AI-powered Nutrition Analyzer for Fitness Enthusiasts

PROJECT REPORT DOCUMENT

Team ID	PNT2022TMID35969
Project Title	AI-powered Nutrition Analyzer for Fitness Enthusiasts
Team Members	D. Sterling Sneha R. Aarthi M. Amara A. Arockia Prathicksha

1. INTRODUCTION

1.1 Project Overview

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

2. LITERATURE SURVEY

2.1 Existing Problem

Food patterns and diet are important factors to improve the lifestyle by preventing diseases. The food industry comprises complexities, and the journey for innovation in the food industry is long, from idea generation to commercialization. It is reported that diet significantly influences the evolution of CNCD (chronic non-communicable diseases), including, cardiovascular diseases, depression, and obesity. Further, product ideas and advanced packaging demand thorough data collection, testing, and certification before approaching consumers. If this work is performed manually, it brings high possibilities of errors that ultimately lead to time and money wastage with no beneficial outcomes. Here AI in nutrition plays a significant role in offering the extraordinary potential for preventing diseases and better treatment methods.

2.2 References

[1] Rojas-Aranda, J.L., Nunez-Varela, J.I., Cuevas-Tello, J.C., Rangel-Ramirez, G. (2020). Fruit Classification for Retail Stores Using Deep Learning. In: Figueroa Mora, K., Anzurez Marín, J., Cerda, J., Carrasco-Ochoa, J., Martínez-Trinidad, J., Olvera-López, J. (eds) Pattern Recognition. MCPR 2020. Lecture Notes in Computer Science(), vol 12088. Springer, Cham. https://doi.org/10.1007/978-3-030-49076-8_1

[2] Jana, S., Parekh, R., Sarkar, B. (2020). Automatic Classification of Fruits and Vegetables: A Texture-Based Approach. In: Mandal, J., Mukhopadhyay, S., Dutta, P., Dasgupta, K. (eds) Algorithms in Machine Learning Paradigms. Studies in Computational Intelligence, vol 870. Springer, Singapore. https://doi.org/10.1007/978-981-15-1041-0_5

[3] C. Liu, X. Wang, J. Ni, Y. Cao and B. Liu, "An Edge Computing Visual System for Vegetable Categorization," 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019, pp. 625-632, doi: 10.1109/ICMLA.2019.00115.

[4] D. G. Savakar and A. K. Talawar, "Fuzzy C-Means Clustering based Identification of Indian Common Non-Leafy Vegetables," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 2021, pp. 858-863.

[5] G. Zeng, "Fruit and vegetables classification system using image saliency and convolutional neural network," 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), 2017, pp. 613-617, doi: 10.1109/ITOEC.2017.8122370.

[6] R. S. Chaulagain, S. Pandey, S. R. Basnet and S. Shakya, "Cloud Based Web Scraping for Big Data Applications," 2017 IEEE International Conference on Smart Cloud (SmartCloud), 2017

[7] K. Jaspin, S. Selvan, J. D. Rose, J. Ebenezer and A. Chockalingam, "Real-Time Surveillance for Identification of Fruits Ripening Stages and Vegetables Maturation Stages with Infection Detection," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), 2021, pp. 581-586, doi: 10.1109/ISPCC53510.2021.9609441.

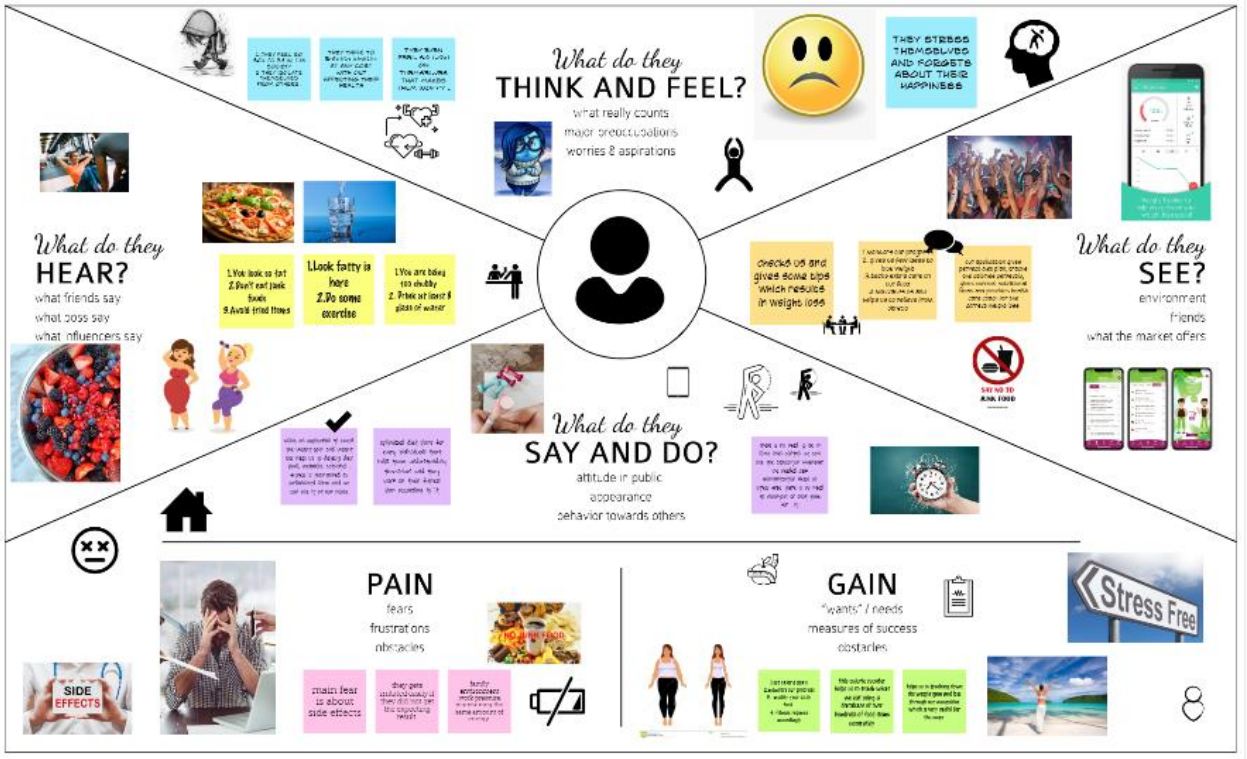
[8] Khatun, Mehenag & Nine, Julker & Ali, Md. Forhad & Sarker, Pritom & Turzo, Nakib. (2020). Fruits Classification using Convolutional Neural Network. 5. 1-6.

2.3 Problem Statement Definition

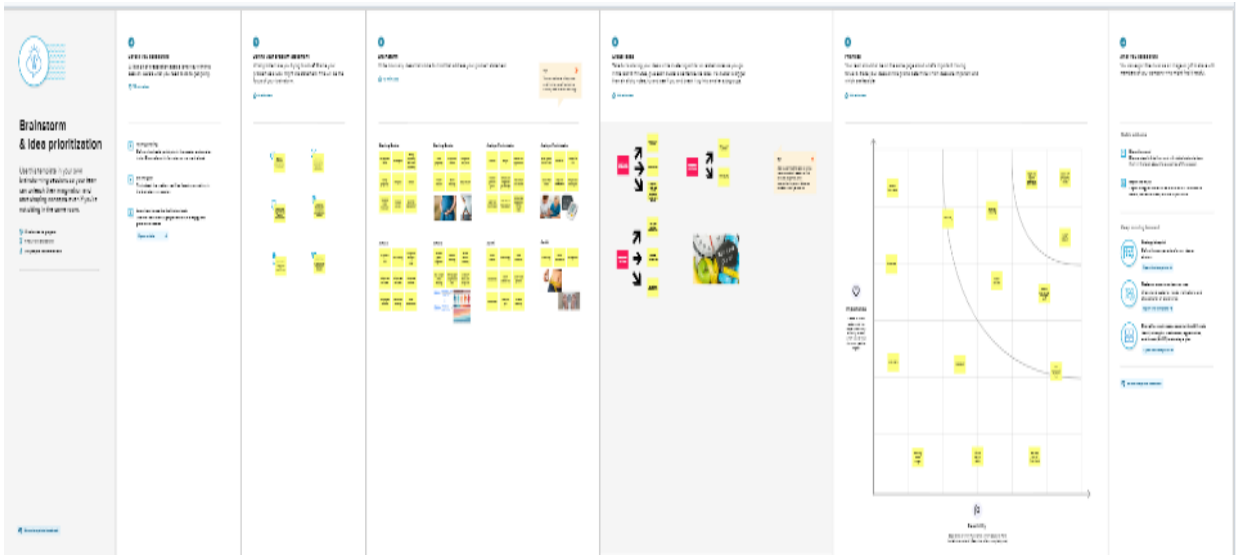
Humans should give equal importance to their health as they do towards other things. But in today's fast moving lifestyle, people aren't conscious about their health and follow questionable eating habits. When followed on a regular basis, they can have detrimental effects on the human body and may even be fatal. Thus it is imperative to maintain good health. A person needs to follow a balanced diet, i.e. consume food containing proteins, vitamins and other vital nutrients that are needed by the human body in suggested proportions on a daily basis. In this project, a system is developed to identify edible products and discern their nutritional information. The users of this system can capture images of the ingredients that go into their food and be informed of their nutritional composition. This way, fitness enthusiasts will be able to keep track of their calorie intake and people will be able to follow a healthy lifestyle of eating. The work proposed is a simple application that can recognize the raw food items based on the input image and provide information regarding their nutritional value to the user

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement(Problem to be solved)	To develop a system that identifies edible products and discerns their nutritional information for the benefit of fitness enthusiasts
2.	Idea/Solution description	The system develop is an application that scans the surroundings to capture images. The image is analyzed to identify the fruits present in the image using machine learning models. Once the raw food items have been identified, their corresponding nutritional values are fetched from a database where the relevant details are stored. The application allows for a user to keep track of the amount of calories they consume in a day versus the total recommended amount for their dietary needs. The data of frequently consumed fruits is stored locally in the database.
3.	Novelty/Uniqueness	The proposed system maintains a personal Nutrition calendar for the user and notifies them when they do not meet the requirements of their diet. Further, the apps inbuilt with features that suggest alternative foods, construct a food chart, develop a workout schedule, and recommend recipes that suit the caloric needs of the user. The system also integrates capabilities of identifying spoilt food items and whether fruits have ripened.
4.	Social Impact/Customer Satisfaction	The proposed application is use full for fitness enthusiasts to keep track of their calorie intakeandthusmaintaintheirphysicalstate.Event hosewhoarenotconsciousabouttheirphysiquem ayusethisapplicationtoleadahealthierlifestyle as it helps to keep track ofwhattheyeat,suggestshealthyalternativesandr ecipes,aswellasworkoutplans.

5.	Business Model(Revenue Model)	The application can be deployed for access by the general public. The application would draw the attention of several users who are determined to lead a healthy lifestyle and wish to undergo a physical transformation. The application could be built in such a way that features are progressively unlocked based onthesubscriptionamountpaidbytheuserstarting fromthegenericnutritionanalyzerfeaturetochartingoutpersonalplansforusers.
6.	Scalability of the Solution	The proposed application has several features. It can be further enhanced to integrate more features based on feedback from users and ratings.

3.4 Problem Solution Fit

4. RECUIREMENT ANALYSIS

4.1 Functional Requirements

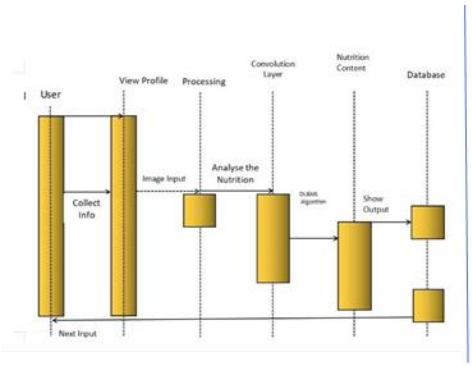
A healthy diet throughout life promotes healthy pregnancy outcomes, supports normal growth, development and ageing, helps to maintain a healthy body weight, and reduces the risk of chronic disease leading to overall health and well-being.

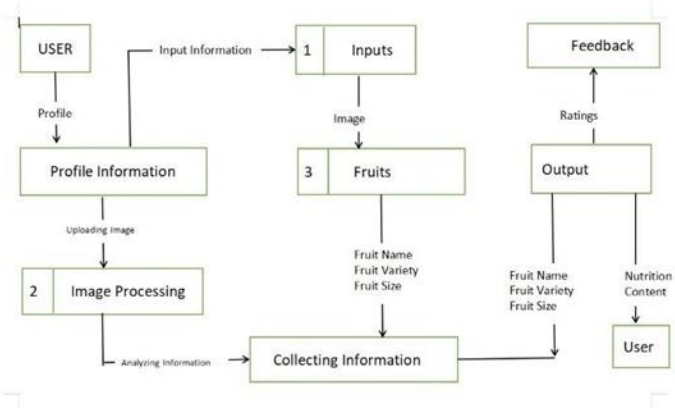
4.2 Non-Functional Requirements

Nonfunctional Requirements Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. Also known as system qualities, nonfunctional requirements are just as critical as functional Epics, Capabilities, Features, and Stories. They ensure the usability and effectiveness of the entire system. Failing to meet any one of them can result in systems that fail to satisfy internal business, user, or market needs, or that do not fulfill mandatory requirements imposed by regulatory or standards agencies. In some cases, non-compliance can cause significant legal issues (privacy, security, safety, to name a few). NFRs are persistent qualities and constraints that, unlike functional requirements, are typically revisited as part of the Definition of Done (DoD) for each Iteration, Program Increment (PI), or release.

5. PROJECT DESIGN

5.1 Data Flow Diagram





5.2 Solution & Technical Architecture

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	An application where users create a profile, capture images of the ingredients they use in their food, and have access to a bespoke nutrition calendar	HTML, CSS, JavaScript
2.	Image Capture	Users must capture the image(s) of the ingredients they consume	IBM Maximo Image Inspection
3.	Ingredient Detection Model	The ingredients used must be identified from the captured image	Machine Learning & Image Processing using Python
4.	Calorie Consumption Monitoring	The application keeps track of the calories consumed by the user in a day and notifies when there is over-consumption	IBM Push Notifications
5.	Database of Ingredients	The data of ingredients and their corresponding calories are stored	MySQL
6.	Cloud Database for Back-up	Data used by the application is stored here for back-up and monthly calendars are stored as consolidated reports	IBM Cloudant
7.	File Storage	Per-day calorie consumption along with items consumed is kept track using a file system. This is used to generate a personal calorie calendar as well	IBM Block Storage
8.	Calorie Value Consolidation	A web-scraping API is employed to find the calorie values of ingredients which are stored in the database	Beautiful Soup
9.	Machine Learning Model	Captured images are processed using machine learning models to identify ingredients	Object Recognition Model to Label Ingredients
10.	Infrastructure (Server / Cloud)	The application is deployment on cloud for use Cloud Server Configuration :	Cloud Foundry

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Google Colab, VS Code, Online Websites	Python, HTML, CSS, JavaScript
2.	Security Implementations	E-mail based authentication for data access and encryption of text before storing in files	SMTP, Encryption Algorithms
3.	Scalable Architecture	Application is revised based on user experience and feedback including updates, bug fixes, and inclusion of new features	Customer feedback, reviews, and ratings
4.	Availability	Users should be able to access the application that is hosted on the cloud at all times and should not face any issues such as application crash	IBM Cloud
5.	Performance	Application should handle large number of requests and should not compromise on quality of results and time taken	Testing - Black, White, and Beta Revise application in spiral model

5.3 User Stories

User Type	Functional Requirement(Epic)	User Story Number	User Story/Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can do registration by my mobile and get confirmation email	I can access my account/dashboard	High	Sprint-1
Customer (Web user)	Registration	USN-2	As a user, I can register for the application By entering my email, password, and confirming my password	I can access my account/dashboard	High	Sprint-1
		USN-3	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email &click confirm	High	Sprint-1
		USN-4	As a user, I can register for the application through other social media	I can register &access the dashboard with Login	Low	Sprint-2
	Login	USN-5	As a user, I can login to the application by entering email &password	I can successfully logged into the homepage	High	Sprint-1
	Dashboard	USN-6	As a user, I can select the list of options provided in the dashboard	I can access the options according to my need	Medium	Sprint-1
	Search	USN-7	As a user, I can search for different variety of fruits	I can get the nutrition content of different fruits	High	Sprint-2
	View	USN-8	As a user, I can view the list of fruits	I will get the information such as calories, vitamins etc	High	Sprint-2
	Notifications	USN-9	As a user, I will receive notification about variety and textures of different fruits	I will get the frequent updates of different fruits	Low	Sprint-2
Customer Care Executive	Mediator	USN-10	As a customer care executive ,they could take care of customer feed backs and solve User requirements	Users can get help and support from customer care Executives	Medium	Sprint-2
Administrator	Database	USN-11	As a admin, I will store the user database confidentially	I can store and access data if it is needed in future	High	Sprint-1
	Data Information	USN-12	As a admin, I will include the dataset for performing various processes	I can store data set and analysis it	High	Sprint-2
	Processing	USN-13	As a admin, I will use various convolution layers for image analysis	I can process using various convolution layers	High	Sprint-2
	Nutrition Analyzer	USN-14	As a admin, I will predict the fruit that has send as input	I can get the nutrition content of particular food after processing and display it	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning &Estimation

Planning and Estimation are essential in software projects to achieve predictability, reduce the risks involved, and set a basic expectation for all stakeholders. Planning brings a lot of focus on preparation and forecasting whereas Estimation is a process to forecast project-related variables i.e., effort, scope, schedule, etc.

Planning: Planning is required irrespective of the project management methodologies that the team follows, whether it is Waterfall or Agile. Planning gives the project team a perspective on how to meet the objective in a systematic way and helps project stakeholders to keep a tab on the project progress and investments done.

As Mike Cohn defines it, “Agile planning balances the effort and investment in planning with the knowledge that we will revise the plan through the course of the project. An agile plan is one that we are not only willing but also eager to change”. This concept exists mainly to avoid the weakness of the planning.

6.2 Sprint Delivery Schedule

Image result for sprint delivery schedule In case you're unfamiliar, a sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication.

6.3 Reports From JIRA

One part of ensuring the success and smooth operations of your projects in JIRA is reporting. It involves gaining the knowledge about the health, progress and overall status of your JIRA projects through Gadgets, report pages or even third party applications. The goal of this guide is to provide an overview of the tools available to JIRA users today and how they can be used to fulfill the different types of reporting needs that users face today

7.CODING & SOLUTIONING

7.1 FEATURE 1

```
const sign_in_btn =
document.querySelector("#sig
n-in-btn");
                                const sign_up_btn =
                                document.querySelector("#sign-
                                up-btn");
                                const container =
                                document.querySelector(".conta
                                iner");
                                });
                                sign_in_btn.addEventListener("
                                click", () => {
```

```
                                container.classList.remove("si
                                gn-up-mode");
                                });
{
                    "require": {
                        "phpmailer/phpmailer":
                "^6.6"
                    }
                }
{
    "_readme": [
      "This file locks the dependencies of your project to a known state",
      "Read more about it at https://getcomposer.org/doc/01-basic-
usage.md#installing-dependencies",
      "This file is @generated automatically"
    ],
    "content-hash": "9c3d62e919c2177ee2d1eff04c5d4018",
    "packages": [
      {
        "name": "phpmailer/phpmailer",
        "version": "v6.6.5",
        "source": {
          "type": "git",
          "url": "https://github.com/PHPMailer/PHPMailer.git",
          "reference":
"8b6386d7417526d1ea4da9edb70b8352f7543627"
        },
        "dist": {
          "type": "zip",
          "url":
"https://api.github.com/repos/PHPMailer/PHPMailer/zipball/8b6386d7
417526d1ea4da9edb70b8352f7543627",
          "reference":
"8b6386d7417526d1ea4da9edb70b8352f7543627",
          "shasum": ""
        },
        "require": {
```

```
            "ext-ctype": "*",
            "ext-filter": "*",
            "ext-hash": "*",
            "php": ">=5.5.0"
        },
        "require-dev": {
            "dealerdirect/phpcodesniffer-composer-installer": "^0.7.0",
            "doctrine/annotations": "^1.2",
            "php-parallel-lint/php-console-highlighter": "^1.0.0",
            "php-parallel-lint/php-parallel-lint": "^1.3.2",
            "phpcompatibility/php-compatibility": "^9.3.5",
            "roave/security-advisories": "dev-latest",
            "squizlabs/php_codesniffer": "^3.6.2",
            "yoast/phpunit-polyfills": "^1.0.0"
        },
        "suggest": {
            "ext-mbstring": "Needed to send email in multibyte encoding
charset or decode encoded addresses",
            "hayageek/oauth2-yahoo": "Needed for Yahoo XOAUTH2
authentication",
            "league/oauth2-google": "Needed for Google XOAUTH2
authentication",
            "psr/log": "For optional PSR-3 debug logging",
            "symfony/polyfill-mbstring": "To support UTF-8 if the Mbstring
PHP extension is not enabled (^1.2)",
            "thenetworg/oauth2-azure": "Needed for Microsoft XOAUTH2
authentication"
        },
        "type": "library",
        "autoload": {
            "psr-4": {
                "PHPMailer\\PHPMailer\\": "src/"
            }
        },
        "notification-url": "https://packagist.org/downloads/",
        "license": [
            "LGPL-2.1-only"
```

```
        ],
        "authors": [
          {
            "name": "Marcus Bointon",
            "email": "phpmailer@synchromedia.co.uk"
          },
          {
            "name": "Jim Jagielski",
            "email": "jimjag@gmail.com"
          },
          {
            "name": "Andy Prevost",
            "email": "codeworxtech@users.sourceforge.net"
          },
          {
            "name": "Brent R. Matzelle"
          }
        ],
        "description": "PHPMailer is a full-featured email creation and
transfer class for PHP",
        "support": {
          "issues": "https://github.com/PHPMailer/PHPMailer/issues",
          "source":
"https://github.com/PHPMailer/PHPMailer/tree/v6.6.5"
        },
        "funding": [
          {
            "url": "https://github.com/Synchro",
            "type": "github"
          }
        ],
        "time": "2022-10-07T12:23:10+00:00"
      }
    ],
    "packages-dev": [],
    "aliases": [],
    "minimum-stability": "stable",
```

```
        "stability-flags": [],
        "prefer-stable": false,
        "prefer-lowest": false,
        "platform": [],
        "platform-dev": [],
        "plugin-api-version": "2.3.0"
    }
```

7.2 FEATURE 2

```
?php
    include 'config.php';
    use PHPMailer\PHPMailer\PHPMailer;
    use PHPMailer\PHPMailer\SMTP;
    use PHPMailer\PHPMailer\Exception;
    require 'vendor/autoload.php';
    session_start();
    error_reporting(0);
    if (isset($_SESSION["user_id"])) {
     header("Location: welcome.php");
    }
    if (isset($_POST["signup"])) {
     $full_name = mysqli_real_escape_string($conn,
    $_POST["signup_full_name"]);
     $email = mysqli_real_escape_string($conn, $_POST["signup_email"]);
     $password = mysqli_real_escape_string($conn,
    md5($_POST["signup_password"]));
     $cpassword = mysqli_real_escape_string($conn,
    md5($_POST["signup_cpassword"]));
     $token = md5(rand());
     $check_email = mysqli_num_rows(mysqli_query($conn, "SELECT email
    FROM users WHERE email='$email'"));
     if ($password !== $cpassword) {
      echo "<script>alert('Password did not match.');</script>";
     } elseif ($check_email > 0) {
      echo "<script>alert('Email already exists in out database.');</script>";
     } else {
      $sql = "INSERT INTO users (full_name, email, password, token, status)
    VALUES ('$full_name', '$email', '$password', '$token', '0')";
```

```
    $result = mysqli_query($conn, $sql);
    if ($result) {
      $_POST["signup_full_name"] = "";
      $_POST["signup_email"] = "";
      $_POST["signup_password"] = "";
      $_POST["signup_cpassword"] = "";
      $to = $email;
      $subject = "Email verification - Scan My Nutri";
      $message = "
      <html>
      <head>
      <title>{$subject}</title>
      </head>
      <body>
      <p><strong>Dear {$full_name},</strong></p>
      <p>Thanks for registration! Verify your email to access our website.
Click below link to verify your email.</p>
      <p><a href='{$base_url}verify-email.php?token={$token}'>Verify
Email</a></p>
      </body>
      </html>
      ";
      //Create an instance; passing `true` enables exceptions
      $mail = new PHPMailer(true);
      try {
        //Server settings
        $mail->SMTPDebug = 0;                    //Enable verbose debug output
        $mail->isSMTP();                                 //Send using SMTP
        $mail->Host       = $smtp['host'];                 //Set the SMTP server to
send through
        $mail->SMTPAuth   = true;                          //Enable SMTP
authentication
        $mail->Username   = $smtp['user'];                 //SMTP username
        $mail->Password   = $smtp['pass'];                          //SMTP password
        $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;
//Enable implicit TLS encryption
        $mail->Port       = $smtp['port'];                          //TCP port to
```

```
connect to; use 587 if you have set `SMTPSecure =
PHPMailer::ENCRYPTION_STARTTLS`
        //Recipients
        $mail->setFrom($my_email);
        $mail->addAddress($email, $full_name);     //Add a recipient
        //Content
        $mail->isHTML(true);                                  //Set email format to HTML
        $mail->Subject = $subject;
        $mail->Body    = $message;
        $mail->send();
        echo "<script>alert('We have sent a verification link to your email -
{$email}.');</script>";
      } catch (Exception $e) {
        echo "<script>alert('Mail not sent. Please try again.');</script>";
      }
    } else {
      echo "<script>alert('User registration failed.');</script>";
    }
  }
}
if (isset($_POST["signin"])) {
  $email = mysqli_real_escape_string($conn, $_POST["email"]);
  $password = mysqli_real_escape_string($conn,
md5($_POST["password"]));
  $check_email = mysqli_query($conn, "SELECT id FROM users WHERE
email='$email' AND password='$password' AND status='1'");
  if (mysqli_num_rows($check_email) > 0) {
    $row = mysqli_fetch_assoc($check_email);
    $_SESSION["user_id"] = $row['id'];
    header("Location: welcome.php");
  } else {
    echo "<script>alert('Login details is incorrect. Please try
again.');</script>";
  }
}
?>
<!DOCTYPE html>
```

```
<html lang="en">
<head>
 <meta charset="UTF-8" />
 <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
 <link rel="stylesheet" href="style.css" />
 <title>Scan My Nutri</title>
</head>
<body>
 <div class="container">
  <div class="forms-container">
   <div class="signin-signup">
    <form action="" method="post" class="sign-in-form">
     <h2 class="title">Sign in</h2>
     <div class="input-field">
      <i class="fas fa-user"></i>
      <input type="text" placeholder="Email Address" name="email"
value="<?php echo $_POST['email']; ?>" required />
     </div>
     <div class="input-field">
      <i class="fas fa-lock"></i>
      <input type="password" placeholder="Password"
name="password" value="<?php echo $_POST['password']; ?>" required
/>
     </div>
     <input type="submit" value="Login" name="signin" class="btn solid"
/>
     <p style="display: flex;justify-content: center;align-items:
center;margin-top: 20px;"><a href="forgot-password.php" style="color:
#4590ef;">Forgot Password?</a></p>
    </form>
    <form action="" class="sign-up-form" method="post">
     <h2 class="title">Sign up</h2>
     <div class="input-field">
      <i class="fas fa-user"></i>
      <input type="text" placeholder="Full Name"
name="signup_full_name" value="<?php echo
```

```
$_POST["signup_full_name"]; ?>" required />
        </div>
        <div class="input-field">
          <i class="fas fa-envelope"></i>
          <input type="email" placeholder="Email Address"
name="signup_email" value="<?php echo $_POST["signup_email"]; ?>"
required />
        </div>
        <div class="input-field">
          <i class="fas fa-lock"></i>
          <input type="password" placeholder="Password"
name="signup_password" value="<?php echo
$_POST["signup_password"]; ?>" required />
        </div>
        <div class="input-field">
          <i class="fas fa-lock"></i>
          <input type="password" placeholder="Confirm Password"
name="signup_cpassword" value="<?php echo
$_POST["signup_cpassword"]; ?>" required />
        </div>
        <input type="submit" class="btn" name="signup" value="Sign up" />
      </form>
    </div>
  </div>
  <div class="panels-container">
    <div class="panel left-panel">
      <div class="content">
        <h3>New here ?</h3>
        <br>

        <button class="btn transparent" id="sign-up-btn">
          Sign up
        </button>
      </div>
      <img src="img/download.png" class="image" alt="" />
    </div>
    <div class="panel right-panel">
```

```
        <div class="content">
          <h3>One of us ?</h3>
          <br>

          <button class="btn transparent" id="sign-in-btn">
            Sign in
          </button>
        </div>
        <img src="img/image.png" class="image" alt="" />
      </div>
    </div>
  </div>
  <script src="https://kit.fontawesome.com/64d58efce2.js"
crossorigin="anonymous"></script>
  <script src="app.js"></script>
</body>
</html>
```

8. TESTING

8.1 Test Cases

A mineral panel is a type of nutrition test that is used to evaluate the levels of various minerals in the body, and to diagnose and monitor mineral deficiencies. Some of the most common mineral tests include: Calcium Blood Test: Measures the level of calcium in the blood.

8.2 User Acceptance Testing

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

9. RESULTS

9.1 Performance Metrics

Taking a food systems approach is a promising strategy for improving diets. Implementing such an approach would require the use of a comprehensive set of metrics to characterize food systems, set meaningful goals, track food system performance, and evaluate the impacts of food system interventions. Food system metrics are also useful to structure debates and communicate to policy makers and the general public. This paper provides an updated analytical framework of food systems and uses this to identify systematically relevant metrics and indicators based on data availability in low and middle income countries. We conclude that public data are relatively well available for food system drivers and outcomes, but not for all of the food system activities. With only minor additional investments, existing surveys could be extended to cover a large part of the required additional data. For some indicators, however, targeted data collection efforts are needed. As the list of indicators partly overlaps with the indicators for the Sustainable Development Goals (SGDs), part of the collected data could serve not only to describe and monitor food systems, but also to track progress towards attaining the SDGs.

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

- Monitor Your Diet Easily. Weight watchers or people who want to gain weight can mention the type and amount of foods consumed at each meal.
- Monitor Your Progress.
- Give Free Health and Fitness Tips.
- Track Your Foot Steps.
- Provide Personal Health Coaches.
- All in One Health Tool.
- Keep You Motivated.

10.2 Disadvantages

- Expensive. Fitness trackers can be expensive to buy. If you don't use it, you could have spent your money on something more useful.
- Battery. Most trackers have limited battery life.
- Accuracy. Some trackers do not provide 100% accurate metrics/information that may be shown on the device

11. CONCLUSION

It is important to maintain health and fitness of not just our physical self but our mental self as well, to have a happy and fulfilled life. Engaging in regular physical activity may produce improvements in an individual's physical health, cognitive performance, and psychological well-being. Physical benefits include, but are not limited to, reduced risk for diseases, and improvements in physical functioning, fitness, and overall quality of life.

12. FUTURE SCOPE

Future is a fitness coaching app that let's you work with a real trainer, one-on-one. I've never been good at working out alone. Whether that's at home, in a gym or outdoors, I've always needed a trainer or coach to motivate me and give me that extra push. If you love technology, fitness apps and using an Apple watch, you'll probably love the Future app. The sleek app integrates beautifully with the watch, and our testers found it to be very easy to use. Each workout also offers video demonstrations with instructions on how to complete the exercise with good form.

13. APPENDIX

Source Code

```
{
        "cells": [
         {
          "cell_type": "code",
```

```
   "execution_count": null,
   "metadata": {
    "id": "-4U2x7XApAPv"
   },
   "outputs": [],
   "source": [
    "#import keras libraries\n",
    "from keras.models import Sequential\n",
    "from keras.layers import Dense\n",
    "from keras.layers import Convolution2D\n",
    "from keras.layers import MaxPooling2D\n",
    "from keras.layers import Flatten"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "GUqs8zuap0Ro"
   },
   "outputs": [],
   "source": [
    "#image preprocessing(or) image augmentation\n",
    "from keras.preprocessing.image import ImageDataGenerator"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "t44vJdxpqO67"
   },
   "outputs": [],
   "source": [
    "train_datagen =
ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,hori
zontal_flip=True,vertical_flip=True)\n",
```

```
      "#rescale => rescaling pixel value from 0 to 255 to 0 to 1\n",
      "#shear_range=> counter clock wise rotation(anti clock)"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "metadata": {
      "id": "bPtjB_31qZLl"
     },
     "outputs": [],
     "source": [
      "test_datagen = ImageDataGenerator(rescale=1./255)"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "id": "ltTuui5Kqdtp",
      "outputId": "2f168c3f-c51e-4c92-dc28-3d4ea011d4da"
     },
     "outputs": [
      {
       "output_type": "stream",
       "name": "stdout",
       "text": [
        "Found 4118 images belonging to 5 classes.\n"
       ]
      }
     ],
     "source": [
      "x_train =
train_datagen.flow_from_directory(\"/content/drive/MyDrive/ibm
```

```
project/TRAIN_SET\",target_size=(64,64),batch_size=32,class_mode=\"binar
y\")"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "id": "U9WzDTJHuiAh",
      "outputId": "87f6e98f-1cba-473a-b803-faa60d4eeb7d"
     },
     "outputs": [
      {
       "output_type": "stream",
       "name": "stdout",
       "text": [
        "Found 929 images belonging to 3 classes.\n"
       ]
      }
     ],
     "source": [
      "x_test =
test_datagen.flow_from_directory(\"/content/drive/MyDrive/ibm
project/TEST_SET\",target_size=(64,64),batch_size=32,class_mode=\"binary
\")"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
```

```
      "id": "bApCdADGup8T",
      "outputId": "d57ab51e-f9c3-47b2-f19c-f25f10a7aec7"
     },
     "outputs": [
      {
       "output_type": "execute_result",
       "data": {
        "text/plain": [
         "{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3,
'WATERMELON': 4}"
        ]
       },
       "metadata": {},
       "execution_count": 7
      }
     ],
     "source": [
      "x_train.class_indices"
     ]
    },
    {
     "cell_type": "code",
     "source": [
      "#checking the number of classes\n",
      "print(x_test.class_indices)"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "id": "9A3kmlgHz0Q7",
      "outputId": "d2e6daaa-dbe2-4552-ef65-d5e8bbe0d9ea"
     },
     "execution_count": null,
     "outputs": [
      {
       "output_type": "stream",
```

```
      "name": "stdout",
      "text": [
       "{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2}\n"
      ]
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "from collections import Counter as c\n",
     "c(x_train .labels)"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "yGeKS68E0bSP",
     "outputId": "cd5bac4d-ffb6-464b-d6f0-841ef62e776d"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "Counter({0: 995, 1: 1354, 2: 1019, 3: 275, 4: 475})"
       ]
      },
      "metadata": {},
      "execution_count": 11
     }
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
```

```
   "metadata": {
    "id": "dx_5gTSAu0hY"
   },
   "outputs": [],
   "source": [
    "#Initializing the model\n",
    "model = Sequential()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "ufSbk5LVu9qU"
   },
   "outputs": [],
   "source": [
    "# add First convolution layer"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "62dYvr9WvHlF"
   },
   "outputs": [],
   "source": [

"model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation=\"relu
\"))\n",
    "# 32 indicates => no of feature detectors\n",
    "#(3,3)=> kernel size (feature detector size)"
   ]
  },
  {
   "cell_type": "code",
```

```
  "execution_count": null,
  "metadata": {
   "id": "0RoS09jlvROB"
  },
  "outputs": [],
  "source": [
   "# add Maxpooling layer"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "7tIjlFq_vaMc"
  },
  "outputs": [],
  "source": [
   "model.add(MaxPooling2D(pool_size=(2,2)))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "lnioOB-s9CaM"
  },
  "outputs": [],
  "source": [
   "#Second convolution layer and pooling\n",
   "model.add(Convolution2D(32,(3,3),activation='relu'))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "bAcEug9x-Rqm"
```

```
  },
  "outputs": [],
  "source": [
   "model.add(MaxPooling2D(pool_size=(2,2)))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "hFOgQQQb_Inn"
  },
  "outputs": [],
  "source": [
   "#Flattening the layers\n",
   "model.add(Flatten())"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "v1LSVWYs_g2v"
  },
  "outputs": [],
  "source": [
   "model.add(Dense(units=128,activation='relu'))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "DKg4TBZZ_zT6"
  },
  "outputs": [],
  "source": [
```

```
    "model.add(Dense(units=5,activation='softmax'))"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "eCB4ZIxOvh4G"
   },
   "outputs": [],
   "source": [
    "# add flatten layer => input to your ANN"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "agjb4SXivnq_"
   },
   "outputs": [],
   "source": [
    "model.add(Flatten())"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "fGDMWXyMwSWs",
    "outputId": "e6a3a789-c1aa-406c-886a-6a40f77b71b7"
   },
   "outputs": [
    {
```

```
        "output_type": "stream",
        "name": "stdout",
        "text": [
         "Model: \"sequential\"\n",
         "_________________________\n",
         " Layer (type)              Output Shape            Param #   \n",

"=================================================================
===\n",
         " conv2d (Conv2D)           (None, 62, 62, 32)       896       \n",
         "                                                    \n",
         " max_pooling2d (MaxPooling2D  (None, 31, 31, 32)      0        \n",
         " )                                                  \n",
         "                                                    \n",
         " conv2d_1 (Conv2D)          (None, 29, 29, 32)       9248      \n",
         "                                                    \n",
         " max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)      0        \n",
         " 2D)                                                \n",
         "                                                    \n",
         " flatten (Flatten)          (None, 6272)             0        \n",
         "                                                    \n",
         " dense (Dense)              (None, 128)              802944    \n",
         "                                                    \n",
         " dense_1 (Dense)            (None, 5)                645       \n",
         "                                                    \n",
         " flatten_1 (Flatten)        (None, 5)                0        \n",
         "                                                    \n",

"=================================================================
===\n",
         "Total params: 813,733\n",
         "Trainable params: 813,733\n",
         "Non-trainable params: 0\n",
         "_________________________\n"
        ]
       }
      ],
```

```
   "source": [
    "model.summary()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "EQirf5FewdjE"
   },
   "outputs": [],
   "source": [
    "# adding dense layer"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "2tPWSWhNwgGB"
   },
   "outputs": [],
   "source": [
    "#hidden layer"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "gE4dkAxfwlQU"
   },
   "outputs": [],
   "source": [

"model.add(Dense(units=300,kernel_initializer=\"random_uniform\",activation=\"relu\"))"
```

```
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "id": "Qa_XY5iiwwnX"
    },
    "outputs": [],
    "source": [

"model.add(Dense(units=200,kernel_initializer=\"random_uniform\",activati
on=\"relu\"))"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "id": "LK3wwTiKw5D0"
    },
    "outputs": [],
    "source": [
     "#output layer"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "0tEhMxf-w9mU",
     "outputId": "75ff58d8-a81d-4a9e-d08b-669a7ad64c10"
    },
    "outputs": [
```

```
      {
       "output_type": "execute_result",
       "data": {
        "text/plain": [
         "129"
        ]
       },
       "metadata": {},
       "execution_count": 30
      }
     ],
     "source": [

"model.add(Dense(units=4,kernel_initializer=\"random_uniform\",activation
=\"softmax\"))\n",
      "len(x_train)"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "metadata": {
      "id": "yV6nAWK2xC2e"
     },
     "outputs": [],
     "source": [
      "#Ann starts so need to add dense layers"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "metadata": {
      "id": "ej3QucuhxImk"
     },
     "outputs": [],
     "source": [
```

```
"model.add(Dense(units=128,activation=\"relu\",kernel_initializer=\"rando
m_uniform\"))"
     ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "id": "f_cjd0eTxXa1"
    },
    "outputs": [],
    "source": [

"model.add(Dense(units=1,activation=\"sigmoid\",kernel_initializer=\"rando
m_uniform\"))"
     ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "id": "q846LaeFx3BK"
    },
    "outputs": [],
    "source": [
     "#Compile the model\n",

"model.compile(loss=\"binary_crossentropy\",optimizer=\"adam\",metrics=
['accuracy'])"
     ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "id": "4fAss-XEyHCe"
```

```
   },
   "outputs": [],
   "source": [
    "#Train the model"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "hgVQdW_cyb9l",
    "outputId": "01e2b5a1-f81a-4547-bf21-21e5814100dc"
   },
   "outputs": [
    {
     "metadata": {
      "tags": null
     },
     "name": "stderr",
     "output_type": "stream",
     "text": [
      "/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.\n",
      "  \"\"\"Entry point for launching an IPython kernel.\n"
     ]
    },
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "Epoch 1/20\n",
      "129/129 [==============================] - 2459s 19s/step - loss: -0.0526 - accuracy: 0.3273 - val_loss: 0.1126 - val_accuracy: 0.4467\n",
```

"Epoch 2/20\n",
"129/129 [==============================] - 36s 277ms/step - loss: -3.0746 - accuracy: 0.3288 - val_loss: 0.2155 - val_accuracy: 0.4467\n",
"Epoch 3/20\n",
"129/129 [==============================] - 35s 268ms/step - loss: -8.7866 - accuracy: 0.3288 - val_loss: 0.5095 - val_accuracy: 0.4467\n",
"Epoch 4/20\n",
"129/129 [==============================] - 36s 281ms/step - loss: -17.7107 - accuracy: 0.3288 - val_loss: 0.9337 - val_accuracy: 0.4467\n",
"Epoch 5/20\n",
"129/129 [==============================] - 36s 282ms/step - loss: -29.8704 - accuracy: 0.3288 - val_loss: 1.4811 - val_accuracy: 0.4467\n",
"Epoch 6/20\n",
"129/129 [==============================] - 36s 277ms/step - loss: -45.0273 - accuracy: 0.3288 - val_loss: 2.1422 - val_accuracy: 0.4467\n",
"Epoch 7/20\n",
"129/129 [==============================] - 35s 269ms/step - loss: -62.9152 - accuracy: 0.3288 - val_loss: 2.9106 - val_accuracy: 0.4467\n",
"Epoch 8/20\n",
"129/129 [==============================] - 40s 309ms/step - loss: -83.5868 - accuracy: 0.3288 - val_loss: 3.7855 - val_accuracy: 0.4467\n",
"Epoch 9/20\n",
"129/129 [==============================] - 36s 281ms/step - loss: -106.7443 - accuracy: 0.3288 - val_loss: 4.7640 - val_accuracy: 0.4467\n",
"Epoch 10/20\n",
"129/129 [==============================] - 36s 278ms/step - loss: -132.3641 - accuracy: 0.3288 - val_loss: 5.8398 - val_accuracy: 0.4467\n",
"Epoch 11/20\n",
"129/129 [==============================] - 35s 271ms/step - loss: -160.3758 - accuracy: 0.3288 - val_loss: 7.0081 - val_accuracy: 0.4467\n",
"Epoch 12/20\n",
"129/129 [==============================] - 35s 269ms/step - loss: -190.6966 - accuracy: 0.3288 - val_loss: 8.2454 - val_accuracy: 0.4467\n",

```
          "Epoch 13/20\n",
          "129/129 [==============================] - 36s 279ms/step - loss: -223.1146 - accuracy: 0.3288 - val_loss: 9.6145 - val_accuracy: 0.4467\n",
          "Epoch 14/20\n",
          "129/129 [==============================] - 36s 280ms/step - loss: -257.9082 - accuracy: 0.3288 - val_loss: 11.0088 - val_accuracy: 0.4467\n",
          "Epoch 15/20\n",
          "129/129 [==============================] - 37s 290ms/step - loss: -294.5687 - accuracy: 0.3288 - val_loss: 12.5175 - val_accuracy: 0.4467\n",
          "Epoch 16/20\n",
          "129/129 [==============================] - 34s 266ms/step - loss: -333.2441 - accuracy: 0.3288 - val_loss: 14.1130 - val_accuracy: 0.4467\n",
          "Epoch 17/20\n",
          "129/129 [==============================] - 36s 279ms/step - loss: -374.0325 - accuracy: 0.3288 - val_loss: 15.7641 - val_accuracy: 0.4467\n",
          "Epoch 18/20\n",
          "129/129 [==============================] - 36s 278ms/step - loss: -416.7053 - accuracy: 0.3288 - val_loss: 17.5287 - val_accuracy: 0.4467\n",
          "Epoch 19/20\n",
          "129/129 [==============================] - 35s 267ms/step - loss: -461.2285 - accuracy: 0.3288 - val_loss: 19.3238 - val_accuracy: 0.4467\n",
          "Epoch 20/20\n",
          "129/129 [==============================] - 34s 265ms/step - loss: -507.5266 - accuracy: 0.3288 - val_loss: 21.2192 - val_accuracy: 0.4467\n"
        ]
       },
       {
        "output_type": "execute_result",
        "data": {
```

```
          "text/plain": [
            "<keras.callbacks.History at 0x7f5c66ea6f50>"
          ]
        },
        "metadata": {},
        "execution_count": 36
      }
    ],
    "source": [
      "model.fit_generator(x_train,steps_per_epoch=len(x_train), validation_data=x_test, validation_steps=len(x_test), epochs= 20)"
    ]
  },
  {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
      "id": "5nrwRs8k5rSf"
    },
    "outputs": [],
    "source": [
      "model.save(\"nutrition.h5\")"
    ]
  },
  {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
      "id": "JR93P4teGyAb"
    },
    "outputs": [],
    "source": [
      "#Prediction the result"
    ]
  },
  {
    "cell_type": "code",
```

```
   "execution_count": null,
   "metadata": {
    "id": "qCIJVUjdGzw9"
   },
   "outputs": [],
   "source": [
    "from tensorflow.keras.models import load_model\n",
    "from keras.preprocessing  import image\n",
    "model =load_model(\"nutrition.h5\")"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "2f9AzoEwKLqB"
   },
   "outputs": [],
   "source": [
    "import numpy as np\n"
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "from tensorflow.keras.utils import load_img\n",
    "from tensorflow.keras.utils import img_to_array\n",
    "#loading of the image\n",
    "img = load_img(r'/content/drive/MyDrive/ibm project/Sample_Images-
20221102T071233Z-001/Sample_Images/Test_Image3.jpg',
grayscale=False,target_size=(64,64))\n",
    "#image to array \n",
    "x = img_to_array(img)\n",
    "#changing the shape\n",
    "x= np.expand_dims(x,axis = 0)\n",
    "predict_x=model.predict(x)\n",
    "classes_x=np.argmax(predict_x,axis = -1)\n",
```

```
      "classes_x"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "id": "CPvf0dfowTAL",
      "outputId": "1855f68a-13eb-4a61-9baa-93b3e31eb9f9"
     },
     "execution_count": null,
     "outputs": [
      {
       "output_type": "stream",
       "name": "stdout",
       "text": [
        "1/1 [==============================] - 0s 166ms/step\n"
       ]
      },
      {
       "output_type": "execute_result",
       "data": {
        "text/plain": [
         "array([0])"
        ]
       },
       "metadata": {},
       "execution_count": 48
      }
     ]
    },
    {
     "cell_type": "code",
     "source": [
      "index=['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE',
'WATERMELON']\n",
      "result=str(index[classes_x[0]])\n",
      "result"
```

```
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 36
     },
     "id": "3LzViysVEDln",
     "outputId": "0c9c54b0-fe74-479e-9a7c-51083f302ff4"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "'APPLES'"
       ],
       "application/vnd.google.colaboratory.intrinsic+json": {
        "type": "string"
       }
      },
      "metadata": {},
      "execution_count": 49
     }
    ]
   }
  ],
  "metadata": {
   "colab": {
    "provenance": []
   },
   "kernelspec": {
    "display_name": "Python 3",
    "name": "python3"
   },
   "language_info": {
    "name": "python"
```

```
    }
   },
   "nbformat": 4,
   "nbformat_minor": 0
}
```