# Lebanese International University

School of Arts & Sciences

Department of Computer Science and Information Technology

***-Project: Food-Image Classification and Calorie Detect WebApp***

***-Built and integrated by: Karim Fayad***

***-Instructor: Mr. Oussama Hatoum***

***Beirut Campus   Fall 2024-2025***

# ***Table Of Content:***

# ***Chapter One: Introduction To Food Image Classification and Calorie Detect Web App***

## Problem:

Maintaining a healthy diet and understanding nutritional values is crucial for overall well-being. However, many individuals struggle to monitor their food intake due to busy lifestyles and a lack of accessible tools for calorie tracking.

A common question arises: *What am I eating, and how many calories does it contain?* Existing solutions require manual input or extensive effort to log meals, which can be tedious and discouraging for users. This challenge is even more pronounced for individuals new to calorie tracking or those without a strong nutritional background.

The process of identifying food types and calories from a meal can be overwhelming, especially with the vast variety of cuisines and food items. This lack of intuitive and efficient solutions often leaves users without a practical way to maintain their dietary goals.

## Solution: Introducing the Food Image Classification and Calorie Detect Web App

Your one-stop platform for effortless calorie tracking. Our web application uses cutting-edge machine learning to identify food items from images and retrieve their calorie information from a reliable database. Designed to simplify the process, this app caters to users of all skill levels, making healthy eating more accessible and achievable.

## Why Choose This App?

We understand the frustration of managing dietary habits and tracking calories. This app offers a streamlined experience by combining advanced technology with an intuitive interface. Simply upload a photo of your meal, and the app will classify the food and provide accurate calorie details in seconds.

## Benefits:

Effortlessly track your calorie intake.

Identify food items from images with ease.

Receive accurate and reliable calorie information.

User-friendly interface designed for everyone.

Maintain a healthier lifestyle with minimal effort.

Personalized recommendations for consistent dietary goals.

## Features:

User authentication (Login/Sign up).

Upload and analyze food images.

Database integration for calorie information.

Responsive design for desktop and mobile use.

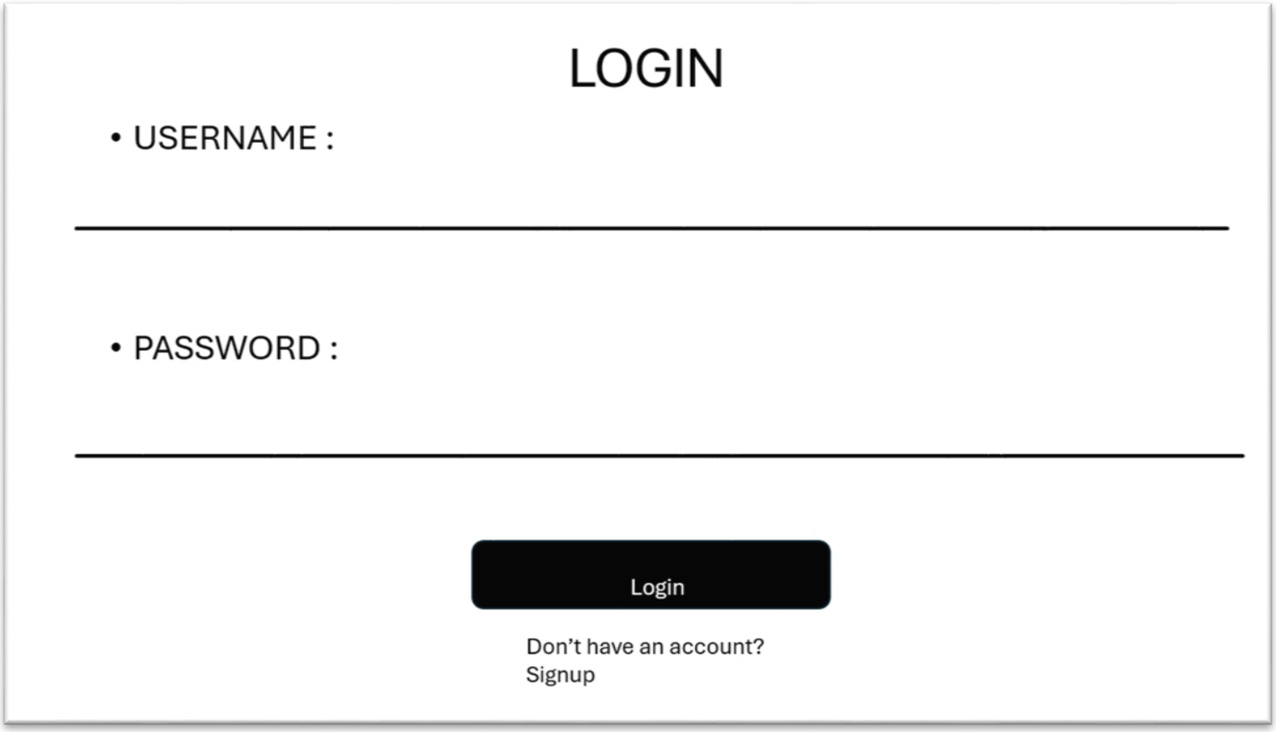
Personalized calorie tracking dashboard.

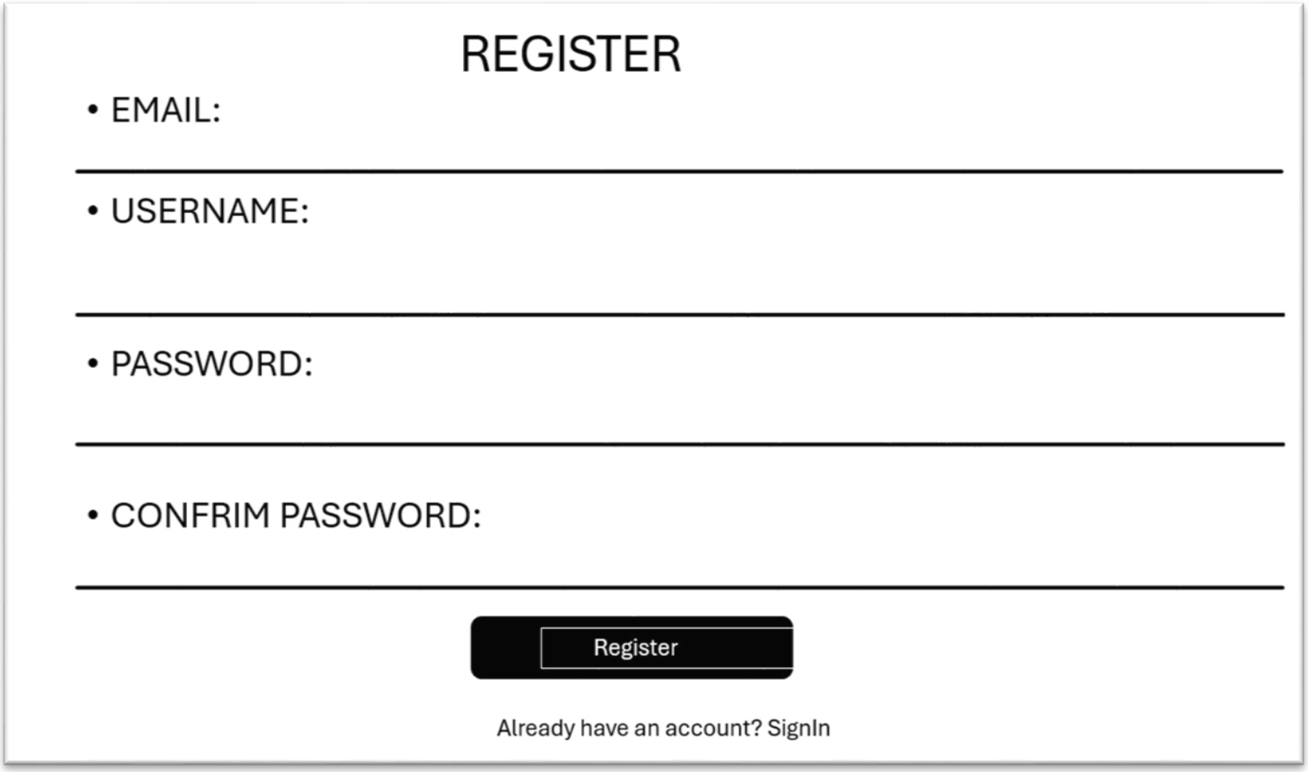
Secure and scalable free hosting.

# ***CHAPTER TWO: STORYBOARD IMPLEMENTATION***

## USER AUTHENTICATION

### LOGIN





REGISTER

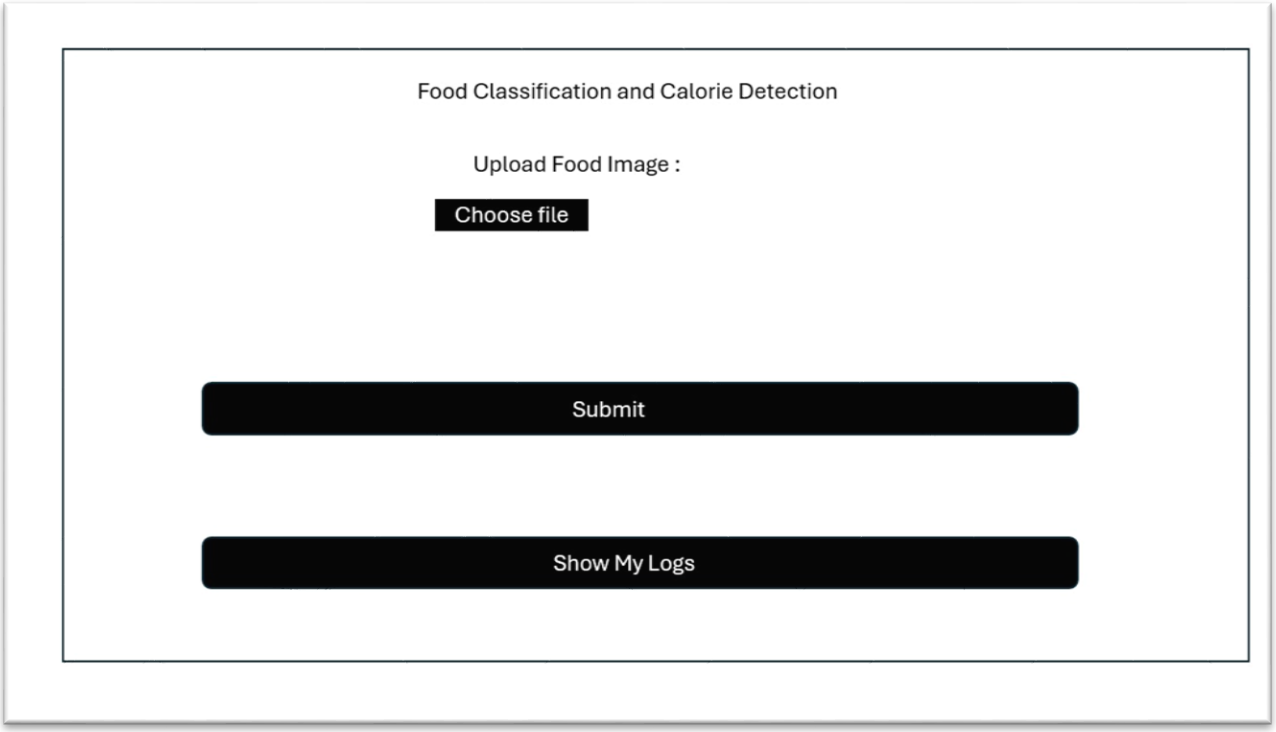
- EMAIL:
- USERNAME:
- PASSWORD:
- CONFRIM PASSWORD:

Already have an account? SignIn

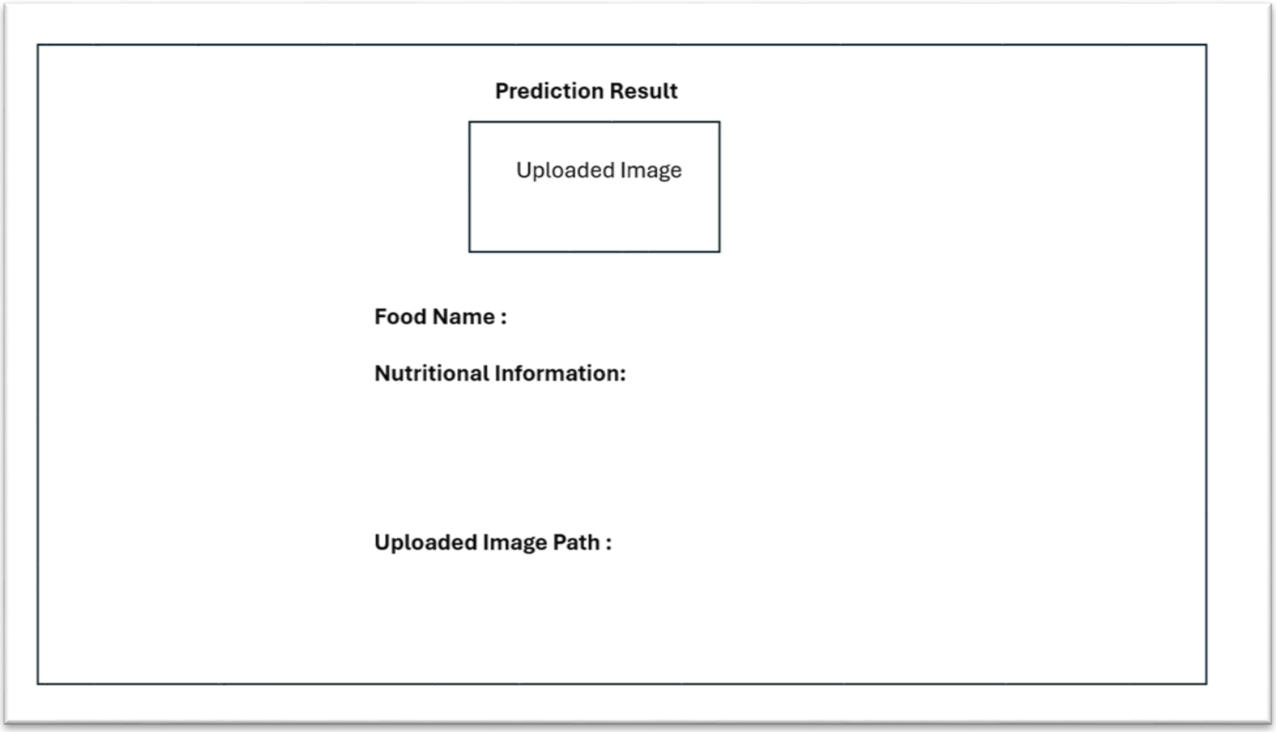
**Register**

## Food Image Classification and Calorie Detect Main-Page

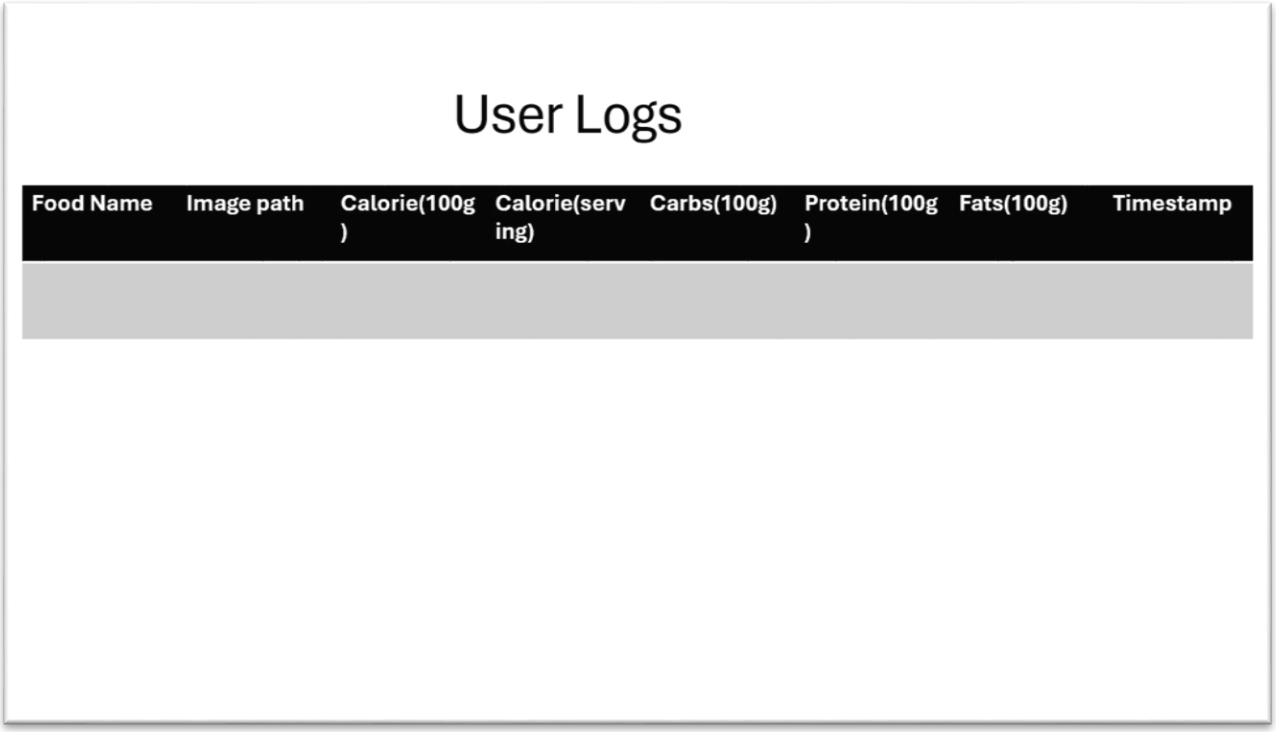
**Upload File and Submit:**



**Prediction Results and Nutritional Information:**



**User logs Table:**



User Logs

| Food Name | Image path | Calorie(100g) | Calorie(serving) | Carbs(100g) | Protein(100g) | Fats(100g) | Timestamp |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

# ***CHAPTER THREE: ENTITY RELATIONSHIP DIAGRAM***

**Entities and Attributes:**

- **Users**

  UserID (Primary Key)

  Name

  Email (Unique)

  Password

  RegistrationDate

- **FoodCalories**

  FoodID (Primary Key)

  FoodName

  CaloriesInfo

- **Logs**

  LogID (Primary Key)

  UserID (Foreign Key)

  FoodID (Foreign Key)

  ImagePath (if storing uploaded image reference)

  CaloriesConsumed

  LogDate

**Relationships:**

**Users** can have multiple **Logs** (1-to-Many).

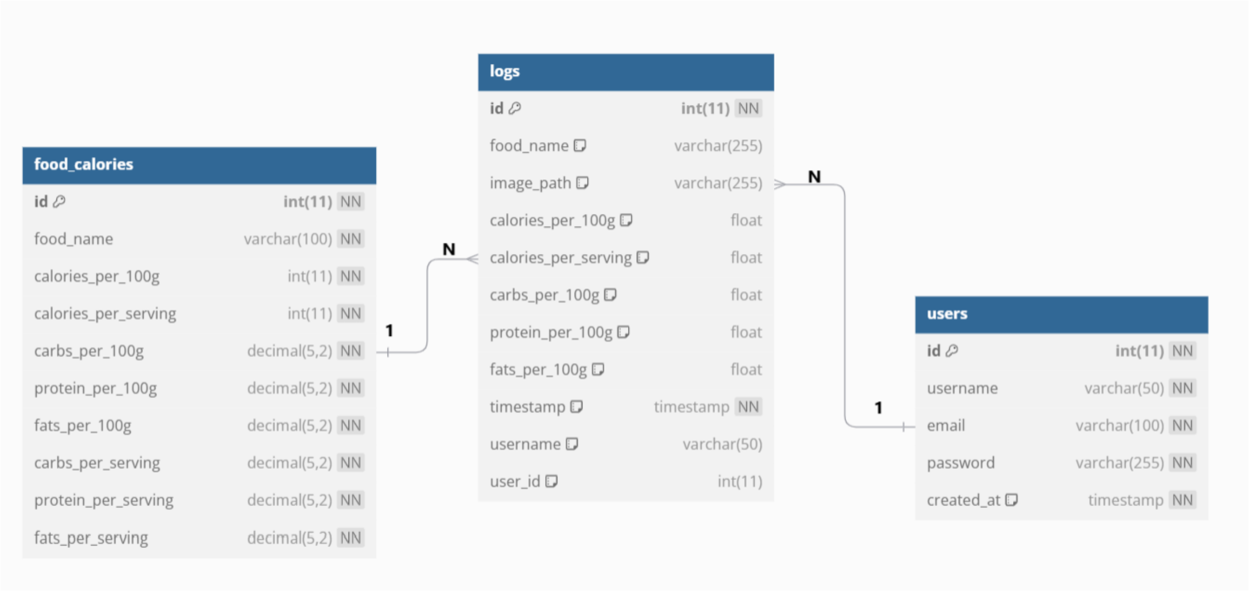
Each log entry records a food item the user interacted with.

**Logs** link to **FoodCalories** (Many-to-1).

Each log associates a food item with calorie information from the **FoodCalories** table.

### Diagram Description

- **Users**: Stores user details for authentication and personalization.
- **FoodCalories**: Maintains a list of food items and their calorie data.
- **Logs**: Tracks user activity, linking users to the foods they upload or consume.



| food_calories | | |
|---|---|---|
| id | int(11) | NN |
| food_name | varchar(100) | NN |
| calories_per_100g | int(11) | NN |
| calories_per_serving | int(11) | NN |
| carbs_per_100g | decimal(5,2) | NN |
| protein_per_100g | decimal(5,2) | NN |
| fats_per_100g | decimal(5,2) | NN |
| carbs_per_serving | decimal(5,2) | NN |
| protein_per_serving | decimal(5,2) | NN |
| fats_per_serving | decimal(5,2) | NN |

| logs | | |
|---|---|---|
| id | int(11) | NN |
| food_name | varchar(255) | |
| image_path | varchar(255) | |
| calories_per_100g | float | |
| calories_per_serving | float | |
| carbs_per_100g | float | |
| protein_per_100g | float | |
| fats_per_100g | float | |
| timestamp | timestamp | NN |
| username | varchar(50) | |
| user_id | int(11) | |

| users | | |
|---|---|---|
| id | int(11) | NN |
| username | varchar(50) | NN |
| email | varchar(100) | NN |
| password | varchar(255) | NN |
| created_at | timestamp | NN |

# ***CHAPTER FOUR: DATA DICTIONARY & IMPLEMENTATION***

**`food_calories` Table**

| Attribute Name | Data Type | Description | Constraints |
|---|---|---|---|
| `id` | INT (11) | Unique identifier for each food item. | Primary Key, Not Null |
| `food_name` | VARCHAR (100) | Name of the food item. | Not Null |
| `calories_per_100g` | INT (11) | Calories per 100 grams of the food item. | Not Null |
| `calories_per_serving` | INT (11) | Calories per serving size of the food item. | Not Null |
| `carbs_per_100g` | DECIMAL (5,2) | Carbohydrates per 100 grams of the food item. | Not Null |
| `protein_per_100g` | DECIMAL (5,2) | Protein content per 100 grams of the food item. | Not Null |
| `fats_per_100g` | DECIMAL (5,2) | Fat content per 100 grams of the food item. | Not Null |
| `carbs_per_serving` | DECIMAL (5,2) | Carbohydrates per serving of the food item. | Not Null |
| `protein_per_serving` | DECIMAL (5,2) | Protein per serving of the food item. | Not Null |
| `fats_per_serving` | DECIMAL (5,2) | Fat content per serving of the food item. | Not Null |

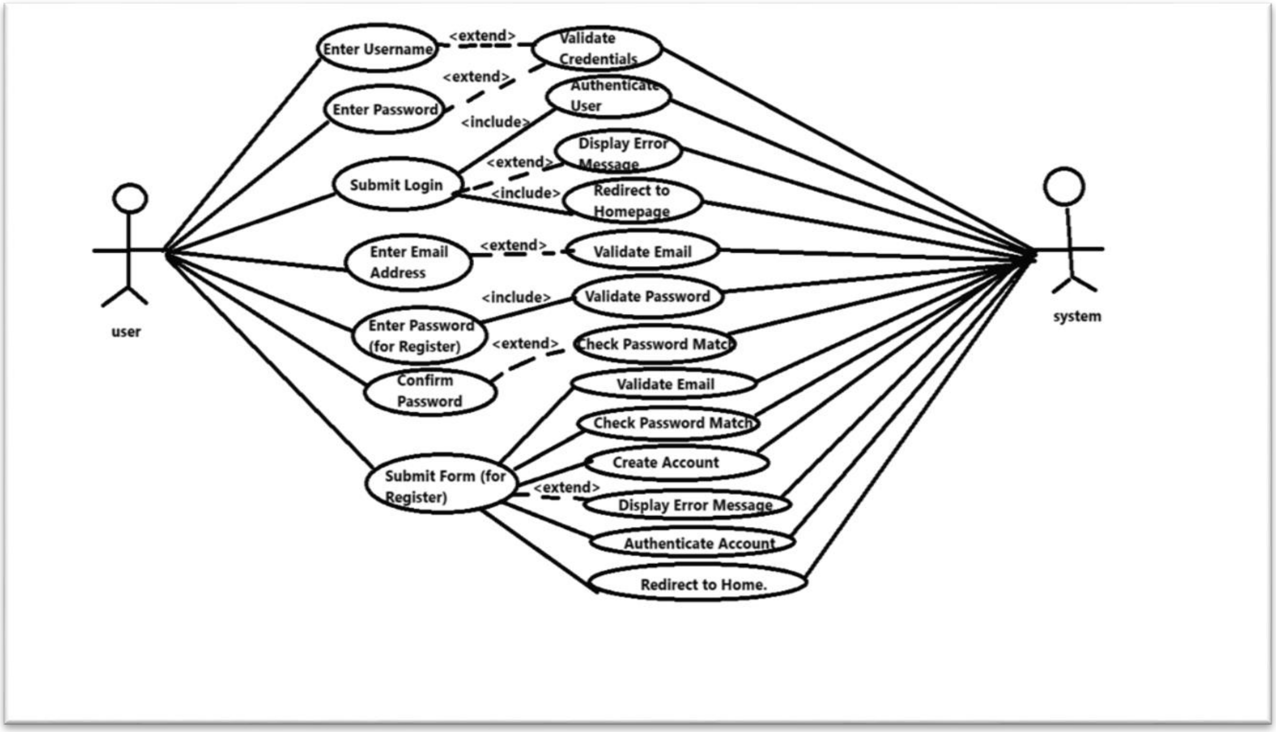
### logs Table

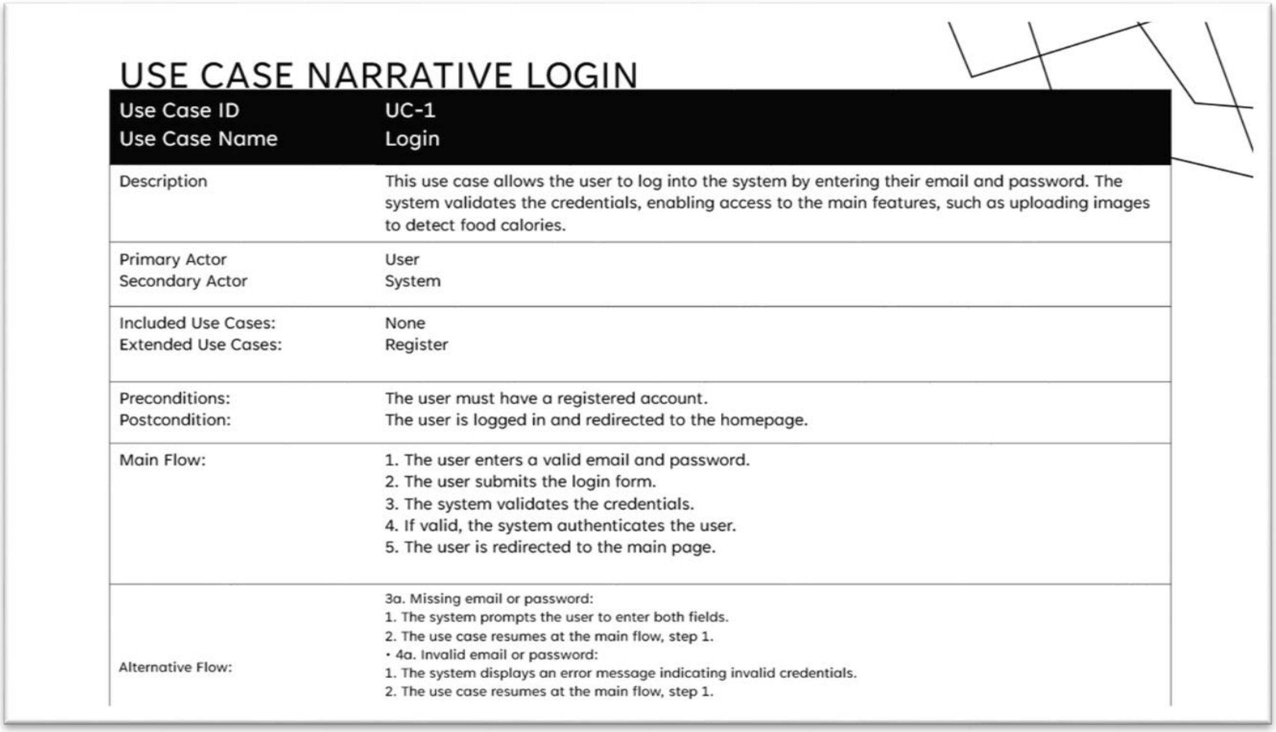
| Attribute Name | Data Type | Description | Constraints |
|---|---|---|---|
| id | INT (11) | Unique identifier for each log entry. | Primary Key, Not Null |
| food_name | VARCHAR (255) | Name of the food logged by the user. | Null Allowed |
| image_path | VARCHAR (255) | Path to the uploaded food image (optional). | Null Allowed |
| calories_per_100g | FLOAT | Calories per 100 grams of the food item logged. | Null Allowed |
| calories_per_serving | FLOAT | Calories per serving of the food item logged. | Null Allowed |
| carbs_per_100g | FLOAT | Carbohydrates per 100 grams of the food item logged. | Null Allowed |
| protein_per_100g | FLOAT | Protein per 100 grams of the food item logged. | Null Allowed |
| fats_per_100g | FLOAT | Fat per 100 grams of the food item logged. | Null Allowed |
| timestamp | TIMESTAMP | Timestamp of the log entry (automatically assigned). | Not Null, Default: CURRENT_TIMESTAMP |
| username | VARCHAR (50) | Username of the user who logged in the food. | Null Allowed |
| user_id | INT (11) | User ID of the user who logged in the food item. | Foreign Key (Users) |

### users Table

| Attribute Name | Data Type | Description | Constraints |
|---|---|---|---|
| id | INT (11) | Unique identifier for each user. | Primary Key, Not Null |
| username | VARCHAR (50) | Username for the user. | Not Null |
| email | VARCHAR (100) | Email address for the user (unique). | Not Null |
| password | VARCHAR (255) | Hashed password for user authentication. | Not Null |
| created at | TIMESTAMP | Timestamp of when the user account was created. | Not Null, Default: CURRENT_TIMESTAMP |

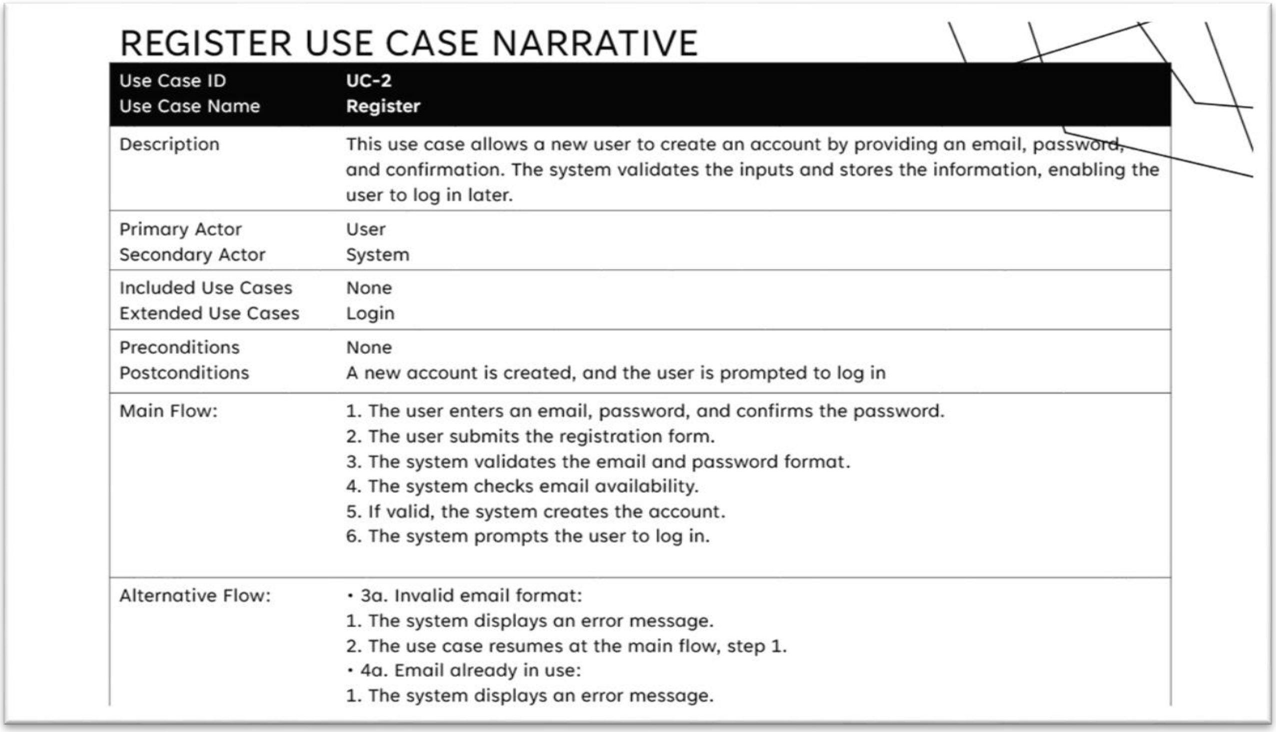
# ***CHAPTER FIVE: USE CASE DIAGRAM***





## USE CASE NARRATIVE LOGIN

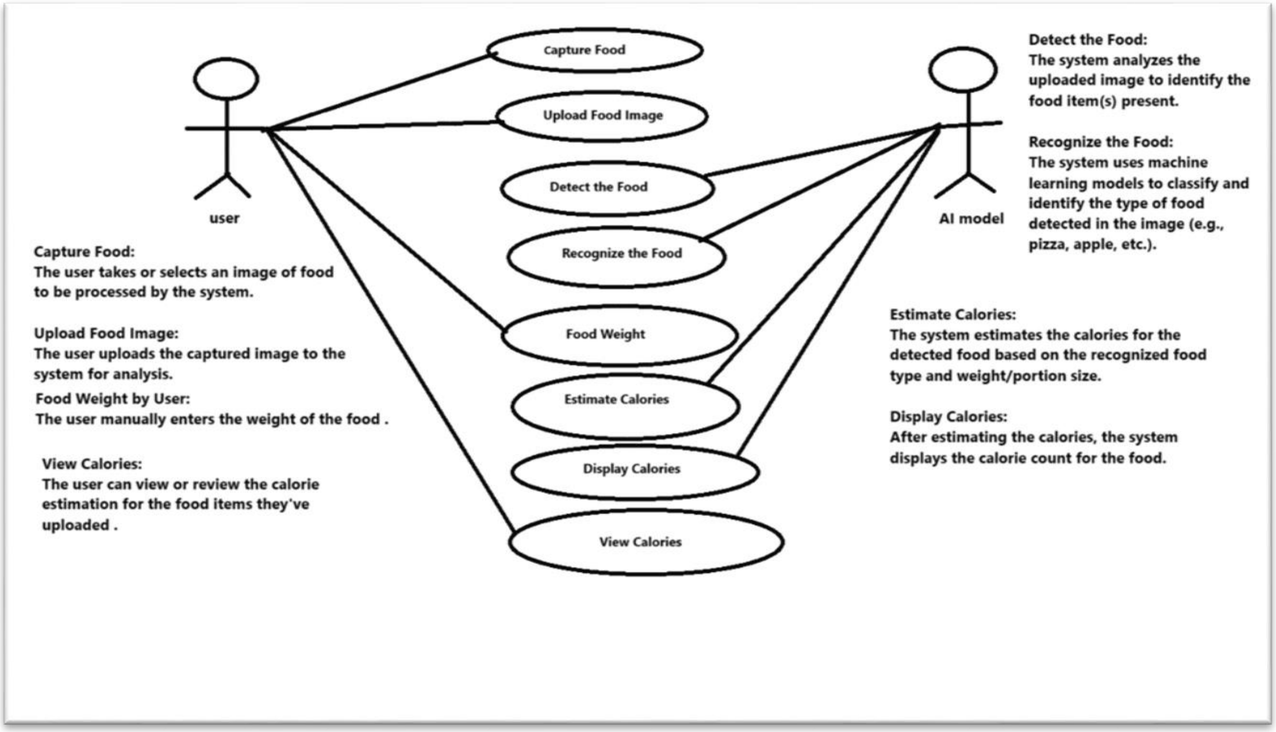
| | |
|---|---|
| Use Case ID<br>Use Case Name | UC-1<br>Login |
| Description | This use case allows the user to log into the system by entering their email and password. The system validates the credentials, enabling access to the main features, such as uploading images to detect food calories. |
| Primary Actor<br>Secondary Actor | User<br>System |
| Included Use Cases:<br>Extended Use Cases: | None<br>Register |
| Preconditions:<br>Postcondition: | The user must have a registered account.<br>The user is logged in and redirected to the homepage. |
| Main Flow: | 1. The user enters a valid email and password.<br>2. The user submits the login form.<br>3. The system validates the credentials.<br>4. If valid, the system authenticates the user.<br>5. The user is redirected to the main page. |
| Alternative Flow: | 3a. Missing email or password:<br>1. The system prompts the user to enter both fields.<br>2. The use case resumes at the main flow, step 1.<br>• 4a. Invalid email or password:<br>1. The system displays an error message indicating invalid credentials.<br>2. The use case resumes at the main flow, step 1. |



## REGISTER USE CASE NARRATIVE

| Use Case ID<br>Use Case Name | **UC-2**<br>**Register** |
|---|---|
| Description | This use case allows a new user to create an account by providing an email, password, and confirmation. The system validates the inputs and stores the information, enabling the user to log in later. |
| Primary Actor<br>Secondary Actor | User<br>System |
| Included Use Cases<br>Extended Use Cases | None<br>Login |
| Preconditions<br>Postconditions | None<br>A new account is created, and the user is prompted to log in |
| Main Flow: | 1. The user enters an email, password, and confirms the password.<br>2. The user submits the registration form.<br>3. The system validates the email and password format.<br>4. The system checks email availability.<br>5. If valid, the system creates the account.<br>6. The system prompts the user to log in. |
| Alternative Flow: | • 3a. Invalid email format:<br>1. The system displays an error message.<br>2. The use case resumes at the main flow, step 1.<br>• 4a. Email already in use:<br>1. The system displays an error message. |

## CALORIE DETECTION BY AI USECASE DIAGRAM



**Capture Food:**
The user takes or selects an image of food to be processed by the system.

**Upload Food Image:**
The user uploads the captured image to the system for analysis.

**Food Weight by User:**
The user manually enters the weight of the food .

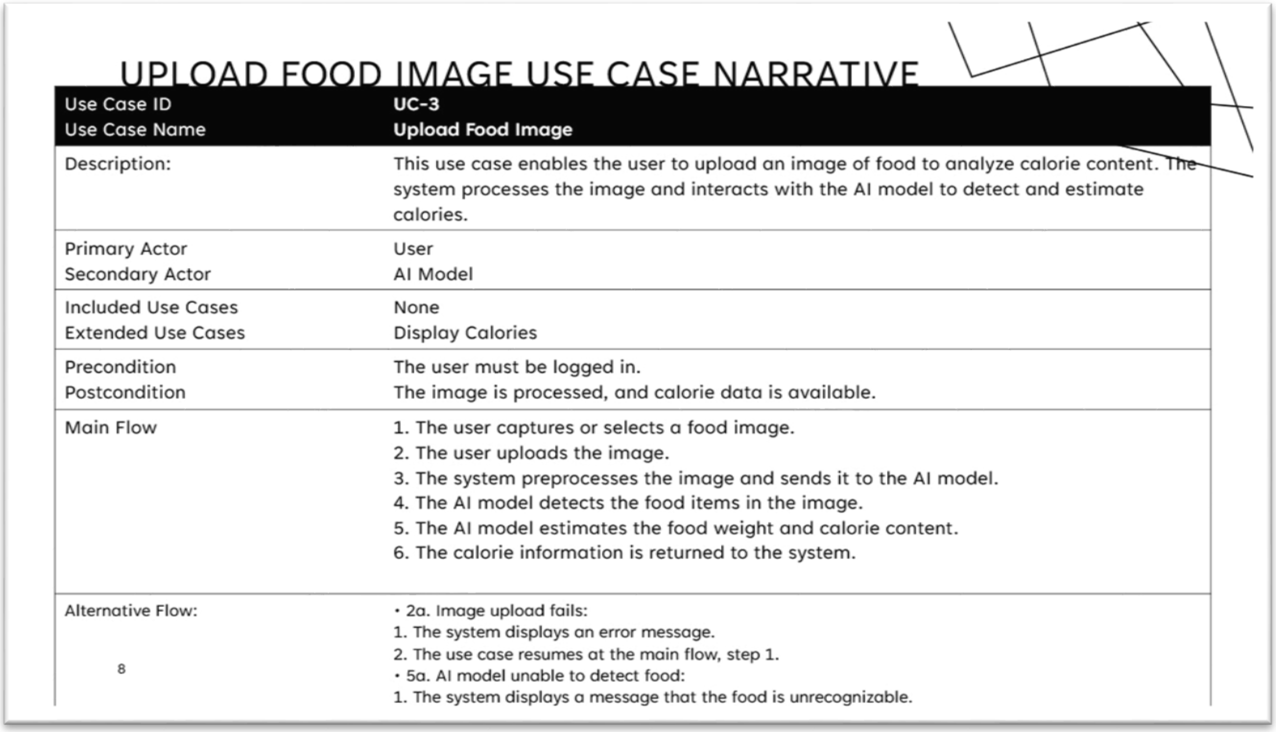
**View Calories:**
The user can view or review the calorie estimation for the food items they've uploaded .

**Detect the Food:**
The system analyzes the uploaded image to identify the food item(s) present.

**Recognize the Food:**
The system uses machine learning models to classify and identify the type of food detected in the image (e.g., pizza, apple, etc.).

**Estimate Calories:**
The system estimates the calories for the detected food based on the recognized food type and weight/portion size.

**Display Calories:**
After estimating the calories, the system displays the calorie count for the food.



# UPLOAD FOOD IMAGE USE CASE NARRATIVE

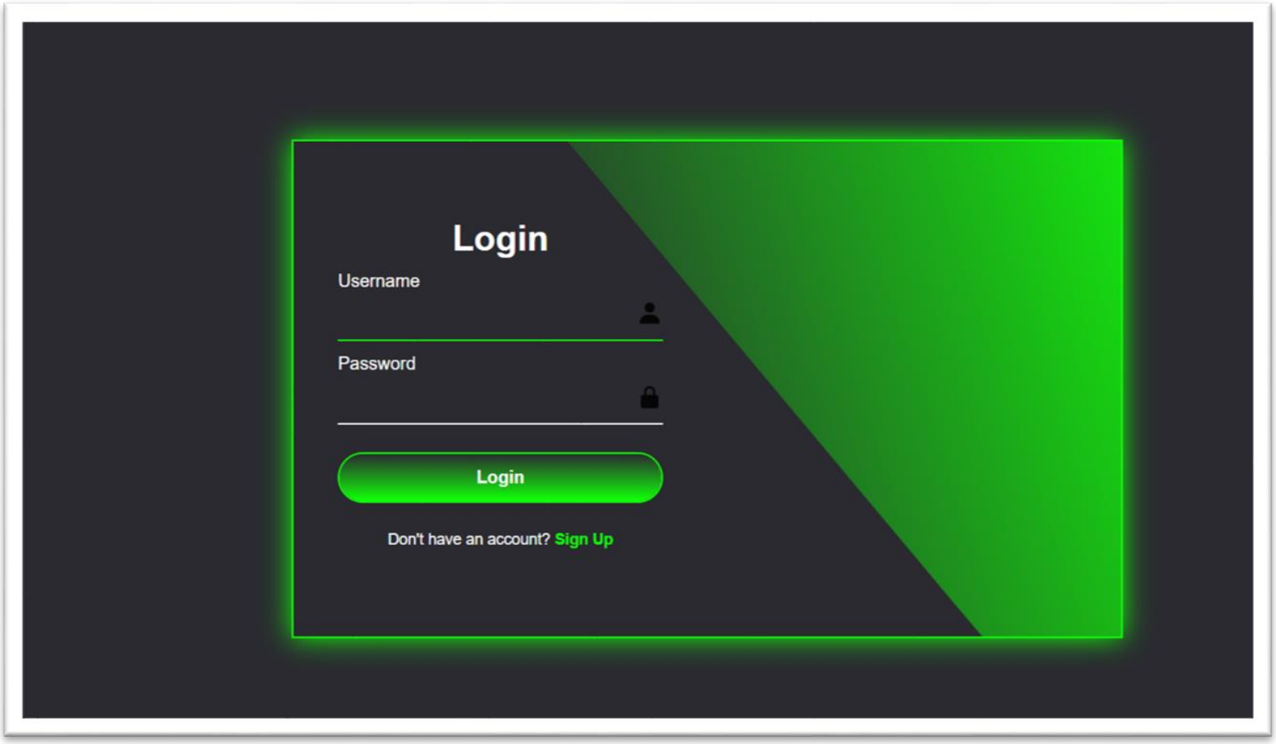
| Use Case ID<br>Use Case Name | UC-3<br>**Upload Food Image** |
|---|---|
| Description: | This use case enables the user to upload an image of food to analyze calorie content. The system processes the image and interacts with the AI model to detect and estimate calories. |
| Primary Actor<br>Secondary Actor | User<br>AI Model |
| Included Use Cases<br>Extended Use Cases | None<br>Display Calories |
| Precondition<br>Postcondition | The user must be logged in.<br>The image is processed, and calorie data is available. |
| Main Flow | 1. The user captures or selects a food image.<br>2. The user uploads the image.<br>3. The system preprocesses the image and sends it to the AI model.<br>4. The AI model detects the food items in the image.<br>5. The AI model estimates the food weight and calorie content.<br>6. The calorie information is returned to the system. |
| Alternative Flow: | • 2a. Image upload fails:<br>1. The system displays an error message.<br>2. The use case resumes at the main flow, step 1.<br>• 5a. AI model unable to detect food:<br>1. The system displays a message that the food is unrecognizable. |

# ***Chapter SIX: Documentation***

1. User Authentication (Login and Registration)

***Login Page***

- When you visit the login page, you will be prompted to **enter your username**.
- The system will check if the username exists in the **user's** database.
- If the **username exists**, you will be redirected to the **main page**.

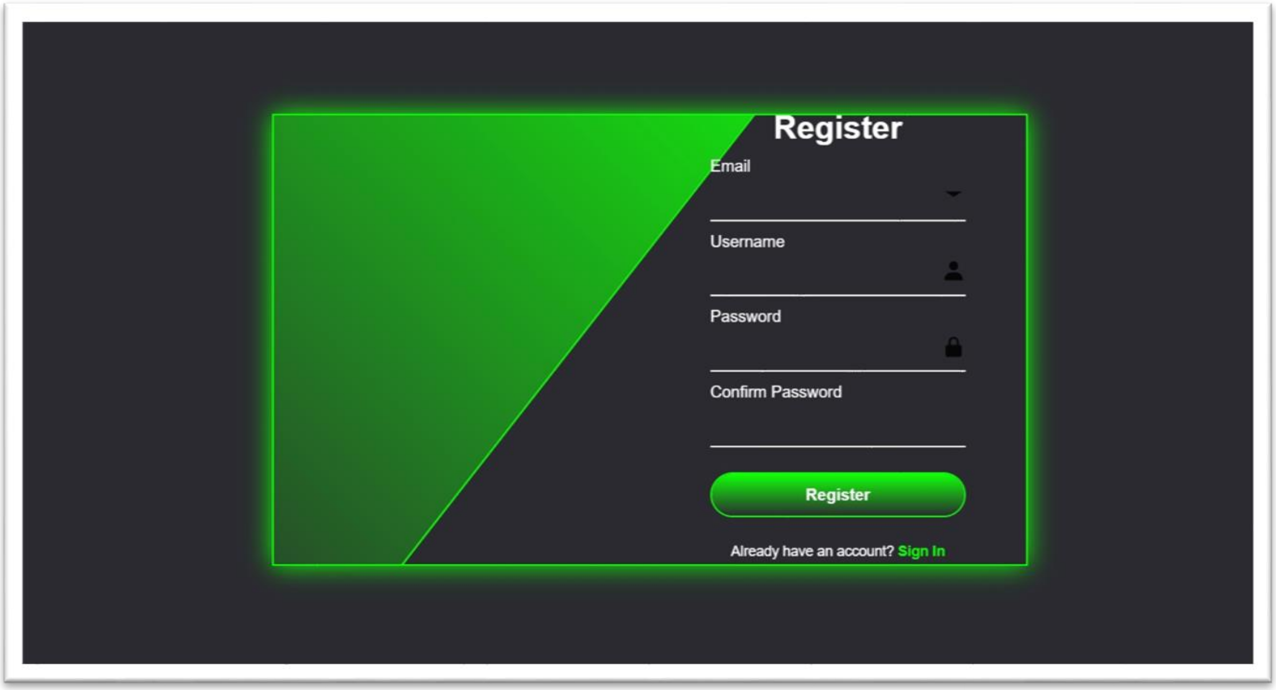


Don't have an account? Sign Up

***Registration Page***

- If you don’t have an account, the system will prompt you to **fill in your details**:
    - **Email**: Enter your email address.
    - **Username**: Choose a unique username for your account.
    - **Password**: Set your password. It should be at least 8 characters long.
    - **Confirm Password**: Re-enter your password to confirm it matches.

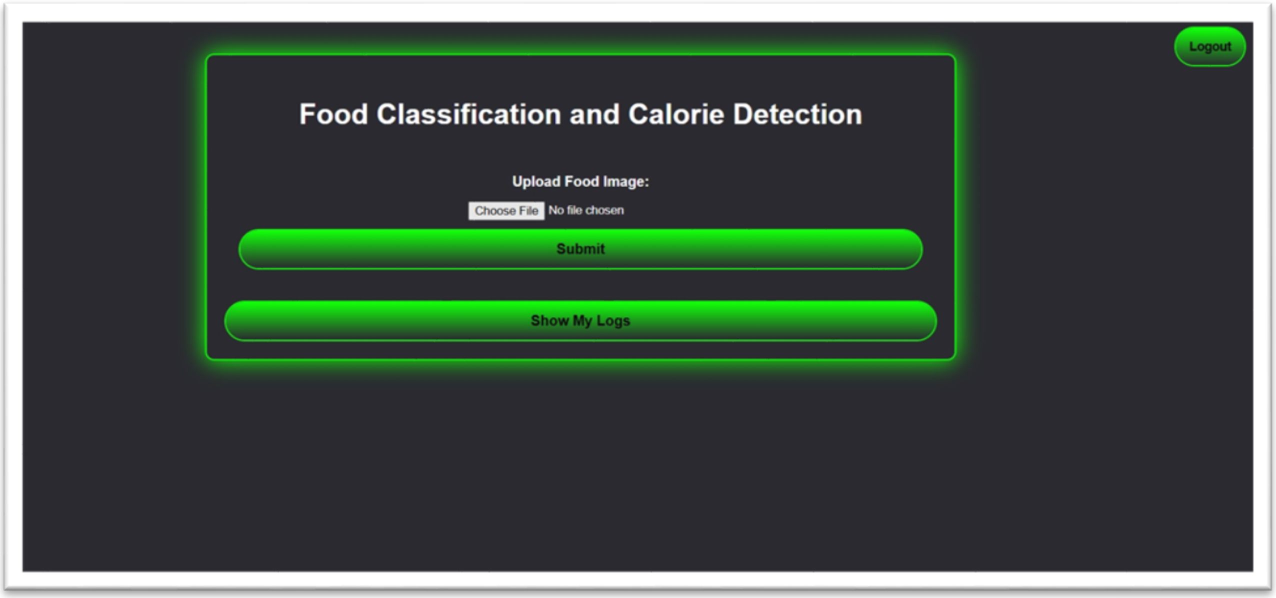
- After successfully completing the registration form, **submit the details**.
- Upon successful registration, you will be redirected to the **main page**.



## 2. Main Page Features

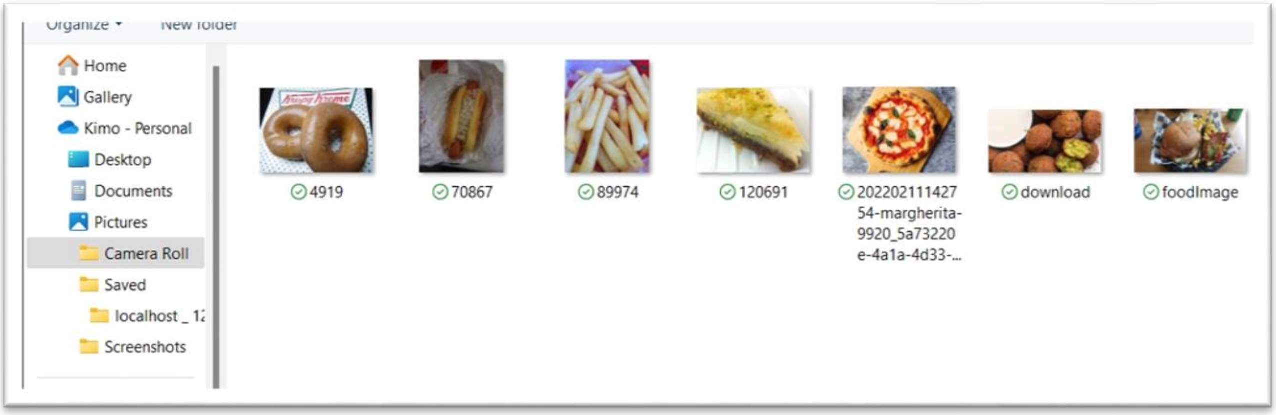
*Upload Food Image for Prediction*

- On the **main page**, you will see an **Upload File** button.
- **Click on this button** to upload a food image from your device.
- Once the image is uploaded, the system will automatically **submit the image** to process it and provide **calorie predictions**.



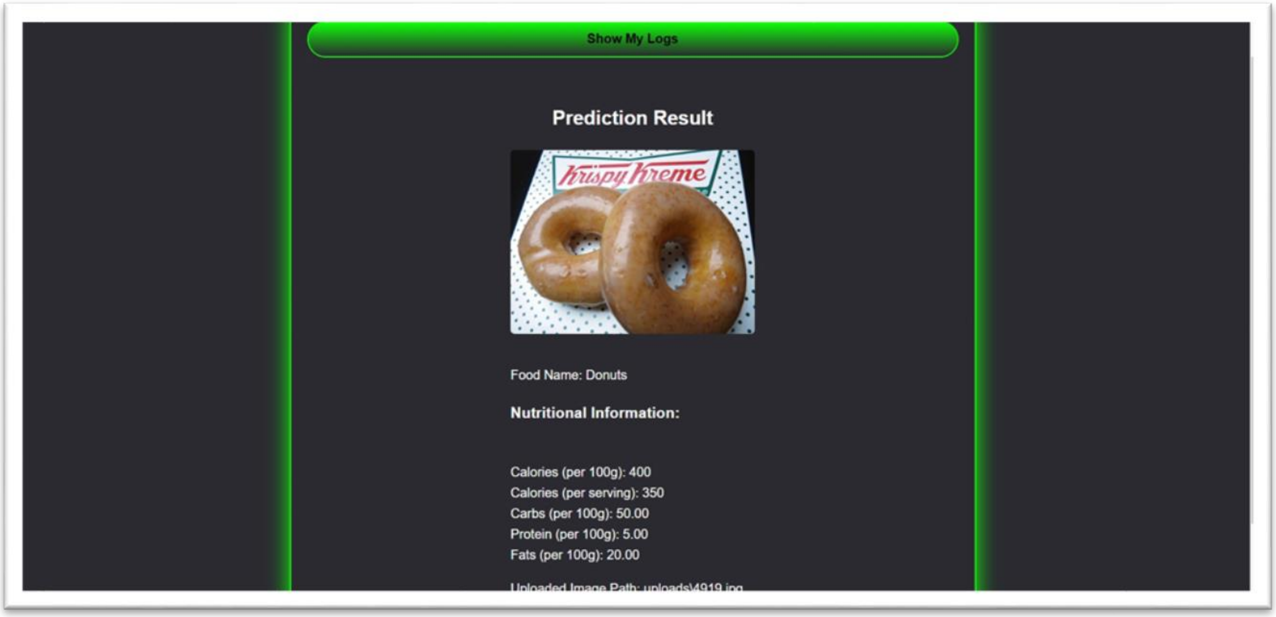
Food Classification and Calorie Detection

Upload Food Image:



***Food Prediction Results***

- After submitting the uploaded food image, the system will classify the food type and **display the prediction results**.
- These results will include:
  - **Food Name**
  - **Calories per Serving**
  - **Carbs, Protein, and Fats Content**
- The results will be shown to the user in a readable format on the main page.



Prediction Result

Food Name: Donuts

Nutritional Information:

Calories (per 100g): 400
Calories (per serving): 350
Carbs (per 100g): 50.00
Protein (per 100g): 5.00
Fats (per 100g): 20.00

Uploaded Image Path: uploads\4919.jpg

## 3. View Logs

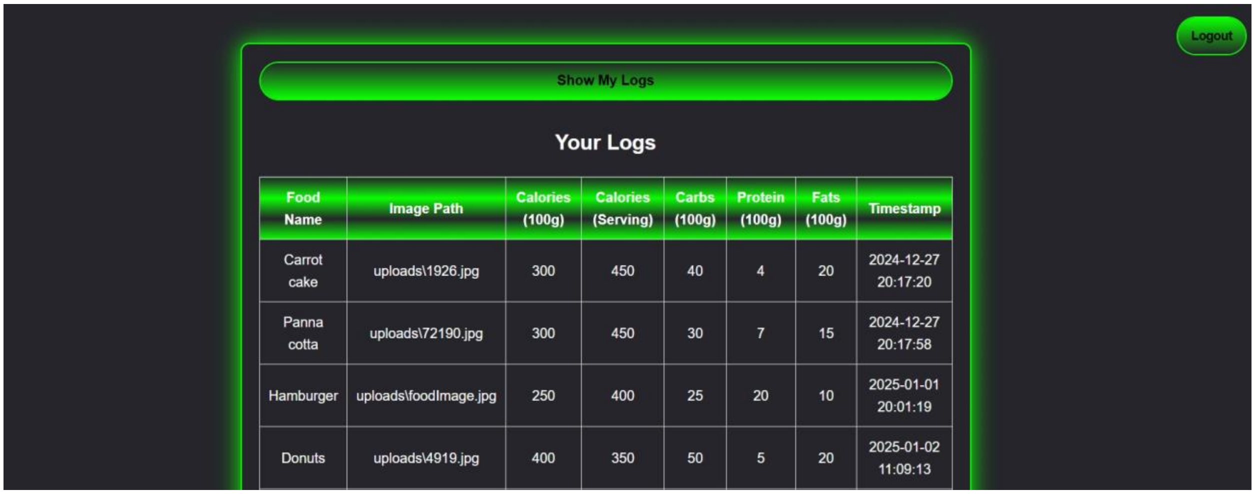
***Show Logs Page***

- To view the logs of your past food uploads and their predictions, click on the **Show Logs** link/button on the main page.
- This will redirect you to a page where you can **view all your previous logs**.

***Log Details:***

- **Food Name**
- **Image** (if uploaded)
- **Calories per Serving**
- **Carbs, Protein, and Fats Content**
- **Timestamp** of when the food was logged.
- **Username** of the user who logged into the food item.

- You will be able to **see your entire history** of logged food items along with their nutritional information.



Your Logs

| Food Name | Image Path | Calories (100g) | Calories (Serving) | Carbs (100g) | Protein (100g) | Fats (100g) | Timestamp |
|---|---|---|---|---|---|---|---|
| Carrot cake | uploads\1926.jpg | 300 | 450 | 40 | 4 | 20 | 2024-12-27 20:17:20 |
| Panna cotta | uploads\72190.jpg | 300 | 450 | 30 | 7 | 15 | 2024-12-27 20:17:58 |
| Hamburger | uploads\foodImage.jpg | 250 | 400 | 25 | 20 | 10 | 2025-01-01 20:01:19 |
| Donuts | uploads\4919.jpg | 400 | 350 | 50 | 5 | 20 | 2025-01-02 11:09:13 |

# ***CHAPTER SEVEN: FUTURE IMPELEMENTATIONS***

## Future Implementation Plan

As we continue to improve the **food tracking web app**, there are two main areas of focus for future work:

**1. Improving the Accuracy of the Machine Learning Model**

Enhancing the machine learning model to improve the precision of food classification for a wider variety of food items.

**2. Web App Hosting**

To make the app accessible to users and showcase its features, we will host the web app on a reliable platform.

**3. Feature Expansion**:

Adding more user-centric features like meal planning, nutritional goal setting, and integration with fitness tracking apps.

**4. Mobile App Development**:

Expanding the platform to include a mobile app for greater accessibility and convenience.

# ***CHAPTER EIGHT: CONCLUSION***

**Key Achievements**

1. **Machine Learning Integration**:
    a. Implemented an image classification model capable of recognizing a wide variety of food items.
2. **User Authentication**:
    a. Developed a robust authentication system for user registration, login, and secure access to personalized features such as log tracking.
3. **Database Management**:

    a. Designed and implemented a relational database to store user information, food nutritional data, and logs efficiently.

4. **Seamless User Experience**:
    a. Created an intuitive interface for users to upload food images, view results, and track their food logs with minimal effort.
5. **Responsive Design**:
    a. Ensured the application is accessible across devices, providing a consistent experience for all users.

***Acknowledgment***

A huge thanks to **Mr. Oussama Hatoum** for his invaluable guidance and support throughout the course. His mentorship has played a significant role in shaping this project and enhancing its overall quality. His expertise, insights, and encouragement have been instrumental in overcoming challenges and ensuring the successful completion of this project.

***Final Thoughts***

This project has been an insightful journey into the intersection of machine learning, web development, and user experience design. By combining these domains, the application showcases the potential of technology to simplify daily tasks and promote healthier lifestyle choices. With further improvements, this app has the potential to become a powerful tool for dietary management and nutritional education.