

Course 61998: Extended Project in Software Engineering

**Capstone Project Phase B**

**24-1-R-16**

**NutriSmart: A Personalized Recipe App for Optimal Health**

Students:

Khalid Sweid-324264530 Khalidsw02@gmail.com

Inbar Mizrahi-316582709 Inbar1289@gmail.com

Supervisor:

Dr. Natali Levi

[**Abstract 3**](#_ry3k1llt5lmh)

[**Introduction 3**](#_pwjuvkyl63c5)

[**System Architecture 5**](#_9eetilmdv7nh)

[**Process of development 6**](#_myc2d72g1gdd)

[GUI creation 6](#_rzcdmjqcws3j)

[Struggles along the way 6](#_geaoz85r71f3)

[Image Recognition Model 7](#_exnigbz8ho5e)

[UEC FOOD100 dataset 7](#_9r3braxsp176)

[**Challenges 8**](#_kd56ddybwx7)

[**Tests 10**](#_wsd96kiht8h3)

[Sign-up 10](#_e3pfcm8jhdw5)

[Log-in 10](#_8rl2t86rvwte)

[Update health condition 11](#_rinzw7juh5q6)

[Upload a photo 11](#_22m32wgv8bf9)

[Ask a nutrition 12](#_qsw5h2btue0g)

[**Flow Chart 13**](#_4o24cgzfzo37)

[**Activity Diagram 13**](#_j8m6o1xdkufs)

[**Class Diagram 14**](#_5mvkroeh9ox0)

[**User guide 15**](#_ko88tsuck42c)

[first page 15](#_rj207gtotgc9)

[Signup 16](#_sk5z8hodbhyn)

[Log in 16](#_s0030ucuq7cd)

[Continue as a guest 17](#_pjaesra4ccuh)

[Best recipe and desert of the month 18](#_pm3bidyaxmp0)

[Registered user screen 20](#_9zubwtqmvqfh)

[Menu user screen 21](#_fv7btg5zrmm4)

[Input Medical Conditions 22](#_rf31zekjwtty)

[Customize meal plan 23](#_kpchl5my7axe)

[Ask Nutritionist 24](#_67s58rqasdgb)

[Recipes 25](#_bfdph2x77mrm)

[Insert Image 26](#_ly0o0extmii0)

[**Programmer guide 27**](#_aha47ylgskof)

[Developer Manual: AI-Powered Nutritional Recipe Application 27](#_v1fy29ze73nf)

[1. Introduction 27](#_368mcv2g1zi8)

[2. Backend 27](#_16x5gbetd7bs)

[3. Frontend 28](#_jxkfmkgcsvwr)

[Key Features and Implementation Details 30](#_ioysoiecf0qv)

[AI-Powered Image Recognition 30](#_vcapfc9kmcg0)

[Personalized Nutritional Advice 30](#_n8q0yg75gxl3)

[Meal Plan Customization 30](#_hpqervs6exlc)

[Database 31](#_sht5s3whhw72)

[Testing 31](#_merwh0727krl)

[**Conclusion 32**](#_56ua253fsoot)

# Abstract

Proper nutrition is a vital aspect of overall health and well-being, emphasizing the critical importance of a balanced diet for optimal development and disease prevention. Adequate intake of essential nutrients supports growth, enhances cognitive function, and strengthens the immune system, underscoring the need for comprehensive dietary planning. Additionally, proper nutrition plays a crucial role in training and physical performance, providing the energy and nutrients necessary for muscle recovery and endurance.

In this project, we propose a web application that utilizes image recognition and analysis technology, specifically YOLO V10 algorithm, to analyze food images. This enables users to insert images alongside text, allowing them to inform the application about specific dietary needs and preferences, such as pregnancy, allergies, medical conditions, gain weight, muscle gain, and diseases, to receive personalized meal recommendations. Additionally, our application will integrate with ChatGPT-4 through APIs to process user illustrations and provide accurate and consistent recipes.

By leveraging deep learning techniques and AI, our solution aims to enhance and give accurate and helpful meal plans. Subsequently, our application provides accurate results by allowing users to insert images alongside text, a new feature designed to enhance recipe outcomes. Experimental results demonstrate that our application delivers superior performance, surpassing many of the latest state-of-the-art methods by a substantial margin.

Our project lays a solid foundation for future developments, and the potential to expand dietary features for even more personalized nutrition guidance.

# Introduction

With the advent of the digital era, it has become progressively difficult for consumers to uphold a nutritious diet using nutrition applications. Although these apps may seem attractive at first, they generally prioritize calorie counting as the main approach to monitor food consumption, which can be both monotonous and unnecessarily simplistic. Moreover, a significant number of these applications do not possess the functionality to enable users to upload photographs of their meals, hindering their capacity to precisely monitor and control their individualized nutritional requirements. Consequently, numerous consumers forsake these digital tools in order to seek professional counsel from a nutritionist. While there is a financial cost involved, collaborating with a nutritionist provides a personalized and thorough method for attaining and sustaining a healthy diet.

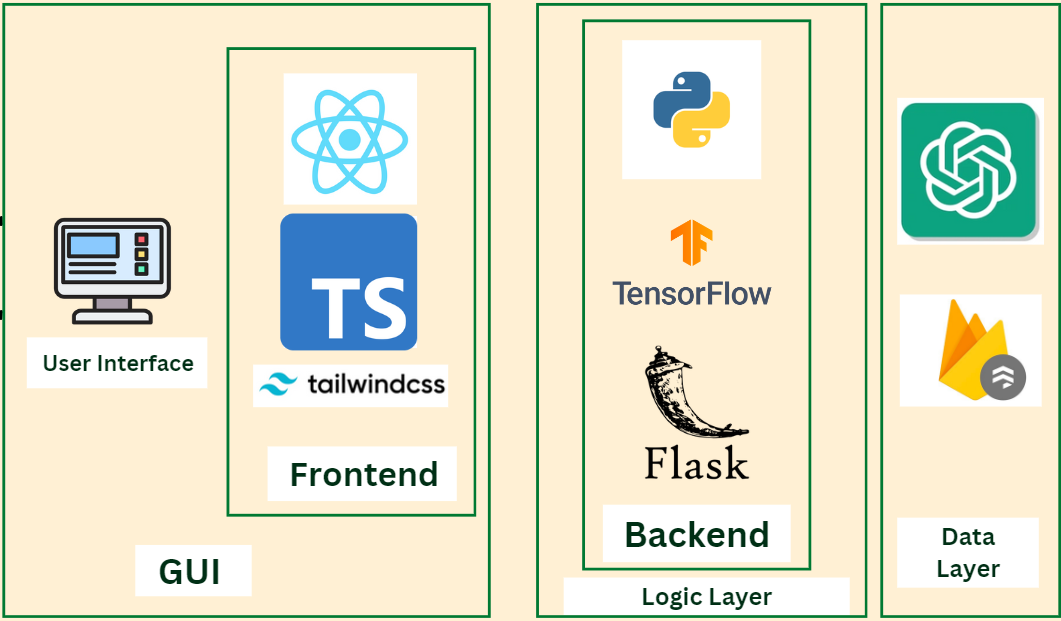
Nutrition is a crucial factor in human health, impacting many aspects of our well-being. Nutrition plays a crucial role in preventing and managing various diseases, with inadequate nutrition frequently being a major cause of ailments including obesity, diabetes, and cardiovascular diseases. A diet that does not contain necessary nutrients can cause shortages and weaken the immune system, making existing health problems worse. Considering its significant influence, it is essential to embrace appropriate and personalized dietary strategies that address our distinct nutritional requirements. Customized nutrition plans assist in maintaining a well-rounded consumption of vitamins, minerals, and essential nutrients, thereby fostering optimal well-being and averting illness. The significance of nutrition in human existence emphasizes the need for meticulous dietary strategizing and expert counsel to sustain general health and welfare.

The use of artificial intelligence (AI) into nutrition apps has the capacity to transform dietary management and greatly enhance user outcomes. Nutrition apps powered by artificial intelligence may process extensive data to offer tailored dietary suggestions, considering specific health measurements, tastes, and objectives of each individual. Our program utilizes state-of-the-art AI technologies to offer an unmatched nutrition management experience. By connecting OpenAI's API with ChatGPT-4, users may submit visual representations of their dietary preferences and obtain customized recipes that are specifically designed to meet their individual requirements. In addition, we employ YOLO V10, an advanced object detection algorithm, to evaluate and identify the food items in users' photographs. By integrating text-based illustrations and visual recognition, our program is able to provide precise and tailored nutritional advice with a high level of accuracy. This advanced AI integration enables users to effortlessly and confidently accomplish their nutritional goals.

# 

# System Architecture

We have developed a recipe application that is powered by artificial intelligence and offers individualized meal suggestions. Through the program, users are able to engage in conversation with a bot that is connected to the ChatGPT API in order to obtain recipes and share pictures of their meals. The system makes use of photo recognition to determine the amount of food that users consume. If the users have specific dietary goals, such as increasing their protein intake, the system will prescribe precise meal additions in grams to aid them in accomplishing their goals.

The system was designed from the ground up, with careful consideration given to selecting the tools and technology that would be best suitable for bringing this concept to life. 

We went with React because of its component-based architecture, which makes it possible to create a user experience that is both responsive and dynamic during the development process. Because of its widespread use and robust ecosystem, it is an excellent choice for the development of web applications that are both scalable and offer good performance.

By utilizing TypeScript, type safety and code predictability were significantly improved, which resulted in a significant improvement in the development process. This was accomplished by discovering problems at an earlier stage and making codebase maintenance easier.

For the purpose of modifying our interface, Tailwind CSS offers an approach that is both highly adaptive and highly efficient, hence removing the requirement for considerable bespoke CSS. A flawless design was guaranteed by the utility-first philosophy, which also made it possible to speed up the development process.

Because of its adaptability, widespread use, and extensive library resources for the development of machine learning models and application programming interfaces (APIs), Python was an excellent choice for handling backend logic.

We decided to use Flask as our backend framework because of its ease of use and adaptability. Flask allows for smooth interaction with a number of APIs while also maintaining a lightweight and speedy development process.

Because of its powerful machine learning capabilities, TensorFlow was the best option for identifying pictures. TensorFlow was the best choice. It made it possible for us to carefully examine images of meals and determine the categories of food components that were contained inside them.

Users are able to interact with the bot in order to receive real-time meal recommendations and instruction thanks to the integration of OpenAI's application programming interface (API), which made it possible for the app to provide a conversational interface.

Firestore Database: Firestore was chosen because of its ability to synchronize data in real time, its scalability, and its seamless integration with our React application. It made it possible for us to keep user data in a secure and efficient manner.

Each technology made a significant contribution to the development of a unified and intelligent application, which facilitated a more streamlined development process and enabled us to create an advanced solution that was tailored to the nutritional requirements of the user.

# Process of development

We are now in the phase of system development. We will document the stages we are undertaking and the challenges we are encountering.

This is our application link: <https://nutri-smart-app.vercel.app>

## GUI creation

We began the process of writing the scripts and creating the graphical user interface (GUI) when we realized that we had utilized a private key for interacting with GPT. Upon attempting to upload our code to Git, we encountered a problem as GitHub detected the secret credentials that were hardcoded in the script. Consequently, we initiated an investigation into how to address this issue and determine the appropriate approach for managing API keys. We utilized a .env file, a basic text document employed to store environment variables for the application. These files are typically utilized to segregate configuration settings and sensitive information from the application code. This facilitates a more organized and safe method for handling environment-specific data such as API keys, database credentials, and other configurations that should not be hard-coded into the source code.

## Struggles along the way

During the writing process, we realized that our file organization is improper and we subsequently refined our classification by specifically identifying the client server. We have modified the method by which the client delivers a request to the server. As a result, the server now handles all the cognitive processing, backend operations, and API queries. Initially, in order to get familiarity and understanding of the functioning of things, we mistakenly performed certain backend tasks within the frontend files. Upon recognizing the correct method of dividing the files and assigning duties, we made efforts to modify it, resulting in the expenditure of a portion of our time.

## Image Recognition Model

The development of the image recognition model outlined in the code entails many essential stages. Firstly, the required libraries are imported, and the dataset, including photos and their associated labels, is loaded and subjected to preprocessing. After cleaning and formatting the labels, the photos are resized and normalized in order to make them suitable for training. Next, the dataset is partitioned into training, validation, and test sets to guarantee accurate evaluation of the model. A Keras-based convolutional neural network (CNN) architecture is constructed, consisting of several layers such as convolutional, pooling, and dropout layers, to extract pertinent information and mitigate the risk of overfitting. The model undergoes training using the training dataset, and real-time data augmentation is implemented using the ImageDataGenerator to improve the model's capacity to generalize. Checkpoints are employed during training to preserve the model from which the highest performance is achieved. Following training, the model is loaded and employed to generate predictions on fresh photos, with the outcomes visually represented and assessed to determine the predicted categories.

## UEC FOOD100 dataset

The UECFOOD100 dataset constitutes a substantial advancement in food image identification and nutritional analysis. This extensive compilation, created by academics at the University of Electro-Communications in Tokyo, Japan, includes a varied selection of 100 unique food categories, each carefully picked to represent traditional Japanese cuisine. The dataset's strength resides in its broad diversity, encapsulating the intricate visual attributes of foods from traditional Japanese cuisine to widely recognized worldwide dining. Each category in UECFOOD100 has roughly 100 photographs, culminating in a substantial corpus of around 10,000 food images, all photographed under real-world situations to guarantee practical applicability.

The photographs in UECFOOD100 are distinguished by their variety in lighting conditions, plate styles, and perspectives, accurately replicating the obstacles encountered in actual food recognition situations. This heterogeneity is essential for creating resilient machine learning models that can generalize across many presentation situations. The dataset's producers have meticulously included many instances of the same meal served in diverse fashions, thus addressing the intrinsic visual variability within food categories. This method greatly improves the dataset's efficacy in training algorithms to identify dishes despite discrepancies in preparation and presentation.

Additionally, UECFOOD100 includes extensive metadata, featuring bounding box annotations for each food item included in the photos. This supplementary information is essential for tasks beyond simple classification, facilitating the creation of advanced algorithms for food localization and segmentation in intricate, multi-item meal photos. The dataset's architecture and annotation framework enable its implementation in many computer vision tasks, ranging from fundamental food categorization to more sophisticated applications like nutritional content calculation and portion size analysis.

The use of UECFOOD100 in nutritional analysis and dietary monitoring is paramount. Given the persistent global difficulties of obesity and diet-related health disorders, the automatic recognition and analysis of food intake via picture analysis offers a promising opportunity for intervention and research. UECFOOD100 offers a standardized benchmark dataset that allows researchers and developers to evaluate the efficacy of various algorithms and methodologies in food picture identification, hence promoting innovation and advancement in this vital field of research.

UECFOOD100 serves as a fundamental resource in food picture recognition, providing researchers and developers with a comprehensive, diversified, and meticulously annotated dataset for training and assessing machine learning models. Its thoroughness, consideration of real-world variability, and meticulous annotations render it an indispensable resource for enhancing the field of automated dietary assessment and nutritional analysis. As research in this domain progresses, datasets such as UECFOOD100 will be crucial in fostering innovation and enhancing the precision and relevance of food recognition technology across diverse contexts, ranging from individual health management to extensive nutritional epidemiology investigations.

# Challenges

While developing our recipe application, we addressed technical challenges and individual working methods to establish an innovative platform. One individual favored autonomous work, while the other excelled in a cooperative setting. Ultimately, we devised a method to integrate various methodologies, optimizing productivity and enhancing problem-solving efficacy. This equilibrium enabled us to advance, despite significant external adversities.

Our efforts occurred under significant stress, as both of us reside near the border in Israel during a period of war. The incessant noises of bombs and explosions established an atmosphere of fear and uncertainty; however, we remained focused and committed to delivering a functional system.

The primary objective of our application was to include contemporary AI technologies to assess uploaded meal photographs and provide tailored nutritional guidance. Utilizing TensorFlow, we developed a system that identifies meal contents and, through integration with the ChatGPT API, offers personalized recommendations aligned with the user's nutritional objectives, such as enhancing protein consumption. A distinctive feature of our application is its ability to "remember" the user's health problems, utilizing this knowledge in subsequent encounters with the bot to enhance the relevance and personalization of each tip. This is a distinguishing characteristic that differentiates our application from competitors in the market.

Nonetheless, although TensorFlow facilitated the implementation of image recognition, it concurrently presented obstacles. The system occasionally misinterpreted objects in the images, concentrating on the plate or utensils rather than the food itself, thereby compromising the accuracy of our meal recommendations. Notwithstanding efforts to optimize the model, this continues to be a domain requiring enhancement.

We encountered substantial challenges with the integration of the OpenAI API for the chatbot. At one juncture, we inadvertently posted the API key to GitHub, resulting in the system's malfunction until the key was adequately secured and reset. The delay hindered our efforts, as we devoted extra time to resolving the issue.

Furthermore, owing to budget limitations, we still lack a dedicated server executing the Python code required for TensorFlow image identification. Although the application functions flawlessly on our local PCs, consumers evaluating the system will not utilize the picture recognition feature due to its absence on a live server. Operating such services incurs significant expenses, and regrettably, we were unable to maintain those costs throughout the development phase.

Moreover, while our intentions were to develop a full nutritionist page, time constraints necessitated the prioritization of more essential features, such as the integration of the GPT chatbot and the resolution of API difficulties. This feature is hence unfinished.

A major problem faced during the development process pertained to version control and collaboration. The team initially created a Git repository to oversee the project's source code and promote collaborative development. Nevertheless, unexpected issues emerged during the publication and deployment of the application from this repository. In light of these challenges, the team strategically opted to transfer the project to a new Git repository. The switch, albeit essential, presented an unforeseen challenge: one team member utilizing an Apple operating system encountered continual difficulties in submitting changes to the new repository. Despite diligent attempts, the team could not rectify these compatibility concerns.

To address this difficulty and maintain momentum, the team implemented an alternative collaborative technique. They chose to implement changes manually and assign one team member to oversee the Git repository. This method required heightened dependence on face-to-face collaboration and the utilization of alternate communication platforms to synchronize development activities. This workaround facilitated project progression, highlighting the necessity of predicting and resolving potential cross-platform compatibility concerns in collaborative software development. The team's experience with deployment challenges further shaped their prudent approach to version control, as they aimed to maintain the stability of the successfully launched program. This difficulty underscored the essential requirement for resilient, platform-independent collaboration tools and accentuated the importance of flexibility in addressing unexpected technological difficulties in software development initiatives.

# 

# Tests

## Sign-up

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Description | | Successful Login | | Result |
| **#** | **Action** | **Input** | **Expected Result** | All Tests  Passed |
| 1 | Sign up with wrong email format | test.com | “Invalid email format, email must be domain@example.com” |
| 2 | Different passwords | Password: 123456  Confirm Password: 1234 | “Password do not match” |
| 3 | Sign up with too short password | 123 | “Password should be at least 6 characters” |
| 5 | Normal Sign-up |  | Move to the login screen |

## Log-in

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Description | | Successful Login | | Result |
| **#** | **Action** | **Input** | **Expected Result** | All Tests  Passed |
| 1 | Login with wrong email | test.com | “Invalid email format, email must be domain@example.com” |
| 2 | Login with wrong password | 123456 | “Incorrect Password!” |
| 3 | Login with to short password | 123 | “Incorrect Password!” |
| 4 | Email matches the saved password |  | “Welcome!” |
| 5 | Login a nutritionist | [nut@email.com](mailto:nut@email.com)  123123 | “Welcome!” |  |

## 

## Update health condition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Description | | Successful Login | | Result |
| **#** | **Action** | **Input** | **Expected Result** | All Tests  Passed |
| 1 | Click on Obesity | Obesity and | “Submit successfully”  Transfer to main page |
| 2 | Text Only | Insert text - can’t eat lactose | “Submit successfully”  Transfer to main page |
| 3 | Click and text | Click on cancer and add text- “reduce amount of salt” | “Submit successfully”  Transfer to main page |

## 

## Upload a photo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Description | | Successful Login | | Result |
| **#** | **Action** | **Input** | **Expected Result** | All Tests  Passed |
| 1 | Upload a photo of food with demand of some nutrition values and text | Picture of meat and rice, goals of 600 cal and 35g protein, free text- add vegetables | The nutrismart analyze the photo and presented recommendations to add broccoli and zucchini based on my health condition saved in the system |
| 2 | Upload a photo with text | Picture of fish, free text requesting to add some side dish | Based on my health condition saved in the system, it recommended me to add rice and salad since I am diabetic and can’t eat bread. |
| 3 | Upload only a picture with protein target | Picture of soup | The system recommend me to add 200 grams of chicken breast to meet my goals |
| 4 | Upload a photo not related to food | Picture of an empty plate  50g protein | The system will recognize the object in the picture, then recommend food based on the values that filled in, here - to eat 3 eggs, pack of tuna and 2 spoons of white cheese. |

## Ask a nutrition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Description | | Successful Login | | Result |
| **#** | **Action** | **Input** | **Expected Result** | All Tests  Passed |
| 1 | Ask one question | Title, email and content | “Submit successfully”  Transfer to main page |
| 2 | Add more than one question | Insert another question | The question submitted and is appears in the page |
| 3 | Add questions that out the limits of the page space | more questions | The questions are piling up and you can scroll them. |

# Flow Chart

# 

# 

# 

# 

# 

# Activity Diagram

# 

# Class Diagram

# 

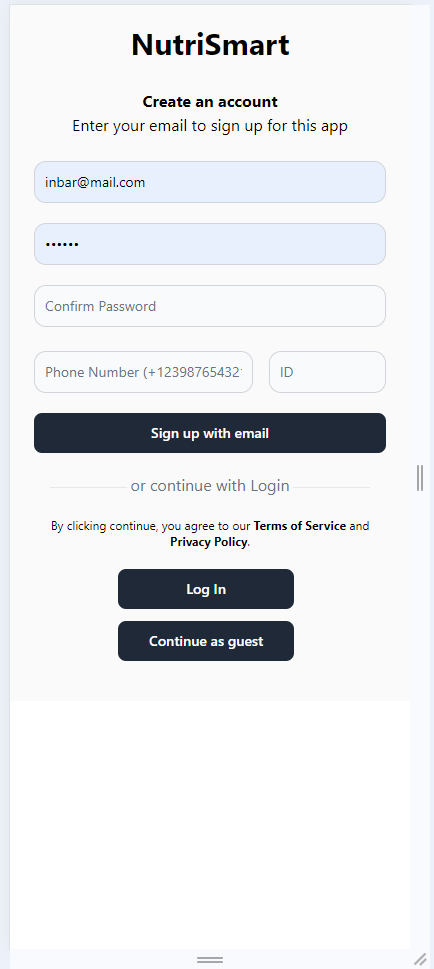
# User guide

This is our application link: <https://nutri-smart-app.vercel.app>

## first page

This is the initial page that visitors will encounter while entering our website.

You may register on our website, sign in, or proceed as a guest.



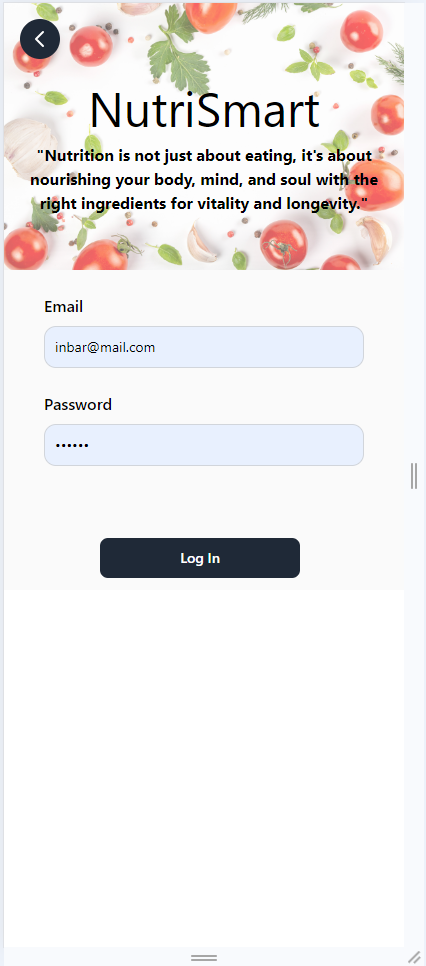
## 

## Signup

Accurately completing all fields (valid email, sufficiently lengthy password) and selecting ‘Signup with email’ will redirect you to the login screen.

## Log in

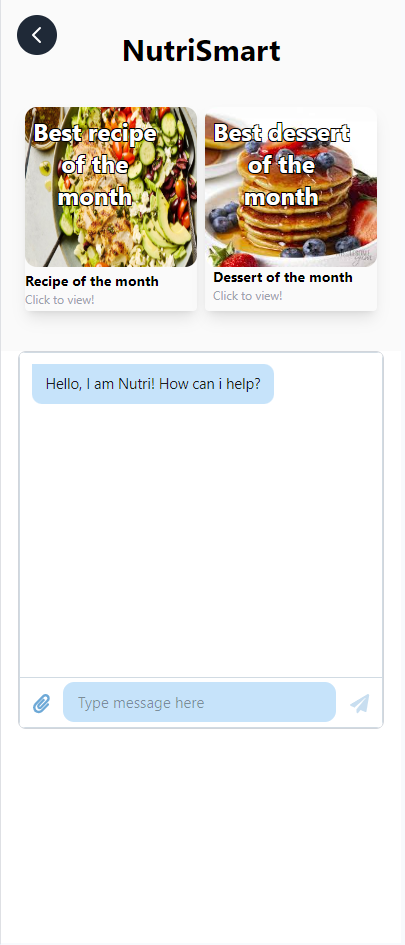
This page is intended for those who are already registered on our website. Users can log in and utilize the application using their email and password.



## Continue as a guest

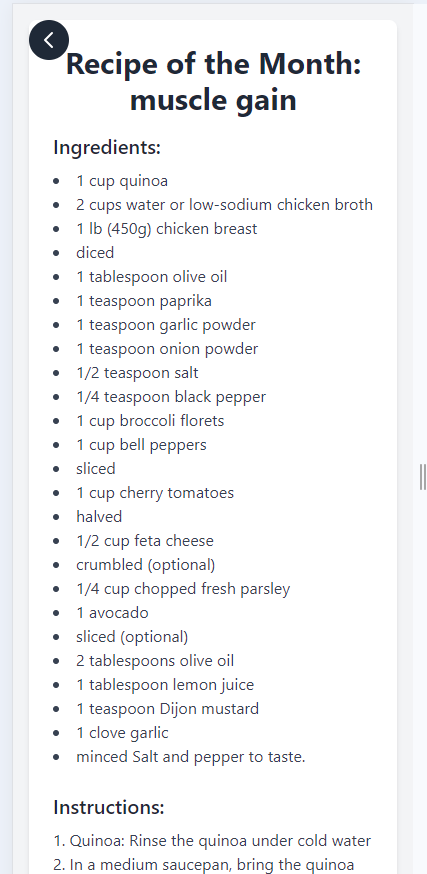
To participate as a guest in our application, registration with email and personal information is not required. You may proceed as a guest, where you can engage in conversation with Nutri the Bot and get the recipe and the tastiest dessert of the month.

Chatting - guest

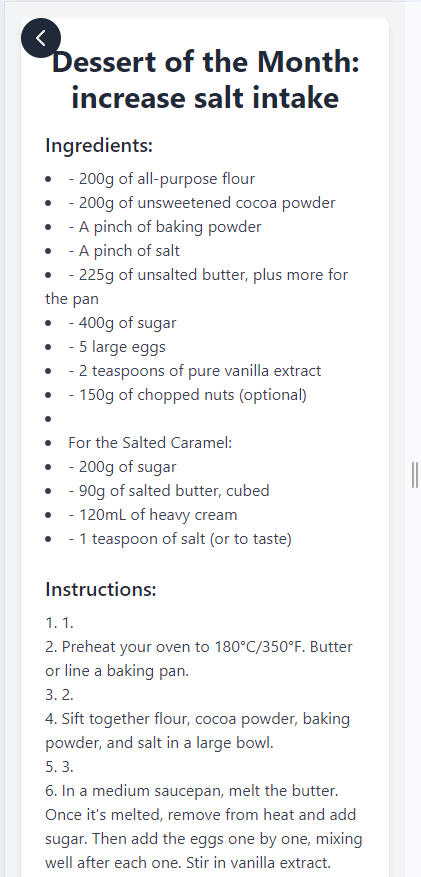
Here, you may input the desired text and request delectable meals that meet your specifications. If it pertains to a conditioned circumstance such as allergies, or if you are an athlete seeking to augment your protein intake. Note that, as a guest, your health information will not be preserved for future interactions with the bot.

## Best recipe and desert of the month

It is the recipe that has been selected as the best of the month. The choice that is most heavily favored by users.

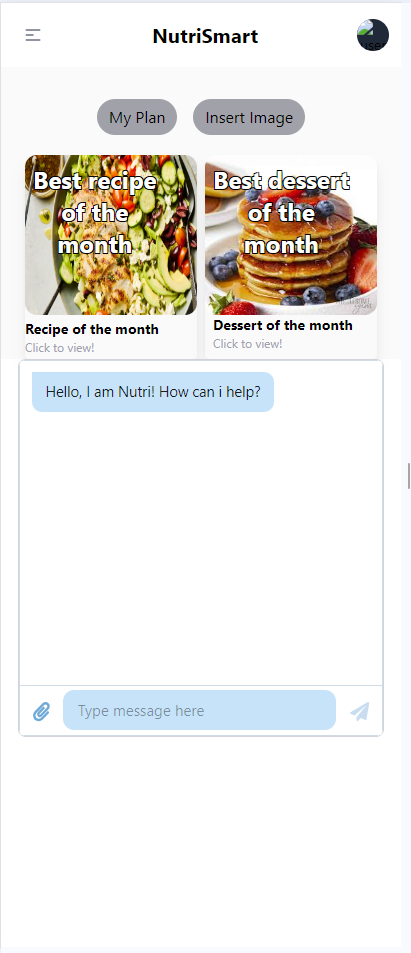


The dessert recipe of the month is this one, and it is of the highest quality. The choice that is most heavily favored by users.



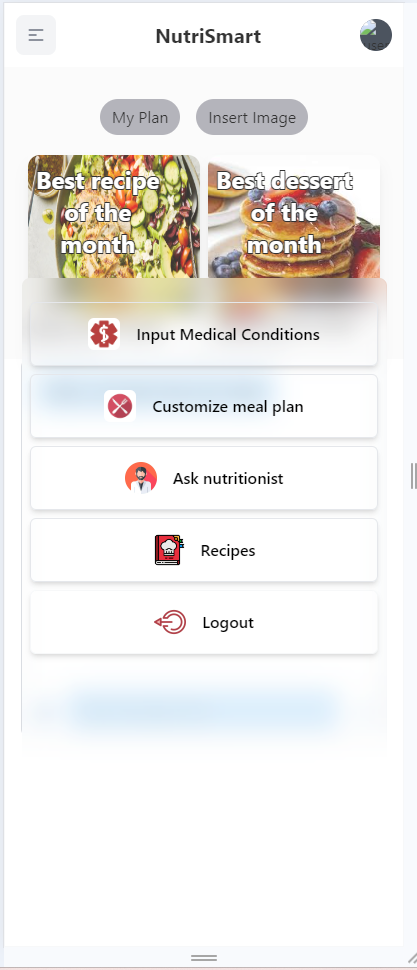
## Registered user screen

As a user, you have the ability to search for recipes based on topics, upload an image, review the recipe and dessert of the month, engage in conversation with nutri the bot, and browse the primary menu.



## Menu user screen

This menu allows the user to input their current medical condition, ensuring that each interaction with the bot accounts for their health status. A meal plan can be tailored to individual preferences. Users will have the opportunity to submit questions to a nutritionist for a response, browse recipes by category, and log out of the system.



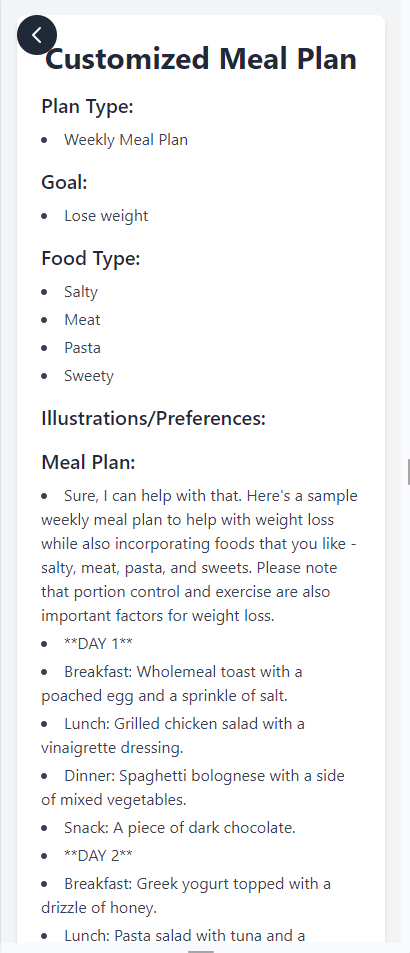
## Input Medical Conditions

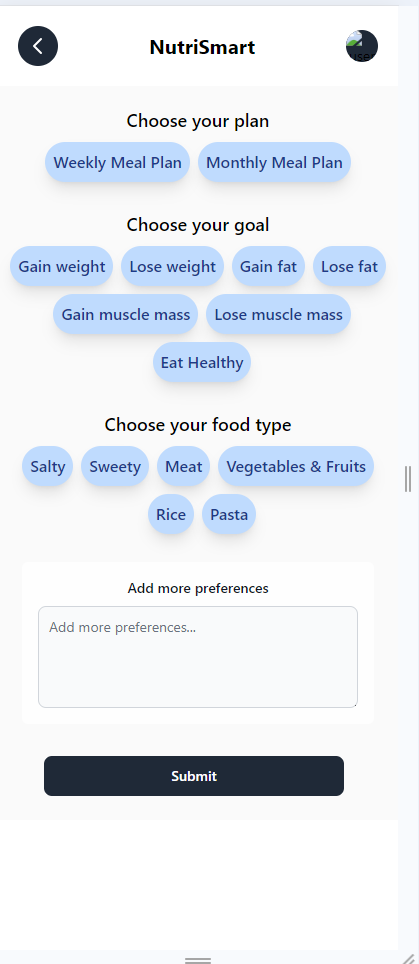
This screen allows you to input your medical status, any allergies, and diseases that may restrict your diet; this is the appropriate section to mention them. You may select from the provided conditions or compose your own, your health status, or a combination of both. Your medical condition will be considered in every interaction with Nutri, the bot.

## 

## Customize meal plan

This website allows you to design your own food plan. You may select either a weekly or monthly program, determine the objective of your new dietary habits, and specify your preferred food types. You may select multiple choices. Input additional text with specific constraints or particular requirements, then proceed to click submit. The application will formulate the plan and deliver it to you. You can return to it from the home screen by selecting “My Plan.”

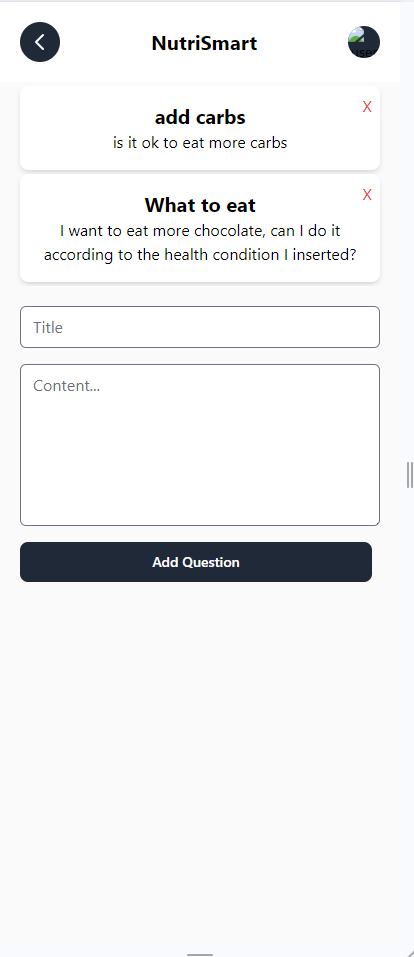




## 

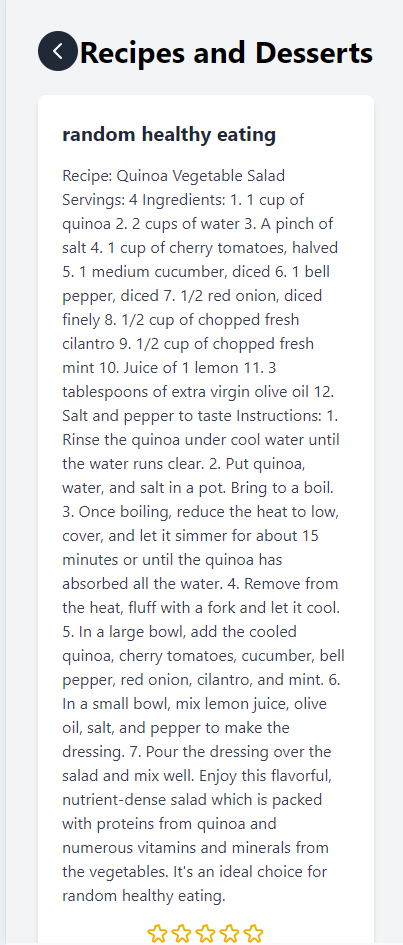
## Ask Nutritionist

If you encounter any issues with your plan, have uncertainties, or wish to seek professional advice, you may effortlessly click “Ask Nutritionist” to submit a question to one of our nutritionists, who will provide a response. You may submit multiple inquiries and access your questions here.



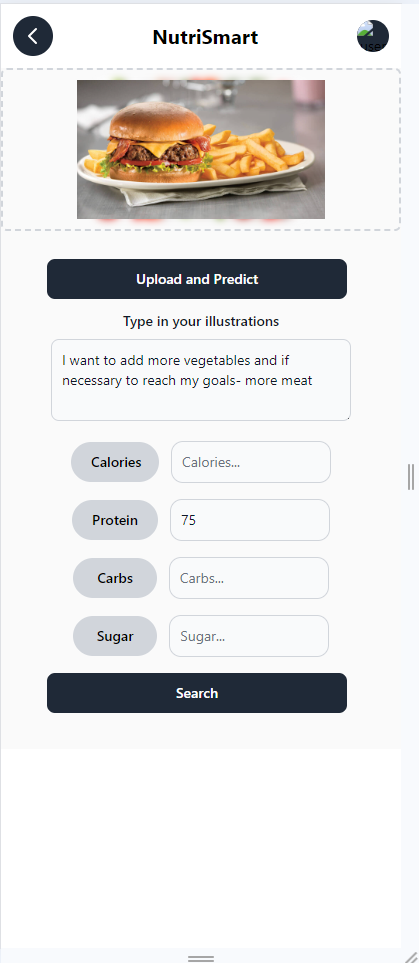
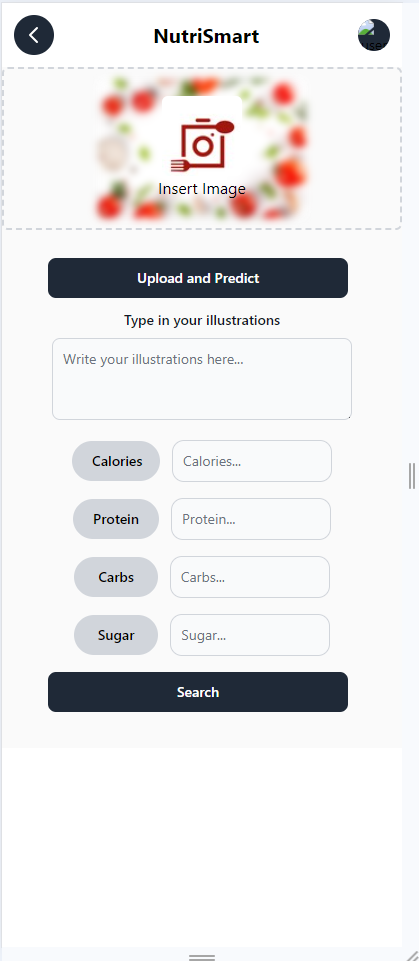
## Recipes

You can search for general recipes by category; options such as Keto diet and Gluten-free diet will be displayed upon clicking "Generate More Recipes," prompting the system to present additional recipes by category.



## Insert Image

The option of inserting an image and request for nutrition tips based on your current meal, is the renewal of our project. Here you upload a photo, the system will analyze the content of it and based on your request, goals, and health condition will advise you what food you should add to your meal.



# Programmer guide

## Developer Manual: AI-Powered Nutritional Recipe Application

### Introduction

This developer manual offers a complete overview of the application powered by artificial intelligence and utilizes nutritious recipes. The project is divided into two primary components: the Backend, which is developed in Python, and the Frontend, which is built with TypeScript and React. The application incorporates cutting-edge artificial intelligence technology to analyze photographs of meals, provide individualized nutritional guidance, and generate individualized meal plans depending on the user's health problems and dietary objectives.

### Backend

The backend resides in the src/Backend/ directory and comprises multiple Python scripts that manage diverse functionalities:

src/

└── Backend/

├── main.py

├── CheckOnlineUser.ts

├── ContentProcessor.ts

├── CustomizeMealPlanContentProcessor.ts

├── DatabaseUtils.ts

├── ExtractCustomizeMealPlanFromDatabase.ts

├── ExtractDessertOfTheMonthFromDatabase.ts

├── ExtractRecipeOfTheMonthFromDatabase.ts

├── GenerateRecipesAndDesserts.ts

├── GoalsManager.ts

├── HandleAIMsg.tsx

├── MedicalConditionsManager.ts

├── Multi-label-classification.ipynb

├── SaveCustomizeMealPlanToDatabase.ts

├── SaveIllustrationsToDatabase.ts

├── SaveRecipesAndDessertsToDatabase.ts

└── UpdateUserQuestions.tsx

main.py: Image analyzing and food recognition python file.

CheckOnlineUser.ts: Manages user online status, check, set and reset online status for user.

ContentProcessor.ts: Prompt creation for ChatGPT.

CustomizeMealPlanContentProcessor.ts: Handles customization of meal plans.

DatabaseUtils.ts: Provides utility functions for database operations. (get users information, check correct passwords, check user existence, etc.)

ExtractCustomizeMealPlanFromDatabase.ts: Retrieves customized meal plans from the database.

ExtractDessertOfTheMonthFromDatabase.ts: Fetches the dessert of the month.

ExtractRecipeOfTheMonthFromDatabase.ts: Retrieves the recipe of the month.

GenerateRecipesAndDesserts.ts: Generates random new recipes and desserts.

GoalsManager.ts: Manages user goals and preferences.

HandleAIMsg.tsx: Handles AI-generated response and requests.

MedicalConditionsManager.ts: Manages user medical conditions.

Multi-label-classification.ipynb: Jupyter notebook for multi-label classification

SaveCustomizeMealPlanToDatabase.ts: Stores customized meal plans in the database.

SaveIllustrationsToDatabase.ts: Saves illustrations to the database.

SaveRecipesAndDessertsToDatabase.ts: Stores recipes and desserts in the database.

UpdateUserQuestions.tsx: Updates user-related questions.

### Frontend

The frontend is in the `Frontend` directory and comprises multiple TypeScript React components:

src/

└── Frontend/

├── App.tsx

├── BestDessertOfTheMonth.tsx

├── BestRecipeOfTheMonth.tsx

├── CustomizeMealPlan.tsx

├── CustomizeMealPlanResults.tsx

├── GuestPage.tsx

├── InputGoals.tsx

├── InputMedicalConditions.tsx

├── LoadingSpinner.tsx

├── Login.tsx

├── NSChatBot.tsx

├── NutritionistChat.tsx

├── NutritionistPage\_ClientSide.tsx

├── PrivacyPolicy.tsx

├── RecipesAndDesserts.tsx

├── ResultsPage.tsx

├── Terms.tsx

├── UserContext.tsx

├── UserLoginPage.tsx

├── UserPage.tsx

├── UserPlan.tsx

├── firebase.ts

├── index.css

└── main.tsx

App.tsx: The main application component.

BestDessertOfTheMonth.tsx: Displays the best dessert of the month.

BestRecipeOfTheMonth.tsx: Shows the best recipe of the month.

CustomizeMealPlan.tsx: Allows users to customize meal plans.

CustomizeMealPlanResults.tsx: Displays results of customized meal plans.

GuestPage.tsx: Landing page for guest users.

InputGoals.tsx: Component for users to input their goals.

InputMedicalConditions.tsx: Allows users to input medical conditions.

LoadingSpinner.tsx: Displays a loading animation.

Login.tsx: Handles user login.

NSChatBot.tsx: Implements the nutritionist chatbot.

NutritionistChat.tsx: Provides a chat interface with a nutritionist.

NutritionistPage\_ClientSide.tsx: Client-side component for the nutritionist page.

PrivacyPolicy.tsx: Displays the privacy policy.

RecipesAndDesserts.tsx: Shows available recipes and desserts.

ResultsPage.tsx: Displays various results to the user.

Terms.tsx: Shows the terms of service.

UserContext.tsx: Manages user context throughout the application.

UserLoginPage.tsx: Handles user login process.

UserPage.tsx: Displays user-specific information.

UserPlan.tsx: Shows the user's meal plan.

firebase.ts: Configures and initializes Firebase for the application.

index.css: Contains global CSS styles.

main.tsx: The entry point for the React application.

## 

## Key Features and Implementation Details

### AI-Powered Image Recognition

The program utilizes TensorFlow for image recognition, with the implementation located in the `Multi-label-classification.ipynb` notebook combined with YOLO algorithm to improve results. This robust tool allows users to upload images of their meals for comprehensive examination. Properly configuring TensorFlow in the backend environment is essential for implementing this capability. The trained model is subsequently employed to analyze uploaded photos. To sustain and enhance the system's precision, it is essential to address any misidentifications and perpetually train the model with updated data. This continuous enhancement guarantees that the image recognition functionality remains strong and dependable for users evaluating their meals.

### Personalized Nutritional Advice

The application’s primary function encompasses tailored nutritional guidance, utilizing the ChatGPT API via HandleAIMsg.tsx. This feature necessitates meticulous management of API keys, integration of user-specific data from GoalsManager.ts and MedicalConditionsManager.ts, and comprehensive error handling. User profile management is executed across multiple components, with UserContext.tsx overseeing application-wide user state, and CRUD operations managed through DatabaseUtils.ts. Input validation and sanitization are essential in components such as InputGoals.tsx and InputMedicalConditions.tsx. The meal plan customization feature enables customers to personalize their nutritional plans, employing CustomizeMealPlan.tsx for the interface and CustomizeMealPlanContentProcessor.ts for backend functionality, with data storage handled by specialized database interaction components.

### Meal Plan Customization

The application emphasizes recipe and dessert administration, encompassing the creation, storage, and retrieval of culinary information. This procedure is enabled by elements like GenerateRecipesAndDesserts.ts for content generation, along with other database interaction scripts for storage and retrieval. Featured goods are displayed with specialized components such as BestRecipeOfTheMonth.tsx and BestDessertOfTheMonth.tsx. The nutritionist chat feature offers customers an interactive platform for nutritional guidance, developed with NutritionistChat.tsx and NSChatBot.tsx, which integrates with an AI backend for intelligent responses and includes real-time updates and message history management.

### Database

The design of the database is essential for effective data management, necessitating a well-organized schema, appropriate indexing for performance enhancement, and strong safeguards for data consistency and integrity. Security is essential, requiring robust authentication and permission systems, deployment of HTTPS, input sanitization, regular updates of dependencies, and effective error handling to protect sensitive data. Strategies for performance optimization encompass caching methods, query optimization, lazy loading of substantial information, pagination for large data sets, and the implementation of server-side rendering to enhance initial load times.

### Testing

A thorough testing method includes unit tests for discrete components, integration tests for system coherence, and end-to-end tests for user scenario simulation. Prospective advancements may encompass enhancements in image recognition precision, augmentation of AI functionalities within the nutritionist chat feature, establishment of a recipe recommendation system, creation of a mobile application (as opposed to a web-based platform), and incorporation with wearable devices for real-time health data monitoring.

This AI-driven nutritional recipe application exemplifies a sophisticated amalgamation of several technologies designed to deliver individualized dietary guidance and meal planning. The developer handbook functions as an exhaustive reference for team members, detailing the system architecture, essential feature implementations, and optimal techniques for contributing to the project's continuous development and enhancement. The success and advancement of this project will rely on consistent code evaluations, compliance with best practices, and a dedication to ongoing education and innovation in the swiftly evolving domain of AI-driven nutritional technology.

# 

# Conclusion

In spite of the many challenges we faced, our team was able to effectively complete the key objectives of our recipe application project, demonstrating remarkable resilience and adaptability in the face of adversity. Our accomplishments are particularly noteworthy when one considers the extraordinary circumstances under which we operated, such as the ongoing battle that took place in close proximity to our homes and the psychological strain that resulted from it.

The program that we developed represents a significant step forward in the use of artificial intelligence technology to the making of dietary recommendations. We developed a system that analyzes food images and provides personalized nutritional advice that are linked with the specific dietary goals and health situations of users. This system was constructed by utilizing TensorFlow for image recognition and the ChatGPT API for customized guidance. The capability of our software to remember the health condition of its users and incorporate this information into subsequent interactions results in an increase in the degree to which each recommendation is relevant and personalized. This is an essential innovation that our program possesses. Due to the presence of this functionality, our application stands out in the highly competitive field of nutritional apps by offering a user experience that is both exceptionally customizable and user-focused.

In the field of software development, our research highlights the necessity of an adaptable and inventive approach to problem-solving practices. We persisted in providing a working system that accomplishes its primary goals, despite the fact that we encountered a number of technical barriers, such as difficulty in integrating APIs and TensorFlow, which included issues with accuracy. Significant learning opportunities were presented to us as a result of the challenges we encountered, which included oversights in terms of security and limitations on resources. These challenges highlighted areas that may be improved and optimized in the future.

The development process presented evidence of the benefits that may be gained from including a variety of working styles within a group. We were able to successfully leverage our individual strengths and offset each other's limitations by striking a balance between working independently and working together. This resulted in an increase in both our overall productivity and our capacity to solve problems if we were successful.

We were able to successfully complete our core goals in spite of the fact that we faced a number of significant hurdles, such as TensorFlow making incorrect identifications, API key issues, and the absence of a server that was operational. The ability to efficiently analyze meal images, the utilization of artificial intelligence to provide individualized assistance, and the retention of user health information for the purpose of providing continuous personalized recommendations are all features of our system. In extremely challenging conditions, this achievement displays our ability to adjust to new situations and overcome challenges, whether they be technical or personal in nature.

Taking into consideration the future, there are obvious avenues that can be pursued in order to continue improving and optimizing our application. The enhancement of the precision of the image recognition system, the implementation of full functionality on a live server, and the completion of the nutritionist page are all important focal points for future initiatives. In addition, if resources are available, enhancing the app's functionalities and improving the user experience could significantly boost the app's market viability and user engagement.

In spite of the significant obstacles that were encountered, this project displays endurance, innovation, and collaboration. We were able to construct a fully functional and one-of-a-kind application while simultaneously gaining valuable insights into the challenges associated with incorporating advanced artificial intelligence technologies into practical applications. Because of this experience, our problem-solving skills have been strengthened, our understanding of how artificial intelligence is implemented has been enhanced, and we are now better prepared for future endeavors in software development. When we think back on this journey, we are proud of the unique functionality that we were able to deliver. The application has a tremendous potential for future extension, and it could be further refined and expanded with further resources. This endeavor has not only resulted in a high-quality product, but it has also ingrained in us a sense of resiliency and adaptability that will be beneficial to us in our future professional endeavors.