FOODSNAP - A DEEP LEARNING POWERED DIETARY MANAGEMENT AND FOOD ANALYSIS APPLICATION

Submitted for partial fulfillment of the requirements

for the award of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING – ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

by

Shaik Esub	-	**21BQ5A4206**
Pathan Sattar Khan	-	**21BQ5A4204**
Appala Sai Srija	-	**21BQ5A4201**
Prathipati Sandeep	-	**21BQ5A4205**

Under the guidance of

Dr. V. Muralidhar

Associate Professor

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING - ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Permanently Affiliated to JNTU Kakinada, Approved by AICTE

Accredited by NAAC with 'A' Grade, ISO 9001:2008 Certified

NAMBUR (V), PEDAKAKANI (M), GUNTUR – 522 508

Tel no: 0863-2118036, url: www.vvitguntur.com

March-April 2024

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY
Permanently Affiliated to JNTUK, Kakinada, Approved by AICTE
Accredited by NAAC with 'A' Grade, ISO 9001:20008 Certified
Nambur, Pedakakani (M), Guntur (Gt) -522508

DEPARTMENT OF CSE-ARTIFICIAL INTELLIGENCE& MACHINE LEARNING

CERTIFICATE

This is to certify that this **Project Report** is the bonafide work of **Mr. Shaik Esub, Mr. Pathan Sattar Khan, Ms. Appala Sai Srija, Mr. Prathipathi Sandeep**, bearing Reg. No. **21BQ5A4206, 21BQ5A4204, 21BQ5A4201, 21BQ5A4205** respectively who had carried out the project entitled **"FOODSNAP - A Deep Learning Powered Dietary Management And Food Analysis Application "** under our supervision.

Project Guide
(Dr. V. Muralidhar, Associate Professor)

Head of the Department
(Dr. K. Suresh Babu , Professor)

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, Mr. Shaik Esub, Mr. Pathan Sattar Khan, Ms. Appala Sai Srija, Mr. Prathipati Sandeep, hereby declare that the Project Report entitled **"FOODSNAP - A Deep Learning Powered Dietary Management And Food Analysis Application"** done by us under the guidance of Dr. V. Muralidhar, Associate Professor, CSE-Artificial Intelligence & Machine Learning at Vasireddy Venkatadri Institute of Technology is submitted for partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science Engineering - Artificial Intelligence & Machine Learning. The results embodied in this report have not been submitted to any other University for the award of any degree.

DATE :

PLACE : Nambur

SIGNATURE OF THE CANDIDATE (S)

Shaik Esub,

Pathan Sattar Khan,

Appala Sai Srija,

Prathipati Sandeep

ACKNOWLEDGEMENT

We take this opportunity to express my deepest gratitude and appreciation to all those people who made this project work easier with words of encouragement, motivation, discipline, and faith by offering different places to look to expand my ideas and helped me towards the successful completion of this project work.

First and foremost, we express my deep gratitude to **Sri. Vasireddy Vidya Sagar**, Chairman, Vasireddy Venkatadri Institute of Technology for providing necessary facilities throughout the B.Tech programme.

We express my sincere thanks to **Dr. Y. Mallikarjuna Reddy**, Principal, Vasireddy Venkatadri Institute of Technology for his constant support and cooperation throughout the B.Tech programme.

We express my sincere gratitude to **Dr. K. Suresh Babu**, Professor & HOD, Computer Science Engineering – Artificial Intelligence & Machine Learning Vasireddy Venkatadri Institute of Technology for his constant encouragement, motivation and faith by offering different places to look to expand my ideas.

We would like to express my sincere gratefulness to our Guide **Dr. V. Muralidhar**, Associate Professor, CSE-Artificial Intelligence & Machine Learning for his insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project.

We would like to express our sincere heartfelt thanks to our Project Coordinator **Mr. N. Balayesu**, Associate Professor, CSE-Artificial Intelligence & Machine Learning for his valuable advices, motivating suggestions, moral support, help and coordination among us in successful completion of this project.

We would like to take this opportunity to express my thanks to the **Teaching and Non-Teaching** Staff in the Department of Computer Science Engineering -Artificial Intelligence and Machine Learning, VVIT for their invaluable help and support.

Name (s) of Students

Shaik Esub

Pathan Sattar Khan

Appala Sai Srija

Prathipati Sandeep

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

DL	Deep Learning
BMI	Body Mass Index
GPT	Generative Pre-trained transformer
ML	Machine Learning
HTTP	Hyper Text Transfer Protocol
API	Application Programming Interface
YOLO v8	You Only Look Once (Version 8.0)

ABSTRACT

The main concept is to develop an innovative and comprehensive app that aims to revolutionize dietary management through the use of Deep Learning and Natural Language Processing (NLP) techniques. The app utilizes a GPT Based Sentence Classifier and a custom-trained Deep Learning Model for food detection, enabling users to accurately record and monitor their food intake in a user-friendly manner. The app's diverse database provides users with detailed nutritional information, including calories, carbohydrates, fats, and proteins. It also incorporates a variety of charts and graphs to display variations in nutritional information for each food intake and perform analytics over the food data.

Additionally, the app offers personalized dietary recommendations, Food Allergen Alerts, Report Generation, and goal setting for weight gain and weight loss, empowering users to make informed choices about their dietary habits and health.

Keywords: Deep Learning, Dietary Management, Natural Language Processing, Food Allergen Alerts, Dietary Recommendations, Generative pre-trained transformer.

CHAPTER 1
INTRODUCTION

1.1 WHAT IS DIETARY MANAGEMENT?

Dietary management is a proactive approach to fostering health and well-being through conscious choices in food consumption. It involves the deliberate selection of foods and beverages, considering factors such as nutrient balance, caloric control, and individual health goals. By tailoring dietary habits to specific needs, individuals can address medical conditions, achieve weight management objectives, and support overall health. This practice extends beyond mere food choices, encompassing considerations like portion sizes, hydration, and the incorporation of lifestyle factors such as exercise and stress management. Nutritionists and dietitians often play a pivotal role in providing personalized guidance, ensuring that individuals make informed decisions aligned with their unique health requirements. In essence, dietary management is a holistic and proactive approach to nourishing the body, promoting optimal health, and preventing or managing various health conditions.

Some key components of dietary management are:

1. **Nutritional Assessment:** Evaluating an individual's diet, health history, and lifestyle to identify deficiencies or imbalances, using methods like dietary recalls and measurements.
2. **Goal Setting:** Establishing specific dietary objectives based on assessment findings, such as weight management or addressing medical conditions.
3. **Meal Planning:** Creating balanced meal plans considering nutritional needs, preferences, and cultural background, selecting foods from different groups.
4. **Portion Control:** Managing caloric intake and weight by controlling serving sizes, utilizing tools or cues to estimate appropriate portions.
5. **Dietary Modifications:** Adjusting diets to meet health goals, like reducing certain ingredients and increasing nutrient-rich foods, tailored to individual needs.
6. **Monitoring and Adjustments:** Regularly tracking dietary intake and progress, making changes as needed based on feedback and health status, with ongoing support for adherence.

1.2 WHAT IS FOOD ANALYSIS?

Food analysis is a multidisciplinary field that involves examining various aspects of food to understand its composition, quality, safety, and nutritional value. It involves the use of various scientific techniques and methodologies to analyze the chemical, physical, and microbiological properties of foods. Food analysis is crucial for ensuring food safety, meeting regulatory standards, verifying label claims, and maintaining product quality.

Food analysis involves determining the nutrient content of food products, including energy (calories), macronutrients (carbohydrates, proteins, fats), micronutrients (vitamins, minerals), and other bioactive compounds. Nutritional labeling regulations require accurate assessment of nutrient levels in packaged foods. Nutritional analysis may be performed using laboratory methods, databases, or software programs that calculate nutrient values based on ingredient composition.

1.3 WHAT IS DEEP LEARNING?

Deep learning is a subset of machine learning which is based on artificial neural network architecture. An artificial neural network or ANN uses layers of interconnected nodes called neurons that work together to process and learn from the input data.

In a fully connected Deep neural network, there is an input layer and one or more hidden layers connected one after the other. Each neuron receives input from the previous layer neurons or the input layer. The output of one neuron becomes the input to other neurons in the next layer of the network, and this process continues until the final layer produces the output of the network. The layers of the neural network transform the input data through a series of nonlinear transformations, allowing the network to learn complex representations of the input data.

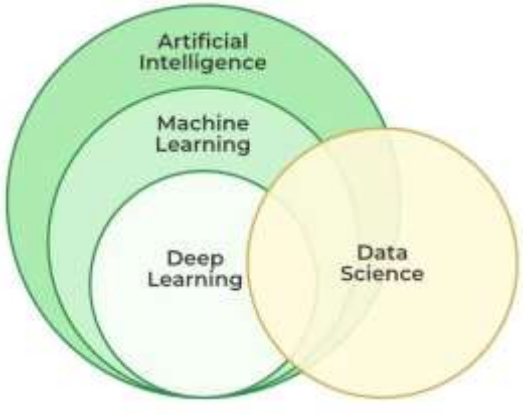


Fig 1.1: Deep Learning

Deep learning can be used for supervised, unsupervised as well as reinforcement machine learning. it uses a variety of ways to process these.

Supervised Machine Learning: Supervised machine learning is the machine learning technique in which the neural network learns to make predictions or classify data based on the labeled datasets. Here we input both input features along with the target variables. the neural network learns to make predictions based on the cost or error that comes from the difference between the predicted and the actual target, this process is known as backpropagation. Deep learning algorithms like Convolutional neural networks, Recurrent neural networks are used for many supervised tasks like image classifications and recognization, sentiment analysis, language translations, etc.

Unsupervised Machine Learning: Unsupervised machine learning is the machine learning technique in which the neural network learns to discover the patterns or to cluster the dataset based on unlabeled datasets. Here there are no target variables. while the machine has to self-determined the hidden patterns or relationships within the datasets. Deep learning algorithms like autoencoders and generative models are used for unsupervised tasks like clustering, dimensionality reduction, and anomaly detection.

Reinforcement Machine Learning: Reinforcement Machine Learning is the machine learning technique in which an agent learns to make decisions in an environment to maximize a reward signal. The agent interacts with the environment by taking action and observing the resulting rewards. Deep learning can be used to learn policies, or a set of actions, that maximizes the cumulative reward over time. Deep reinforcement learning algorithms like Deep Q networks and Deep Deterministic Policy Gradient (DDPG) are used to reinforce tasks like robotics and game playing etc.

Advantages:

1. High accuracy
2. Automated feature engineering
3. Scalability
4. Flexibility
5. Continual improvement

Disadvantages:

1. High Computational Cost
2. Overfitting
3. Limited to Training Data
4. Black-Box Models

1.4 DEEP LEARNING APPLICATIONS

The main applications of deep learning can be divided into computer vision, natural language processing (NLP), and reinforcement learning.

Computer vision:

In computer vision, Deep learning models can enable machines to identify and understand visual data. Some of the main applications of deep learning in computer vision include:

- **Object detection and recognition**: Deep learning model can be used to identify and locate objects within images and videos, making it possible for machines to perform tasks such as self-driving cars, surveillance, and robotics.
- **Image classification:** Deep learning models can be used to classify images into categories such as animals, plants, and buildings. This is used in applications such as medical imaging, quality control, and image retrieval.
- **Image segmentation:** Deep learning models can be used for image segmentation into different regions, making it possible to identify specific features within images.

Natural language processing (NLP):

In NLP, the Deep learning model can enable machines to understand and generate human language. Some of the main applications of deep learning in NLP include:

- **Automatic Text Generation :** Deep learning model can learn the corpus of text and new text like summaries, essays can be automatically generated using these trained models.
- **Language translation:** Deep learning models can translate text from one language to another, making it possible to communicate with people from different linguistic backgrounds.

- **Sentiment analysis:** Deep learning models can analyze the sentiment of a piece of text, making it possible to determine whether the text is positive, negative, or neutral. This is used in applications such as customer service, social media monitoring, and political analysis.
- **Speech recognition:** Deep learning models can recognize and transcribe spoken words, making it possible to perform tasks such as speech-to-text conversion, voice search, and voice-controlled devices.

Reinforcement learning:

In reinforcement learning, deep learning works as training agents to take action in an environment to maximize a reward. Some of the main applications of deep learning in reinforcement learning include:

- **Game playing**: Deep reinforcement learning models have been able to beat human experts at games such as Go, Chess, and Atari.
- **Robotics:** Deep reinforcement learning models can be used to train robots to perform complex tasks such as grasping objects, navigation, and manipulation.
- **Control systems:** Deep reinforcement learning models can be used to control complex systems such as power grids, traffic management, and supply chain optimization.

1.5 DEEP LEARNING TECHNIQUES

Deep learning encompasses a variety of techniques and architectures that enable computers to learn from data and make predictions or decisions. Here are some common deep learning techniques

- **Artificial Neural Networks (ANNs):** ANNs are the foundation of deep learning. They consist of interconnected nodes organized in layers, including an input layer, one or more hidden layers, and an output layer. Each node applies a nonlinear activation function to its input, allowing the network to model complex relationships in the data.

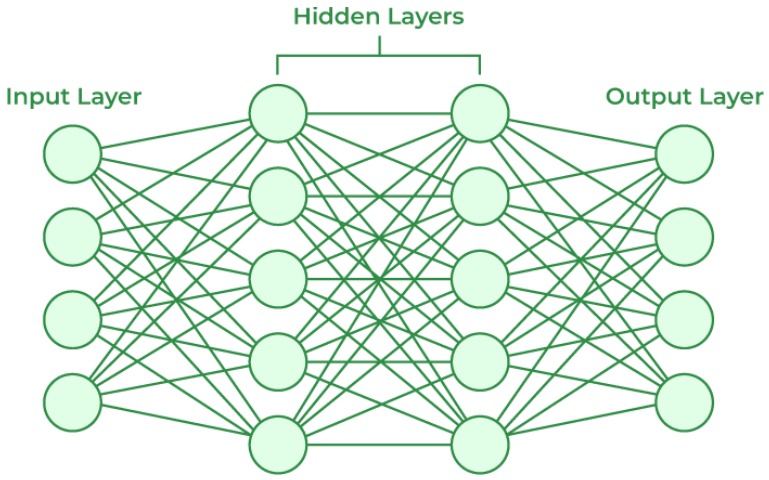


Fig 1.2: Artificial Neural Networks (ANNs)

- **Convolutional Neural Networks (CNNs):** CNNs are specialized neural networks designed for processing grid-like data such as images and videos. They use convolutional layers to extract spatial hierarchies of features from the input data, enabling tasks such as image classification, object detection, and image segmentation.

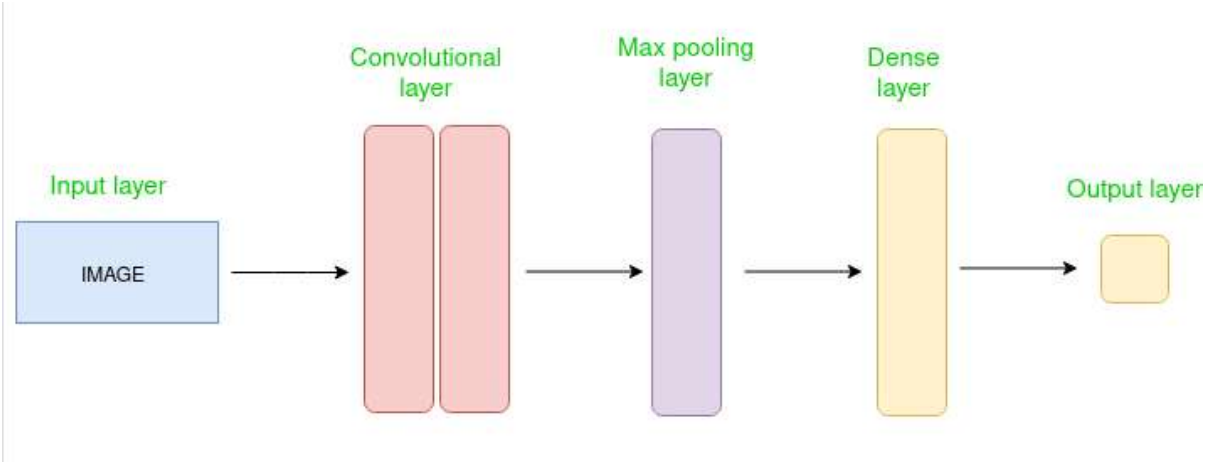


Fig 1.3: Convolutional Neural Networks (CNNs)

- **Recurrent Neural Networks (RNNs):** RNNs are designed to handle sequential data with temporal dependencies, such as time series, text, and speech. They have feedback connections that allow information to persist over time, making them suitable for tasks such as language modeling, speech recognition, and machine translation.

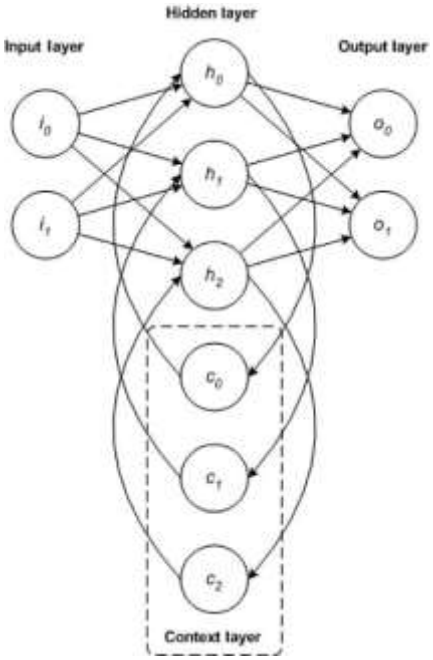


Fig 1.4: Recurrent Neural Networks (RNNs)

- **Long Short-Term Memory (LSTM) Networks:** LSTMs are a type of RNN designed to address the vanishing gradient problem and capture long-range dependencies in sequential data. They incorporate memory cells and gating mechanisms to selectively retain and update information over time, making them effective for tasks such as speech recognition, handwriting recognition, and sentiment analysis.

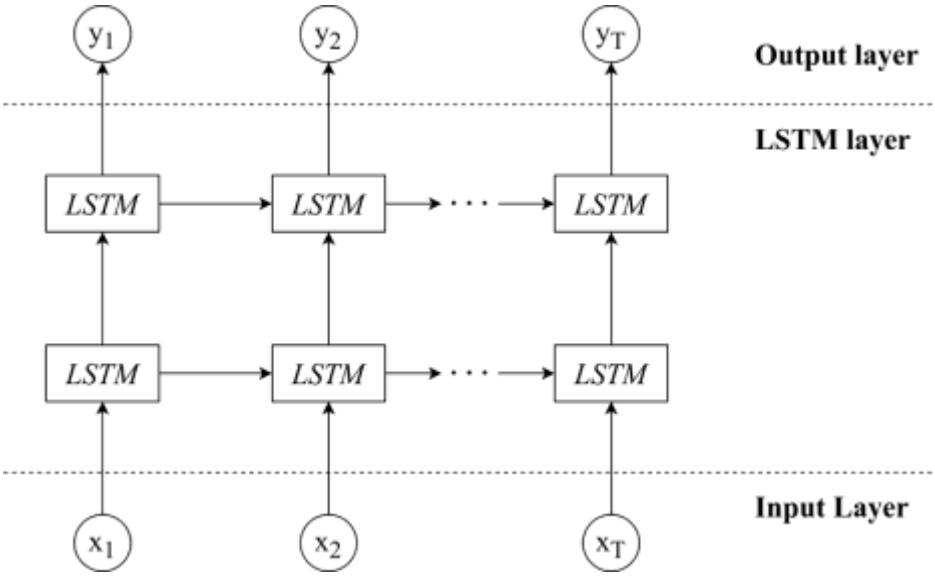


Fig 1.5: Long Short-Term Memory (LSTM) Networks

- **Generative Adversarial Networks (GANs):** GANs consist of two neural networks, a generator and a discriminator, trained adversarially to generate realistic data samples. They are used for tasks such as image generation, image-to-image translation, and data augmentation.
- **Autoencoders:** Autoencoders are neural networks trained to reconstruct input data at the output layer, typically with a bottleneck layer that learns a compressed representation of the data. They are used for tasks such as data compression, denoising, and feature learning.

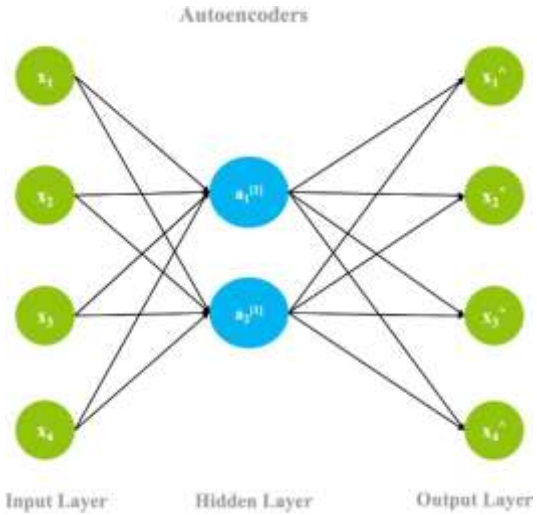


Fig 1.6: Autoencoders

- **Deep Reinforcement Learning (DRL):** DRL combines deep learning with reinforcement learning, where an agent learns to interact with an environment to maximize a reward signal. DRL has achieved remarkable success in areas such as game playing, robotics, and autonomous systems.
- **Transformers:** Transformers are attention-based models originally designed for natural language processing tasks. They use self-attention mechanisms to capture long-range dependencies in sequences, enabling tasks such as machine translation, text generation, and document summarization.
- **Capsule Networks:** Capsule networks are an alternative architecture to CNNs, designed to capture hierarchical relationships between parts of objects in images. They aim to overcome some limitations of traditional CNNs in tasks such as object recognition and pose estimation.

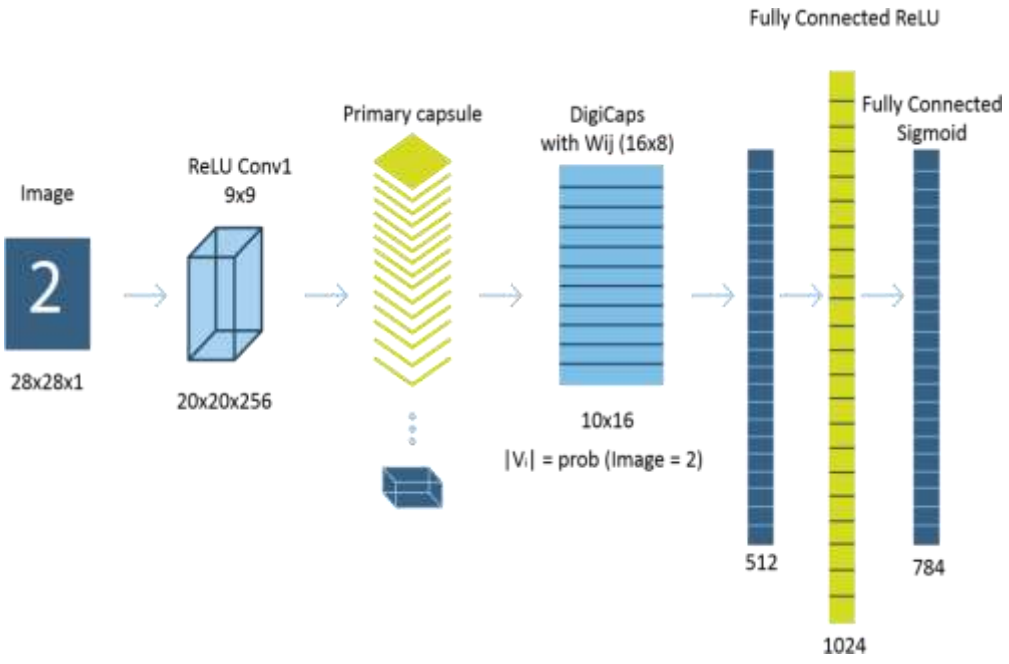


Fig 1.7: Capsule Networks

1.6 AIM:

The main concept is to develop an innovative and comprehensive app that aims to revolutionize dietary management using Deep Learning and Natural Language Processing (NLP) techniques. The app utilizes a GPT Based Sentence Classifier and a custom-trained Deep Learning Model for food detection, enabling users to accurately record and monitor their food intake in a user-friendly manner. The app's diverse database provides users with detailed nutritional information, including calories, carbohydrates, fats, and proteins. It also incorporates a variety of charts and graphs to display variations in nutritional information for each food intake and perform analytics over the food data.

Additionally, the app offers personalized dietary recommendations, Food Allergen Alerts, Report Generation, and goal setting for weight gain and weight loss, empowering users to make informed choices about their dietary habits and health.

1.7 PROCESS:

1. First, we develop a custom-trained Deep Learning Model for food detection and fine-tune the model using labeled food images to train it to recognize different food items.

2. Next, we developed a GPT-based sentence classifier to predict the food label from a description of food, and integrated both the food detection model and the GPT-based Sentence Classifier into the application.
3. We also used Food Recommendation algorithms to recommend food for a user based on their unique body constitution, such as BMI. We collected important information on food items enrolled in our database and displayed their advantages and disadvantages to the user while recommending the food item.
4. We also developed GPT-based Allergen Alerts to alert the user when a food item is not recommended to eat, using pre-recorded health issues of the user, enabling informed choices.
5. We developed a better UI including advanced navigation features to navigate to the required date to view their food log and history.
6. We included a Diet Report Generator in the application itself to generate a PDF report about their diet history, including Analytical Pie Charts and Nutrient Ratio. We utilized Cache to store the downloaded Reports in the application itself for viewing and deletion later.
7. We added a Goals feature to set goals such as Weight Loss and Weight Gain, enabling users to track progress over time and reach their goals within a prescribed duration based on their sedentary levels and some universal formulas.
8. We included publicly available Meal Plans in our application, enabling users to download and follow them for healthier lifestyles.
9. We added points for each food log to encourage users by providing a community ranking feature.
10. Overall, we have improved the UI for updating the User profile with new Information and added a Support ticket feature in case of any issues for the user.

1.8 Features:

- This project shows allows users to record their daily food intake accurately.
- The app maintains a diverse database of foods, providing detailed nutritional information for each item.
- Users can access data on calories, carbohydrates, fats, proteins, vitamins, and minerals
- The Features are classified as follows:

1. **User Registration and Profile:** The feature allows users to create accounts and personalize profiles with dietary preferences and personal information, facilitating a tailored experience.

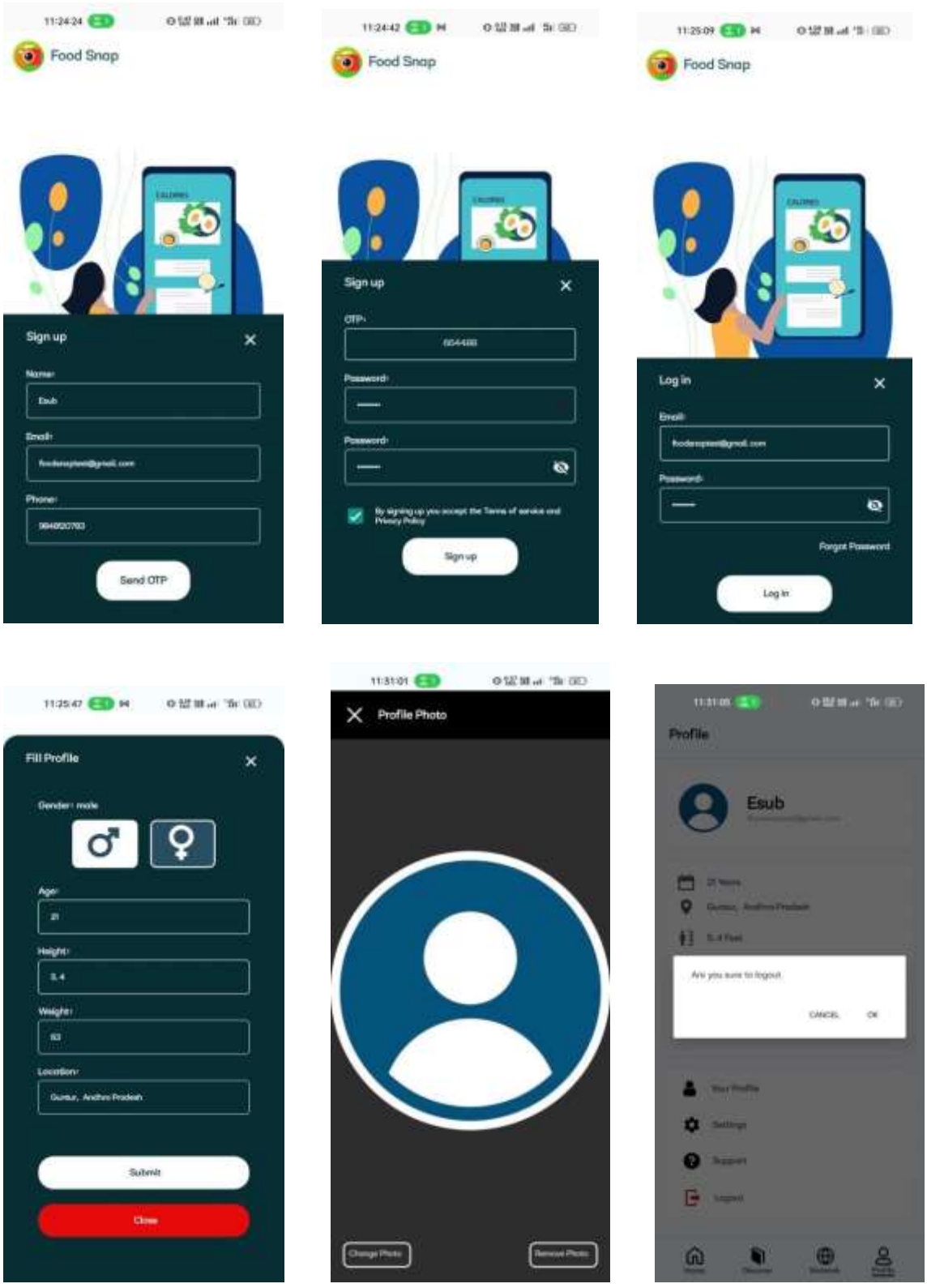


Fig 1.8: User Registration and Profile

2. **Nutritional Analysis:** Through advanced Deep Learning algorithms and a vast food database the application provides detailed nutritional information for food items, including calories, macronutrients, micronutrients composition, ensuring users have comprehensive insights into their dietary intake.

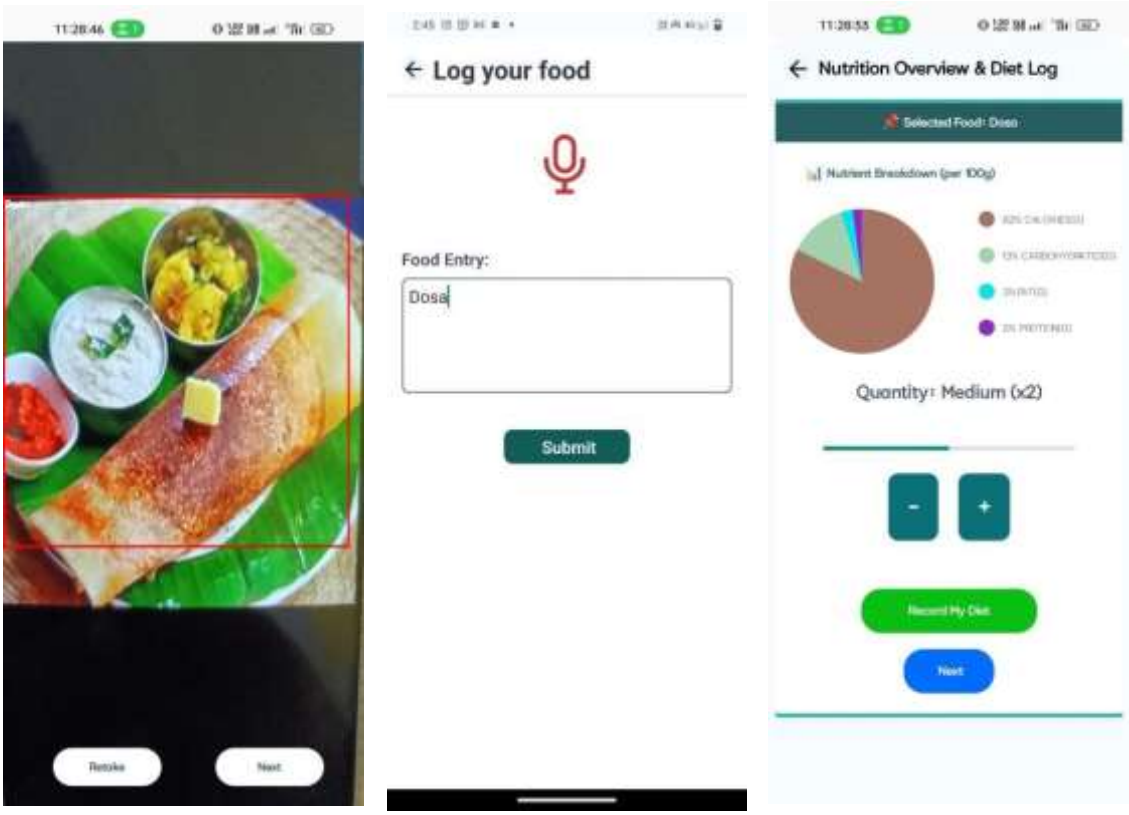


Fig 1.9: Nutritional Analysis

3. **Meal History and Tracking:** With this feature users can easily track their food intake over time, accessing a detailed history of meals consumed, aiding in understanding eating patterns and making informed dietary choices.

Fig 1.10 Meal History and Tracking

4. **Goals Setting and Tracking:** This feature empowers users to set dietary goals such as calorie intake, macronutrient distribution, or specific dietary targets, and tracks their progress towards achieving these goals, fostering accountability and motivation.

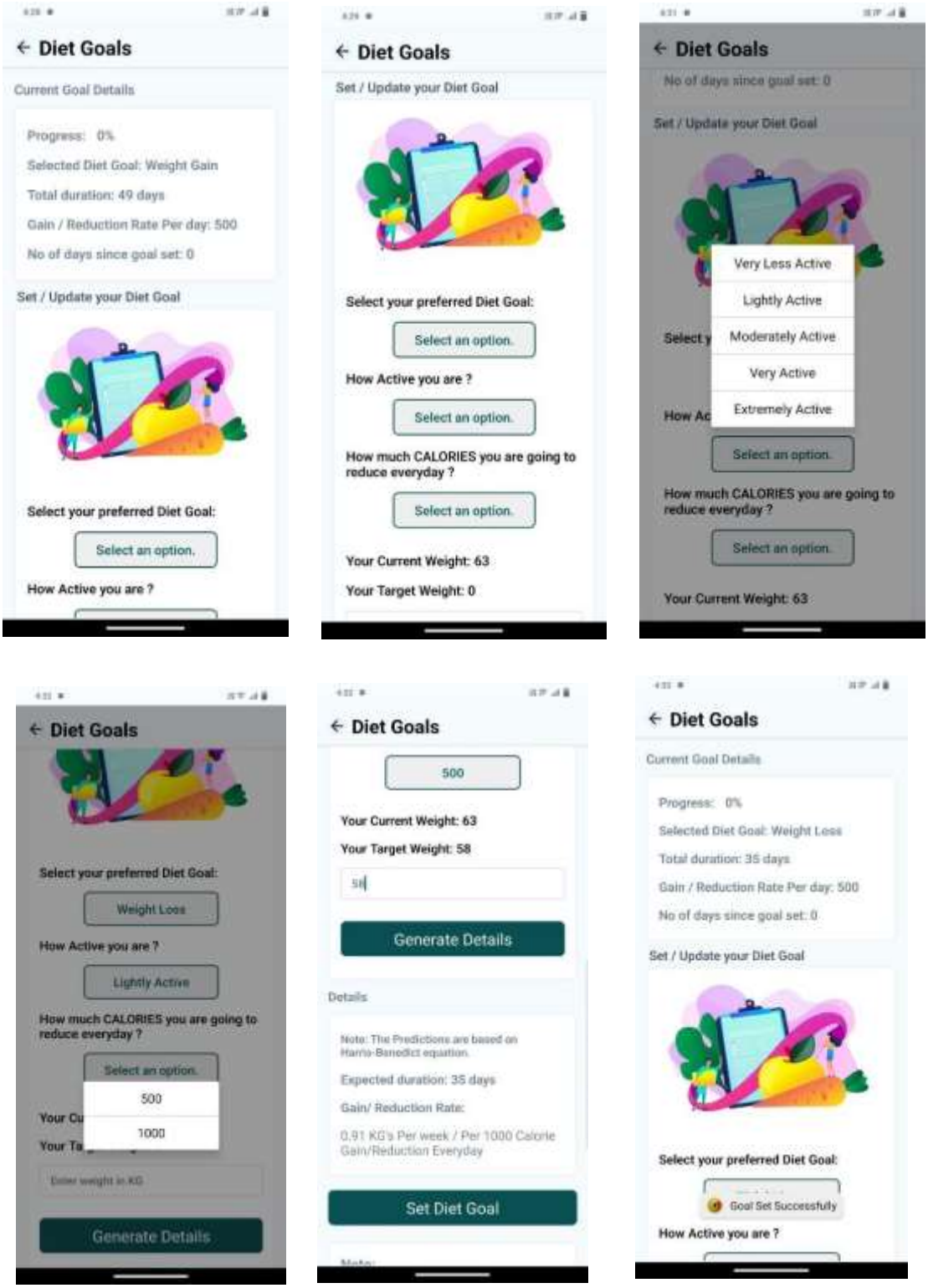


Fig 1.11: Goal Setting and Tracking

5. **Offline Functionality:** We included an Offline Functionality feature to log food data even when a user is not connected to the internet. It uploads the data to the cloud once an internet connection is established.

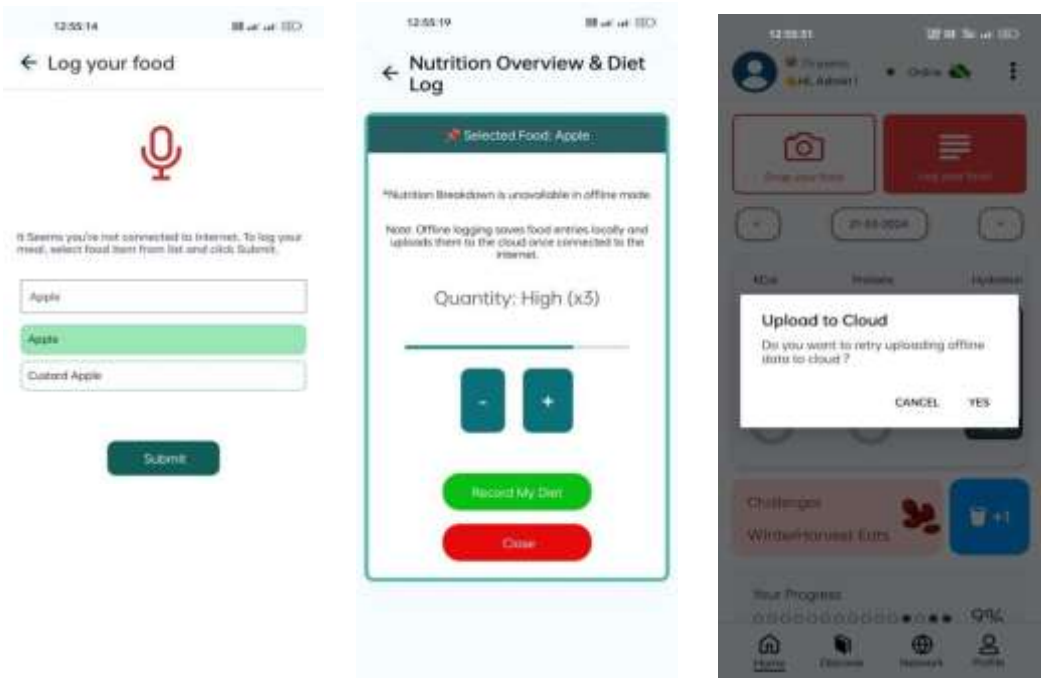


Fig 1.12: Offline Functionality

6. **Personalized Recommendations:** Leveraging user profiles and dietary data, the application generates personalized meal suggestions and dietary recommendations tailored to individual preferences and nutritional needs, guiding users towards healthier eating habits.

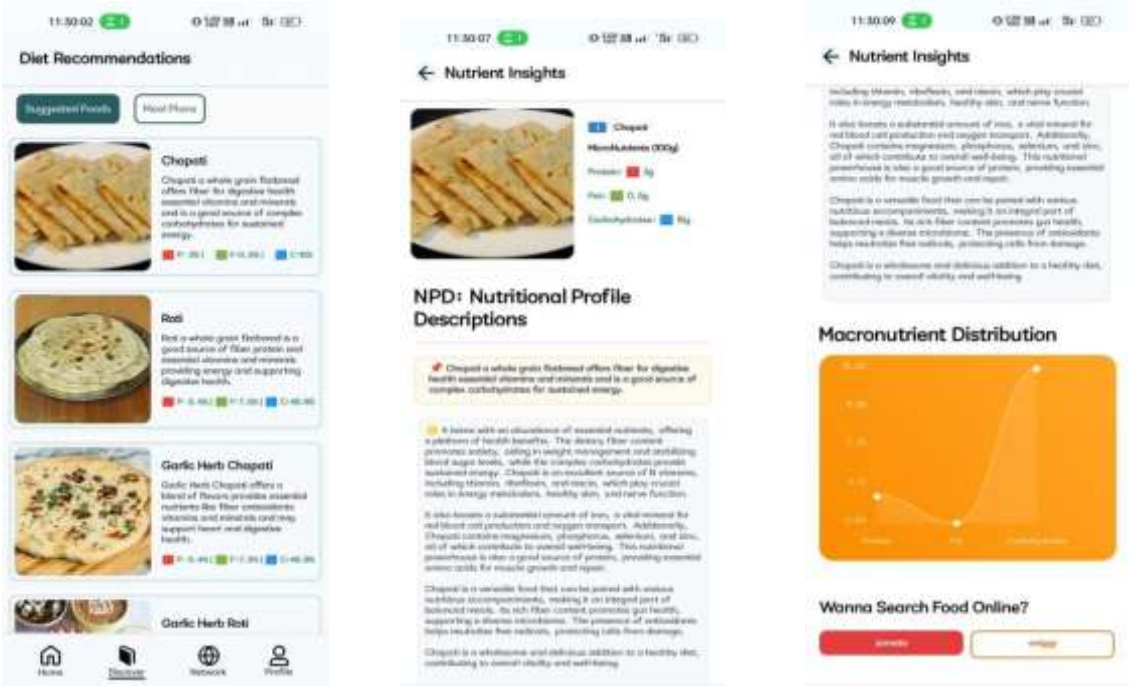


Fig 1.13: Personalized Recommendation

7. **Visual Analytics:** The analytics are displayed in Pie charts for each food intake. The intuitive navigation feature enables us to retrieve information with just a single click, and changes the charts and graphs dynamically based on the selected date.

Fig 1.14: Visual Analytics

8. **Reports and Reminders:** Reports Feature allows users to generate a diet report that include Protein: Fat: Carbohydrate Ratio, Personal Details and Analytical Information with our required Start and End dates. The Reminders allows the user to get alerts when a food item is not recommended to eat based on our health condition and to notify about the food log to the users.

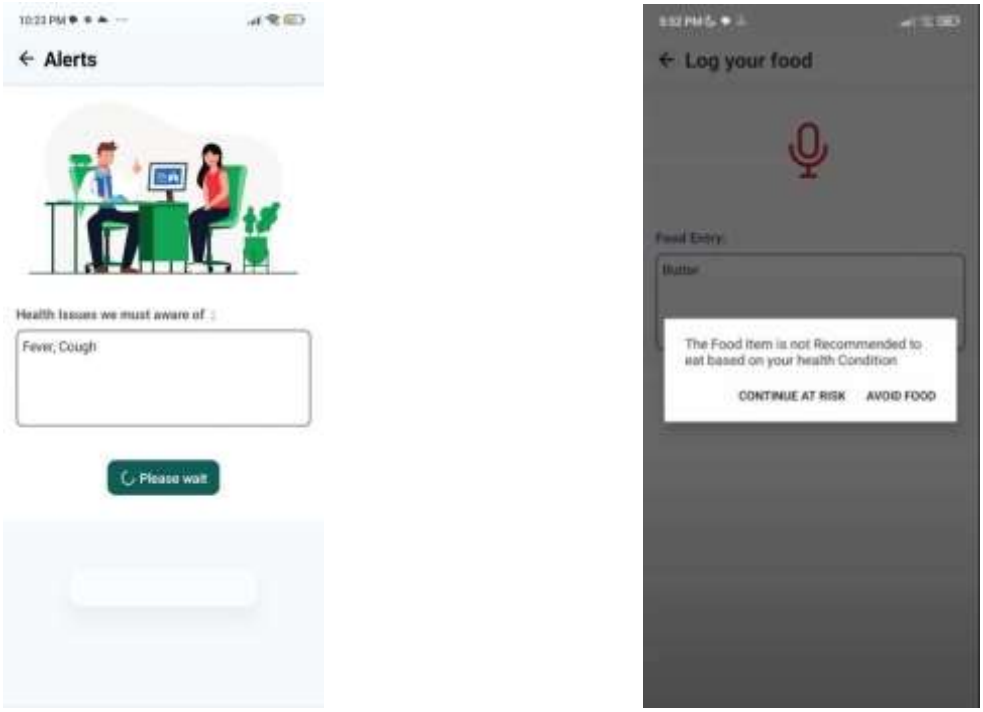
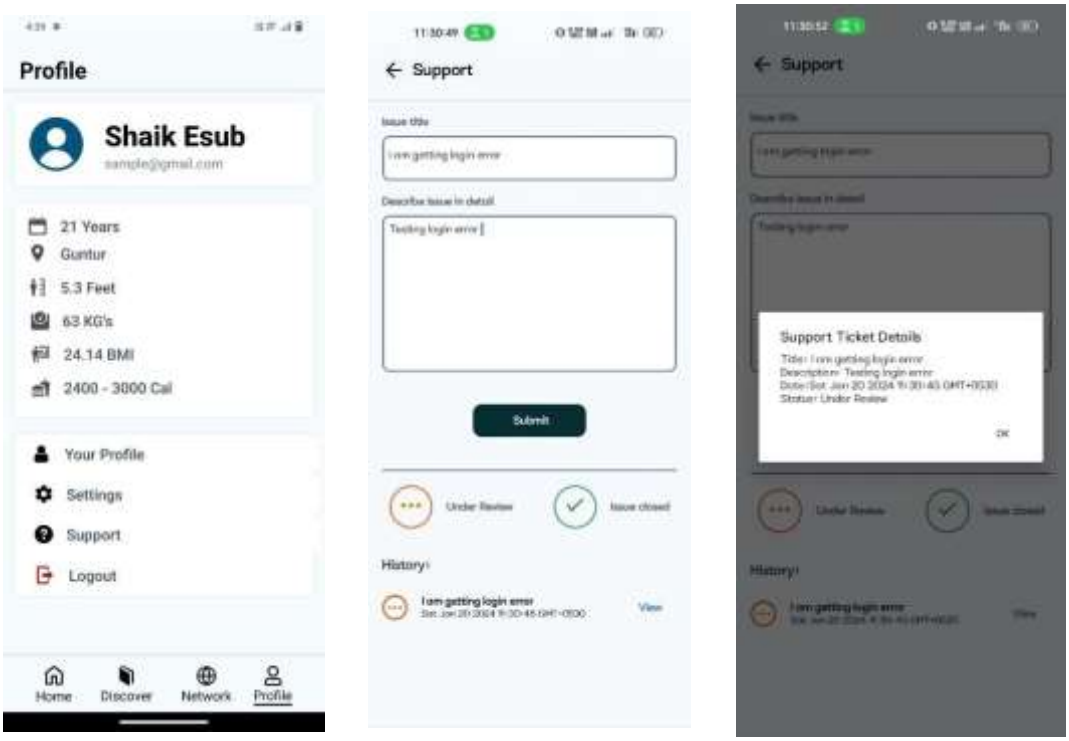


Fig 1.15: Reports and Remainder

9. **Optional Features:** The Profile page allows the user to update their personal details, The support page is used to describe any issues or difficulties they face while using the application to the developers/administrators and it also provides meal plans to follow for users and Community achievement points to encourage user participation.



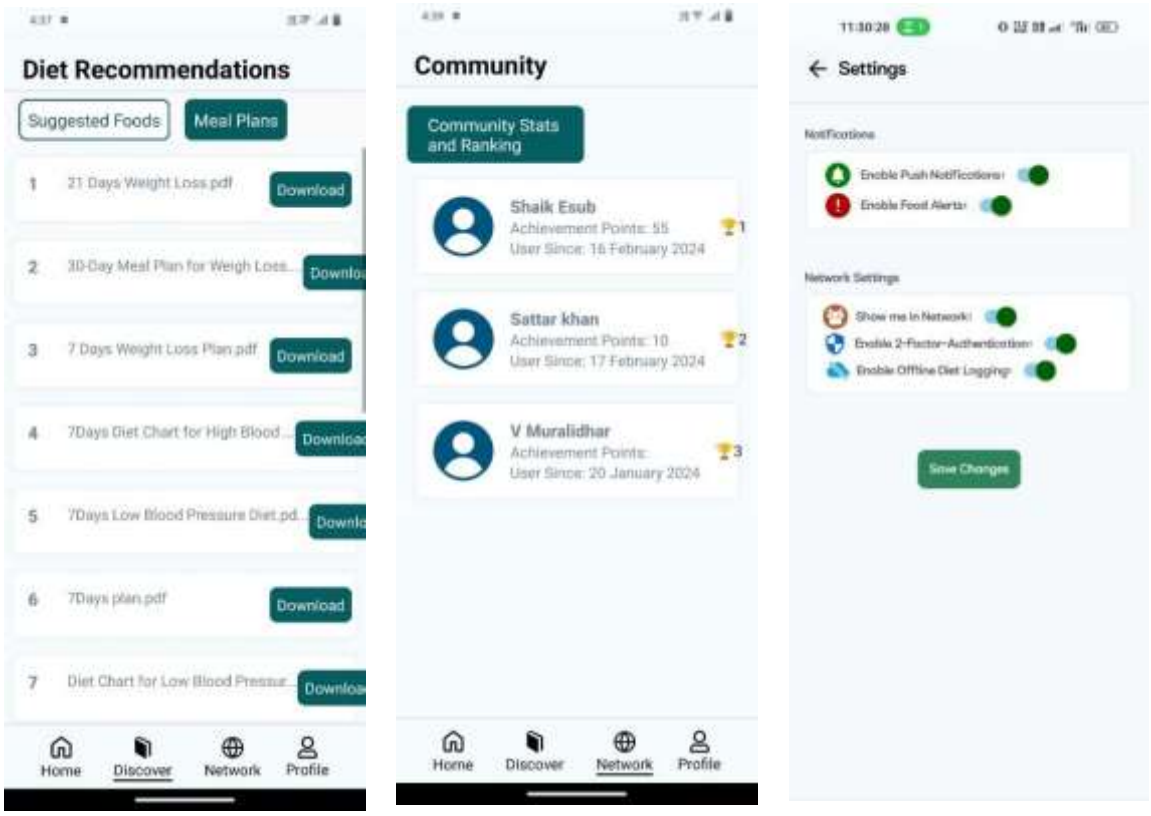


Fig 1.16: Optional Features

1.9 EXISTING SYSTEM:

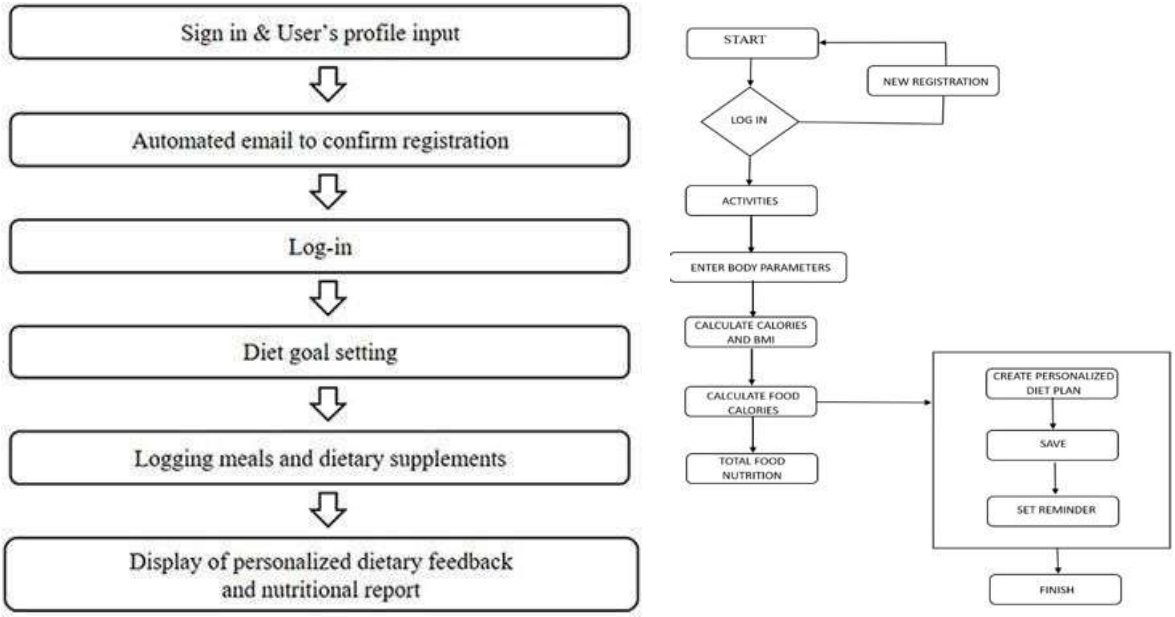


Fig: 1.17: Existing system

Foodsnap offers a unique approach to dietary management, it's important to acknowledge existing diet recall apps and their limitations. Here is an analysis of some common features and shortcomings:

Existing Apps:

- **MyFitnessPal (MFP):** A popular calorie tracker with a vast food database and features for logging meals, exercise, and weight. However, MFP relies on manual food entry, which can be time-consuming and error-prone. Additionally, it lacks features like allergen alerts support and personalized recommendations.
- **Cronometer:** Offers detailed nutritional information beyond just calories, including micronutrients and bioavailable vitamins/minerals. However, like MFP, it relies on manual food entry and may not cater to users with specific dietary needs.
- **MyPlate:** A government-backed app promoting healthy eating habits with functionalities like meal planning and recipe suggestions. MyPlate might lack the in-depth nutritional analysis and customization options offered by Foodsnap.

Limitations of Existing Apps:

- **Manual Food Entry:** Most existing apps rely on manual food logging, which can be tedious and inaccurate, especially for users with busy lifestyles.
- **Generic Recommendations:** One-size-fits-all dietary advice might not be suitable for everyone. Foodsnap addresses this by incorporating gender and universal formulas and for personalized recommendations.
- **Allergen and Diet Restriction Awareness:** Existing apps might not have robust features for managing allergies and dietary restrictions effectively.

These existing diet recall apps typically rely on manual data entry for recording food consumption and lack advanced features such as image recognition or personalized dietary recommendations. However, they remain widely used due to their established user base and comprehensive tracking capabilities.

Foodsnap aims to address these limitations by:

- **Image Recognition:** Eliminates the need for manual food entry, saving time and improving accuracy.
- **Personalized Recommendations:** Tailored dietary advice based on user profiles and their unique body constitution.
- **Allergen and Diet Restriction Management:** Integrates alerts and reminders to promote informed dietary choices.

By addressing these shortcomings, Foodsnap presents a comprehensive and user-friendly solution for dietary management, fostering healthier eating habits for a wider range of users.

1.10 GPT BASED SENTENCE CLASSIFIER:

In our project, we leverage the power of GPT (Generative Pre-trained Transformer) models to develop a sophisticated sentence classifier tailored for food description input. This focuses on how we utilize GPT models to enhance the user experience when manually entering meal details into our application**.**

1.10.1 Introduction to GPT Model:

Before delving into the specifics of our GPT-based sentence classifier, it's essential to understand what GPT models are and why they are instrumental in our project. GPT models are state-of-the-art language models developed by OpenAI, capable of understanding and generating human-like text based on the input they receive. These models are pre-trained on vast amounts of text data and fine-tuned for specific tasks, making them adept at understanding and generating natural language text.

1.10.2 Algorithm Overview:

Our GPT-based sentence classifier operates on a sophisticated algorithm that harnesses the power of transformer architectures. The algorithm can be broken down into several key steps:

1. **Tokenization:** The input text is tokenized, breaking it down into individual units such as words or subwords.
2. **Embedding:** Each token is converted into a high-dimensional vector representation called an embedding, capturing its semantic meaning.
3. **Self-Attention:** The model calculates attention scores between each pair of tokens, allowing it to focus on relevant parts of the input sequence.
4. **Feedforward Networks:** The attention-weighted representations are passed through feedforward neural networks, applying non-linear transformations.
5. **Output Generation:** The final layer generates a probability distribution over the vocabulary, predicting the likelihood of each token in the output sequence.

1.10.3 Integration into Our Application:

In our project, we integrate the GPT-based sentence classifier into our application's interface to enhance the manual entry of meal details. When users input food descriptions, the classifier

analyses the text in real time and categorizes it based on its relevance to food items. This seamless integration allows users to describe their meals in natural language, ensuring efficient and accurate logging of dietary intake.

Example Usage:

For instance, a user might input the description "I ate food item which color is red and tastes sweet." Our GPT-based sentence classifier would recognize the description's characteristics and generate the most likely corresponding food item as the output, in this case, "apple." This example demonstrates the model's ability to process natural language input and generate relevant text output based on its learned knowledge and patterns in the training data.

1.11 YOLO v8 FOOD DETECTION MODEL [6]:

In our project, we have integrated the YOLOv8 (You Only Look Once version 8) pre-trained model for food detection, enabling our application to accurately identify food items in images captured by the user's smartphone camera.

1.11.1 Definition of YOLOv8:

YOLOv8 is a state-of-the-art object detection algorithm that belongs to the YOLO (You Only Look Once) [6] family of models. It is designed to perform real-time object detection by simultaneously predicting bounding boxes and class probabilities for multiple objects in an image. YOLOv8 builds upon previous iterations of the YOLO model, incorporating improvements in accuracy and speed.

1.11.2 Algorithm Overview:

The YOLOv8 algorithm can be summarized as follows:

1. **Input Image:** The algorithm takes an input image captured by the user's smartphone camera.
2. **Image Processing*:*** The input image is preprocessed to prepare it for object detection.
3. **Feature Extraction*:*** The preprocessed image is passed through a convolutional neural network (CNN), which extracts features at multiple scales.
4. **Detection:** YOLOv8 divides the input image into a grid of cells and predicts bounding boxes and class probabilities for objects within each cell. Unlike traditional object

detection methods that apply a sliding window approach, YOLOv8 performs detection using a single neural network evaluation, resulting in faster inference times.

5. **Non-Maximum Suppression (NMS*):*** After detecting objects, YOLOv8 applies NMS to remove redundant bounding boxes and retain only the most confident detections for each object class.
6. **Output*:*** The final output of the YOLOv8 model is a set of bounding boxes along with their associated class labels and confidence scores, indicating the presence and location of food items in the input image.

1.11.3 Integration into Our Application:

In our project, we have trained the YOLOv8 pre-trained model with a custom food image dataset, fine-tuning it specifically for food detection. This trained model is seamlessly integrated into our application, allowing users to capture images of their meals using their smartphone cameras. The YOLOv8 model then analyzes these images in real-time and identifies the food items present, providing valuable information for dietary tracking and analysis.

1.11.4 Architecture of YOLOv8 Model:

The architecture of YOLOv8 is characterized by its deep convolutional neural network structure, consisting of multiple layers of convolutional, pooling, and fully connected layers. It typically comprises a backbone network, such as Darknet or ResNet, followed by detection heads responsible for predicting bounding boxes and class probabilities. YOLOv8 employs feature pyramid networks and skip connections to capture multi-scale features, enabling it to detect objects of varying sizes and shapes effectively.

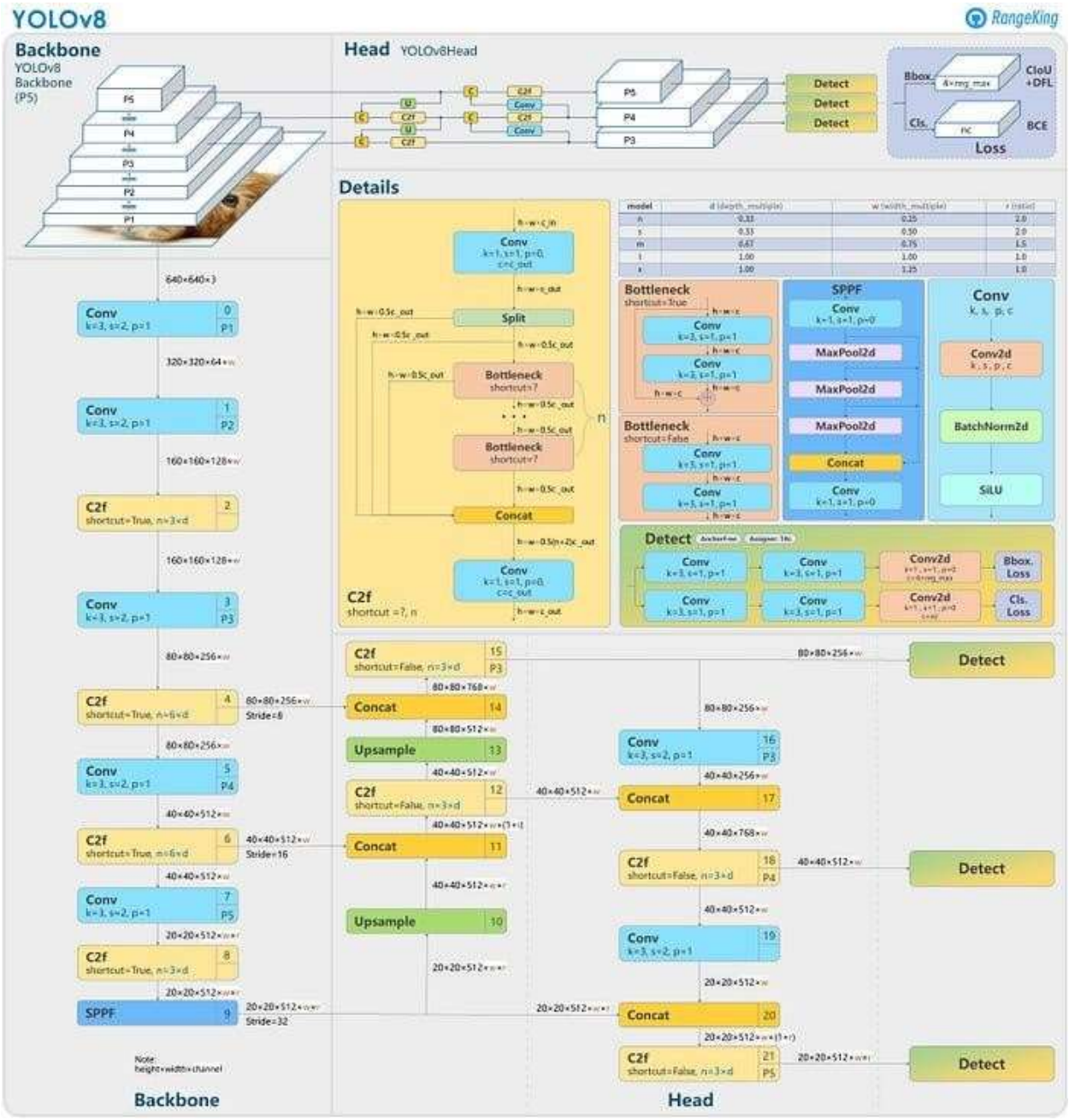


Fig.1.18: Architecture of YOLOv8

Example Usage:

For example, a user may take a picture of their breakfast plate using the application's camera feature. The YOLOv8 food detection model analyzes the image and identifies various food items present on the plate, such as eggs, bacon, toast, and orange slices. It generates bounding boxes around each detected food item and assigns class labels and confidence scores, indicating the model's certainty in its predictions. This information is then used to automatically log the

user's dietary intake, providing them with valuable insights into their nutrition and meal composition.

Results:

FIG.1.19: Food Logging using YOLOv8

1.12 DIET RECOMMENDATION SYSTEM & MEAL PLANS:

In this section, we outline the workings of our diet recommendation system, which provides personalized dietary advice and meal plans based on the user's body mass index (BMI) category. We'll define the system, describe its algorithm, and offer an example of its functionality.

1.12.1 Definition of Diet Recommendation System:

Our diet recommendation system utilizes the user's BMI to categorize them into different weight categories: Underweight, Normal Weight, Overweight, or Obese. Based on this classification, the system suggests appropriate foods and meal plans tailored to the user's nutritional needs and weight management goals.

1.12.2 Algorithm:

The algorithm for our diet recommendation system follows a series of steps to calculate the user's total daily calorie requirements and generate personalized meal plans:

1. **Calculation of Basal Metabolic Rate (BMR):** Depending on the user's gender, weight, height, and age, the system calculates the BMR using specific formulas for males and females using harris benedict equation [5].
   For males: BMR_male = 88.362 + (13.397 × weight) + (4.799 × height) - (5.677 × age) For females: BMR_female = 447.593 + (9.247 × weight) + (3.098 × height) - (4.330 × age)

2. **Determination of Total Calories:** The BMR is multiplied by an activity factor corresponding to the user's activity level (e.g., Sedentary, Lightly Active, etc.) to calculate the total daily calorie requirements.
3. **Food Suggestions**: Based on the user's BMI category, the system suggests foods that align with their nutritional needs and weight management goals.

   - ***Underweight****: Foods with high protein (>10g), high carbohydrates (>10g), or high calories (>200).*
   - ***Normal Weight****: Foods with high protein (>10g), high carbohydrates (>10g), low fat (<10g), or moderate calories.*
   - ***Overweight or Obese****: Foods with high protein (>10g), low carbohydrates (<10g), low fat (<10g), or low calories (<200).*
4. **Generation of Meal Plans:** The system generates personalized meal plans comprising recommended foods and portion sizes to meet the user's daily calorie and nutritional requirements.

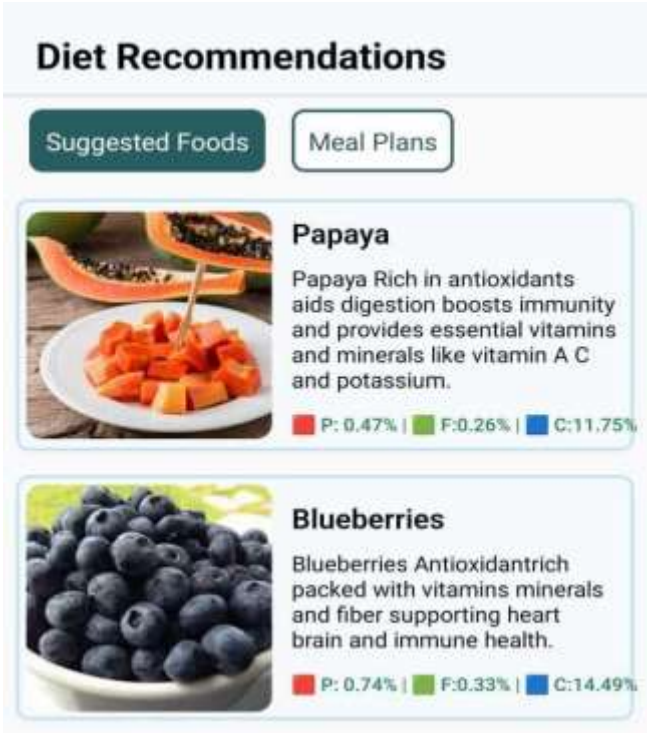


Fig.1.20 Diet Recommendations

Example:
Suppose a 30-year-old female with a height of 160 cm and a weight of 60 kg is categorized as "Normal Weight" based on her BMI. Given that she is moderately active, the system calculates her BMR as follows:

BMR = 447.593 + (9.247 × 60) + (3.098 × 160) - (4.330 × 30) = 1368.193 kcal

Assuming an activity factor of 1.55, the system determines her total daily calorie requirement.
Total Calories = BMR × Activity Factor = 2120.384 kcal
Based on her BMI category, the system suggests foods with high protein, high carbohydrates, and low fat, ensuring a balanced and nutritious diet conducive to maintaining a healthy weight.

Hence, Our diet recommendation system provides personalized guidance and meal plans tailored to the user's BMI category and nutritional needs.
By leveraging algorithms and filtering criteria, we offer users comprehensive support in achieving their weight management and health goals through optimal nutrition.

1.13 GOALS SETTINGS AND TRACKING [4]:

In this section, we outline the functionality of goal setting and tracking within our dietary management system, which aims to empower users in achieving their dietary objectives effectively.

The system allows users to set personalized diet goals based on their desired outcome, whether it's weight gain or weight loss. Additionally, users specify their activity level (lightly, moderately, highly) to tailor the recommendations accordingly. Moreover, users can choose their daily calorie reduction targets, typically 500 or 1000 calories, to align with their weight management goals.

Finally, users input their target weight, providing a clear objective to work towards. All these preferences are stored securely in the user profile within the database.

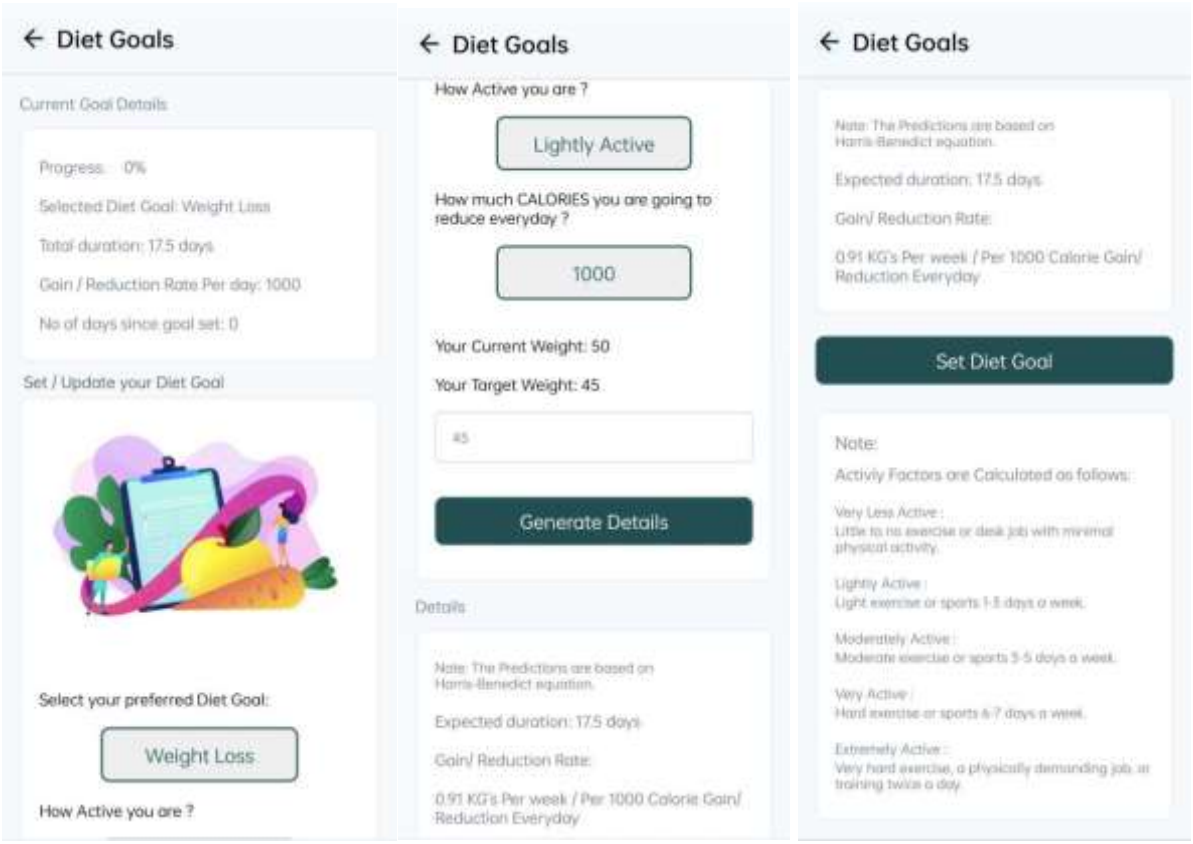


Fig.1.21: Goal Settings

Usage of Goal Settings: The goal settings feature ensures that the dietary recommendations provided by the system are precisely aligned with the user's specific objectives and preferences. By customizing the recommendations based on factors such as diet goals, activity level, and calorie targets, users receive personalized guidance that maximizes the likelihood of success in achieving their dietary objectives. This tailored approach enhances user engagement and adherence to the dietary plan, ultimately leading to more effective outcomes.

The system presents users with a visual representation of their daily calorie consumption using a bar chart. The x-axis of the chart displays the food items consumed by the user, while the y-axis indicates the corresponding calorie counts.

This intuitive visualization allows users to monitor their calorie intake easily, identify potential areas for improvement, and stay accountable to their dietary goals. Furthermore, the tracking feature enables users to make real-time adjustments to their dietary habits, ensuring ongoing progress towards their objectives.

Fig.1.22: Food Tracking

1.14 PROPOSED SYSTEM:

The proposed system builds upon the foundation of existing dietary management solutions, incorporating cutting-edge technologies and novel approaches to provide users with a comprehensive and user-friendly experience.

It aims to revolutionize the way individuals track, analyze, and manage their dietary intake, empowering them to make informed choices about their health and well-being.

The proposed system represents a significant advancement in dietary management technology, offering users a comprehensive and personalized solution for tracking, analyzing, and managing their dietary intake. By harnessing the power of AI, deep learning, and image recognition, the system empowers users to take control of their health and adopt healthier eating habits for life.

CHAPTER 2
REVIEW OF LIERATURE

The paper "**Smart Diet Diary: Real-Time Mobile Application for Food Recognition [1]** " authored by **Muhammad Nadeem, Henry Shen, Lincoln Choy, and Julien Moussa H. Barakat** addresses the global issue of obesity by introducing a smartphone application aimed at assisting obese individuals and patients in managing their dietary intake for healthier living. Utilizing deep learning techniques, specifically a pre-trained faster R-CNN model, the application recognizes food items and calculates their nutritional value, particularly calorie count, based on a dataset of 16,000 images across 14 categories. Achieving an overall accuracy of approximately 80.1% in food recognition and an average calorie computation accuracy within 10% of the real calorie value, the study highlights the limitations of existing dietary assessment methods and proposes image-based recognition systems for more accurate and convenient dietary tracking. The paper also discusses challenges in food recognition, particularly in estimating food volume, and proposes solutions to enhance accuracy and usability. Major contributions include the development of a semi-automated diet tracking application, creation of a customized food image dataset, and significant accuracy in food recognition and calorie estimation. The paper is structured into sections covering literature review, methodology, system evaluation, and conclusion, with discussions on future research directions.

Merieme Mansouri, Samia Benabdellah Chaouni, Said Jai Andaloussi, and Ouaïl Ouchetto proposed **A comprehensive system for food image recognition and dietary assessment [2]**. They introduced a methodology that utilizes deep learning models, including Inception v3, v4, and ResNet, achieving an impressive accuracy of 92%. The system encompasses food image segmentation, feature extraction, classification, volume estimation, and calorie value calculation for dietary assessment. It leverages pre-trained models and transfer learning techniques to enhance performance and scalability. However, the system has limitations as it does not consider individual factors such as BMI and other health issues while calculating calorie counts. This oversight could affect the accuracy of nutritional assessments, particularly for individuals with specific dietary needs or health conditions. Therefore, future enhancements should focus on integrating personalized factors into the calorie estimation process to improve the system's overall effectiveness in dietary assessment and management.

Rodrigo Alfaro and Héctor Allende-Cid proposed A novel approach for text classification in their paper titled "**Evaluation of a New Weighting Function for Text Representation in Multilabel Classification [3]**" The study evaluates a weighting function called relevance frequency for a label (rf1) to modify text representation during multilabel classification. They conducted experiments on ten widely referenced multilabel text datasets, including Reuters, Ohsumed, Enron, Slashdot, Langdot, Bibtex, Medical, TMC2007, and Science and Education. The proposed approach, which combines ensemble machine learning and shallow classification models, demonstrated significant improvements in performance, particularly in datasets with fewer labels, documents, and smaller vocabulary sizes. They compared the performance of their method with alternative techniques based on three performance measures: Hamming Loss, Accuracy, and macro-F1. The results showed improvements of more than 10% in terms of macro-F1 in classifiers based on their method in almost all analyzed cases. Furthermore, they evaluated the impact of modifying the representation using two different linear classifiers: Support Vector Machine (SVM) and Artificial Neural Networks (ANN). While their approach achieved promising results, with improvements ranging from 50% to 60% in some categories of data, it lacks the advanced contextual understanding and semantic representation capabilities of models like BERT, limiting its ability to capture intricate relationships within long text inputs.

CHAPTER 3

PROPOSED SOLUTION

3.1 OVERVIEW:

The proposed system offers an innovative approach to dietary management, leveraging machine learning and deep learning techniques for personalized recommendations. It aims to address the complexities of dietary choices by considering users' physical characteristics, ailments, and preferences. Through a comprehensive dataset and advanced algorithms, the system generates tailored meal plans and dietary guidelines. Users can input their personal details and dietary preferences, allowing the system to provide customized recommendations for healthier eating habits. With its emphasis on user-centricity and data-driven insights, the proposed system represents a significant advancement in promoting individual well-being through nutrition.

3.2 DATASET DESCRIPTION:

We have meticulously crafted our datasets to meet the unique needs of our dietary management system, eschewing reliance on existing datasets for greater control and precision in our data collection process.

Dataset 1: YOLOv8 Object Detection: Our first dataset comprises images meticulously collected and annotated using the Roboflow website. This dataset serves as the foundation for training the YOLOv8 pre-trained model, enabling accurate food item detection in images captured by the user's smartphone camera.

Dataset 2: Nutritional Information: Dataset 2 consists of detailed nutritional information for 140 food items, including calories (g), carbohydrates (g), protein (g), fat (g), and quantity (per 100g). This comprehensive dataset facilitates precise nutritional analysis and meal planning within our system, ensuring users receive accurate dietary recommendations.

FOOD	CALORIES(G)	CARBOHYDRATES(G)	PROTEIN(G)	FAT(G)	QUANTITY(G)
Chapati	297	15	3	0.5	100
Roti	297	15	3	0.5	100
Garlic Herb Chapati	273	48.8	9.4	7.5	100

Dataset 3: Sentence Classification: For sentence classification tasks within our system, we have compiled a dataset tailored to the nuances of food-related language. This dataset enables our custom-trained GPT model to classify user inputs accurately, enhancing the natural language processing capabilities of our system. This model is used to understand user input when describing food during meal logging. For instance, if a user describes their food as "round and red," the model would classify it into potential categories like "apple" or "tomato" based on the learned patterns from the training data.

Description	FOOD
Chapati a whole grain flatbread offers fiber for digestive health essential vitamins and minerals and is a good source of complex carbohydrates for sustained energy	Chapati
Roti a whole grain flatbread is a good source of fiber protein and essential vitamins and minerals providing energy and supporting digestive health	Roti
Garlic Herb Chapati offers a blend of flavors provides essential nutrients like fiber antioxidants vitamins and minerals and may support heart and digestive health	Garlic Herb Chapati

3.3 WORKING PROCESS:

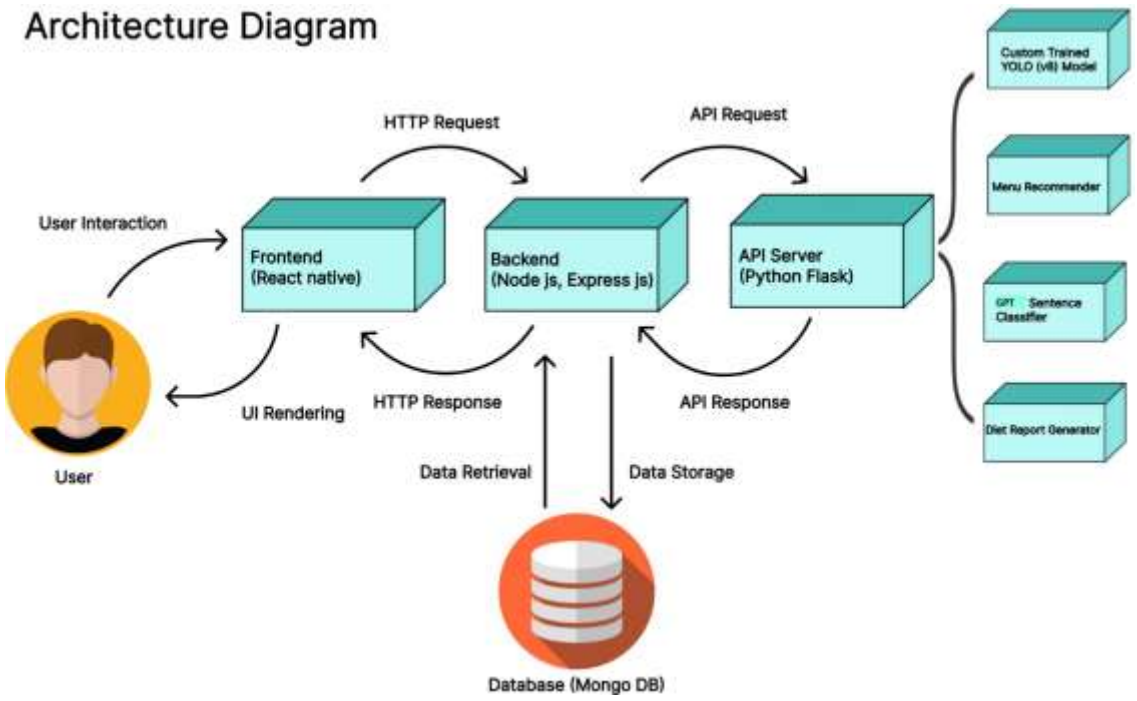


Fig:3.1 System Architecture

1. **User Interaction:**

   The user interaction within our application encompasses a range of functionalities aimed at facilitating effective dietary management. Users can effortlessly record their food intake by tapping on the "Log your Food" or “Snap your Food” button, capturing meal details with ease. Moreover, they have the capability to set personalized diet goals,

tailoring their approach to healthy eating based on preferences and requirements such as weight gain or loss, activity levels, and calorie reduction targets. The application provides comprehensive diet recommendations customized to individual needs and preferences, allowing users to receive tailored meal suggestions aligned with their health goals by inputting personal details like age, weight, and dietary restrictions. Additionally, users can track their meal history and monitor dietary progress over time, accessing past meal entries, nutritional intake, and adherence to diet goals through intuitive tracking features within the app. Overall, our application empowers users to take control of their dietary habits with intuitive and user-friendly tools for recording, goal-setting, recommendation viewing, and meal tracking.

Example: The user taps on the "Log your Food" button on the app's home screen. Frontend (React Native) to record the food intake.

2. **Frontend (React Native)**

   Functionality: The React Native code handles the user interface elements like buttons, text boxes, and images. It captures user actions and translates them into signals the backend can understand.

   Example: Upon tapping the "Log your Food" button, the React Native code generates an HTTP request specifying the user's action and sends it to the Backend.

3. **Backend (Node.js and Express.js):**

   Functionality: The backend, built with Node.js and Express.js, receives the HTTP request from the React Native UI. It acts as an intermediary between the UI and the API Server.

   Example: The Backend receives the HTTP request indicating the user wants to add a meal. It then formulates an API request containing the user ID and meal addition intent and sends it to the API Server.

4. **API Server (Python Flask):**

   Functionality: The API Server, developed with Python Flask, acts as the central processing unit for the application. It receives API requests from the Backend and communicates with various components like the Database and custom models to fulfill the request.

   Example: The API Server gets the API request from the Backend specifying the user ID and meal addition intent. It then decides how to proceed based on this information.

5. **Data Retrieval (MongoDB):**

   Functionality: To retrieve relevant information for processing the request, the API Server queries the MongoDB database. This could involve fetching user information, food data (calories, nutrients), or past meal entries.

   Example: The API Server queries the MongoDB to get the user's dietary preferences and restrictions stored in their profile.

6. **Custom Trained YOLO (v8) Model:**

   Functionality: If the user opted to add a meal by taking a picture, the image is preprocessed and sent to the custom-trained YOLO (v8) model for object detection. This model is specifically trained to identify food items within an image.

   Example: The user takes a picture of their lunch plate containing rice, chicken, and vegetables. The image is sent to the YOLO model, which identifies and returns the probability that each object in the image is rice, chicken, or vegetables.

7. **GPT Sentence Classifier:**

   Functionality: Alternatively, if the user entered the meal details through text (e.g., "oatmeal for breakfast"), the API Server sends the text data to the GPT Sentence Classifier model, which understands text meaning and identifies entities like food items.

   Example: The user types "I had a bowl of oatmeal for breakfast" in the text box. The API server sends this text to the GPT model, which analyzes the sentence and identifies "oatmeal" as the food item.

8. **Menu Recommender:**

   Functionality: Based on the users BMI and the user's profile information retrieved from MongoDB, the Menu Recommender recommends meals or suggests recipes that align with the user's preferences and restrictions.

   Example: The Application and backend server grabs the users BMI and Profile information. The Menu Recommender, knowing the user's unique body constitution, recommends a recipe for their diet.

9. **Diet Report Generator:**

   Functionality: The Diet Report Generator analyzes the user's recent meal entries and dietary intake. It generates reports with insights and trends to help users monitor their progress towards their health goals.

Example: The user has been using the app for a week, logging their meals regularly. The Diet Report Generator analyzes this data and shows a report that indicates the user is exceeding their daily fat intake but consuming less protein than recommended.

10. **API Response:**

    Functionality: Once all the processing is complete, the API Server generates a response containing the processed data (identified food items, nutritional information, recommendations) in a format the Backend can understand.

11. **Backend Response:**

    Functionality: The Backend receives the API request from the frontend processes it and send the response data back into a format suitable for the frontend (React Native UI).

12. **UI Rendering:**

    Functionality: The React Native UI receives the data from the Backend and uses it to update the application's UI. This could involve displaying the identified food items, their nutritional breakdown, and recommendations from the Menu Recommender.

3.4 CONCLUSION:

In conclusion, the working process of our application demonstrates a comprehensive and user-centric approach to dietary management. By seamlessly integrating various components such as user interaction, frontend and backend functionalities, custom-trained models, and data retrieval mechanisms, we have created a robust platform that empowers users to make informed choices about their dietary habits. Through features like personalized diet goal setting, intuitive meal recording, tailored recommendations, and insightful tracking capabilities, our application offers a holistic solution for individuals striving towards healthier lifestyles. By leveraging advanced technologies and prioritizing user experience, we aim to promote well-being and facilitate positive dietary transformations for our users.

CHAPTER 4

IMPLEMENTATION

4.1 Tool 1: EXPO GO APP

Expo Go is a mobile app developed by Expo for testing and running React Native apps on real devices. It allows developers to quickly preview their projects without having to compile code or install additional software. With Expo Go, developers can scan a QR code generated by the Expo CLI (Command Line Interface) or the Expo DevTools in their web browser, and instantly view their app on their iOS or Android device.

Expo is a free and open-source platform for building React Native apps. It provides various tools and services to streamline the development process, including a robust set of APIs for accessing device features, a command-line interface for managing projects, and a suite of development tools for debugging and testing.

Expo Go is a mobile app available on both the Android Play Store (for Android Lollipop and greater) and the iOS App Store (for iOS 13 and greater). It allows you to run your React Native projects directly on your phone during development. When you run npx expo start in your project, Expo CLI starts a development server and generates a QR code.

On Android, open the Expo Go app on your device and scan the QR code to connect to the development server.

Functionalities:

- **Development Environment Simplification:** Expo Go is part of the Expo development ecosystem, which aims to streamline the development of mobile apps, particularly those built with React Native. React Native itself is a framework for building mobile applications using JavaScript and React. Expo builds on top of React Native by providing additional tools and services to simplify various aspects of development.
- **Previewing React Native Apps:** One of the key features of Expo Go is its ability to quickly preview React Native apps on real devices. Developers can make changes to their code and instantly see those changes reflected on their device without having to

rebuild the entire app. This rapid iteration cycle speeds up development and allows developers to fine-tune their apps more efficiently.

- **QR Code-Based Deployment:** Expo Go works by generating a QR code that developers can scan using the Expo Go app on their mobile device. This QR code contains information about the app being developed, including its JavaScript bundle and any assets it depends on. Once scanned, Expo Go loads the app and allows developers to interact with it as if it were installed from an app store.
- **Live Reload and Hot Reloading:** Expo Go supports live reload and hot reloading, which are mechanisms for automatically refreshing the app when changes are made to the code. Live reload refreshes the entire app whenever a file is saved, while hot reloading preserves the app's state and only updates the components that have changed. These features make the development process more responsive and enable developers to see their changes in real-time.
- **Access to Native Device Features:** Expo Go provides access to a wide range of native device features and APIs, including the camera, geolocation, sensors, and more. Developers can use these APIs to create rich and interactive experiences that leverage the capabilities of the underlying device hardware. Expo abstracts away the complexities of interfacing with native code, making it easier for developers to integrate these features into their apps.
- **Cross-Platform Compatibility**: Expo Go supports both iOS and Android devices, allowing developers to test their apps on multiple platforms simultaneously. This cross-platform compatibility is essential for ensuring that apps look and perform consistently across different devices and operating systems.

Expo Go is a powerful tool for React Native developers that simplifies the development and testing of mobile apps. By providing a fast iteration cycle, easy deployment, and access to native device features, Expo Go empowers developers to build high-quality apps more efficiently. Expo Go provides secure methods for developers to authenticate their Expo projects before running them on a device. This ensures that only authorized individuals can access and test the app during development. Expo also provides mechanisms for securely sharing apps with clients or testers while maintaining control over who can access them.

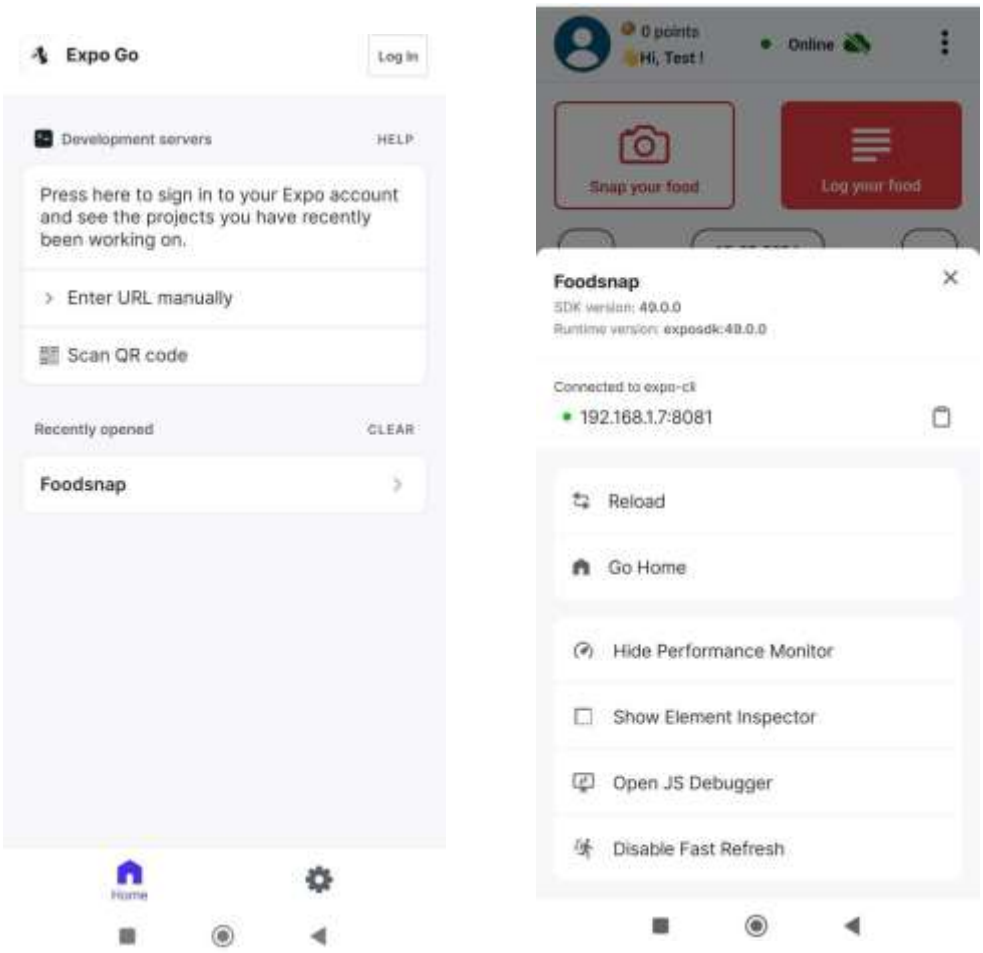


Fig 4.1: Expo GO

4.2 Tool 2: REACT NATIVE PC EMULATOR

React Native is an open-source framework for building cross-platform mobile applications using JavaScript and React. Developed by Facebook, it allows developers to create mobile apps that have a native look and feel while sharing a significant portion of the codebase across multiple platforms, such as iOS and Android.

React Native allows developers to create mobile applications using JavaScript and React, and it provides various tools for testing and debugging on both iOS and Android platforms. While React Native primarily targets mobile platforms, developers can also use it to build applications for desktop platforms like Windows, macOS, and Linux.

For Windows development, there isn't an official PC emulator provided by React Native itself. However, developers can use third-party solutions to run and test React Native apps on PC. Here are a few options:

- **Expo Go**: Although Expo primarily targets mobile development, it offers a feature called "Expo for Web," which allows developers to run React Native applications in

web browsers. While this doesn't emulate the native PC environment directly, it provides a way to quickly preview and test React Native code on desktop browsers.

- **Electron:** Electron is an open-source framework developed by GitHub that allows developers to build cross-platform desktop applications using web technologies such as HTML, CSS, and JavaScript. Developers can integrate React Native with Electron to create desktop applications that leverage the native capabilities of the underlying operating system.
- **React Native for Windows + Microsoft Emulator**: Microsoft has developed "React Native for Windows," an official extension that enables developers to build Windows applications using React Native. Along with this, Microsoft provides an emulator that allows developers to test their React Native apps on Windows environments. This option provides a more native-like experience for PC emulation.
- **Third-Party Emulators**: Developers can also use third-party Android emulators like BlueStacks or Genymotion to run React Native apps on PC. These emulators provide a virtual Android environment on which developers can install and test their React Native apps.
- **Virtual Machines**: Another approach is to set up virtual machines using software like VirtualBox or VMware and install Android or iOS emulators within those virtual machines. This allows developers to simulate mobile environments on their PC for testing React Native apps.

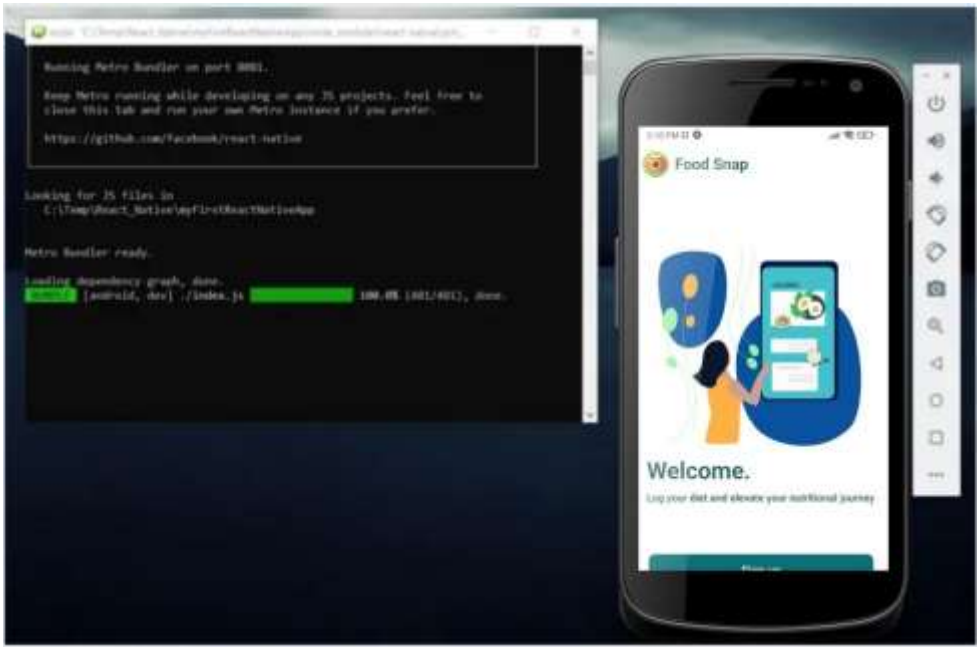


Fig 4.2 React Native PC Emulator

4.3 Tool 3: ANDRIOD PHONE

Android phones in the context of React Native development. React Native allows developers to build mobile applications that can run on Android devices, among other platforms. Here's how React Native works with Android phones:

- **Development Environment Setup:** To develop React Native applications for Android, we need to set up your development environment. This involves installing Node.js, npm (Node Package Manager), and the React Native CLI (Command Line Interface). Additionally, you'll need to install Android Studio, which provides the Android SDK (Software Development Kit) necessary for building Android apps.
- **Creating a React Native Project:** Once your development environment is set up, we can use the React Native CLI to create a new project. We can do this by running a command like npx react-native init MyProject, where "MyProject" is the name of your project. This command will create a new directory with the necessary files and folders for a React Native project.
- **Running the App on an Android Device:** After setting up project, we can run it on an Android device for testing and development. To do this, connect our Android device to our computer via USB and enable USB debugging in the developer options. Then, use the React Native CLI to start the development server (npx react-native start) and run the app on our device (npx react-native run-android). This command will build the app and install it on our connected Android device.
- **Debugging and Testing:** React Native provides various tools for debugging and testing our app on Android devices. We can use Chrome DevTools to debug JavaScript code running in our app, and we can also use Android Studio's built-in tools for debugging and profiling Android apps. Additionally, React Native's hot reloading feature allows to see changes to our code reflected instantly in the running app, making the development process more efficient.
- **Publishing to the Google Play Store:** Once we developed and tested our React Native app on Android devices, we can publish it to the Google Play Store for distribution to users. This involves generating a signed APK (Android Package) file and uploading it to the Google Play Console, where we can manage our app's listing, release updates, and monitor its performance.

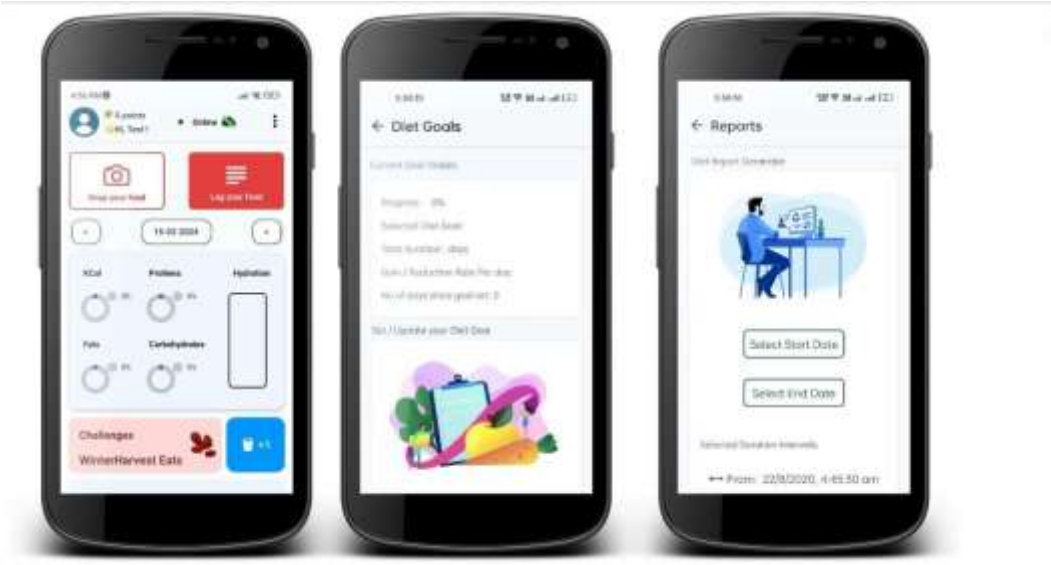


Fig 4.3: Android Phone

4.4 Tool 4: VISUAL STUDIO

Visual Studio is an integrated development environment (IDE) developed by Microsoft. It provides comprehensive tools for software development, including coding, debugging, and testing capabilities. Visual Studio supports multiple programming languages such as C#, C++, Visual Basic .NET, F#, and Python, among others.

Visual Studio Code is a source code editor that can be used with a variety of programming languages. Instead of a project system it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language, contrary to Microsoft Visual Studio which uses the proprietary '.sln' solution file and project-specific project files. It supports a few programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many of Visual Studio Code features are not exposed through menus or the user interface but can be accessed.

Visual Studio Code can be extended via plug-ins, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, debuggers, perform static code analysis, add code linters, using the Language Server Protocol and connect to additional services.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

Visual Studio Code allows users to set the code page in which the active document is saved, the newline character for Windows/Linux, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.

1. **Visual Studio IDE:**

The Visual Studio IDE is a comprehensive environment that allows you to:

- Write, edit, debug, and build code.
- Deploy your applications.
- Utilize compilers, code completion tools, and graphical designers.
- Enhance every stage of the software development process23.

It's like a creative launching pad for developers, providing a rich set of features to streamline development tasks.

2. **Visual Studio Code (VS Code):**

VS Code is a lightweight but powerful source code editor that runs on Windows, macOS, and Linux.

Key features include:

- Code editing: Edit your code efficiently.
- Debugging: Debug directly from the editor with breakpoints and call stacks.
- Extensions: A rich ecosystem of extensions for various languages and runtimes (such as JavaScript, TypeScript, C++, C#, Java, Python, PHP, Go, and .NET).
- Integrated Git: Manage version control seamlessly.

It's a top pick for JavaScript and web developers due to its flexibility and extensibility4.

3. **GitHub Copilot:**

GitHub Copilot is an exciting addition to Visual Studio 2022. It acts as your AI pair programmer, providing multi-line suggestions based on your code and comments.

With Copilot, you can:

- Code faster.
- Work smarter.
- Learn new technologies more easily.

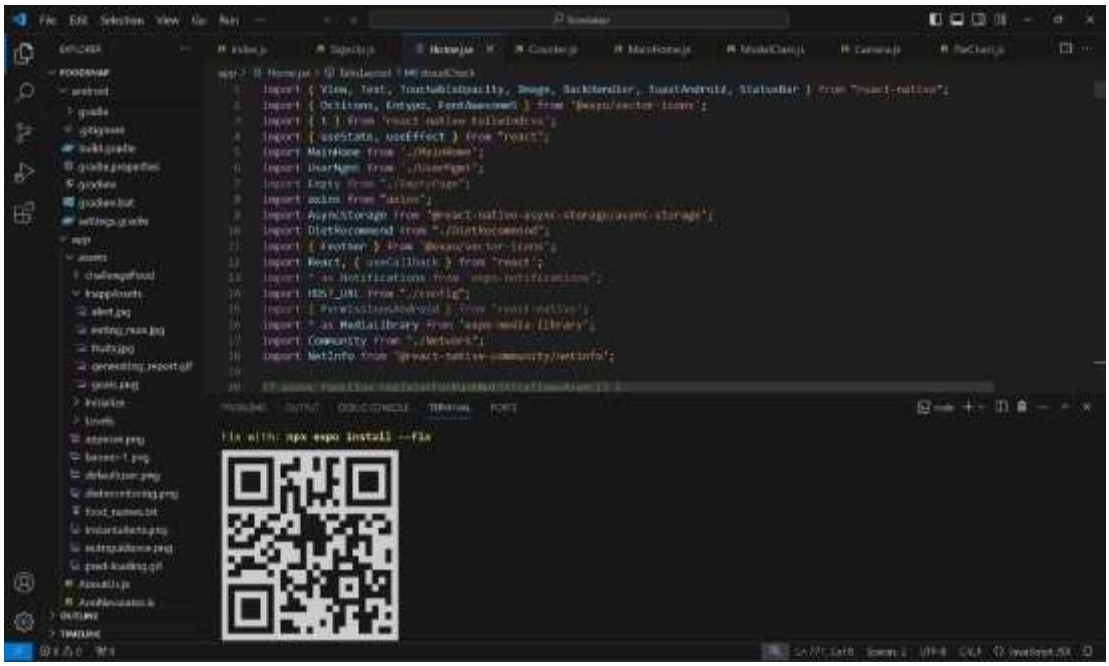


Fig 4.4 Visual Studio

4.5 Tool 5: REST API

REST API (Representational State Transfer Application Programming Interface) is a type of web service that follows the principles of REST architectural style. REST is an architectural style for designing networked applications. It stands for Representational State Transfer and was first presented by Roy Fielding in his doctoral dissertation.

Characteristics of a REST API include:

- **Resource-Based:** REST APIs are centered around resources, which can be any kind of object, data, or service that can be accessed, modified, or deleted via HTTP requests.
- **Uniform Interface:** REST APIs utilize a uniform interface, which typically involves using HTTP methods such as GET, POST, PUT, DELETE, and PATCH to perform actions on resources. Each resource is identified by a unique URI (Uniform Resource Identifier).

- **Statelessness**: REST APIs are stateless, meaning that each request from a client to the server must contain all the information necessary to understand and fulfill the request. The server does not store any client context between requests.
- **Client-Server Architecture:** REST APIs follow a client-server architecture, where the client and server are separate entities that communicate over a network using standard protocols such as HTTP.
- **Cacheability:** REST APIs can leverage caching mechanisms to improve performance and reduce server load. Responses from the server can include cache directives to specify whether the response can be cached and for how long**.**
- **Layered System:** REST APIs can be built using a layered system architecture, where different components (such as load balancers, proxies, and gateways) can be added to the system without affecting the overall functionality.

REST APIs are commonly used for building web services that can be accessed by various client applications, including web browsers, mobile apps, and IoT devices. They provide a scalable and flexible way to expose and consume data and functionality over the internet. Popular web development frameworks such as Flask (Python), Express.js (Node.js), and Django REST Framework (Python) provide tools and libraries for building RESTful APIs.

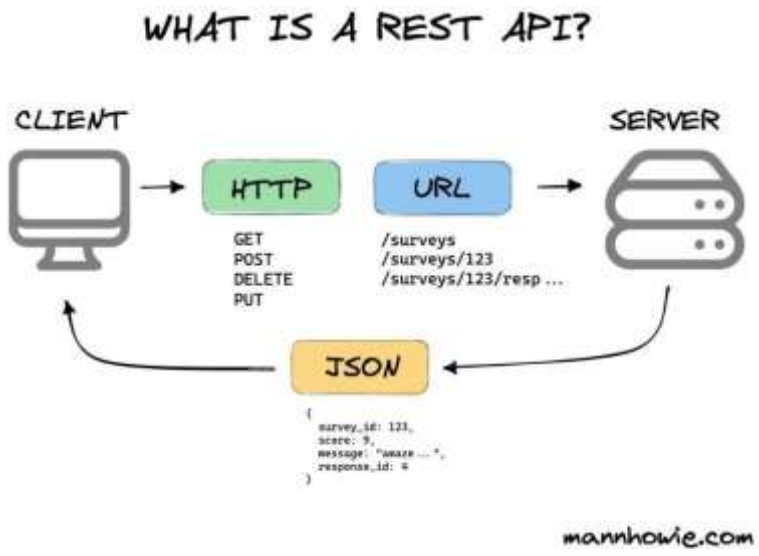


Fig 4.5 Rest API

4.6 Tool 6: MERN STACK WITH REACT NATIVE EXPO

The exciting world of building a Mobile Application using the MERN stack (MongoDB, Express, React Native, and Node.js) with the power of React Native and Expo MERN stack is a popular set of technologies used for building full-stack web applications. The name"MERN" is an acronym representing the four main components:

- **MongoDB**: MongoDB is a NoSQL database that stores data in a flexible, JSON-like format. It is known for its scalability, flexibility, and ease of use. MongoDB is often chosen for MERN stack applications due to its ability to handle large volumes of data and its compatibility with JavaScript, making it seamless to work with in Node.js environments.
- **Express.js**: Express.js is a minimalist web application framework for Node.js. It provides a set of features for building web servers and APIs, such as routing, middleware support, and HTTP utilities. Express.js is commonly used in MERN stack applications to handle server-side logic and create RESTful APIs for communication between the client-side and the database.
- **React**: React is a JavaScript library for building user interfaces. Developed by Facebook, React allows developers to create interactive and dynamic UI components using a component-based architecture. React is a key component of the MERN stack as it enables the creation of modern, single-page web applications with a rich user experience. React works seamlessly with the other components of the stack, allowing for efficient data binding and state management.
- **Node.js**: Node.js is a server-side JavaScript runtime built on Chrome's V8 JavaScript engine. It allows developers to run JavaScript code outside of a web browser, making it possible to build server-side applications using JavaScript. Node.js is the foundation of the MERN stack, providing the runtime environment for running server-side code written in JavaScript. It integrates seamlessly with Express.js to create robust and scalable web servers for MERN applications.

The MERN stack offers a powerful combination of technologies for building modern web applications, with JavaScript being the common language used across the entire stack. This allows for increased developer productivity, code reuse, and seamless communication between client-side and server-side components. Additionally, the MERN stack is highly customizable,

allowing developers to choose from a wide range of libraries, tools, and frameworks to meet the specific requirements of projects.

Fig 4.6 MERN Stack

4.7 Tool 7: PYTHON FLASK SERVER

Flask is a lightweight web framework for building web applications in Python. It allows developers to create web services quickly and easily. Here are the essential steps to get started with Flask:

1. **Install Python:** Make sure we have Python installed on your system. If not, download and install it from the official Python website.
2. **Create a Virtual Environment:** It's a good practice to create a virtual environment for our Flask project. This isolates our project dependencies from the system-wide Python installation. To create a virtual environment, run the following command in our terminal or command prompt**:**python -m venv myenv
3. **Install Flask**: Install Flask using pip. Open our terminal or command prompt and execute: pip install flask
4. Create our First Flask Application
5. Save the code
6. Run Your Flask Application
7. Our Flask app will start, and we can access it by opening a web browser

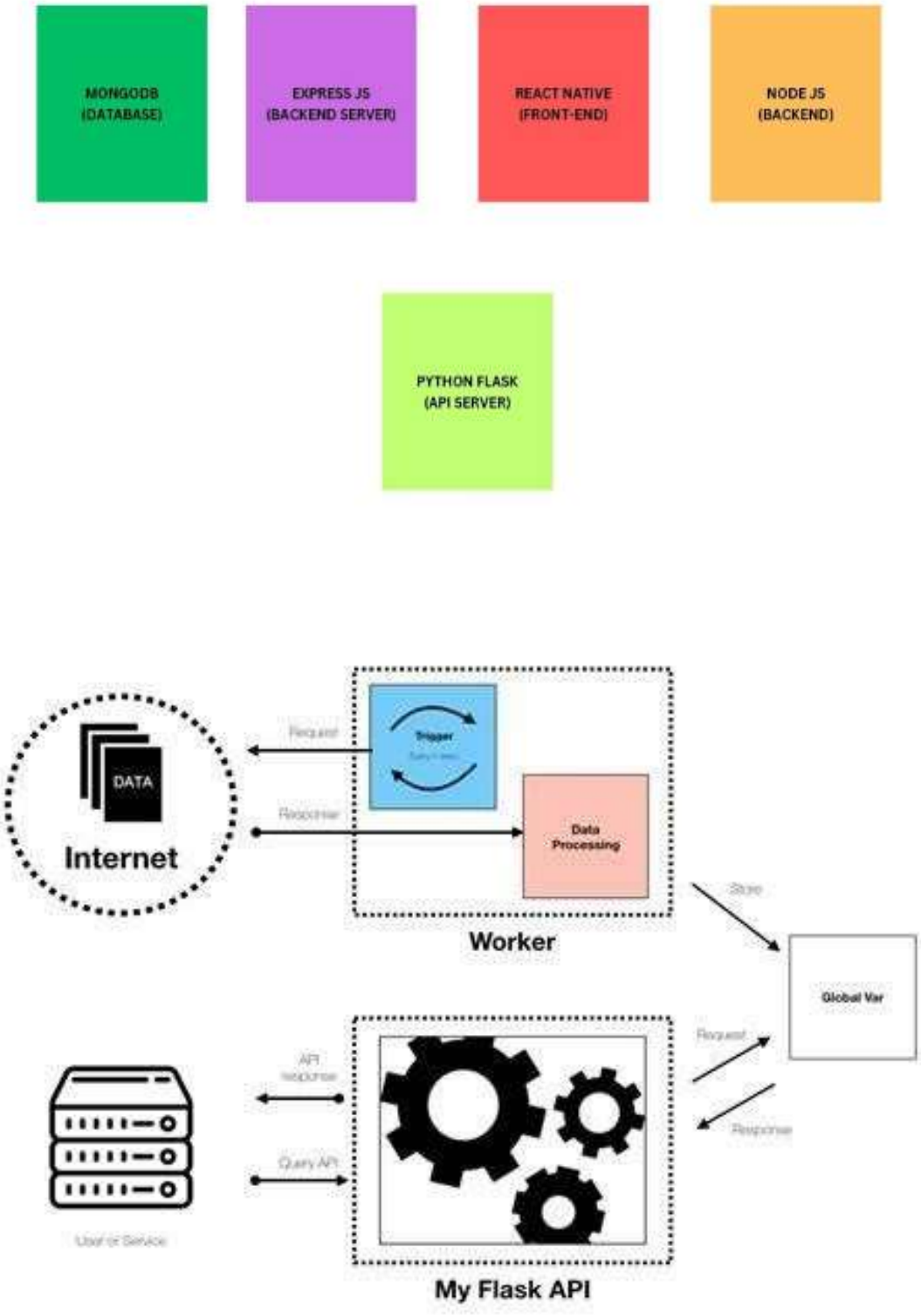


Fig 4.7 Flask Server

CHAPTER 5

RESULTS

The mobile application focuses on dietary habits, ensuring a perfect diet for individuals including health-conscious individuals, fitness enthusiasts, and those who simply want to plan their meals and track their diet over time. It has been developed to include all features such as meal history and tracking, nutritional analysis, goal setting and tracking, offline functionality, personalized recommendations, analytics and reporting, as well as notifications and reminders. Additionally, it provides analytics on data to ensure that all quantities such as calories, fats, proteins, and carbohydrates are within the required range. The application works on a wide range of devices, supporting most Android devices.

Diet recommendations play a major role in suggesting food items for users based on their BMI, and allergen alerts are provided to warn users when certain foods are not recommended. The application also includes a Community Ranking feature to encourage users to log their food and earn achievement points to rank higher. Furthermore, it sends weekly, monthly, and yearly diet reports and offers an option to generate a diet report with included pie charts based on specified start and end dates. Moreover, it boasts a user-friendly UI for an enhanced user experience and allows food logging through a simple snap of the food item.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

Overall, the mobile application offers a comprehensive solution for individuals looking to improve their dietary habits, whether they are health-conscious individuals, fitness enthusiasts, or simply aiming to track their meals. With its wide range of features including meal tracking, nutritional analysis, goal setting, personalized recommendations, and user-friendly interface, the application provides users with the tools they need to achieve their dietary goals. Additionally, features such as allergen alerts, community ranking, and detailed analytics further enhance the user experience and effectiveness of the app.

Future Scope: Looking ahead, there are several avenues for expanding and enhancing the functionality of the mobile application:

1. **Integration with wearable devices:** Incorporating compatibility with wearable fitness trackers and smartwatches would enable users to seamlessly sync their activity data with their dietary information, providing a more holistic approach to health tracking.
2. **Enhanced machine learning algorithms:** Continuously refining and improving the recommendation algorithms based on user feedback and data analysis can lead to more accurate and personalized suggestions for users, catering to their specific dietary needs and preferences.
3. **Collaboration with healthcare professionals:** Partnering with nutritionists, dietitians, and other healthcare professionals can add credibility and expertise to the app, providing users with access to personalized advice and guidance for achieving their health and wellness goals.
4. **Expansion to other platforms:** While the current focus is on Android devices, expanding the application to iOS and other platforms would broaden its reach and accessibility to a larger audience of users.

CHAPTER 7

REFERENCES

[1] Smart Diet Diary: Real-Time Mobile Application for Food Recognition by Muhammad Nadeem, Henry Shen, Lincoln Choy and Julien Moussa H. Barakat - https://www.mdpi.com/2571-5577/6/2/53

[2] Deep Learning for Food Image Recognition and Nutrition Analysis Towards Chronic Diseases Monitoring: A Systematic Review. https://www.researchgate.net/publication/372133738_Deep_Learning_for_Food_Image_Recognition_and_Nutrition_Analysis_Towards_Chronic_Diseases_Monitoring_A_Systematic_Review

[3] Multilabel Text Classification with Label-Dependent Representation by Rodrigo Alfaro, Héctor Allende-Cid, and Héctor Allende - https://www.mdpi.com/2076-3417/13/6/3594

[4] Calorie Calculator: https://www.nasm.org/resources/calorie-calculator

[5] Revised-Harris–Benedict-Equation: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9967803

[6] YOLOv8 Object Detection: https://medium.com/cord-tech/yolov8-for-object-detection-explained-practical-example-23920f77f66a

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed/ Refereed Journal

UGC Journal No. : 7647

Certificate of Publication

Dr. Muralidhar Vejendla

Associate Professor, Department. of CSE- Artificial Intelligence and Machine Learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

TITLE OF PAPER

Foodsnap: A Deep Learning Based Dietary Management and Food Analysis Application

has been published in

IJARESM, Impact Factor: 8.536, Volume 12 Issue 3, March-2024

Paper Id: IJ-2903240205

Date: 29-03-2024

Website: www.ijaresm.com Email: editor.ijaresm@gmail.com

Authorized Signatory

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed/ Refereed Journal

UGC Journal No. : 7647

Certificate of Publication

Shaik Esub

Student, Department. of CSE- Artificial Intelligence and Machine Learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

TITLE OF PAPER

Foodsnap: A Deep Learning Based Dietary Management and Food Analysis Application

has been published in

IJARESM, Impact Factor: 8.536, Volume 12 Issue 3, March-2024

Paper Id: IJ-2903240206

Date: 29-03-2024

Website: www.ijaresm.com Email: editor.ijaresm@gmail.com

Authorized Signatory

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed/ Refereed Journal

UGC Journal No. : 7647

Certificate of Publication

Pathan Sattar Khan

Student, Department. of CSE- Artificial Intelligence and Machine Learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

TITLE OF PAPER

Foodsnap: A Deep Learning Based Dietary Management and Food Analysis Application

has been published in

IJARESM, Impact Factor: 8.536, Volume 12 Issue 3, March-2024

Paper Id: IJ-2903240207

Date: 29-03-2024

Website: www.ijaresm.com Email: editor.ijaresm@gmail.com

Authorized Signatory

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed/ Refereed Journal

UGC Journal No. : 7647

Certificate of Publication

Appala Sai Srija

Student, Department. of CSE- Artificial Intelligence and Machine Learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

TITLE OF PAPER

Foodsnap: A Deep Learning Based Dietary Management and Food Analysis Application

has been published in

IJARESM, Impact Factor: 8.536, Volume 12 Issue 3, March-2024

Paper Id: IJ-2903240208

Date: 29-03-2024

Website: www.ijaresm.com Email: editor.ijaresm@gmail.com

Authorized Signatory

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed/ Refereed Journal

UGC Journal No. : 7647

Certificate of Publication

Prathipati Sandeep

Student, Department. of CSE- Artificial Intelligence and Machine Learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

TITLE OF PAPER

Foodsnap: A Deep Learning Based Dietary Management and Food Analysis Application

has been published in

IJARESM, Impact Factor: 8.536, Volume 12 Issue 3, March-2024

Paper Id: IJ-2903240209

Date: 29-03-2024

Website: www.ijaresm.com Email: editor.ijaresm@gmail.com

Authorized Signatory

International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 12, Issue 3, March-2024, Available online at: www.ijaresm.com

Foodsnap: A Deep Learning Based Dietary Management and Food Analysis Application

Dr. Muralidhar Vejendla[1], Shaik Esub[2], Pathan Sattar Khan[3], Appala Sai Srija[4], Prathipati Sandeep[5]

[1] Associate Professor, Department. of CSE- Artificial Intelligence and Machine Learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India
[2,3,4,5]Student, Department. of CSE- Artificial Intelligence and Machine Learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

---

ABSTRACT

Dietary health and nutrition have become increasingly significant concerns in contemporary society. Amidst rising rates of diet-related diseases, there is a pressing need for innovative solutions to assist individuals in managing their dietary intake effectively.The research presents an innovative and comprehensive app, termed FOODSNAP, which aims to revolutionize dietary management through the seamless integration of cutting-edge technologies such as Deep Learning and Natural Language Processing (NLP). At its core, FOODSNAP leverages a sophisticated Machine Learning-based Sentence Classifier, coupled with a custom-trained Deep Learning Model for food detection. This dual approach ensures precise classification of food image and textual descriptions, enabling users to effortlessly record and monitor their dietary intake with more accuracy and ease. One of the key features of FOODSNAP lies in its extensive and diverse database, which provides detailed nutritional information for a wide array of foods. From essential metrics like calories, carbohydrates, fats, and proteins to specialized dietary requirements, the app offers users comprehensive insights into their food choices. This wealth of information empowers users to make informed decisions about their dietary habits, fostering a deeper understanding of nutritional content and its impact on overall health. It also offers personalized dietary recommendations tailored to individual body constitutions. By analyzing user data and health profiles, the app delivers targeted suggestions for optimal nutrition, guiding users towards healthier eating habits and improved well-being. It is equipped with features such as Food Allergen Alerts, notifying users when a particular food is not recommended based on their allergies or dietary restrictions, thereby ensuring safety and peace of mind. FOODSNAP also facilitates goal setting for weight management, with intuitive tools for both weight gain and weight loss objectives. Through efficient tracking capabilities and insightful analytics, users can monitor their progress in real-time, empowering them to stay on course towards their desired goals. Additionally, the app offers dynamic Diet Report Generation, allowing users to review their dietary patterns and trends over weekly, monthly, and yearly intervals. This holistic approach to dietary management enables users to gain deeper insights into their eating habits and progress towards long-term health objectives. In essence, FOODSNAP represents a paradigm shift in dietary management, offering a comprehensive solution that combines advanced technology with user-centric design principles. By empowering users with actionable insights, personalized recommendations, and intuitive tools, FOODSNAP aims to promote healthier eating behaviors and enhance overall well-being in the digital age.

Keywords: AI-based Smart food analyzer, Dietary tracking, Personalized Dietary Recommendations, Smart Food Analyzer.

INTRODUCTION

In today's fast-paced world, maintaining optimal health can be a challenge, especially when it comes to managing our dietary habits. Traditional methods of food tracking often fall short, plagued by tedious manual recording, limited accessibility, and a lack of personalized insights. This is where the "FOODSNAP-A Deep Learning Based Dietary Management and Food Analysis Application" emerges as a groundbreaking solution, leveraging the power of Deep Learning and Natural Language Processing (NLP) to revolutionize dietary management. Imagine a world where recording your meals is as effortless as taking a picture, where personalized dietary recommendations cater to your unique needs and body constitution, and where health-conscious alerts guide you towards informed choices. By utilizing Deep Learning algorithms, users can simply take a picture of their meals, and the application automatically recognizes and records the

nutritional content using a streamlined approach to eliminate the tedious task of manual data entry, making dietary management more accessible and user-friendly.

Traditional calorie trackers have their limitations. Manual data entry is time-consuming and error-prone, while the lack of support for regional languages and unnamed foods creates barriers for diverse user groups. Furthermore, the project acknowledges the importance of user feedback and continuous improvement. Regular updates and enhancements based on user input ensure that the application evolves in tandem with user needs, reinforcing its commitment to being a dynamic and user-centric dietary management tool.

In envisioning a future where maintaining a healthy lifestyle is seamlessly integrated into daily routines, the "FOODSNAP" introduces a novel approach to food tracking. Imagine capturing the essence of your meals effortlessly, receiving instant nutritional insights, and having a virtual dietary assistant who understands and caters to your unique preferences. This project aspires to be more than just a conventional food diary; it strives to be a personalized companion on the journey towards improved health, providing users with the knowledge and tools they need to make informed decisions about their well-being.

Moreover, the project emphasizes inclusivity by supporting regional languages and accounting for unnamed foods. Recognizing the diverse dietary habits across different cultures, the application ensures that users from various backgrounds can easily incorporate the tool into their daily lives. This inclusivity aligns with the project's goal of reaching a broad user base and promoting healthier dietary choices on a global scale. The limitations of traditional calorie trackers extend beyond the realm of practicality. The arduous process of manual data entry discourages consistent usage, and the lack of support for regional languages and unnamed foods excludes diverse user groups. By addressing these challenges, our project not only embraces technological advancements but also embraces inclusivity and a holistic understanding of nutrition, promising a more effective and user-friendly dietary management solution for individuals from all walks of life.

LITERATURE SURVEY

In recent years, there has been a surge of interest in leveraging deep learning techniques to address the pressing issue of obesity and promote healthier living through dietary management and food recognition systems.

Nadeem et al. (2019) introduced the "Smart Diet Diary," [1] a smartphone application designed to aid obese individuals and patients in monitoring their dietary intake. Utilizing a pre-trained faster R-CNN model, the application identifies food items and computes their nutritional value. The study underscores the limitations of traditional dietary assessment methods and advocates for image-based recognition systems as a more convenient and potentially more accurate alternative.

Similarly, Mansouri et al. (2020) proposed a comprehensive system for food image recognition and dietary assessment [2]. Employing deep learning models such as Inception v3, v4, and ResNet, the system achieves notable accuracy. However, the study acknowledges the challenge of incorporating individual factors like BMI and health issues into calorie estimation, which could impact the accuracy of nutritional assessments.

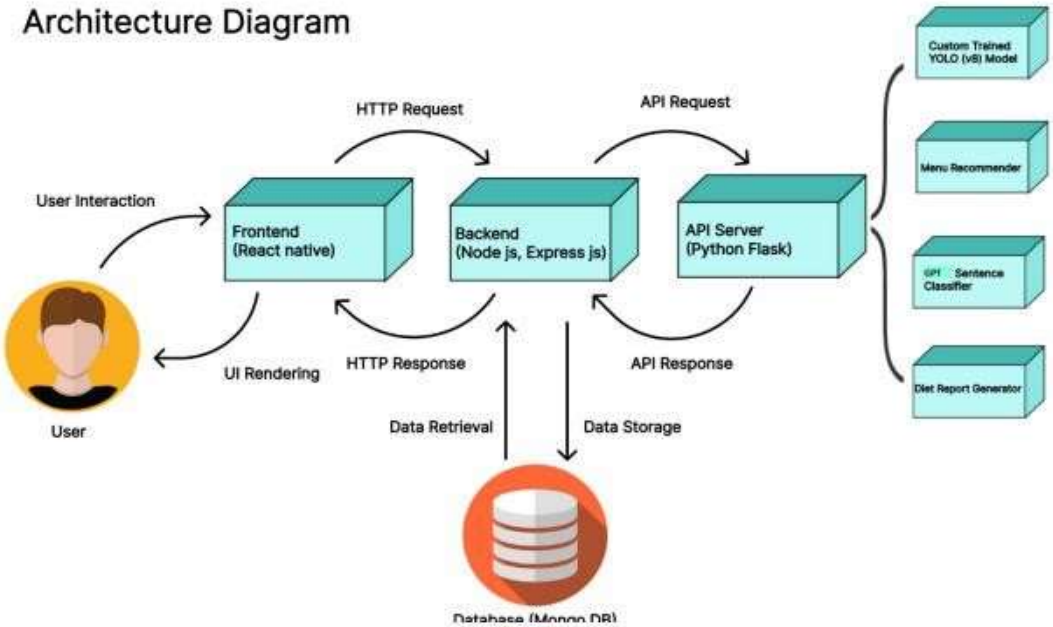
In contrast, Alfaro, and Allende-Cid (2018) introduced a novel approach for text classification. Their study, titled "Evaluation of a New Weighting Function for Text Representation in Multilabel Classification," [3] explores the use of a weighting function called relevance frequency for a label (rf1) to modify text representation during multilabel classification. While their approach shows promise, it lacks the advanced contextual understanding and semantic representation capabilities of models like GPT.

These studies collectively underscore the potential of deep learning techniques in revolutionizing dietary management and food recognition. However, they also highlight the need for further research to address challenges such as incorporating individual factors into dietary assessment models and exploring advanced techniques for text representation and classification.

Proposed system

The envisioned system,"FOODSNAP: A Deep Learning Based Dietary Management and Food Analysis Application" is a cutting-edge mobile application developed on the React Native framework with a robust backend supported by Node.js and MongoDB. The system integrates state-of-the-art technologies to redefine dietary management, providing users with an intuitive, secure, and feature-rich experience.

International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 12, Issue 3, March-2024, Available online at: www.ijaresm.com



Methodology: Development and Implementation of FOODSNAP Features

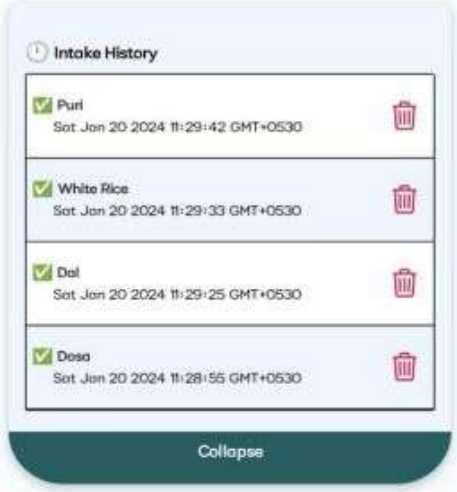
This section outlines the methodology employed in developing and integrating key features within the FOODSNAP application, aimed at enhancing dietary management and user experience.

User-Friendly Dashboard: The proposed system integrates a user-friendly dashboard tailored for efficient dietary management. This dashboard will offer an intuitive snapshot, providing users with comprehensive insights into their nutritional journey. Elements shown as follows

a. A central donut chart elegantly displays the proportional breakdown of calories, proteins, fats, carbohydrates, and water consumed throughout the day. Each segment of the donut represents the percentage contribution of these nutritional elements, providing users with a quick and easy-tounderstand overview of their dietary intake.
b. A bar chart complements the donut chart by presenting a graphical representation of the quantity of food consumed at different times during the day.

Meal History and Tracking: The Intake History feature in the FOODSNAP application seamlessly integrates with MongoDB to record and organize user-generated data. As users input their dietary information through the application's frontend, this data is efficiently stored in the MongoDB database. The Intake History is a dynamic and real-time reflection of the user's dietary journey.

1. User-generated data, including food item names, quantities, date, and time of consumption, are stored in MongoDB. The NoSQL structure of MongoDB facilitates efficient storage and retrieval of this diverse and dynamic dietary information.
2. The intake history serves as a detailed log, presenting a list of consumed food items along with the respective day, date, and time.
3. Users can effortlessly review and analyze their past food entries, gaining valuable insights into their eating patterns.



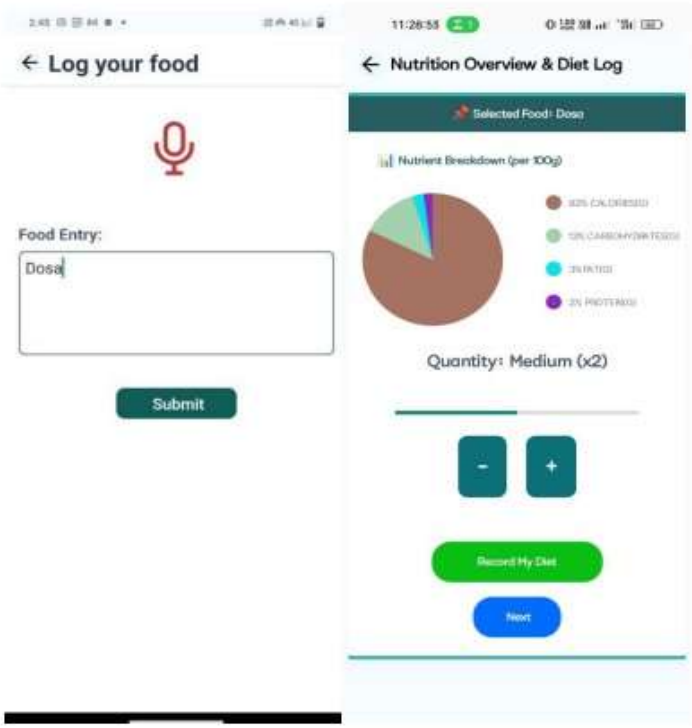
Food Entry Options: The FOODSNAP application introduces versatile methods for users to record their dietary intake seamlessly. Users can choose from two intuitive options such as manual entry via search and image recognition using YOLOv8 [6]. These flexible recording methods cater to different user preferences and enhance the overall user experience.

Manual Entry via Search: Empowering users with control, the manual entry option allows them to search for and add food items directly. Leveraging a user-friendly interface, individuals can effortlessly input the names, quantities, and other details of the consumed items. This method is ideal for users who prefer a hands-on and precise approach to recording their meals.

GPT Based Sentence Classification MODEL: It involves predicting the class label of dietary-related text inputs. Our model, based on the 'GPT architecture, is used to classify texts into one of 138 relevant categories of food we have selected. The model can predict a specific number of labels. A crucial aspect of our methodology is the integration of the GPT tokenizer. The tokenizer preprocesses input texts, converting them into a format suitable for the GPT model and predicts the final output variable among all the different variables using the users input sentence or description.

Image Recognition with YOLOv8:FOODSNAP utilizes YOLOv8 [6], an advanced image recognition model, to automate the process efficiently. With a simple snap/photo of their meal, users can have food items identified and categorized without manual data entry. This streamlines meal logging, particularly useful for users on-the-go. YOLOv8 [6] offers significant advancements in object detection and recognition, ensuring both speed and accuracy in identifying food items within images. During the development of FOODSNAP's image recognition system, a comprehensive dataset of up to 657 images was used to train the model, capturing diverse food items, and ensuring robustness in real-world scenarios. This dataset encompasses variations in shapes, sizes, colors, and contextual settings, enhancing the model's effectiveness.

International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 12, Issue 3, March-2024, Available online at: www.ijaresm.com



Gamification Feature: Incentivizing users to maintain consistent and accurate dietary records, FOODSNAP introduces a gamified system where users earn 5 points for each food entry. This point-based reward system serves as a motivational tool, encouraging users to actively engage with the application and diligently document their meals. Accumulated points not only reflect the user's commitment to their dietary journey but also unlock potential benefits and recognition within the gamified ecosystem. Recognizing the significance of sustained commitment to healthy habits, FOODSNAP incorporates streaks as a gamification feature. Users are rewarded for maintaining consistent engagement with the application over consecutive days. Whether it is logging meals, achieving specific dietary goals, or adhering to recommended nutritional guidelines, users can build and extend their streaks. This element adds an element of challenge, fostering a sense of accomplishment and reinforcing positive behavior patterns.

Personalized Diet Recommendations: In the realm of modern nutritional science, personalized meal recommendations have emerged as a pivotal aspect of promoting individual well-being. Recognizing the inherent diversity among individuals, especially in terms of age, weight, height, and gender, the FOODSNAP application employs a sophisticated approach to tailor dietary guidance for each user. FOODSNAP's personalized meal recommendations begin with user-entered details like age, weight, height, and gender, building a comprehensive picture of their physiology.
BMI, a key metric derived from weight and height, categorizes users into specific ranges like underweight or overweight, guiding tailored meal plans to align with individual health goals. Recognizing gender differences, FOODSNAP incorporates gender-specific considerations in its calculations, ensuring recommended meals are attuned to the unique metabolic rates and nutritional needs of each user.

BMI = weight / (height / 100) **2

Caloric Intake Calculation: The heart of personalized meal recommendations lies in the precise calculation of daily caloric intake. Leveraging renowned Harris-Benedict equations [5], the system computes the Basal Metabolic Rate (BMR) for both males and females. The formulas for calculating BMR are as follows:

- For males: $BMR_{male} = 88.362 + (13.397 \text{ x weight}) + (4.799 \text{x height}) (5.677 \text{ x age})$
- For females: $BMR_{female} = 447.593 + (9.247 \times \text{weight}) + (3.098 \text{ x height}) (4.330 \text{ x age})$

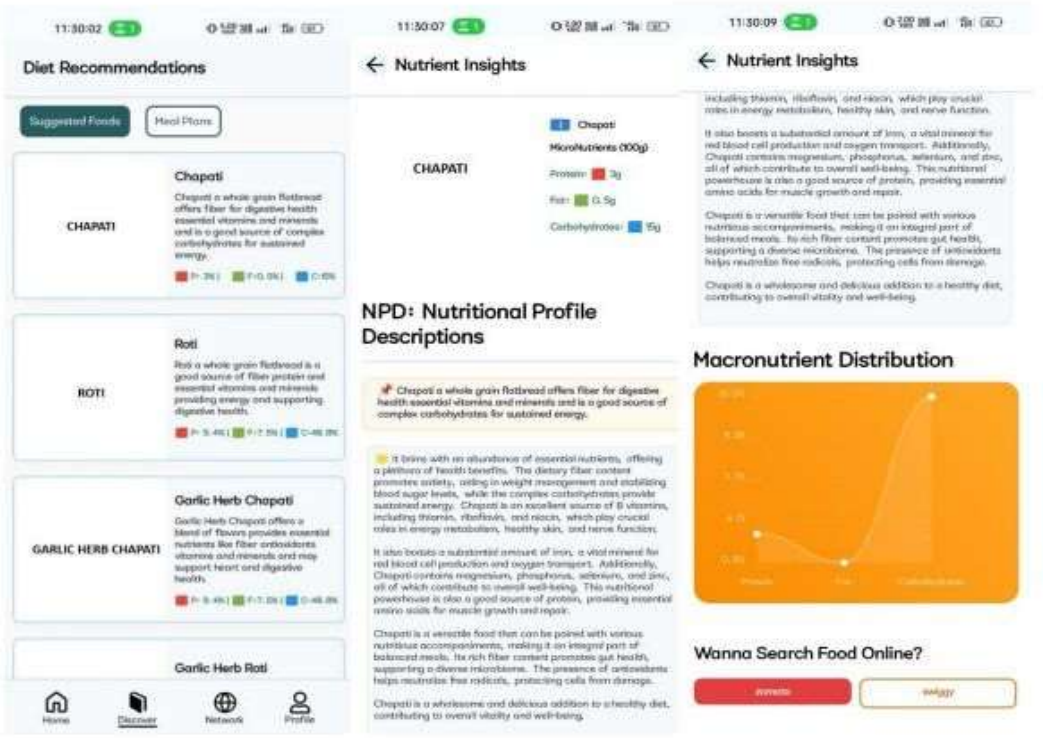
These formulas form the basis for determining the user's BMR, which is then multiplied by an activity factor to provide a personalized estimation of the calories they should consume daily. The activity factor, representing different activity levels, is a crucial component in tailoring recommendations to individual lifestyles

- Total Calories = BMR × Activity Factor
- The activity level multipliers include:

1. Sedentary: 1.2
2. Lightly Active: 1.375
3. Moderately Active: 1.55
4. Very Active: 1.725
5. Extremely Active: 1.9

- Food is suggested from different filtering criteria based on the user's BMI category:

a. **Underweight:** Suggests foods with high protein (>10g), high carbohydrates (>10g), or high calories (>200).
b. **Normal Weight:** Suggests foods with high protein (>10g), high carbohydrates (>10g), low fat (10), moderate calories (>10)
c. **Overweight or Obese**: Suggests foods with high protein (>10g), low carbohydrates (<10g), low fat (<10g) or low calories (<200).



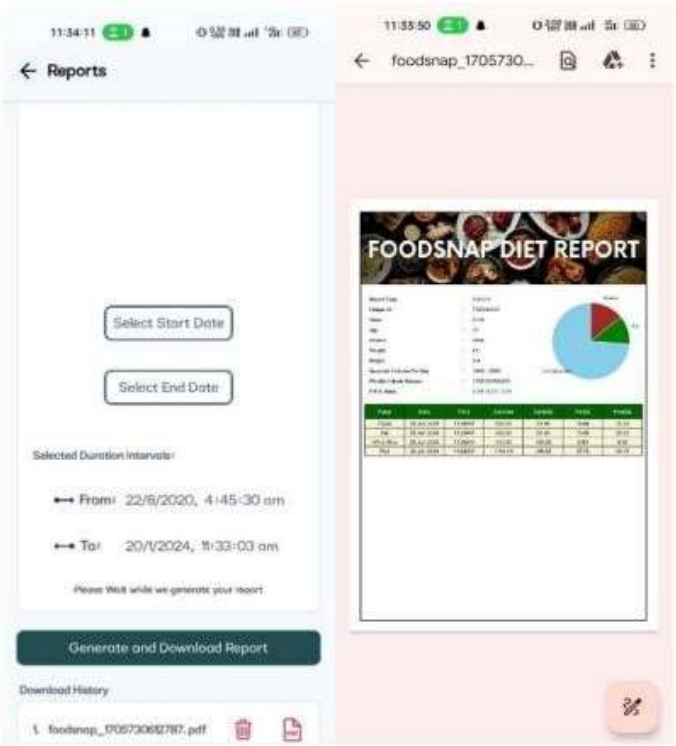
Report Generation: Customized Insights and Nutritional Analysis FOODSNAP's report generation feature introduces a tailored and detailed approach to providing users with personalized insights into their dietary patterns. The custom report, categorized as "Custom," encompasses a range of essential metrics and visual aids to enhance users' understanding of their nutritional habits.

Report Components:

a. **Unique Id:** Every FOODSNAP report is uniquely identified, ensuring accuracy and ease of reference for users seeking specific insights into their dietary behavior.

International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 12, Issue 3, March-2024, Available online at: www.ijaresm.com

b. **Name:** Personalization is prioritized with the inclusion of the user's name, making the report more relatable and usercentric.
c. **Age and Gender**: Crucial demographic information that influences dietary recommendations and nutritional needs is considered in the report.
d. **Weight and Height**: Recorded for precise caloric calculations, these details ensure dietary suggestions align with the user's unique body composition and health goals.
e. **Required Calories Per Day:** This calculated figure serves as a benchmark, representing the user's daily caloric intake requirement for optimal nutrition.
f. **Weekly Caloric Balance**: A summary of the user's caloric intake and expenditure over the week, aiding users in tracking their progress toward dietary and fitness objectives.
g. **P: F:C Ratio& Pie Chart:** Providing insight into macronutrient distribution, the report illustrates the proportion of proteins, fats, and carbohydrates in the user's overall diet.



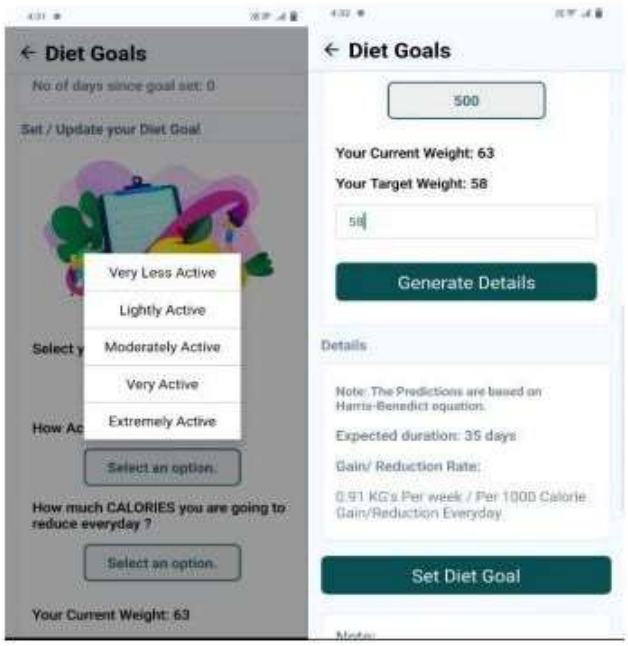
FOODSNAP's report generation is seamlessly integrated with MongoDB, leveraging the power of this robust NoSQL database to fetch and compile user-specific data for insightful reports. MongoDB serves as the centralized repository, storing critical information such as user profiles, food entries, and dietary preferences. Through efficient queries and data retrieval mechanisms, the report generation process dynamically pulls information from MongoDB, ensuring real-time accuracy and personalization in each report. This integration not only enhances the performance and scalability of the report generation feature but also underscores FOODSNAP's commitment to delivering a reliable and datadriven dietary management experience for its users.

Goal Setting and Tracking:

1. The Unique Physical Characteristics serve as crucial parameters for establishing and monitoring Diet Goals. The Harris-Benedict equation [5] is a method to estimate the Basal Metabolic Rate (BMR), indicative of the calories required by the body at rest.
2. As Described before, the total Calories are computed by multiplying BMR with the Activity Factors (Activity levels

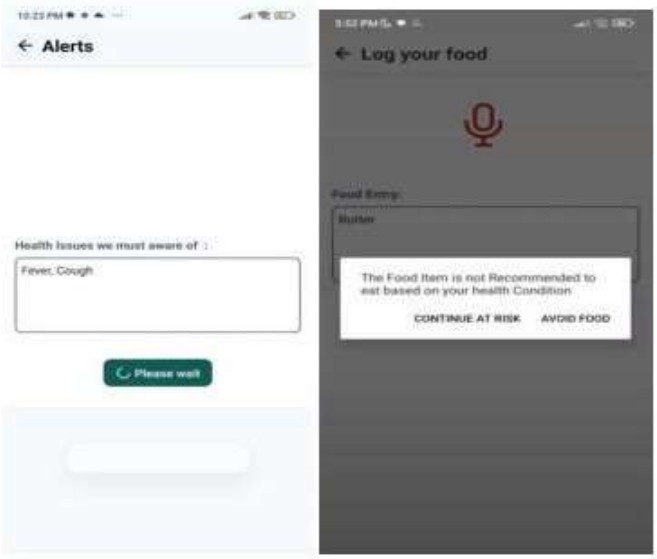
vary, depending on activity levels such as Sedentary, Lightly Active, Moderately Active, Very Active, and Extremely Active)

3. Applying specific criteria, such as a 500-calorie daily deficit, aiming for a 1-pound per week loss, and considering a sedentary activity level, we can effectively set and monitor user goals for weight loss, weight gain, etc. [4]



Allergen Alerts:

Allergen Alerts are delivered to users through Google's Gemini API. When logging food, the system checks against the user's health issues that are previously recorded in user's profile. Then they are used to notify whether a particular food item is recommended or not.



CONCLUSION

The development and implementation of FOODSNAP mark a significant stride toward revolutionizing dietary management through innovative technologies. As evidenced by the initial results and discussions, FOODSNAP exhibits promising outcomes in various aspects, from user engagement to the accuracy of image recognition and the effectiveness of personalized meal recommendations. The amalgamation of YOLOv8 [6] for image recognition, GPT for food name prediction, food allergen alerts, gamification features and MongoDB for robust data management forms the backbone of FOODSNAP success. The application's user-centric approach, encompassing gamification features and personalized recommendations, resonates with users, fostering consistent engagement and adherence to dietary goals. Therefore,FOODSNAP emerges not only as a sophisticated mobile application but as a holistic dietary companion. Its user-friendly interface, coupled with advanced technologies, empowers individuals to make informed choices about their dietary habits and overall well-being. As we continue this journey, the commitment to excellence, user satisfaction, and a healthier lifestyle remains at the core of FOODSNAP's mission. The fusion of cutting-edge technology and user centric design positions FOODSNAP as a transformative force in the realm of dietary management, poised to make a lasting impact on the lives of health-conscious individuals.

REFERENCES

[1]. Smart Diet Diary: Real-Time Mobile Application for Food Recognition by Muhammad Nadeem, Henry Shen, Lincoln Choy, and Julien Moussa H. Barakat - https://www.mdpi.com/2571-5577/6/2/53
[2]. Deep Learning for Food Image Recognition and Nutrition Analysis Towards Chronic Diseases Monitoring: A Systematic Review. https://www.researchgate.net/publication/372133738_Deep_Learning_for_Food_Image_Recognition_and_Nutrition_Analysis_Towards_Chronic_Diseases_Monitoring_A_Systematic_Review
[3]. Multilabel Text Classification with Label-Dependent Representation by Rodrigo Alfaro, Héctor Allende-Cid, and Héctor Allende - https://www.mdpi.com/2076-3417/13/6/3594
[4]. Calorie Calculator: https://www.nasm.org/resources/calorie-calculator
[5]. Revised-Harris–Benedict-Equation: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9967803
[6]. YOLOv8 Object Detection: https://medium.com/cord-tech/yolov8-for-object-detection-explained-practical-example-23920f77f66a