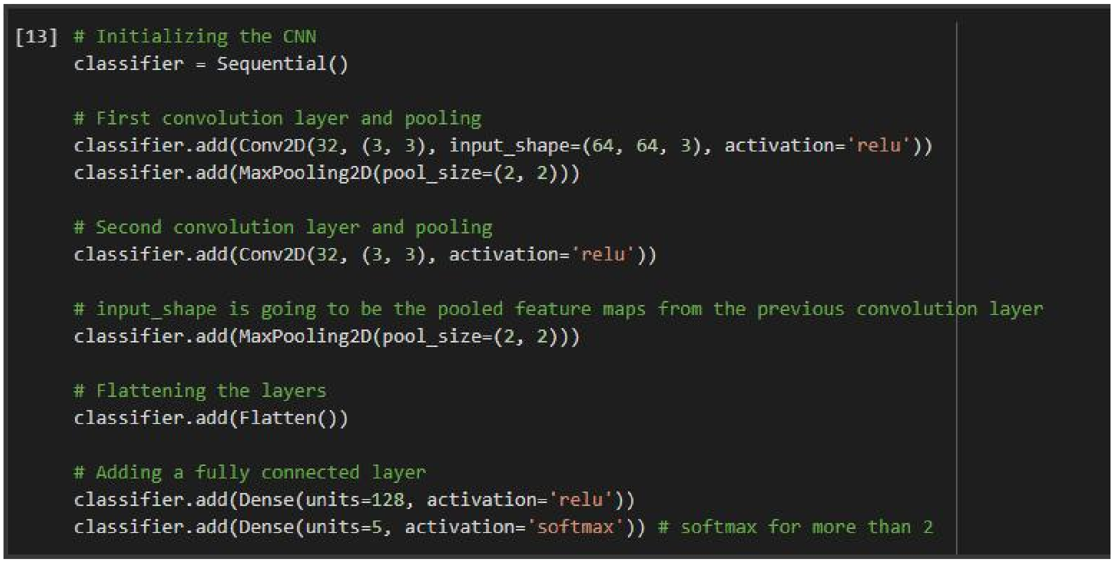
- **Adding Dense layers**

We will be adding three layers for CNN

- Convolution layer
- Pooling layer
- Flattening layer

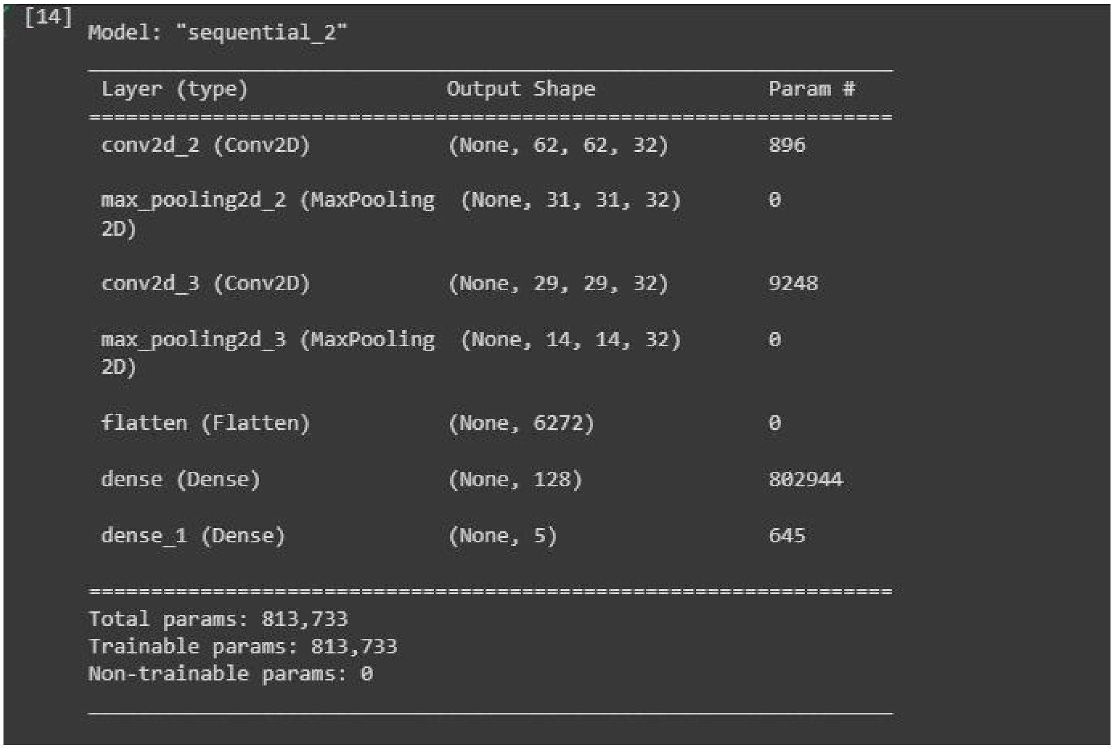
- **Initializing the model**

```
# Initializing the CNN

classifier = Sequential()

# First convolution layer and pooling

classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Second convolution layer and pooling

classifier.add(Conv2D(32, (3, 3), activation='relu'))

# input_shape is going to be the pooled feature maps from the previous convolution layer

classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Flattening the layers

classifier.add(Flatten())

# Adding a fully connected layer

classifier.add(Dense(units=128, activation='relu'))

classifier.add(Dense(units=5, activation='softmax')) # softmax for more than 2
```



```
[13] # Initializing the CNN
     classifier = Sequential()

     # First convolution layer and pooling
     classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
     classifier.add(MaxPooling2D(pool_size=(2, 2)))

     # Second convolution layer and pooling
     classifier.add(Conv2D(32, (3, 3), activation='relu'))

     # input_shape is going to be the pooled feature maps from the previous convolution layer
     classifier.add(MaxPooling2D(pool_size=(2, 2)))

     # Flattening the layers
     classifier.add(Flatten())

     # Adding a fully connected layer
     classifier.add(Dense(units=128, activation='relu'))
     classifier.add(Dense(units=5, activation='softmax')) # softmax for more than 2
```

```
classifier.summary()#summary of our model
```



```
[14]
Model: "sequential_2"
_________________________________________________________________
 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)

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 128)               802944

 dense_1 (Dense)             (None, 5)                 645

=================================================================
Total params: 813,733
Trainable params: 813,733
Non-trainable params: 0
_________________________________________________________________
```

7. Configure the learning process

With both the training data defined and model defined, it's time configure the learning process. This is accomplished with a call to the compile() method of the Sequential model class. Compilation requires 3 arguments: an optimizer, a loss function, and a list of metrics.

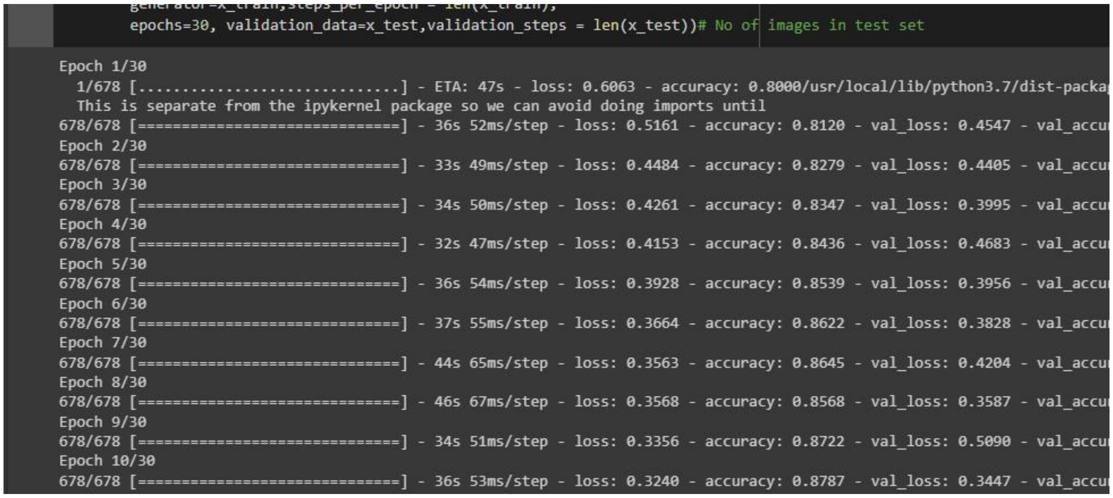
In our example, set up as a multi-class classification problem, we will use the Adam optimizer, the categorical cross entropy loss function, and include solely the accuracy metric.

```
   # Compiling the CNN

   # categorical_crossentropy for more than 2

   classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
```

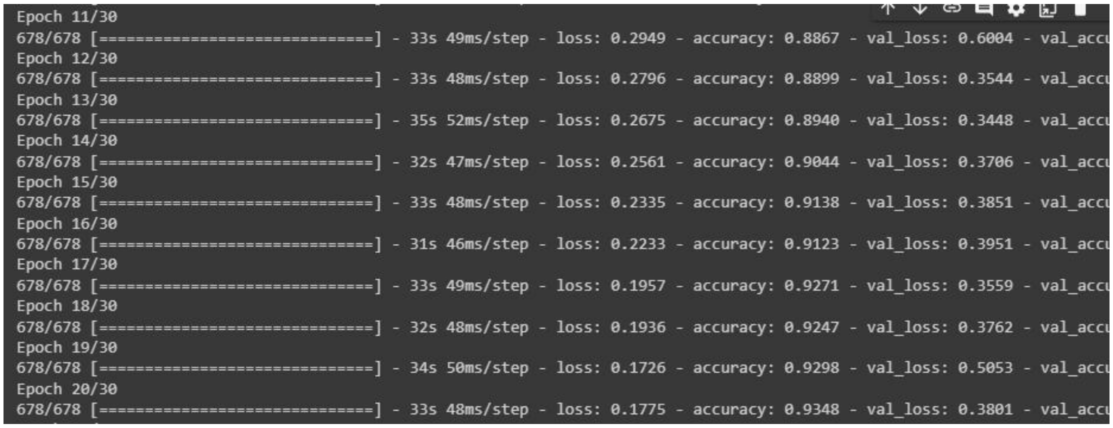
8. Train The Model

At this point we have training data and a fully configured neural network to train with said data. All that is left is to pass the data to the model for the training process to commence, a process which is completed by iterating on the training data. Training begins by calling the fit() method.

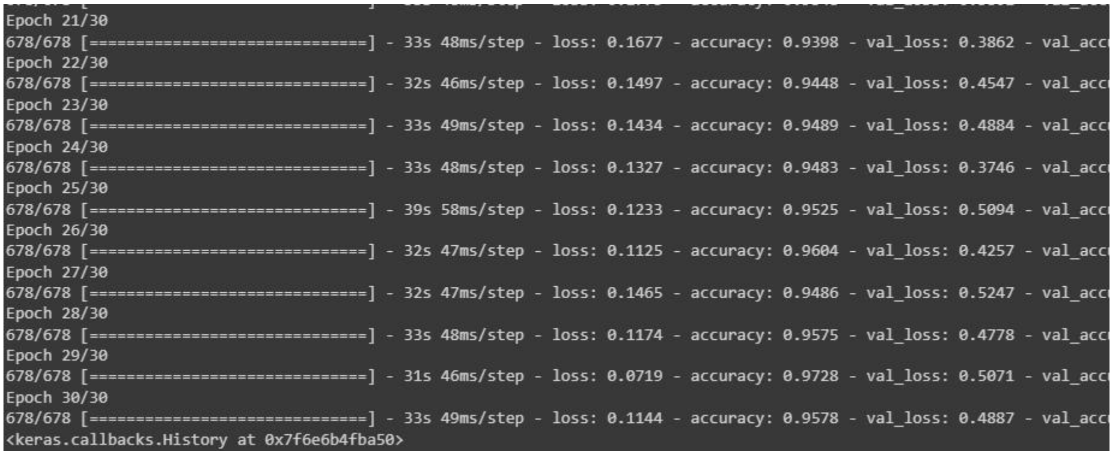
```
>>>

classifier.fit_generator(

  generator=x_train,steps_per_epoch = len(x_train),

epochs=30,validation_data=x_test,validation_steps=len(x_test))
```



```
        generator=x_train,steps_per_epoch = len(x_train),
        epochs=30, validation_data=x_test,validation_steps = len(x_test))# No of images in test set

Epoch 1/30
  1/678 [..............................] - ETA: 47s - loss: 0.6063 - accuracy: 0.8000/usr/local/lib/python3.7/dist-packa
  This is separate from the ipykernel package so we can avoid doing imports until
678/678 [==============================] - 36s 52ms/step - loss: 0.5161 - accuracy: 0.8120 - val_loss: 0.4547 - val_accu
Epoch 2/30
678/678 [==============================] - 33s 49ms/step - loss: 0.4484 - accuracy: 0.8279 - val_loss: 0.4405 - val_accu
Epoch 3/30
678/678 [==============================] - 34s 50ms/step - loss: 0.4261 - accuracy: 0.8347 - val_loss: 0.3995 - val_accu
Epoch 4/30
678/678 [==============================] - 32s 47ms/step - loss: 0.4153 - accuracy: 0.8436 - val_loss: 0.4683 - val_accu
Epoch 5/30
678/678 [==============================] - 36s 54ms/step - loss: 0.3928 - accuracy: 0.8539 - val_loss: 0.3956 - val_accu
Epoch 6/30
678/678 [==============================] - 37s 55ms/step - loss: 0.3664 - accuracy: 0.8622 - val_loss: 0.3828 - val_accu
Epoch 7/30
678/678 [==============================] - 44s 65ms/step - loss: 0.3563 - accuracy: 0.8645 - val_loss: 0.4204 - val_accu
Epoch 8/30
678/678 [==============================] - 46s 67ms/step - loss: 0.3568 - accuracy: 0.8568 - val_loss: 0.3587 - val_accu
Epoch 9/30
678/678 [==============================] - 34s 51ms/step - loss: 0.3356 - accuracy: 0.8722 - val_loss: 0.5090 - val_accu
Epoch 10/30
678/678 [==============================] - 36s 53ms/step - loss: 0.3240 - accuracy: 0.8787 - val_loss: 0.3447 - val_accu
```



```
Epoch 11/30
678/678 [==============================] - 33s 49ms/step - loss: 0.2949 - accuracy: 0.8867 - val_loss: 0.6004 - val_accu
Epoch 12/30
678/678 [==============================] - 33s 48ms/step - loss: 0.2796 - accuracy: 0.8899 - val_loss: 0.3544 - val_accu
Epoch 13/30
678/678 [==============================] - 35s 52ms/step - loss: 0.2675 - accuracy: 0.8940 - val_loss: 0.3448 - val_accu
Epoch 14/30
678/678 [==============================] - 32s 47ms/step - loss: 0.2561 - accuracy: 0.9044 - val_loss: 0.3706 - val_accu
Epoch 15/30
678/678 [==============================] - 33s 48ms/step - loss: 0.2335 - accuracy: 0.9138 - val_loss: 0.3851 - val_accu
Epoch 16/30
678/678 [==============================] - 31s 46ms/step - loss: 0.2233 - accuracy: 0.9123 - val_loss: 0.3951 - val_accu
Epoch 17/30
678/678 [==============================] - 33s 49ms/step - loss: 0.1957 - accuracy: 0.9271 - val_loss: 0.3559 - val_accu
Epoch 18/30
678/678 [==============================] - 32s 48ms/step - loss: 0.1936 - accuracy: 0.9247 - val_loss: 0.3762 - val_accu
Epoch 19/30
678/678 [==============================] - 34s 50ms/step - loss: 0.1726 - accuracy: 0.9298 - val_loss: 0.5053 - val_accu
Epoch 20/30
678/678 [==============================] - 33s 48ms/step - loss: 0.1775 - accuracy: 0.9348 - val_loss: 0.3801 - val_accu
```

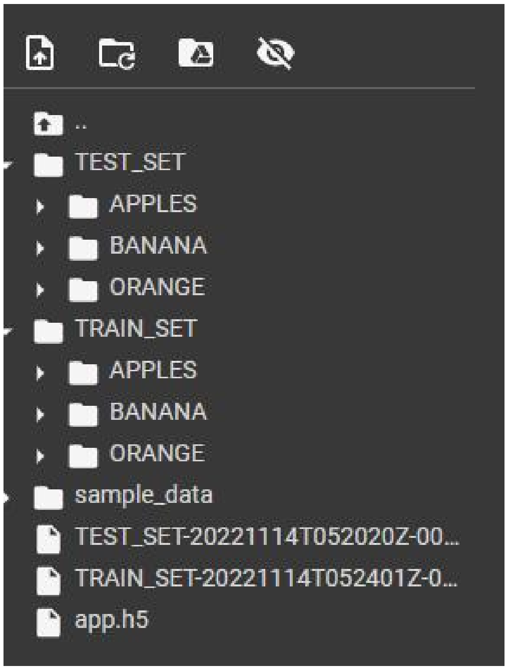


```
Epoch 21/30
678/678 [==============================] - 33s 48ms/step - loss: 0.1677 - accuracy: 0.9398 - val_loss: 0.3862 - val_acc
Epoch 22/30
678/678 [==============================] - 32s 46ms/step - loss: 0.1497 - accuracy: 0.9448 - val_loss: 0.4547 - val_acc
Epoch 23/30
678/678 [==============================] - 33s 49ms/step - loss: 0.1434 - accuracy: 0.9489 - val_loss: 0.4884 - val_acc
Epoch 24/30
678/678 [==============================] - 33s 48ms/step - loss: 0.1327 - accuracy: 0.9483 - val_loss: 0.3746 - val_acc
Epoch 25/30
678/678 [==============================] - 39s 58ms/step - loss: 0.1233 - accuracy: 0.9525 - val_loss: 0.5094 - val_acc
Epoch 26/30
678/678 [==============================] - 32s 47ms/step - loss: 0.1125 - accuracy: 0.9604 - val_loss: 0.4257 - val_acc
Epoch 27/30
678/678 [==============================] - 32s 47ms/step - loss: 0.1465 - accuracy: 0.9486 - val_loss: 0.5247 - val_acc
Epoch 28/30
678/678 [==============================] - 33s 48ms/step - loss: 0.1174 - accuracy: 0.9575 - val_loss: 0.4778 - val_acc
Epoch 29/30
678/678 [==============================] - 31s 46ms/step - loss: 0.0719 - accuracy: 0.9728 - val_loss: 0.5071 - val_acc
Epoch 30/30
678/678 [==============================] - 33s 49ms/step - loss: 0.1144 - accuracy: 0.9578 - val_loss: 0.4887 - val_acc
<keras.callbacks.History at 0x7f6e6b4fba50>
```

9. Save the Model

Your model is to be saved for the future purpose. This saved model ac also be integrated with an android application or web application in order to predict something

```
>># Save the model

classifier.save('app.h5')
```

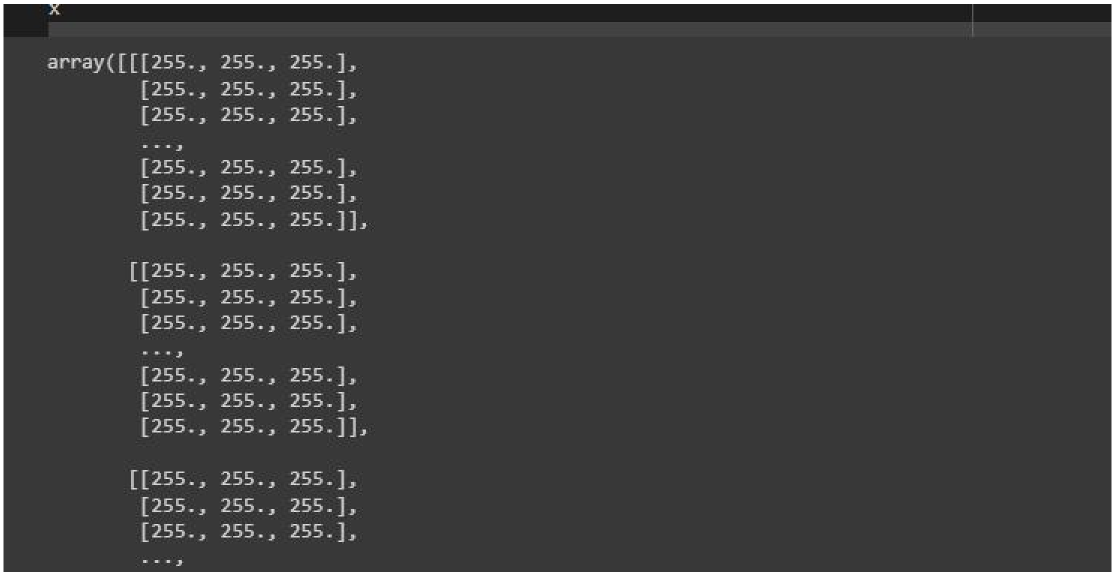


```
..
TEST_SET
    APPLES
    BANANA
    ORANGE
TRAIN_SET
    APPLES
    BANANA
    ORANGE
sample_data
TEST_SET-20221114T052020Z-00...
TRAIN_SET-20221114T052401Z-0...
app.h5
```

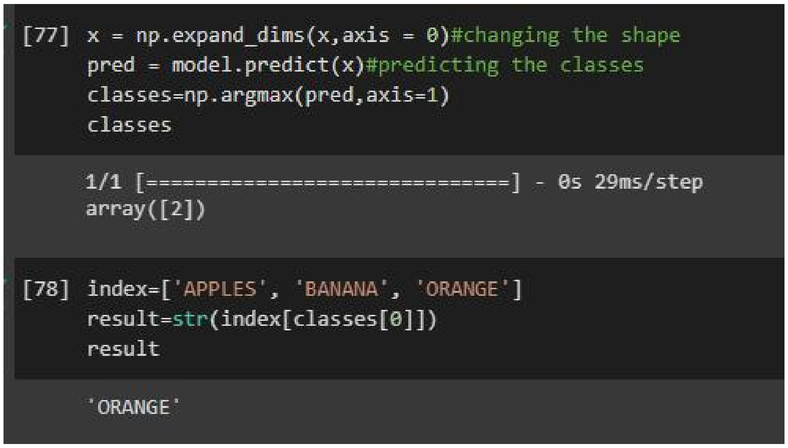
10. Predictions

The last and final step is to make use of the Saved model to make predictions. We use load model class to load the model. We use imread() class from opencv library to read an image and give it to the model to predict the result. Before giving the original image to predict the class, we have to pre-process that image and apply predictions to get accurate result.

```
>>>

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image

model = load_model("app.h5") #loading the model for testing

img=tensorflow.keras.utils.load_img("/content/TEST_SET/ORANGE/n07749192_1081.jpg",grayscale=False,target_size= (64,64))#loading of the image

x = image.img_to_array(img)#image to array

x
```



```
x

array([[[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.]],

       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.]],

       [[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
```



```
[77] x = np.expand_dims(x,axis = 0)#changing the shape
     pred = model.predict(x)#predicting the classes
     classes=np.argmax(pred,axis=1)
     classes

     1/1 [==============================] - 0s 29ms/step
     array([2])
```

```
[78] index=['APPLES', 'BANANA', 'ORANGE']
     result=str(index[classes[0]])
     result

     'ORANGE'
```

SPRINT -3

Application Building

Creating HTML Pages

Home.html

```
<html lang="en" dir="ltr">

<head>

        <meta charset="utf-8">

        <title>Nutrition Image Analysis</title>

        <link rel="shortcut icon" href="{{ url_for('static', filename='diabetes-favicon.ico') }}">

        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css') }}">

        <script src="https://kit.fontawesome.com/5f3f547070.js" crossorigin="anonymous"></script>

        <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">
```

```
    </head>



            <!-- Result -->

            <div class="results">

                              <p style="padding-top: 150px; color:blue;"><h4
style="color:blue;">Food Classified is: <h4><b><h4
style="color:red;"><u>{{showcase1}}<h4><br><h4
style="color:red;"><u>{{showcase}}<h4></p>

            </div>

            <br>

            <br>

     </div>

    </body>

</html>
```

HOME.html

```
<!DOCTYPE html>

<html>

<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>Home</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>

    body

    {

        background-image: url("https://www.livingproofnyc.com/wp-content/themes/livingproof/assets/img/hero-background.jpg");

        background-size: cover;

    }

    .bar

    {

    margin: 0px;

    padding:20px;

    background-color:white;

    opacity:0.6;

    color:black;
```

```
font-family:'Roboto',sans-serif;

font-style: italic;

border-radius:20px;

font-size:25px;

}

h3

{

margin: 0px;

padding:20px;

background-color:#9ACD32;

width: 800px;

opacity:0.6;

color:#000000;

font-family:'Roboto',sans-serif;

font-style: italic;

border-radius:20px;

font-size:25px;

}

a

{
```

```
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:white;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}
.div1{
    background-color: lightgrey;
    width: 500px;
    border: 10px solid peach;
    padding: 20px;
    margin: 20px;
    height: 500px;
```

```
}

.header {   position: relative;

                top:0;

                margin:0px;

                z-index: 1;

                left: 0px;

                right: 0px;

                position: fixed;

                background-color: #8B008B ;

                color: white;

                box-shadow: 0px 8px 4px grey;

                overflow: hidden;

                padding-left:20px;

                font-family: 'Josefin Sans'

                font-size: 2vw;

                width: 100%;

                height:8%;

                text-align: center;

            }

            .topnav {
```

```
    overflow: hidden;

    background-color: #FCAD98;

}

.topnav-right a {

    float: left;

    color: black;

    text-align: center;

    padding: 14px 16px;

    text-decoration: none;

    font-size: 22px;

}

.topnav-right a:hover {

    background-color: #FF69B4;

    color: black;

}

.topnav-right a.active {

    background-color: #DA70D6;

    color: black;

}

.topnav-right {
  float: right;
  padding-right:100px;
}
</style>
</head>
<body>
<!--Brian Tracy-->
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:black; padding-top:1%;padding-left:5%;">Nutrtion Image Analysis</div>
  <div class="topnav-right"style="padding-top:0.5%;">
    <a class="active" href="{{ url_for('home')}}">Home</a>
    <a href="{{ url_for('image1')}}">Classify</a>
  </div>
</div>
</div>
<h1>
<center>
<h3>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
```

```
maintain 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. It ensures compliance with trade and food laws.</h3>

</center>

  </h1>

</body>

</html>
```

Image.html

```
<!DOCTYPE html>

<html lang="en">

<head>

      <meta charset="UTF-8">

      <title>Title</title>

</head>

<body>

{% extends "imageprediction.html" %} {% block content %}

<div style="float:left">

      <br>

      <br>

      <h5>
```

```
    <font color="black" size="3" font-family="sans-serif"><b>Upload image to classify</b></font>
</h5>
<div>
    <form id="upload-file" method="post" enctype="multipart/form-data">
        <label for="imageUpload" class="upload-label">
            Choose...
        </label>
        <input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">
    </form>
    <center>
        <div class="image-section" style="display:none;">
            <div class="img-preview">
                <div id="imagePreview">
                </div>
    </center>
</div>
<center>
    <div>
        <button type="button" class="btn btn-primary btn-lg " id="btn-predict">Classify</button>
        </div>
    </center>
</div>
<div class="loader" style="display:none;margin-left: 450px;"></div>
<h3 id="result">
    <span>
        <p style="padding-top: 25px;">
        <h4>Food Classified is : <h4><b><u>
                        {{showcase}}{{showcase1}}
                        </p>
    </span>
</h3>
</div>
</div>
{% endblock %}
</body>
</html>
```

ImagePrediction.html

```
<!DOCTYPE html>
<html>
```

```
    <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>Predict</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>
    body
    {
        background-image:
url("https://i.pinimg.com/originals/be/21/1a/be211ad5043a8d05757a3538bdd8f450.jpg");
        background-size: cover;
    }
    .bar
    {
    margin: 0px;
```

```
padding:20px;

background-color:white;

opacity:0.6;

color:black;

font-family:'Roboto',sans-serif;

font-style: italic;

border-radius:20px;

font-size:25px;

}

a

{

color:grey;

float:right;

text-decoration:none;

font-style:normal;

padding-right:20px;

}

a:hover{

background-color:black;

color:white;
```

```
border-radius:15px;0

font-size:30px;

padding-left:10px;

}

.div1{

   background-color: lightgrey;

   width: 500px;

   border: 10px solid peach;

   padding: 20px;

   margin: 20px;

   height: 500px;

}

.header {   position: relative;

                        top:0;

                        margin:0px;

                        z-index: 1;

                        left: 0px;

                        right: 0px;

                        position: fixed;

                        background-color: #8B008B ;
                        color: white;

                        box-shadow: 0px 8px 4px grey;

                        overflow: hidden;

                        padding-left:20px;

                        font-family: 'Josefin Sans';

                        font-size: 2vw;

                        width: 100%;

                        height:8%;

                        text-align: center;

                }

                .topnav {

   overflow: hidden;

   background-color: #FCAD98;

}


.topnav-right a {

   float: left;

   color: black;

   text-align: center;

   padding: 14px 16px;
```

```
    text-decoration: none;
    font-size: 18px;
}

.topnav-right a:hover {
    background-color: #FF69B4;
    color: black;
}

.topnav-right a.active {
    background-color: #DA70D6;
    color: black;
}

.topnav-right {
    float: right;
    padding-right:100px;
}
</style>
</head>
```

```
<body>

<div class="header">

<div style="width:50%;float:left;font-size:2vw;text-align:left;color:black; padding-top:1%;padding-left:5%;">Nutrtion Image Analysis</div>

  <div class="topnav-right"style="padding-top:0.5%;">

    <a href="{{ url_for('home')}}">Home</a>

    <a class="active" href="{{ url_for('image1')}}">Classify</a>

  </div>

</div>

<br>

</div>

<div class="container">

        <center>

<div id="content" style="margin-top:2em">{% block content %}{% endblock %}</div></center>

    </div>

</body>

<footer>

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

</footer>
```

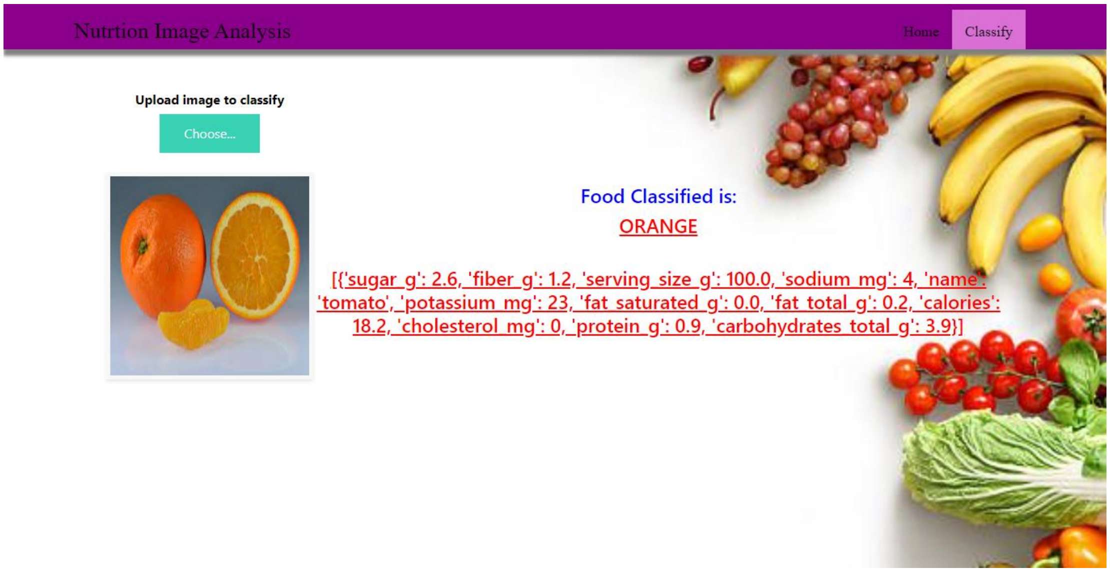
```
</html>
```

Routing to Html Page

App.py

```
from flask import Flask, render_template, request

import os

import numpy as np

import tensorflow

from keras.models import load_model

from keras.preprocessing import image

app = Flask(__name__, template_folder="template")

model = load_model('nutrition.h5')

print("Loaded model from disk")

@ app.route('/')

def home():

        return render_template('home.html')

@ app.route('/image1', methods=['GET', 'POST'])

def image1():

        return render_template("image.html")

@ app.route('/predict', methods=['GET', 'POST'])

def lanuch():
    if request.method == 'POST':

        f = request.files['file']

        basepath = os.path.dirname('__file__')

        filepath = os.path.join(basepath, "uploads", f.filename)

        f.save(filepath)

        img = tensorflow._keras.utils.load_img(filepath, target_size=(64, 64))

        x=image.image_utils.img_to_array(img)

        x = np.expand_dims(x, axis=0)

        pred = np.argmax(model.predict(x), axis=1)

        print("prediction", pred)

        index = ['APPLE', 'BANANA', 'ORANGE', 'PINEAPPLE', '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
```

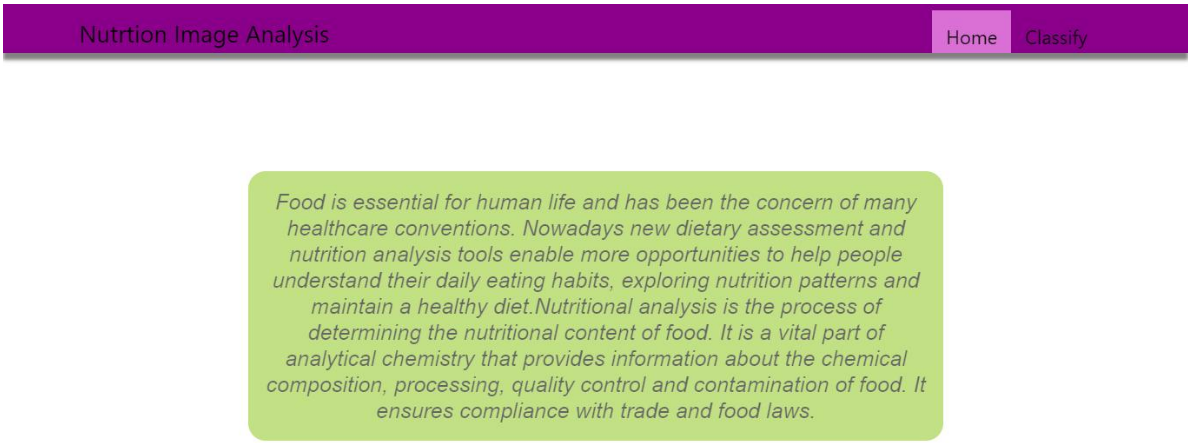
```
        url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"

        querystring = {"query": "tomato"}

        headers = {

            "X-RapidAPI-Key": "e6843bfa04msh49abdb9bbcf3420p1a9e13jsn06dc99b07347",

            "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__":

app.run(debug=TRUE)
```



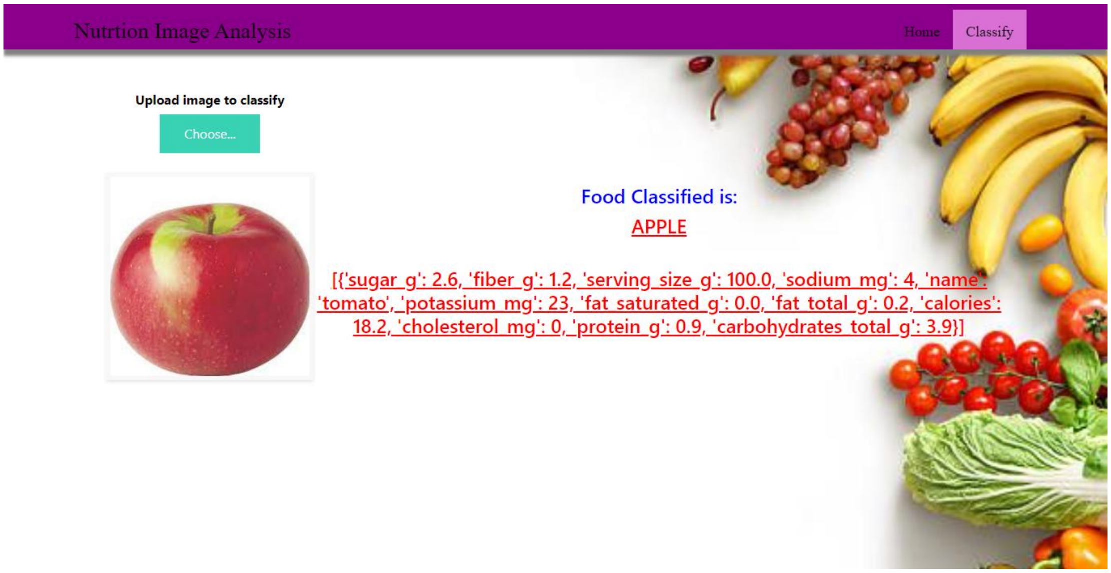
Food Classified is:

ORANGE

```
[{'sugar_g': 2.6, 'fiber_g': 1.2, 'serving_size_g': 100.0, 'sodium_mg': 4, 'name':
'tomato', 'potassium_mg': 23, 'fat_saturated_g': 0.0, 'fat_total_g': 0.2, 'calories':
18.2, 'cholesterol_mg': 0, 'protein_g': 0.9, 'carbohydrates_total_g': 3.9}]
```



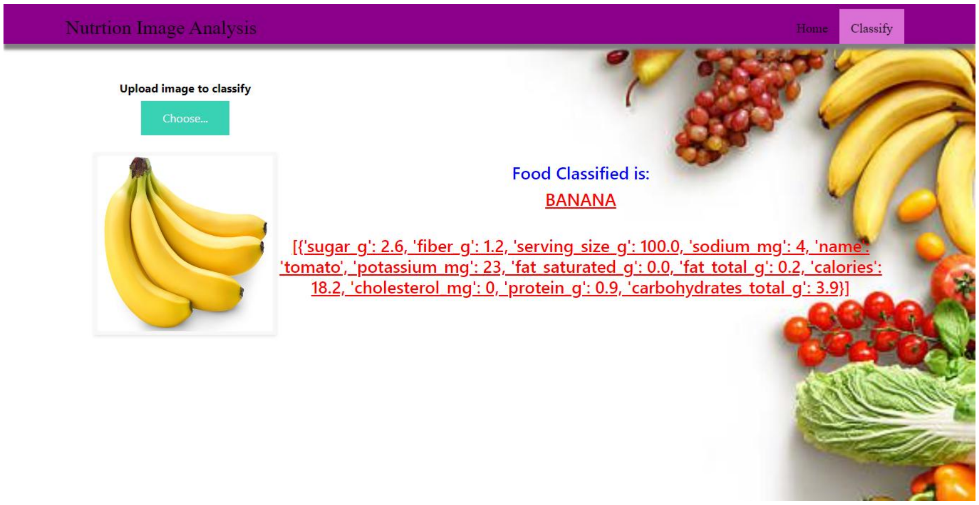
Food is essential for human life and has been the concern of many healthcare conventions. Nowadays new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintain a healthy diet.Nutritional analysis 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. It ensures compliance with trade and food laws.



Food Classified is:

APPLE

```
[{'sugar_g': 2.6, 'fiber_g': 1.2, 'serving_size_g': 100.0, 'sodium_mg': 4, 'name':
'tomato', 'potassium_mg': 23, 'fat_saturated_g': 0.0, 'fat_total_g': 0.2, 'calories':
18.2, 'cholesterol_mg': 0, 'protein_g': 0.9, 'carbohydrates_total_g': 3.9}]
```



Food Classified is:

BANANA

```
[{'sugar_g': 2.6, 'fiber_g': 1.2, 'serving_size_g': 100.0, 'sodium_mg': 4, 'name':
'tomato', 'potassium_mg': 23, 'fat_saturated_g': 0.0, 'fat_total_g': 0.2, 'calories':
18.2, 'cholesterol_mg': 0, 'protein_g': 0.9, 'carbohydrates_total_g': 3.9}]
```

SPRINT 4

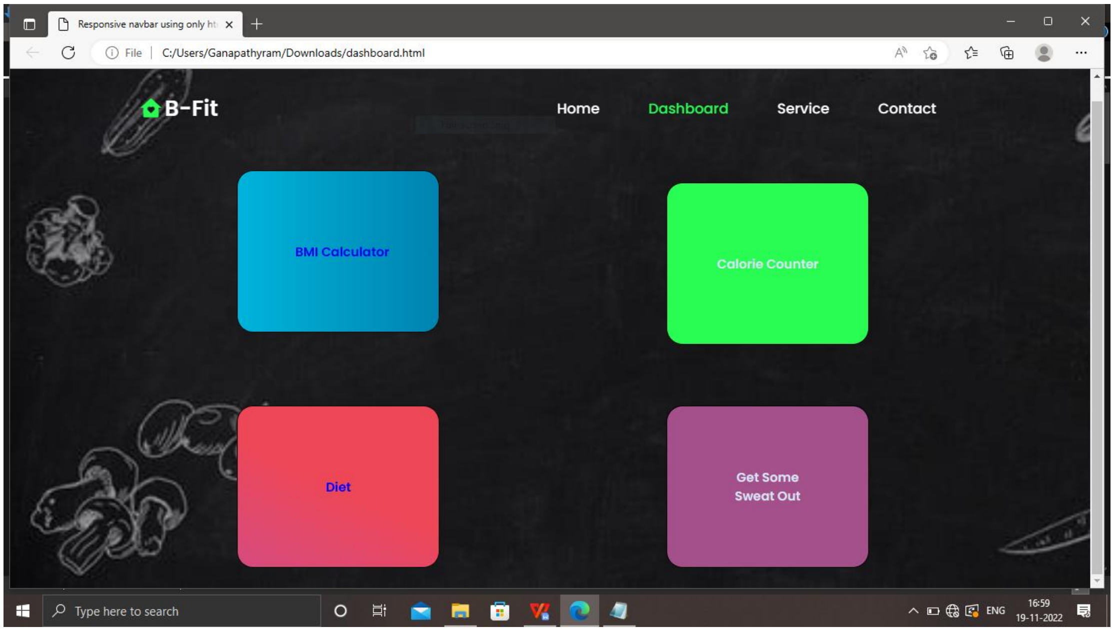
REQUIREMENTS IN THE WEB APPLICATION

DASH BOARD PAGE

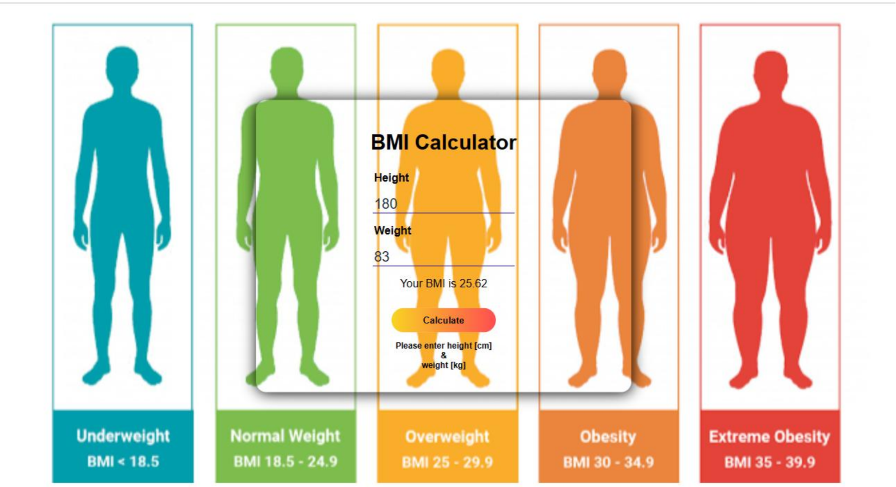
Dashboard page which consists of four different navigation the user can

- **Calculate their BMI value**
- **Know the calorie value in fruits and vegetables**
- **Get a diet plan**
- **Workout plans**

Dash board snap



BMi calculator



BMI Calculator

Height

180

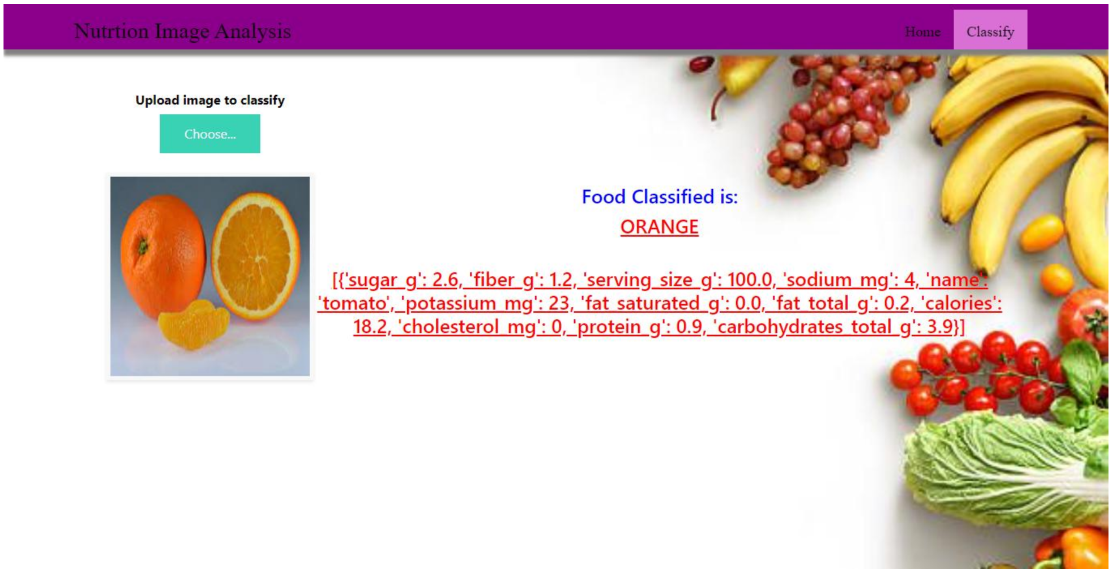
Weight

83

Your BMI is 25.62

Please enter height [cm]
&
weight [kg]

Calorie counter



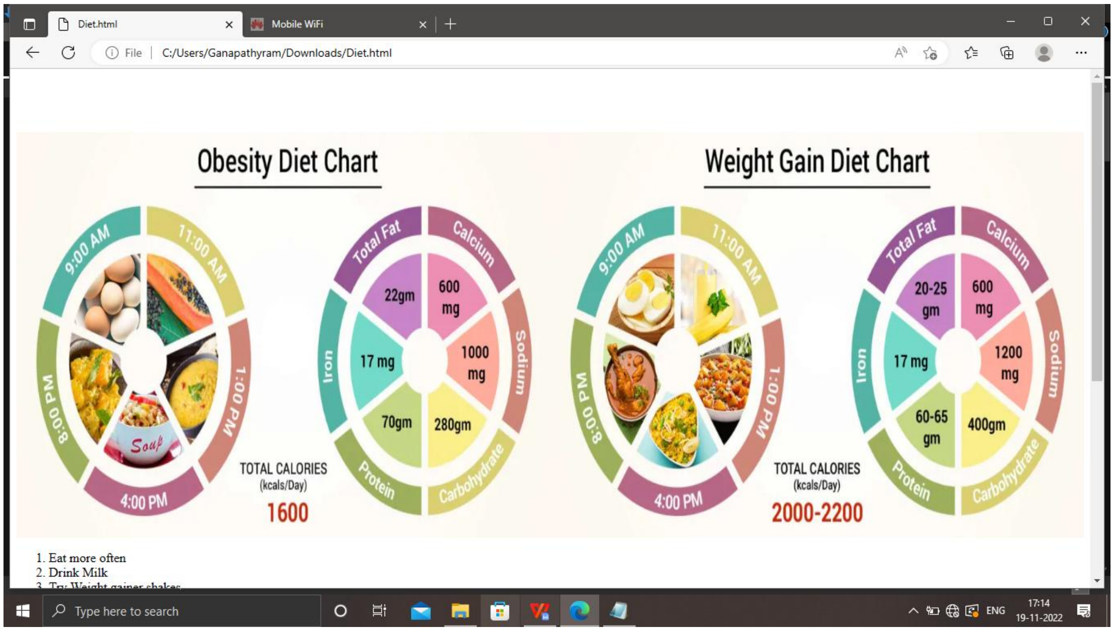
Upload image to classify

Food Classified is:

ORANGE

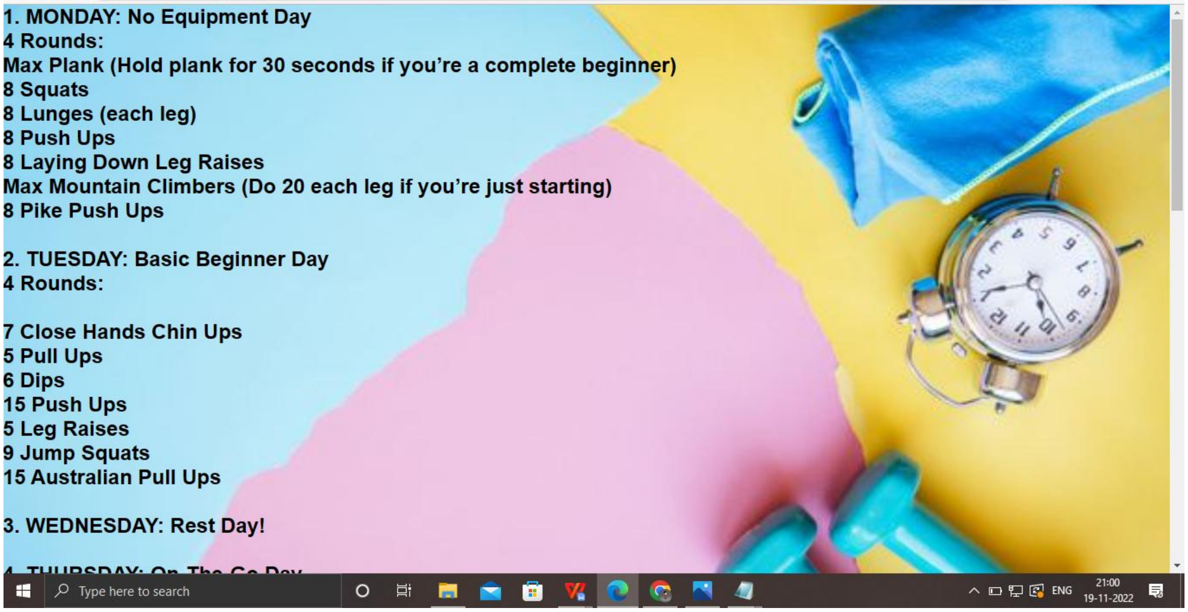
[{'sugar_g': 2.6, 'fiber_g': 1.2, 'serving_size_g': 100.0, 'sodium_mg': 4, 'name': 'tomato', 'potassium_mg': 23, 'fat_saturated_g': 0.0, 'fat_total_g': 0.2, 'calories': 18.2, 'cholesterol_mg': 0, 'protein_g': 0.9, 'carbohydrates_total_g': 3.9}]

Get a diet plan



1. Eat more often
2. Drink Milk
3. Try Weight gainer shakes

Work out plan



1. MONDAY: No Equipment Day
4 Rounds:
Max Plank (Hold plank for 30 seconds if you're a complete beginner)
8 Squats
8 Lunges (each leg)
8 Push Ups
8 Laying Down Leg Raises
Max Mountain Climbers (Do 20 each leg if you're just starting)
8 Pike Push Ups

2. TUESDAY: Basic Beginner Day
4 Rounds:

7 Close Hands Chin Ups
5 Pull Ups
6 Dips
15 Push Ups
5 Leg Raises
9 Jump Squats
15 Australian Pull Ups

3. WEDNESDAY: Rest Day!

4. THURSDAY: On The Go Day

GITHUB & PROJECT DEMO LINK

GITHUB LINK:

AI-POWERED NUTRITION AND FITNESS ENTHUSIASTS

DEMO

Nutrition video