AI-powered Nutrition Analyzer for Fitness Enthusiasts

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Food is essential for human life and has been the concern of many healthcare conventions. Nowadays new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and 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.

1.2 PURPOSE

The main purpose of the project is to building a model which is used for classifying the fruit depends on the different characteristics like color, shape, texture etc. Here the user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fiber, Protein, Calories, etc.).

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

1. To recognize multiple fruits more accurately, the authors **(Pure-CNN: A Framework for Fruit Images Classification)** proposed a Pure Convolutional Neural Network (PCNN) with minimum number of parameters. The PCNN consists of 7 convolutional layers. Additionally, to reduce overfitting and taking average of whole feature maps we employed recently developed Global Average Pooling (GAP) layer that is verified to be very effective. They analysed classification performance using PCNN on recently introduced fruit-360 dataset. The experimental results of the 55244 color fruit images from the 81 categories, show that the PCNN achieved a classification accuracy of 98.88%. The paper presents a new approach to improve fruit image classification using PCNN with Global Average Pooling (GAP).

The highest classification accuracy of 98.88% was obtained when using PCNN with GAP layer. Thus this method can be used for both object recognition and multi-class image classification.

2. In paper **(Fruit Classification using Convolutional Neural Network via Adjust Parameter and Data Enhancement)** the author proposed a method of fruit automatic recognition and classification based on CNN. The paper used 2 data set, one is two color fruit image data set (public data set and the other is self- made data set).

The methods used are preprocessing and Classification Network. They used deep learning module in halcon software. They used the pre-training network built in halcon software to classify and identify fruit images. The CNN used in this paper does not give specific network structure.

According to the experimental results, the method proposed in this paper provides an effective method for automatic recognition and classification of fruit images.

After parameter adjustment, and achieved the highest average classification accuracy of 99.8% on the public data set. In the self-made data set, the classification accuracy is 90.2%.

3. **Improving the Prediction of Rotten Fruit Using Convolutional Neural Network.** This research developed two models for fruit recognition and predicting the freshness or spoilage of fruit using the CNN approach based on VGG16 architecture.

The first model was developed from the RGB images dataset, while the second model has developed from the concatenated images dataset, including RGB image, LoG image, grayscale without background image, and HSV with AGT image. This model gave the validation accuracy of 89.97% and the validation loss of 4.98% during the validation processing.

4. **DeepFood: Food Image Analysis and Dietary Assessment via Deep Model by LANDU JIANG1,2, BOJIA QIU2, XUE LIU2, CHENXI HUANG1,**

AND KUNHUI LIN1 on 2016. The model developed had a three-step algorithm to recognise food and then to create a dietary plan using the dataset available. They used the fat, calories, carbohydrate and proteins as its primary factor to create a dietary plan, which was obtained from the model which can identify the food details. They used Convolutional Neural Network (CNN), Region Proposal Network (RPN).

Dataset used by them were UEC-FOOD100, UEC-FOOD256.

5. **Personalized Classifier for Food Image Recognition by Shota Horiguchi, Sosuke Amano, Makoto Ogawa, and Kiyoharu Aizawa in 2015.** They used a method of incremental learning to get output more personalized to the end user, to increase the accuracy. So each user had a personalized prediction from a personalised model which was trained dynamically with the sample input dataset obtained from them by using a food logging application. They used Convolutional Neural Network (CNN).

6. **Very deep convolutional networks for large-scale image recognition.** In this paper, Author proposed about investigating the effect of the convolution network depth with enormous accuracy. their main contribution is that increasing cn depth using convolution filter (3X3) architecture. That can be achieved by pushing the depth to 16-19 weight layer the Basics of our ImageNet challenge 2014 submission where their team secure first place In training of convNet configuration, their input to convNETS is fixed size 224x224 rgb image (converted as square image) that multiplied by 3x3 convolution matrix

In this work, they evaluated CNN for large scale image classification.

It is beneficial for the classification accuracy. They showed that their model generalized well to wide range of task and datasets matching or outperforming more complex recognition pipeline built around less deep image representations. Their results yet again confirm the importance of depth CNN.

7. **Dropout: A Simple Way to Prevent Neural Network from Overfitting,** the authors proposed a method called Dropout, a simple way to prevent overfitting in neural networks. The key idea is to randomly drop units along with their connections from the neural network during training. This prevents units from co-adapting too much.

The central idea behind the method is to take a large model that overfits easily and repeatedly sample nad train smaller sub-models from it. They implemented this method to variety of application domains including object classification, digit recognition, speech recognition and more and resulted in reducing the error and improves the models performance.

The major drawback of this method is that is increases training time. A dropout network typically takes 2-3 times longer to train than a standard neural network of the same architecture.

2.2 REFERENCES

1. Pure-CNN: A Framework for Fruit Images Classification by Asia Kausar, Mohsin Sharif, JinHyuck Park and Dong Ryeol Shin on 2018 International Conference on Computational Science and Computational Intelligence (CSCI).

Link:
https://www.researchgate.net/publication/338360652_Pure-CNN_A_Framework_for_Fruit_Images_Classification

2. Fruit Classification using Convolutional Neural Network via Adjust Parameter and Data Enhancement by Liuchen Wu, Hui Zhang, Ruibo Chen, Ruibo Chen, Junfei Yi on 12th International Conference on Advanced Computational Intelligence (ICACI).

Link: https://ieeexplore.ieee.org/document/9177518

3. Improving the Prediction of Rotten Fruit Using Convolutional Neural Network by Sumitra Nuanmeesri, Lap Poomhiran, Kunalai Ploydanai on International Journal of Engineering Trends and Technology.

Link: https://www.ijettjournal.org/Volume-69/Issue-7/IJETT-V6917P207.pdf

4. DeepFood: Food Image Analysis and Dietary Assessment via Deep Model by LANDU JIANG1,2 (Member, IEEE), BOJIA QIU2, XUE LIU2, (Fellow, IEEE), CHENXI HUANG1, AND KUNHUI LIN1

Published on:

1. School of Informatics, Xiamen University

2. School of Computer Science, McGill University, Montréal, QC H3A 2A7 CANADA Corresponding author: Xue Liu (xueliu@cs.mcgill.ca), Chenxi Huang (supermonkeyxi@xmu.edu.cn), Kunhui Lin (linkunhuixmu@163.com)

Link: https://ieeexplore.ieee.org/document/8998172

5. Personalized Classifier for Food Image Recognition by Shota Horiguchi, Member, IEEE, Sosuke Amano, Makoto Ogawa, and Kiyoharu Aizawa, Fellow, IEEE.

Published on: JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015 1.

Link: https://ieeexplore.ieee.org/document/8316919

6. Very deep convolutional networks for large-scale image recognition by Karen Simonyan and Andrew Zisserman on arXiv preprint arXiv:1409.1556, 2014.

Link: https://arxiv.org/abs/1409.1556

7. Dropout: a simple way to prevent neural networks from overfitting by Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov on The Journal of Machine Learning Research, 15(1):1929-1958, 2014

Link: https://jmlr.org/papers/v15/srivastava14a.html

2.3 PROBLEM STATEMENT DEFINITION

The main problem faced by fitness enthusiasts is tracking their daily nutrition intake which is important to stay fit. But in today's bustling society and availability of abundant resources online about fitness, tracking nutrition will become more challenging and inaccurate. Fitness enthusiasts normally follow their diet plans but they struggle tracking nutritional contents of the food. Fruits are rich in vitamins, fibers, and minerals which makes them easily digestible, but over-consumption will result in weight gain and even diabetes as fruit contains natural sugar. Fitness enthusiasts follow a diet which contains fruits, vegetables, protein rich foods and low carb foods. But tracking their nutritional contents like fiber, protein and essential nutritions will not be an easy task. Some fruits are allergic to some consumers based on their medical condition which they need to identify before

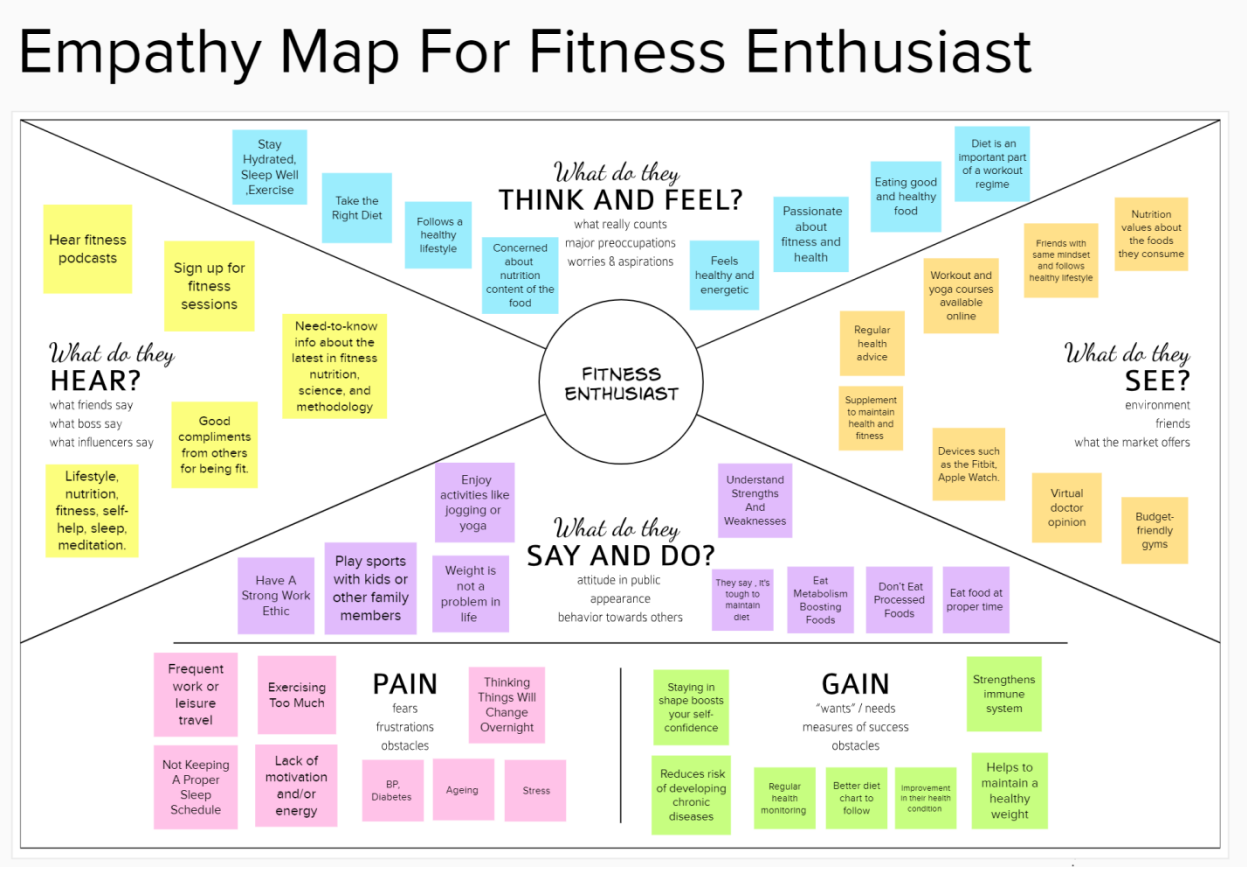
consuming. Identifying nutritional values of unknown food and fruit varieties will become impossible without online technologies as they have no prior knowledge about them.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

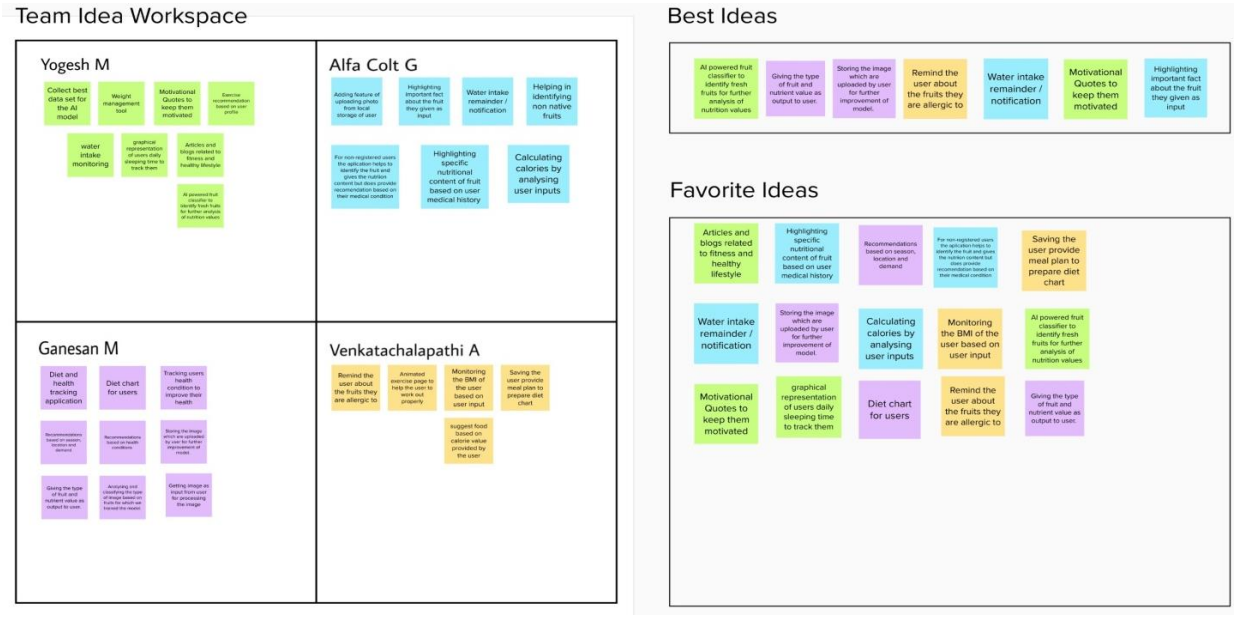
The empathy map represents a principal user and helps teams better understand their motivations, concerns, and user experience. It can be conducted with a variety of different users in mind, anywhere from stakeholders, individual use cases, or entire teams of people.



The problem statement in the empathy map is a user had already tried to know the nutrition details in the fruits but stopped. This helps that user to control is diet on foods. To rectify his problem, he searches for a solution from a different point of view. If he verifies that the solution does not affect his personal thing he moves with the solution or finds the solution which suits him.

3.2 IDEATION & BRAINSTROMING

Brainstorming is a situation where a group of people meet to generate new ideas and solutions around a specific domain of interest by removing inhibitions. People are ableto think more freely, and they suggest as many spontaneous new ideas as possible. All the ideas are noted down without criticism and after the brainstorming session the ideas are evaluated.



In Brainstorming session our team members are trying to solve the problem [How might we attract users to the website]. The new ideas are shared after the team discussion. Each idea given by the individuals is noted. According to the importance, the ideas are grouped and clustered. Ideas are prioritized which are feasible.

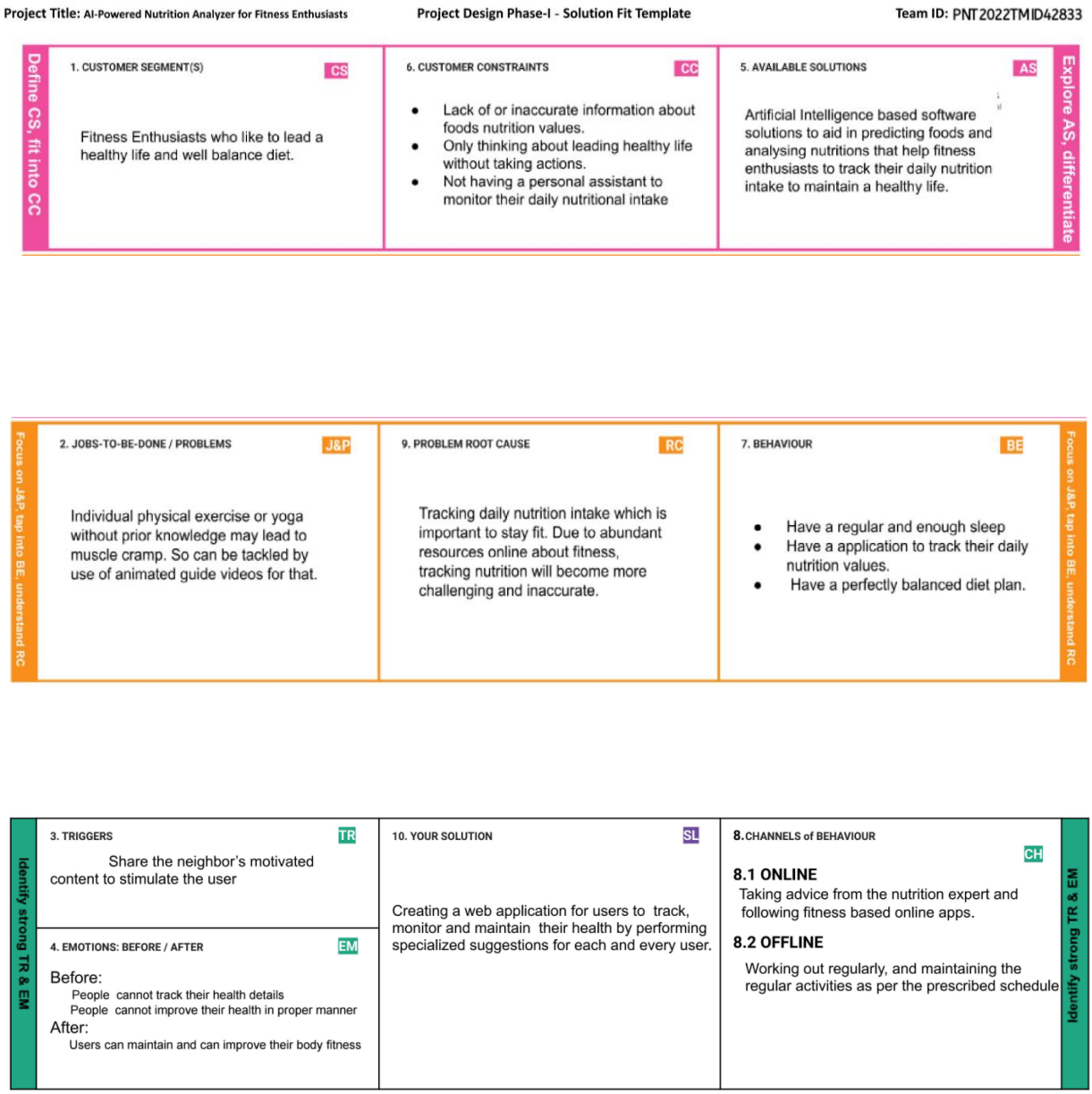
3.3 PROPOSED SOLUTION

Proposed Solution means the technical solution to be provided by the Implementation team in response to the requirements and the objectives of the Project.

S.NO	PARAMETER	DESCRIPTION
1.	**Problem Statement (Problem to be solved)**	How to intake suitable nutrition with correct guidance and weight level should be manage through tracking our day to day fitness.
2.	**Idea / Solution description**	To track fitness level and Analyze the nutrition level of foods like fruits, vegetables. It helps to identify the proportion of vitamins.
3.	**Novelty / Uniqueness**	Giving a individual Food/health Schedule according to their body conditions.
4.	**Social Impact / Customer Satisfaction**	Low expenditure, easy to follow without affecting their personal time.
5.	**Business Model (Revenue Model)**	Low expenditure, easy to follow without affecting their personal time.
6.	**Scalability of the Solution**	Notifying motivational quote's to lead a healthy routine.

3.4 PROBLEM SOLUTION FIT

Problem-Solution Fit - this occurs when you have evidence that customers care about certain jobs, pains, and gains. It helps to identify solutions with higher chances of solution adoption, reduce time spent on testing and get a better overview of the current situation.



CHAPTER 4

REQUIRMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality supposed to accomplish. Behavioral requirements describe all the cases where that uses the functional requirements, these are used in use cases.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Interaction	Interacting the user through web interface and automated voice to answer the user queries and to guide them in a proper way to maintain their fitness. In the web interface, There will be separate and special features for the registered user to get personalized and well defined advice and good practice lectures to maintain their fitness. All the registered users will be verified with either email or mobile number based on their interest in giving their information, but the verification is a must one. For non-registered users, the user can visit the website free of cost and can check the nutrient value in the fruits and vegetables, and also can view the common practices for fitness.

FR-2	User Management	Creating a group of people, who are willing to be fit in their health and making them organized in a sampe place, through which they can collaborate and also can achieve their goals with others, by encouraging each other. The application gives the ability to ask questions about a problem in the fitness groups, through which they can work effectively.
FR-3	User Satisfying	The satisfaction of each user is a must, so UI/UX should be more than enough to engage the user in the platform and the performance of the application should be optimized in order to keep every user for a long time. On an periodic interval (like once in month), we need to interact one to one with each and every user to solve the queries
FR-4	User Engagement	The user should be engaged in the application at least Once a day to get notified about the latest and good practice on fitness which is recommended by the backend model.

4.2 NON-FUNCTIONAL REQUIREMENTS

Non-Functional Requirements are the constraints or the requirements imposed on the system. They specify the quality attribute of the software. Non-Functional Requirements deal with issues like scalability, maintainability, performance, portability, security, reliability, and many more. Non-Functional Requirements address vital issues of quality for software systems.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	60% of the internet users are mobile users, and most of them are only using some common application for communicating based on the features they offer. So the application should be easily accessible by users and also it should have the ability to report an issue by the user to solve it as soon as possible.
NFR-2	Security	While logging the application , the data is encrypted and highly secured which can avoid data plagiarism . Authentication and authorization are to be done properly through the application.
NFR-3	Reliability	Application can offer you to stay focused on your diet plan. It offers to maintain your calories in your desired food. It shows quite accurately calories for the user that makes to sustain in healthy lifestyle

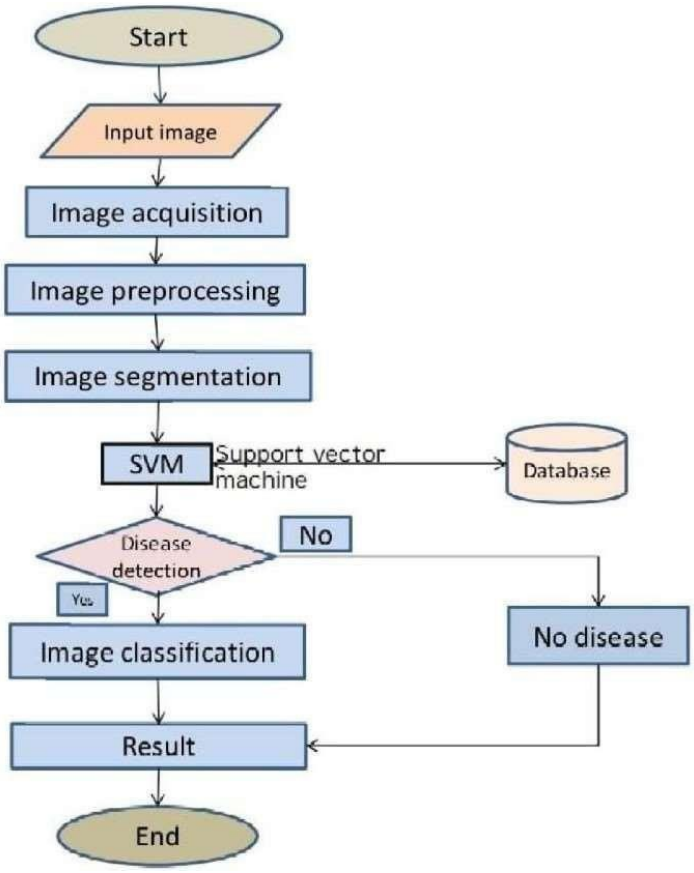
NFR-4	Performance	Performance of the application should be high enough to maintain the user in the application and also to get new users. Performance can be increased by using optimized code and also reducing the redirects and also can by DSA (Data Structures and Algorithms).
NFR-5	Availability	Even though it is a good application for registered users, it has the ability to offer minimum functionality to the non-registered users and also to increase the audience base.
NFR_5	Scalability	The application should be as much as scalable, in order to increase the number of users based on their interest.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops. Specific operations based on the data canbe represented by a flowchart.

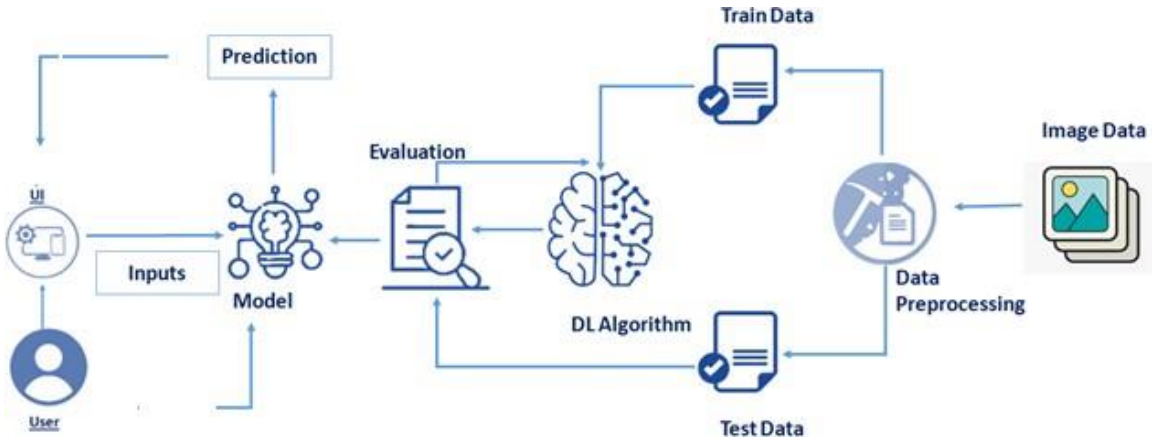


5.2 SOLUTION &TECHNICAL ARCHITECTURE

5.2.1 SOLUTION ARCHITECTURE

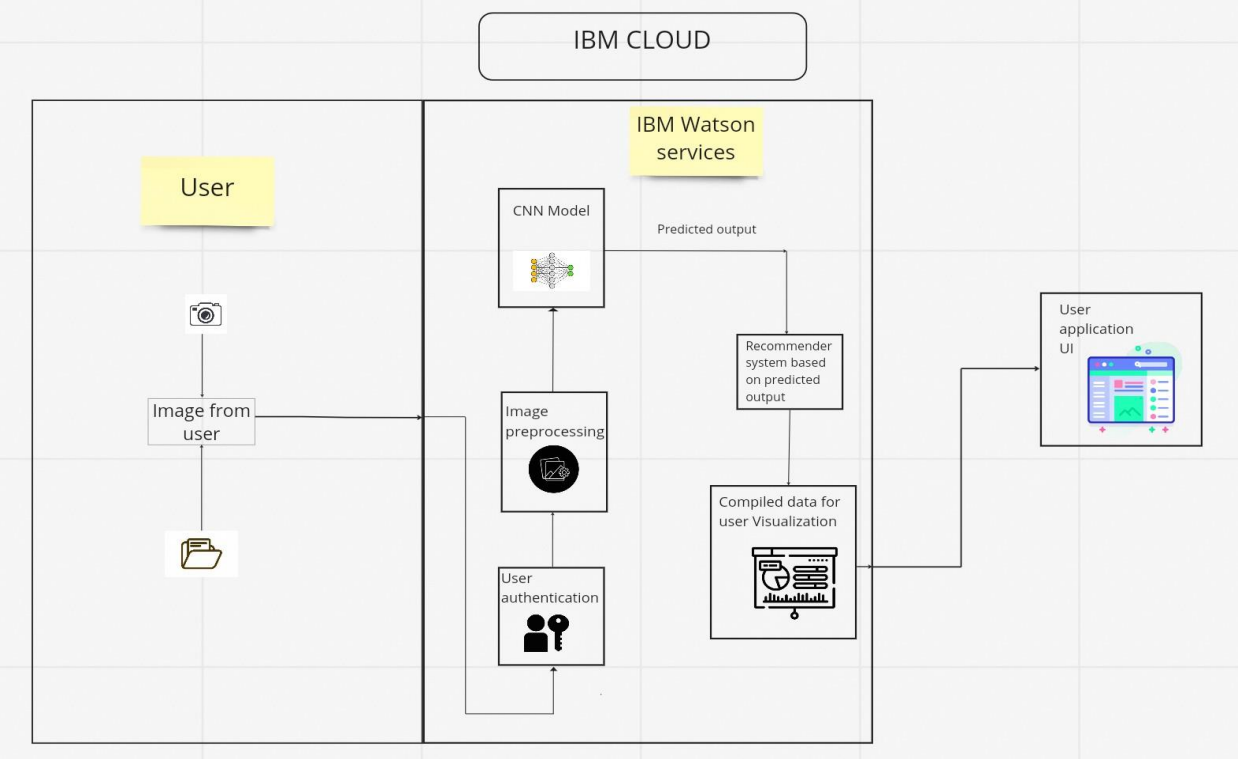
Solution architecture is the process of developing solutions based on predefined processes, guidelines and best practices with the objective that the developed solution fits within the enterprise architecture in terms of information architecture, system portfolios, integration requirements and many more.

It can then be viewed as a combination of roles, processes and documentation that are intended to address specific business needs, requirements or problems through the design and development of applications and information systems.



5.2.2 TECHNICAL ARCHITECTURE

The process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.



5.3 USER STORIES

A user story is the smallest unit of work in an agile framework. It's an end goal, not a feature, expressed from the software user's perspective. A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer.

The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer. Note that "customers" don't have to be external end users in the traditional sense, they can also be internal customers or colleagues within yourorganization who depend on your team.

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance Criteria	Priority	Release
Customer (Mobile user)	Home Page	USN-1	Fitness approval prediction	I can view /access my homepage	Low	Sprint-3
		USN-2	Informatio -n about fitness food details required for the prediction.		Low	Sprint-3
	User Register	USN-3	Enter Email ID and other	I can successfully register by	Medium	Sprint-2

			personal details for Register.	receiving mail.		
	User Login	USN-4	Uses Email ID and Password for login.	I have successfully logged in.	Medium	Sprint-2
	Fitness form	USN-5	Fitness details required for diet should be entered.	I can access the customer details form.	High	Sprint-1
	Result	USN-6	Results will be displayed.	I got my result successfully	High	Sprint-1
		USN-7	After the result displayed the necessary tips will display.	I got useful information.	Low	Sprint-4

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Use the below template to create product backlog and sprint schedule.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-2		USN-2	As a user, I will receive confirmation email once I have registered for the application.	4	High	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-1		USN-3	As a user, I can register for the application	5	Medium	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A

			through Gmail			
Sprint-1	Login	USN-4	As a user, I can log into the application byentering email & password	5	High	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-1	Dashboard	USN-5	As a user I can access the dashboard able to see options to view contents chart, select diet plans, and exercise	5	High	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-2		USN-6	As a user I can see my profile	4	Medium	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A

Sprint-3		USN-7	As a user I can update my profile	3	Low	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-2		USN-8	As a user I can change my password	4	Medium	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-1	Service Request	USN-9	As a user I can request to display nutrition content of food items	5	High	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-2		USN-10	As a user I can request to suggest a diet plan according to my medical details	4	High	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-2		USN-11	As a user I can request to suggest exercise routines according to my medical details	4	Medium	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-3	Notification	USN-12	track the status of diet targets through a	3	Low	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A

			dashboard or email services			
Sprint-3		USN-13	As a user get an email about revised exercise routines based on recent records.	3	Medium	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-1		USN-14	A user noticed after successfully achieved the target workout	5	High	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-3		USN-15	Upload Progress Reports	3	Low	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-4		USN-16	Making UI more interactive	2	Low	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A
Sprint-2		USN-17	As a user I give feedback	4	High	Yogesh M, Alfa Colt G, Ganesan K, Venkatachalapathi A

6.2 SPRINT DELIVERY SCHEDULE

Project Tracker, Velocity & Burn down Chart:

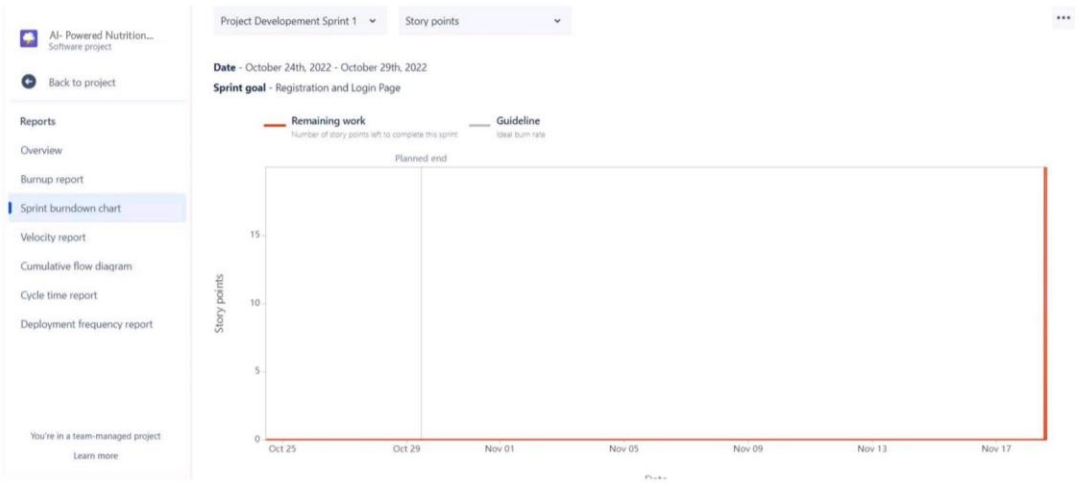
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022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

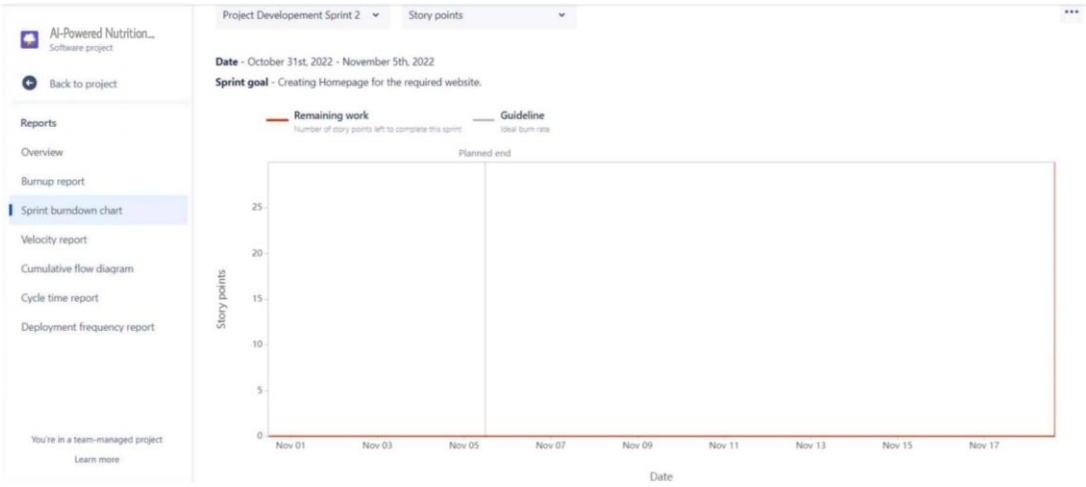
Velocity:

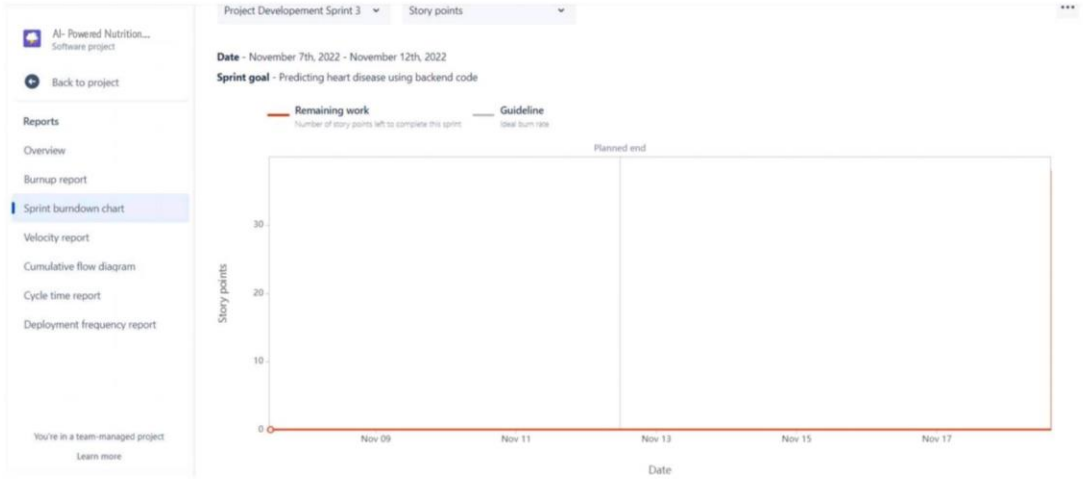
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day).

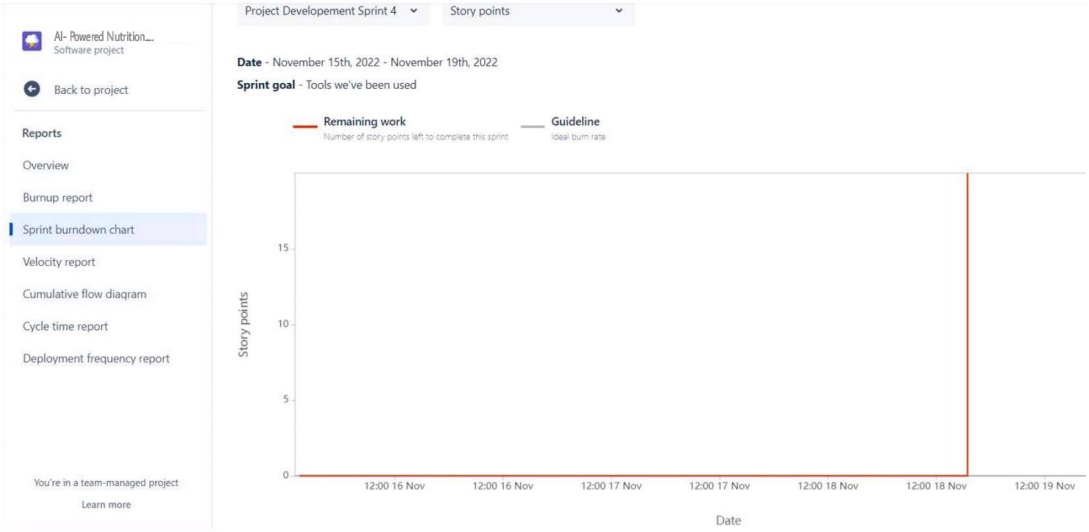
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 REPORTS FROM JIRA






AI- Powered Nutrition...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Project Developement Sprint 3
Story points
Date - November 7th, 2022 - November 12th, 2022
Sprint goal - Predicting heart disease using backend code
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Planned end
Story points
30
20
10
0
Nov 09
Nov 11
Nov 13
Nov 15
Nov 17
Date


AI- Powered Nutrition...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Project Developement Sprint 4
Story points
Date - November 15th, 2022 - November 19th, 2022
Sprint goal - Tools we've been used
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Story points
15
10
5
0
12:00 16 Nov
12:00 16 Nov
12:00 17 Nov
12:00 17 Nov
12:00 18 Nov
12:00 18 Nov
12:00 19 Nov
Date

CHAPTER 7

CODING & SOLUTIONING

7.1 FEATURE 1

```
from flask import Flask,render_template,request,redirect,g,request,session,url_for
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
import tensorflow as tf
import os
import numpy as np #used for numerical analysis
from tensorflow.keras.models import load_model#to load our trained model
from tensorflow.keras.preprocessing import image
import requests
from PIL import Image

app = Flask(_name_,template_folder="templates") # initializing a flask app
# Loading the model
model=load_model("D:\AI nutrition analyzer\Flask\saved_model.h5")
print("Loaded model from disk")

class User:
    def _init_(self, id, username, password):
        self.id = id
        self.username = username
        self.password = password

    def _repr_(self):
        return f'<User: {self.username}>'

users = []
users.append(User(id=1, username='Yogesh', password='yogesh123'))
users.append(User(id=2, username='Ganesh', password='ganesh123'))
users.append(User(id=3, username='venkat', password='venkat'))
```

```
app = Flask(_name_)
app.secret_key = 'somesecretkeythatonlyishouldknow'

@app.before_request
def before_request():
    g.user = None

    if 'user_id' in session:
        user = [x for x in users if x.id == session['user_id']][0]
        g.user = user


@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        session.pop('user_id', None)

        username = request.form['username']
        password = request.form['password']

        user = [x for x in users if x.username == username][0]
        if user and user.password == password:
            session['user_id'] = user.id
            return redirect(url_for('home'))

        return redirect(url_for('login'))

    return render_template('login.html')

@app.route('/home')# route to display the home page
def home():
    if not g.user:
        return redirect(url_for('login'))

    return render_template('home.html')#rendering  the home page

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

```
@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
def launch():
    if request.method=='POST':
        f=request.files['file'] #requesting the file
        basepath=os.path.dirname('_file_')#storing the file directory
        filepath=os.path.join(basepath,r"C:\Users\alfac\OneDrive\Desktop\AI nutrition analyzer\Flask\uploads",f.filename)#storing the file in uploads folder
        f.save(filepath)#saving the file

        img=image.load_img(filepath,target_size=(64,64)) #load and reshaping the image
        x=image.img_to_array(img)#converting image to an array
        x=np.expand_dims(x,axis=0)#changing the dimensions of the image_init_

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

        result=str(index[pred[0]])

        x=result
        print(x)
        result=nutrition(result)
        print(result)

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


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

    querystring = {"query":index}

    headers = {
        'x-rapidapi-key': 'f162f31fa2mshea65fd3f1d29094p16dba6jsn50c59cc02a64',
        'x-rapidapi-host': 'calorieninjas.p.rapidapi.com'
        }

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

```
    print(response.text)
    return response.json()['items']
if _name_ == "_main_":
   # running the app
    app.run(debug=False)
```

7.2 FEATURE 2

```
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
# 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'))
c
# 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
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               802944
_________________________________________________________________
dense_3 (Dense)              (None, 5)                 645
=================================================================
Total params: 813,733
Trainable params: 813,733
Non-trainable params: 0
_________________________________________________________________
# Compiling the CNN
# categorical_crossentropy for more than 2
classifier.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
classifier.fit(
        x_train,steps_per_epoch = len(x_train),
        epochs=20, validation_data=x_test,validation_steps = len(x_test))# No of
images in test set
Epoch 1/20
818/818 [==============================] - 31s 37ms/step - loss: 1.0134
- accuracy: 0.6168 - val_loss: 0.8680 - val_accuracy: 0.6715
```

Epoch 2/20
818/818 [==============================] - 27s 32ms/step - loss: 0.8599 - accuracy: 0.6773 - val_loss: 0.8070 - val_accuracy: 0.7230
Epoch 3/20
818/818 [==============================] - 26s 32ms/step - loss: 0.7904 - accuracy: 0.7042 - val_loss: 0.7145 - val_accuracy: 0.7473
Epoch 4/20
818/818 [==============================] - 26s 32ms/step - loss: 0.7498 - accuracy: 0.7157 - val_loss: 0.7741 - val_accuracy: 0.7201
Epoch 5/20
818/818 [==============================] - 27s 32ms/step - loss: 0.7275 - accuracy: 0.7282 - val_loss: 0.8567 - val_accuracy: 0.6948
Epoch 6/20
818/818 [==============================] - 27s 33ms/step - loss: 0.6779 - accuracy: 0.7536 - val_loss: 0.7427 - val_accuracy: 0.7289
Epoch 7/20
818/818 [==============================] - 26s 32ms/step - loss: 0.6231 - accuracy: 0.7693 - val_loss: 0.7178 - val_accuracy: 0.7425
Epoch 8/20
818/818 [==============================] - 27s 32ms/step - loss: 0.6082 - accuracy: 0.7742 - val_loss: 0.7180 - val_accuracy: 0.7590
Epoch 9/20
818/818 [==============================] - 27s 33ms/step - loss: 0.5508 - accuracy: 0.7977 - val_loss: 0.7001 - val_accuracy: 0.7590
Epoch 10/20
818/818 [==============================] - 28s 35ms/step - loss: 0.5290 - accuracy: 0.8087 - val_loss: 0.7246 - val_accuracy: 0.7979
Epoch 11/20
818/818 [==============================] - 28s 34ms/step - loss: 0.4724 - accuracy: 0.8290 - val_loss: 0.8073 - val_accuracy: 0.7570
Epoch 12/20
818/818 [==============================] - 27s 33ms/step - loss: 0.4518 - accuracy: 0.8380 - val_loss: 0.7389 - val_accuracy: 0.7775
Epoch 13/20
818/818 [==============================] - 27s 33ms/step - loss: 0.4170 - accuracy: 0.8473 - val_loss: 0.8097 - val_accuracy: 0.7347
Epoch 14/20
818/818 [==============================] - 27s 34ms/step - loss: 0.3909 - accuracy: 0.8578 - val_loss: 0.8060 - val_accuracy: 0.7259
Epoch 15/20

```
818/818 [==============================] - 28s 34ms/step - loss: 0.3604
- accuracy: 0.8723 - val_loss: 0.9770 - val_accuracy: 0.7172
Epoch 16/20
818/818 [==============================] - 28s 34ms/step - loss: 0.3177
- accuracy: 0.8823 - val_loss: 0.8997 - val_accuracy: 0.7502
Epoch 17/20
818/818 [==============================] - 28s 34ms/step - loss: 0.3181
- accuracy: 0.8838 - val_loss: 0.8560 - val_accuracy: 0.7502
Epoch 18/20
818/818 [==============================] - 28s 34ms/step - loss: 0.2724
- accuracy: 0.9051 - val_loss: 1.0304 - val_accuracy: 0.7211
Epoch 19/20
818/818 [==============================] - 27s 34ms/step - loss: 0.2568
- accuracy: 0.9053 - val_loss: 1.0279 - val_accuracy: 0.7396
Epoch 20/20
818/818 [==============================] - 28s 34ms/step - loss: 0.2489
- accuracy: 0.9134 - val_loss: 0.9830 - val_accuracy: 0.7668
classifier.save('D:\nutri.h5')#save  the model
from tensorflow.keras.models import load_model
from keras.preprocessing import image
model = load_model("nutri.h5") #loading the model for testing
img = image.load_img(r"D:\Dataset\TRAIN_SET\ORANGE\n07749192_97.jpg",
                     grayscale=False,target_size= (64,64))#loading of the image
x = image.img_to_array(img)#image  to array
x = np.expand_dims(x,axis = 0)#changing the shape
pred=np.argmax(model.predict(x), axis=1)
pred
array([2], dtype=int64)
index=['APPLES', 'BANANA', 'ORANGE','PINEAPPLE','WATERMELON']
result=str(index[pred[0]])
result
'ORANGE'
```

CHAPTER 8

TESTING

8.1 TEST CASES

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

Characteristics of a good test case:

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

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

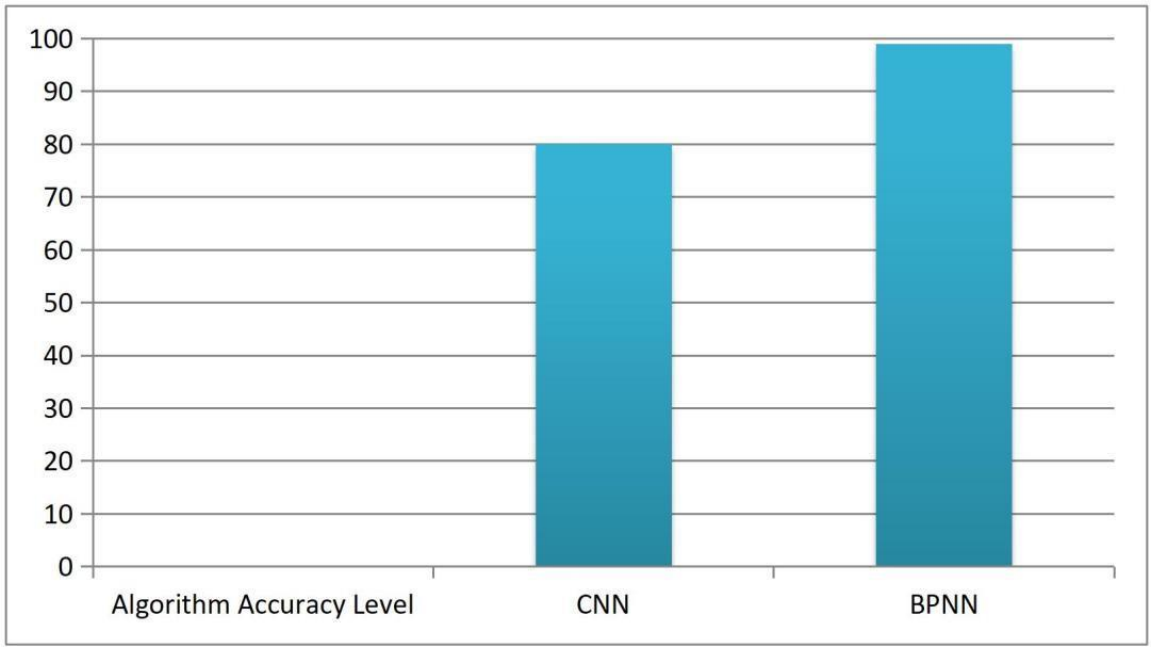
8.2 USER ACCEPTANCE TESTING

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

CHAPTER 9

RESULT

9.1 PERFORMANCE METRICS



CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

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

DISADVANTAGES

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

CHAPTER 11

CONCLUTION

The approach for an automated food nutrition detection system that can determine the amount of nutrients in food is proposed in this project work. The machine has so far been able to place the meal into one of the many categories listed in the dataset. The well-known food dataset was used for the categorization. The classification of the food photos into their appropriate classifications using a deep learning approach. By reducing noise from the dataset, the classification process may be made better. The same research may be done with a larger dataset, more classes, and more photos in each class since a larger dataset increases accuracy by teaching the algorithm additional features and lowers the loss rate. The model's weights may be saved and used to create designs for food categorization, calorie extraction, and picture classification.

CHAPTER 12

FUTURE SCOPE

The food photographs in this research study are categorised into the appropriate groups using a deep learning approach. In terms of future improvement, the classification task may be made better by reducing noise from the dataset. The same research may be done with a larger dataset, more classes, and more photos in each class since a larger dataset increases accuracy by teaching the algorithm additional features and lowers the loss rate. The model's weights may be saved and utilised to create a web or mobile application that classifies images and also extracts the calories from the food that has been identified.

CHAPTER 13

APPENDIX

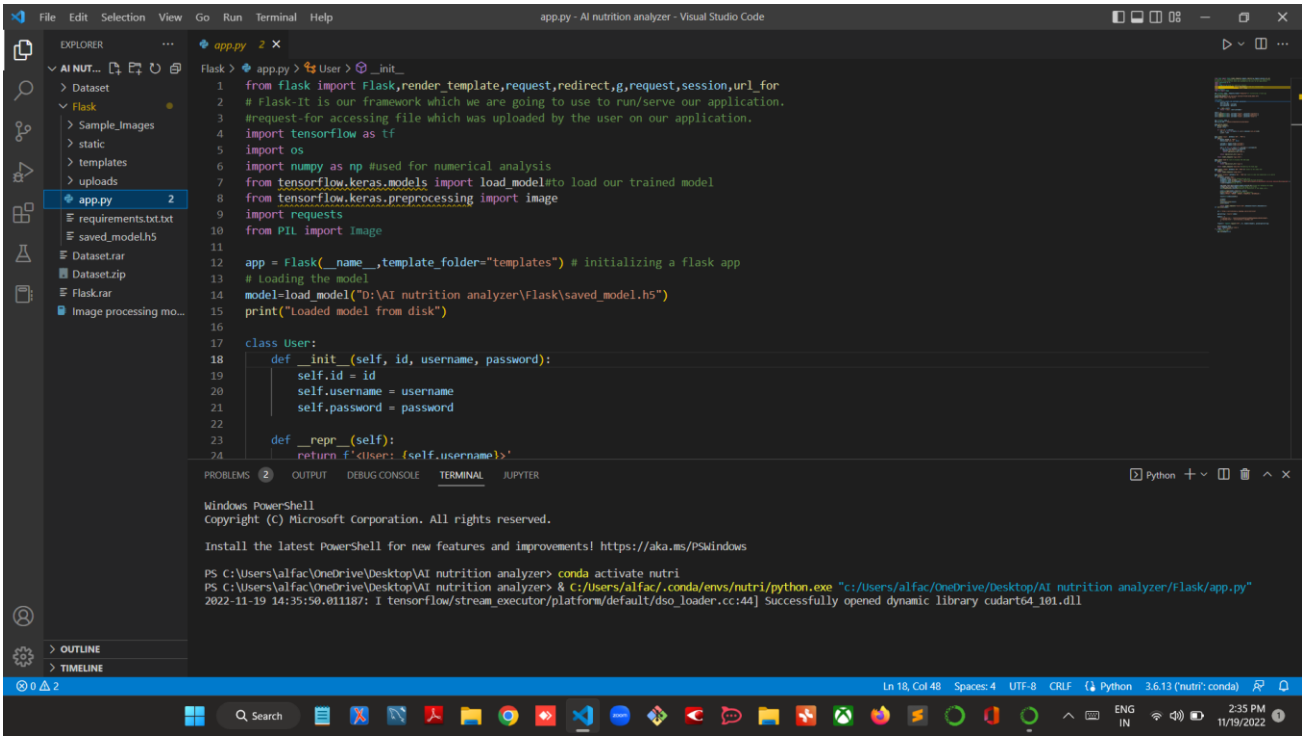
13.1 SOURCE CODE

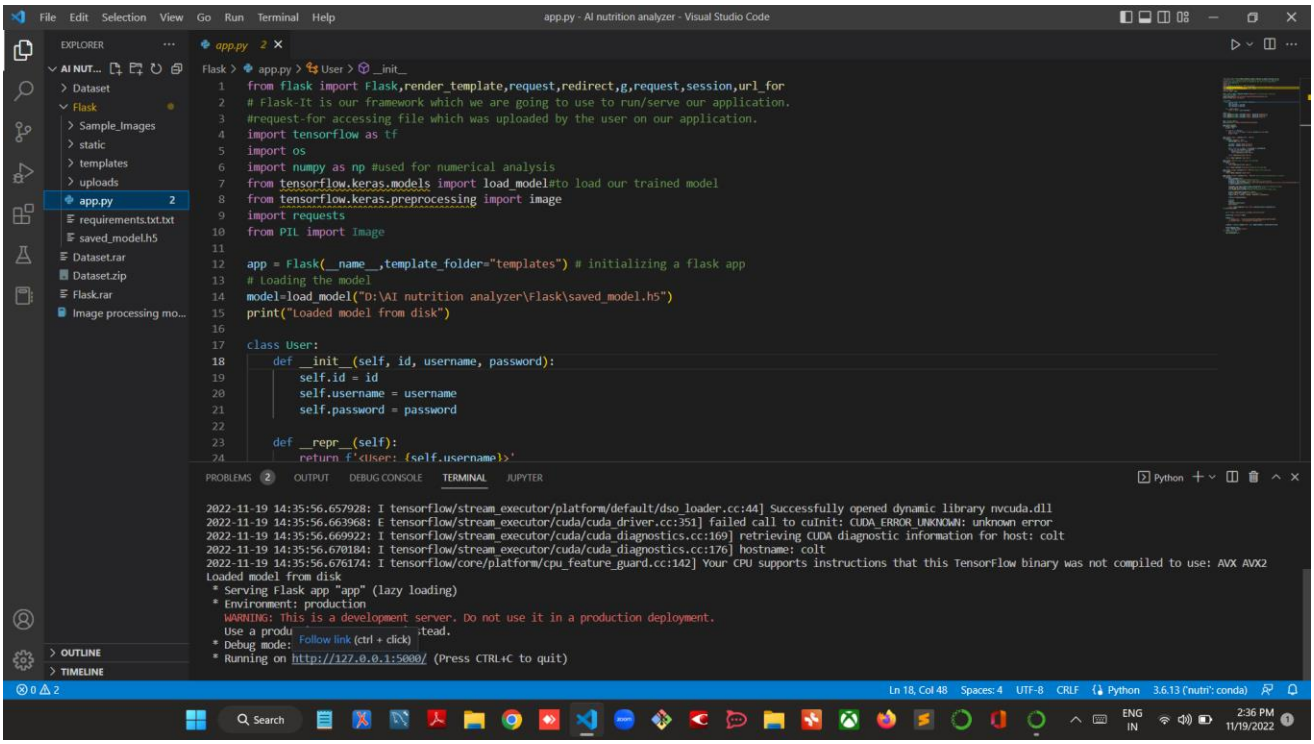
```
from flask import Flask,render_template,request,redirect,g,request,session,url_for
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
import tensorflow as tf
import os
import numpy as np #used for numerical analysis
from tensorflow.keras.models import load_model#to load our trained model
from tensorflow.keras.preprocessing import image
import requests
from PIL import Image

app = Flask(_name_,template_folder="templates") # initializing  a flask app
# Loading the model
model=load_model("D:\AI nutrition analyzer\Flask\saved_model.h5")
print("Loaded model from disk")

class User:
    def _init_(self, id, username, password):
        self.id = id
        self.username = username
        self.password = password

    def _repr_(self):
        return f'<User: {self.username}>'

users = []
users.append(User(id=1, username='Yogesh', password='yogesh123'))
users.append(User(id=2, username='Ganesh', password='ganesh123'))
users.append(User(id=3, username='venkat', password='venkat'))
```

```
app = Flask(_name_)
app.secret_key = 'somesecretkeythatonlyishouldknow'

@app.before_request
def before_request():
    g.user = None

    if 'user_id' in session:
        user = [x for x in users if x.id == session['user_id']][0]
        g.user = user


@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        session.pop('user_id', None)

        username = request.form['username']
        password = request.form['password']

        user = [x for x in users if x.username == username][0]
        if user and user.password == password:
            session['user_id'] = user.id
            return redirect(url_for('home'))

        return redirect(url_for('login'))

    return render_template('login.html')

@app.route('/home')# route to display the home page
def home():
    if not g.user:
        return redirect(url_for('login'))

    return render_template('home.html')#rendering  the home page

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

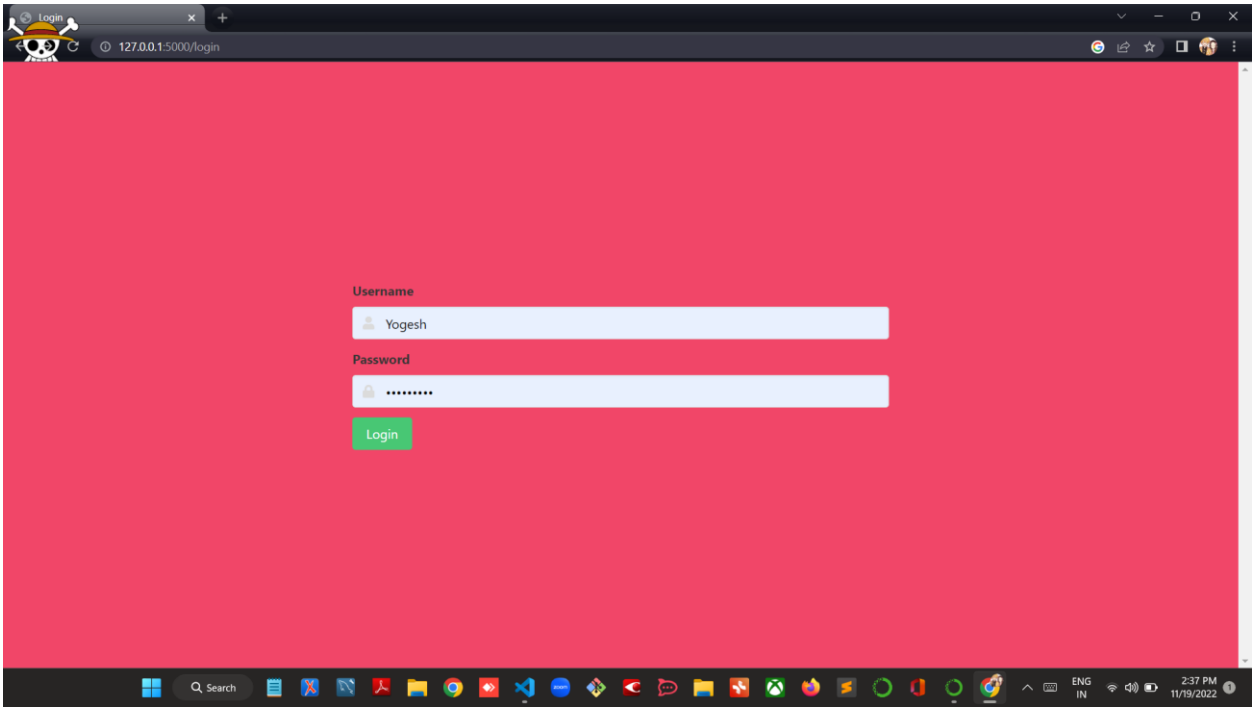
```
@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
def launch():
    if request.method=='POST':
        f=request.files['file'] #requesting the file
        basepath=os.path.dirname('_file_')#storing the file directory
        filepath=os.path.join(basepath,r"C:\Users\alfac\OneDrive\Desktop\AI nutrition analyzer\Flask\uploads",f.filename)#storing the file in uploads folder
        f.save(filepath)#saving the file

        img=image.load_img(filepath,target_size=(64,64)) #load and reshaping the image
        x=image.img_to_array(img)#converting image to an array
        x=np.expand_dims(x,axis=0)#changing the dimensions of the image_init_

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

        result=str(index[pred[0]])

        x=result
        print(x)
        result=nutrition(result)
        print(result)

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


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

    querystring = {"query":index}

    headers = {
        'x-rapidapi-key': 'f162f31fa2mshea65fd3f1d29094p16dba6jsn50c59cc02a64',
        'x-rapidapi-host': 'calorieninjas.p.rapidapi.com'
        }

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

```
    print(response.text)
    return response.json()['items']
if _name_ == "_main_":
   # running the app
    app.run(debug=False)
```

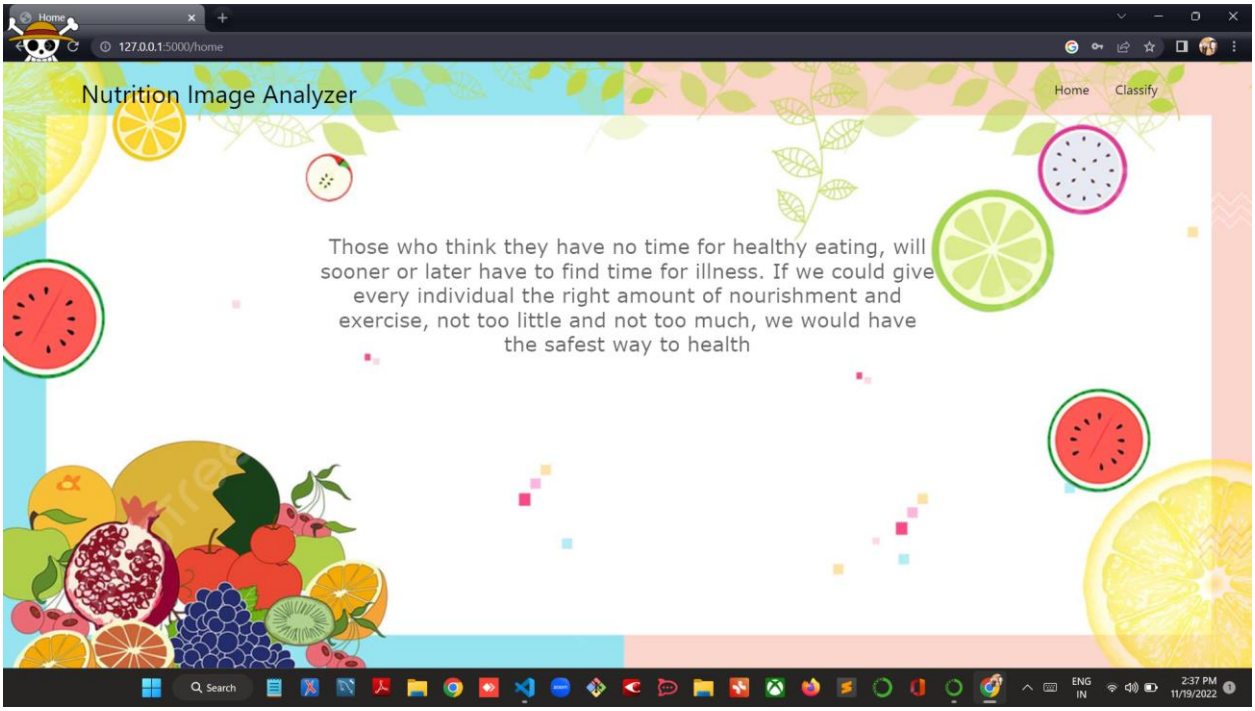

File Edit Selection View Go Run Terminal Help
app.py - AI nutrition analyzer - Visual Studio Code
EXPLORER
AI NUT...
Dataset
Flask
Sample_Images
static
templates
uploads
app.py 2
requirements.txt.txt
saved_model.h5
Dataset.rar
Dataset.zip
Flask.rar
Image processing mo...
app.py 2
Flask > app.py > User > __init__
1 from flask import Flask,render_template,request,redirect,g,request,session,url_for
2 # Flask-It is our framework which we are going to use to run/serve our application.
3 #request-for accessing file which was uploaded by the user on our application.
4 import tensorflow as tf
5 import os
6 import numpy as np #used for numerical analysis
7 from tensorflow.keras.models import load_model#to load our trained model
8 from tensorflow.keras.preprocessing import image
9 import requests
10 from PIL import Image
11
12 app = Flask(__name__,template_folder="templates") # initializing a flask app
13 # Loading the model
14 model=load_model("D:\AI nutrition analyzer\Flask\saved_model.h5")
15 print("Loaded model from disk")
16
17 class User:
18 def __init__(self, id, username, password):
19 self.id = id
20 self.username = username
21 self.password = password
22
23 def __repr__(self):
24 return f'<User: {self.username}>'
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Python
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\alfac\OneDrive\Desktop\AI nutrition analyzer> conda activate nutri
PS C:\Users\alfac\OneDrive\Desktop\AI nutrition analyzer> & C:/Users/alfac/.conda/envs/nutri/python.exe "c:/Users/alfac/OneDrive/Desktop/AI nutrition analyzer/Flask/app.py"
2022-11-19 14:35:50.011187: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library cudart64_101.dll
OUTLINE
TIMELINE
Ln 18, Col 48 Spaces: 4 UTF-8 CRLF Python 3.6.13 ('nutri': conda)
Search
ENG IN
2:35 PM 11/19/2022


File Edit Selection View Go Run Terminal Help
app.py - AI nutrition analyzer - Visual Studio Code
EXPLORER
AI NUT...
Dataset
Flask
Sample_Images
static
templates
uploads
app.py 2
requirements.txt.txt
saved_model.h5
Dataset.rar
Dataset.zip
Flask.rar
Image processing mo...
app.py 2
Flask > app.py > User > __init__
1 from flask import Flask,render_template,request,redirect,g,request,session,url_for
2 # Flask-It is our framework which we are going to use to run/serve our application.
3 #request-for accessing file which was uploaded by the user on our application.
4 import tensorflow as tf
5 import os
6 import numpy as np #used for numerical analysis
7 from tensorflow.keras.models import load_model#to load our trained model
8 from tensorflow.keras.preprocessing import image
9 import requests
10 from PIL import Image
11
12 app = Flask(__name__,template_folder="templates") # initializing a flask app
13 # Loading the model
14 model=load_model("D:\AI nutrition analyzer\Flask\saved_model.h5")
15 print("Loaded model from disk")
16
17 class User:
18 def __init__(self, id, username, password):
19 self.id = id
20 self.username = username
21 self.password = password
22
23 def __repr__(self):
24 return f'<User: {self.username}>'
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Python
2022-11-19 14:35:56.657928: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library nvcuda.dll
2022-11-19 14:35:56.663968: E tensorflow/stream_executor/cuda/cuda_driver.cc:351] failed call to cuInit: CUDA_ERROR_UNKNOWN: unknown error
2022-11-19 14:35:56.669922: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: colt
2022-11-19 14:35:56.670184: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: colt
2022-11-19 14:35:56.676174: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
Loaded model from disk
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a produ Follow link (ctrl + click) itead.
* Debug mode:
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
OUTLINE
TIMELINE
Ln 18, Col 48 Spaces: 4 UTF-8 CRLF Python 3.6.13 ('nutri': conda)
Search
ENG IN
2:36 PM 11/19/2022

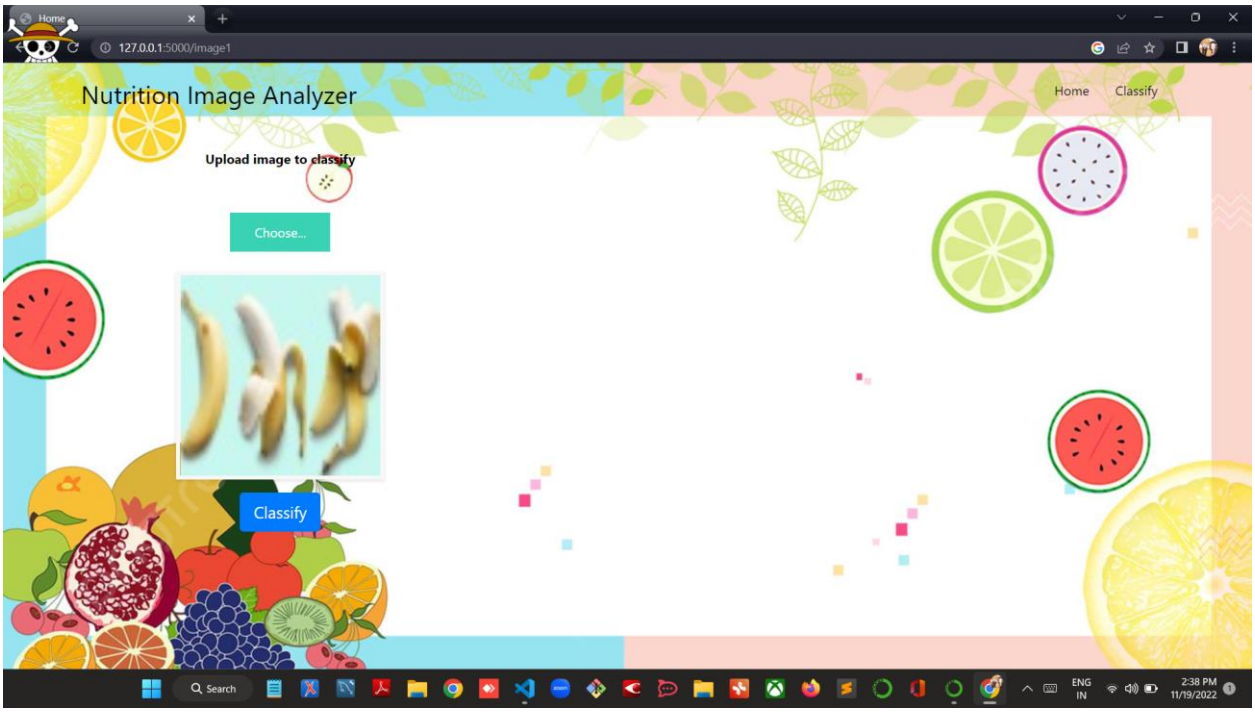
LOG-IN PAGE



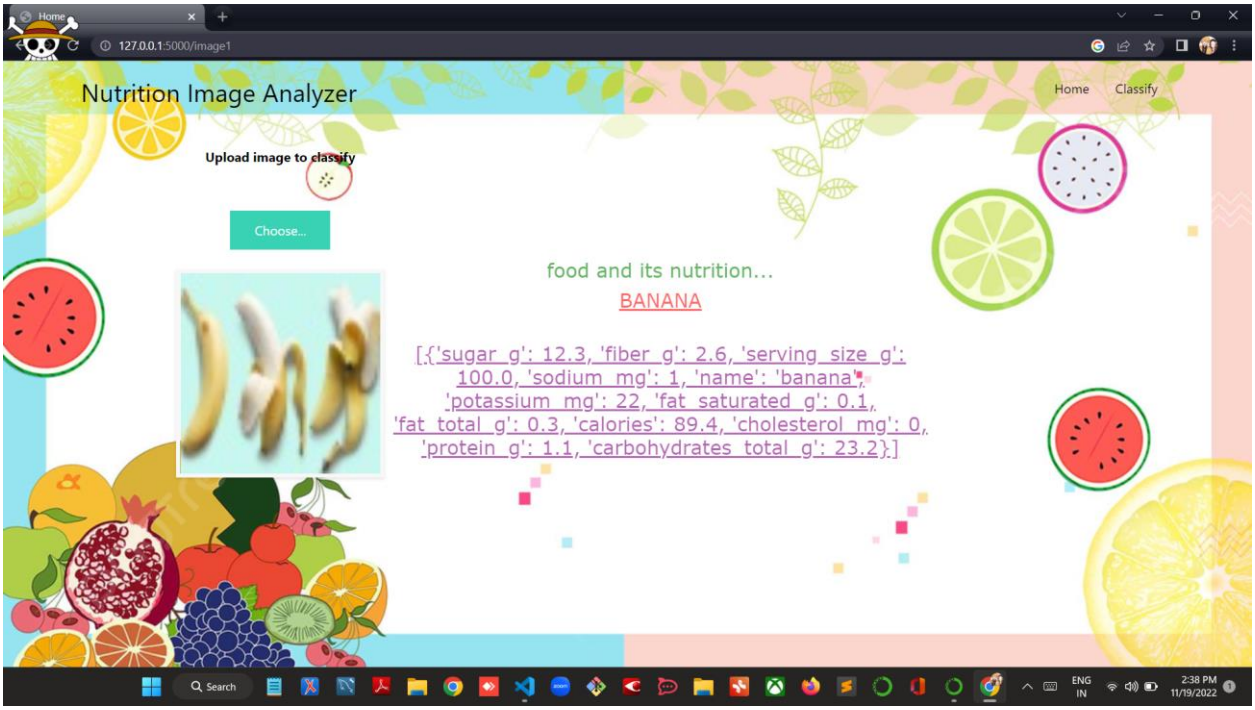
HOME PAGE



TEST PAGE



PREDICT PAGE



13.2 GITHUB & PROJECT DEMO LINK

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-48913-1660814282

PROJECT DRIVE DEMO LINK

https://drive.google.com/file/d/1fQwMbLM8Es_AQIHyTZ1KHdMTzDaYt0iz/view?usp=share_link