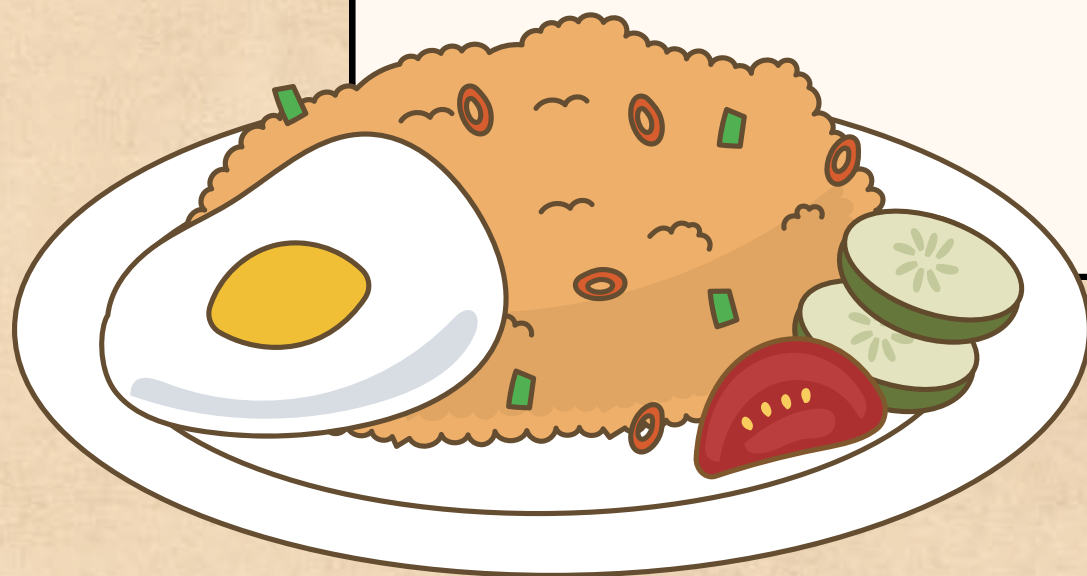
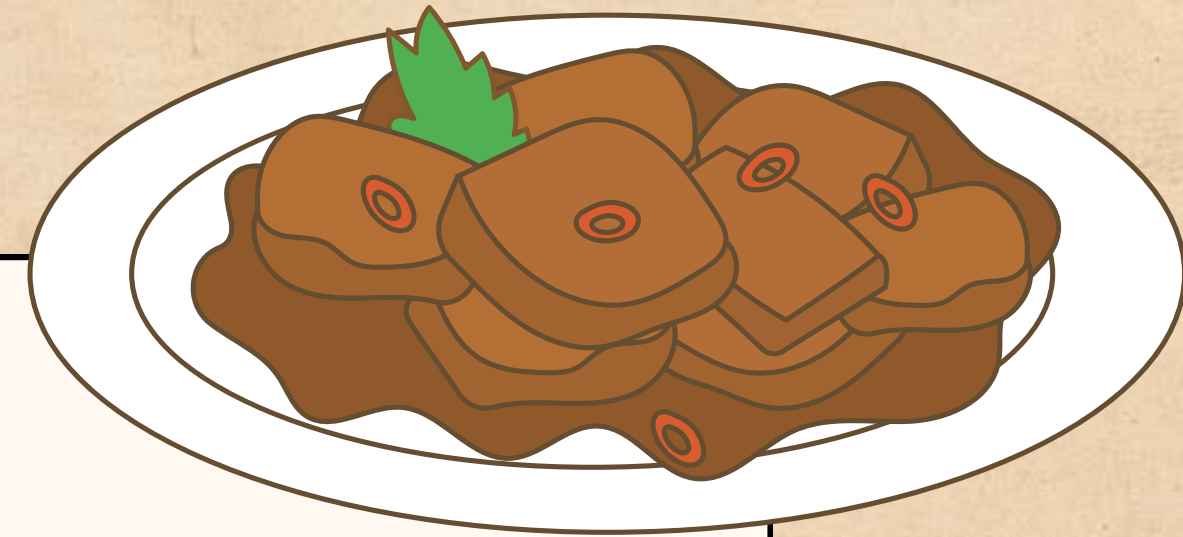


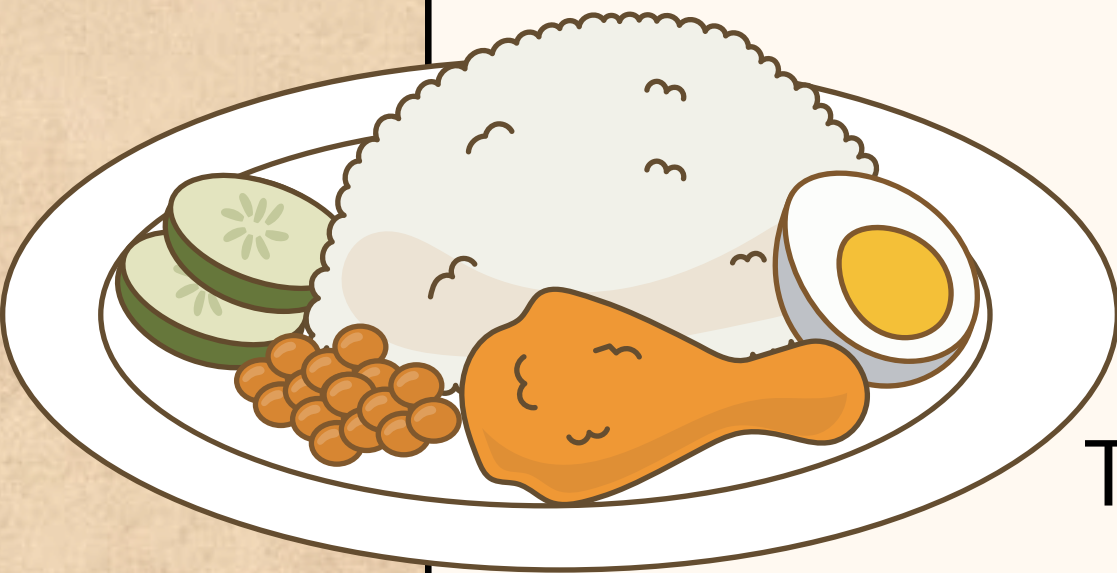


# Nutrition AI Coach

By Singh Parisdeep & Nasui Elisa

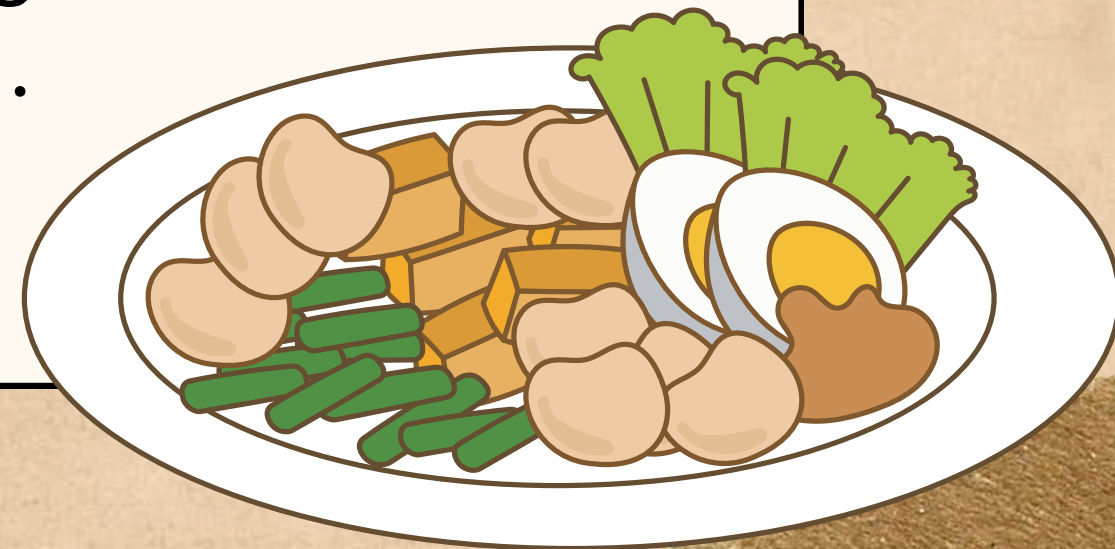






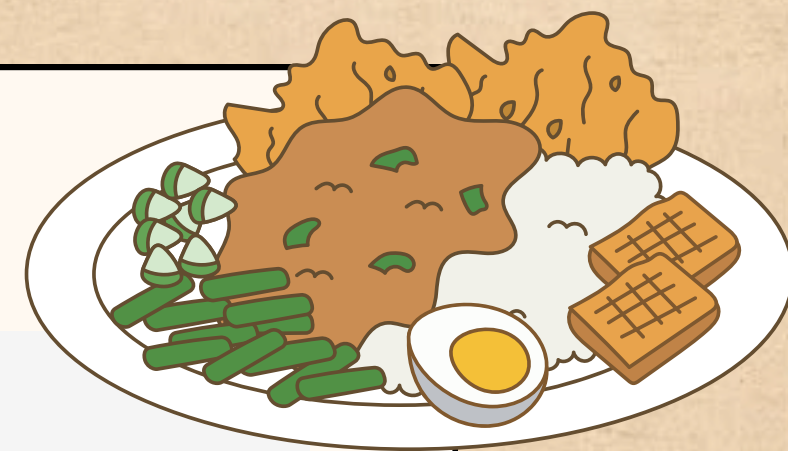
# Our Project

The Nutrition AI Coach is an app that strives to make healthy and nutritious cooking easier, as well as encourage people to try new dishes.

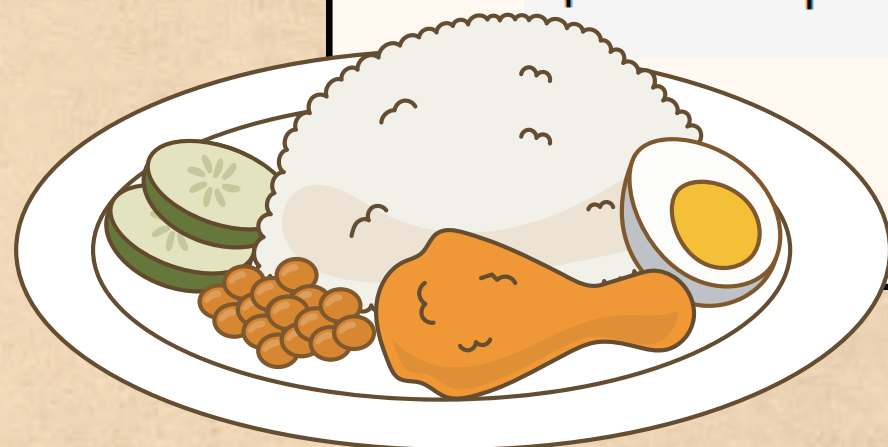




# Basic Architecture



```
app project/
├── run.py           # Main file to run the application
├── app/             # Folder containing the core files of the app
│   ├── chat.py      # Manages chat/messages
│   ├── database.py  # Manages the database
│   ├── models.py    # Defines the data models
│   └── user.py       # Manages user-related functions (login/register)
├── templates/       # Folder containing HTML files
│   └── index.html   # Front-end of the app
├── .env             # Environment variables (e.g., database, API keys)
└── requirements.txt # List of Python dependencies for the project
```





# user.py

This file handles **user registration** and **login**:

- Reads email and password sent from the frontend
- Hashes the password before storing it
- Checks if the email already exists before creating a new user
- Verifies the email + password when logging in
- Stores the user's ID in the Flask session to keep them logged in



# chat.py

This file sets up the connection to **PostgreSQL**:

- Loads the DATABASE\_URL from .env
- Creates a SQLAlchemy engine connected to PostgreSQL
- Provides SessionLocal to interact with the database
- Defines the Base class used to create SQLAlchemy models



# index.html

This file contains **the entire user interface**:

- Displays the pages: Home, Login, Register, Chat
- Includes the forms for registration and login
- Displays the chat area to talk with the AI
- Uses JavaScript to:
  - send HTTP requests to the Flask backend
  - render chat messages
  - switch screens depending on user actions (login, register, chat)



# models.py

This file defines the **database structure** using SQLAlchemy.

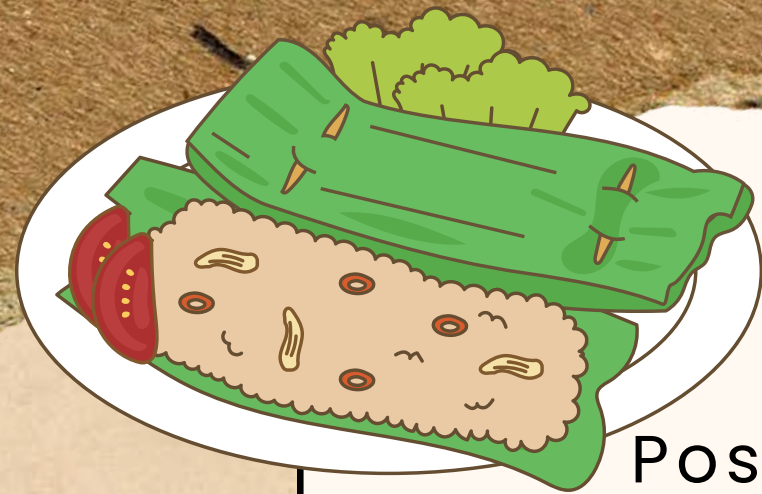
## **Class: User**

Represents a user in the database with the following fields:

- **id** → Unique ID for each user (primary key)
- **email** → The user's email (must be unique)
- **password** → The hashed password stored in the database
- **chat\_history** → A text field storing the entire chat history as JSON

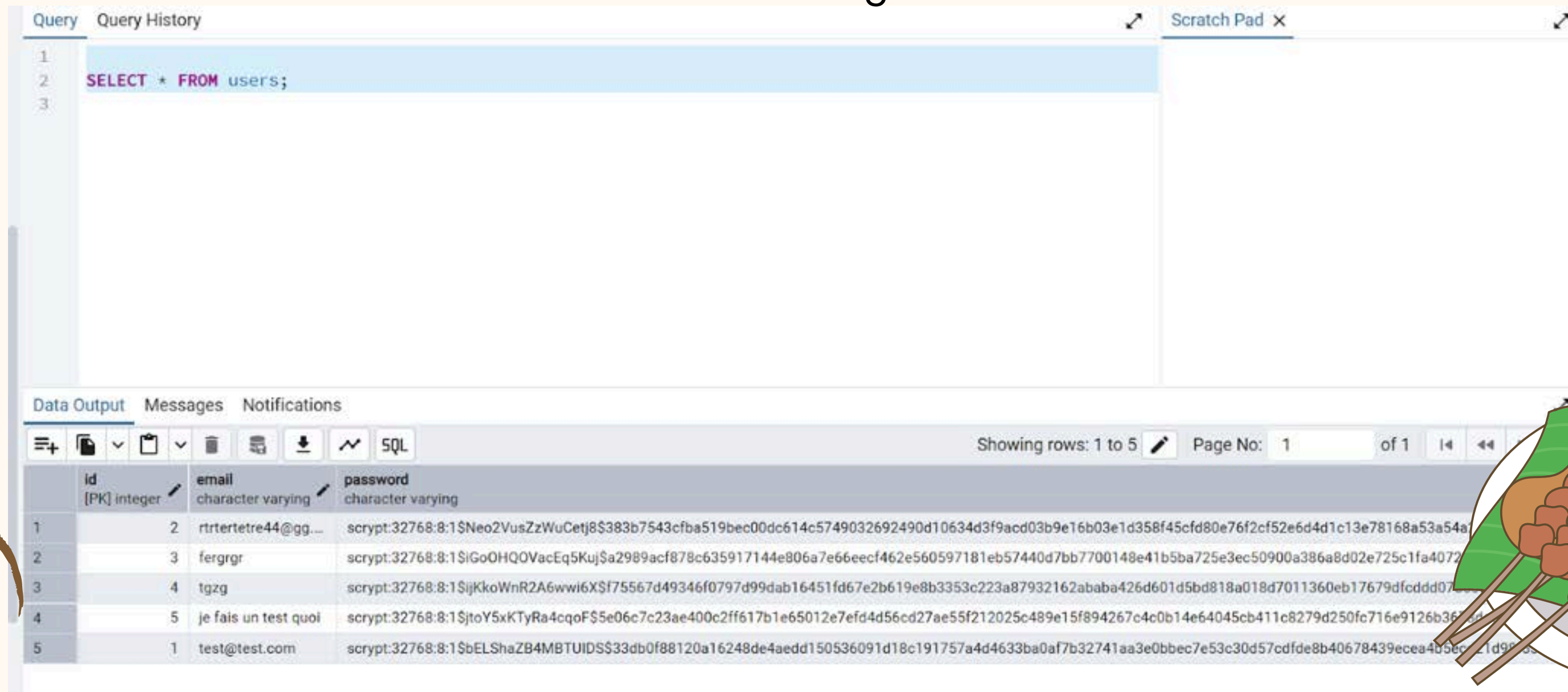
This model is used by the rest of the app to create, read, and update user data in the database.





# The Database

PostgreSQL is a free and open-source relational database management system (RDBMS) emphasizing extensibility and SQL compliance. Below, you can see how we utilized it for account management:

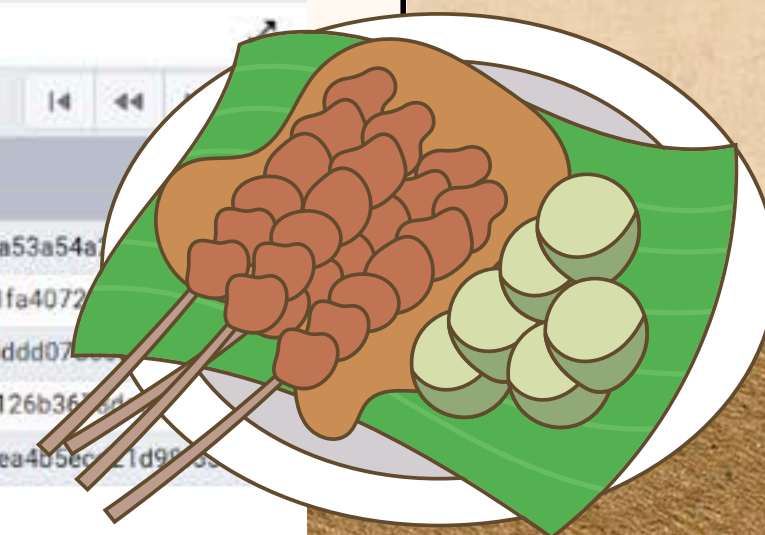


The screenshot shows a PostgreSQL query editor with a query window and a data output window. The query window contains the following SQL query:

```
SELECT * FROM users;
```

The data output window shows the results of the query, which are 5 rows of user data. The columns are: id (integer), email (character varying), and password (character varying). The data is as follows:

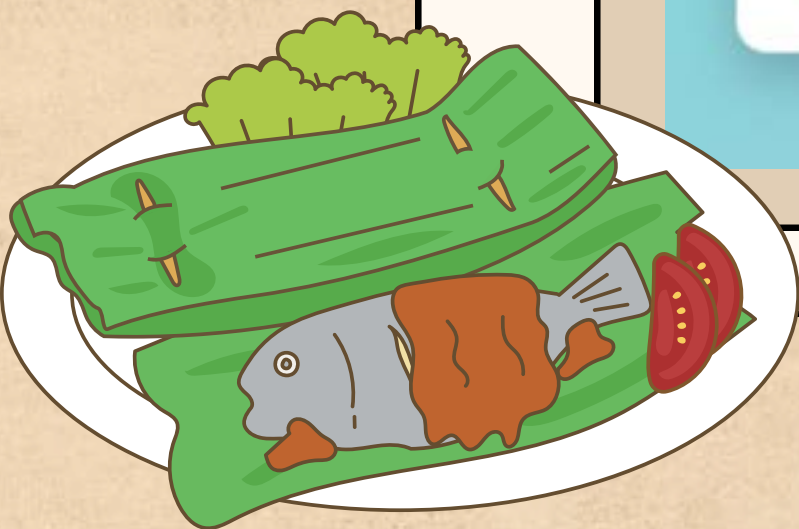
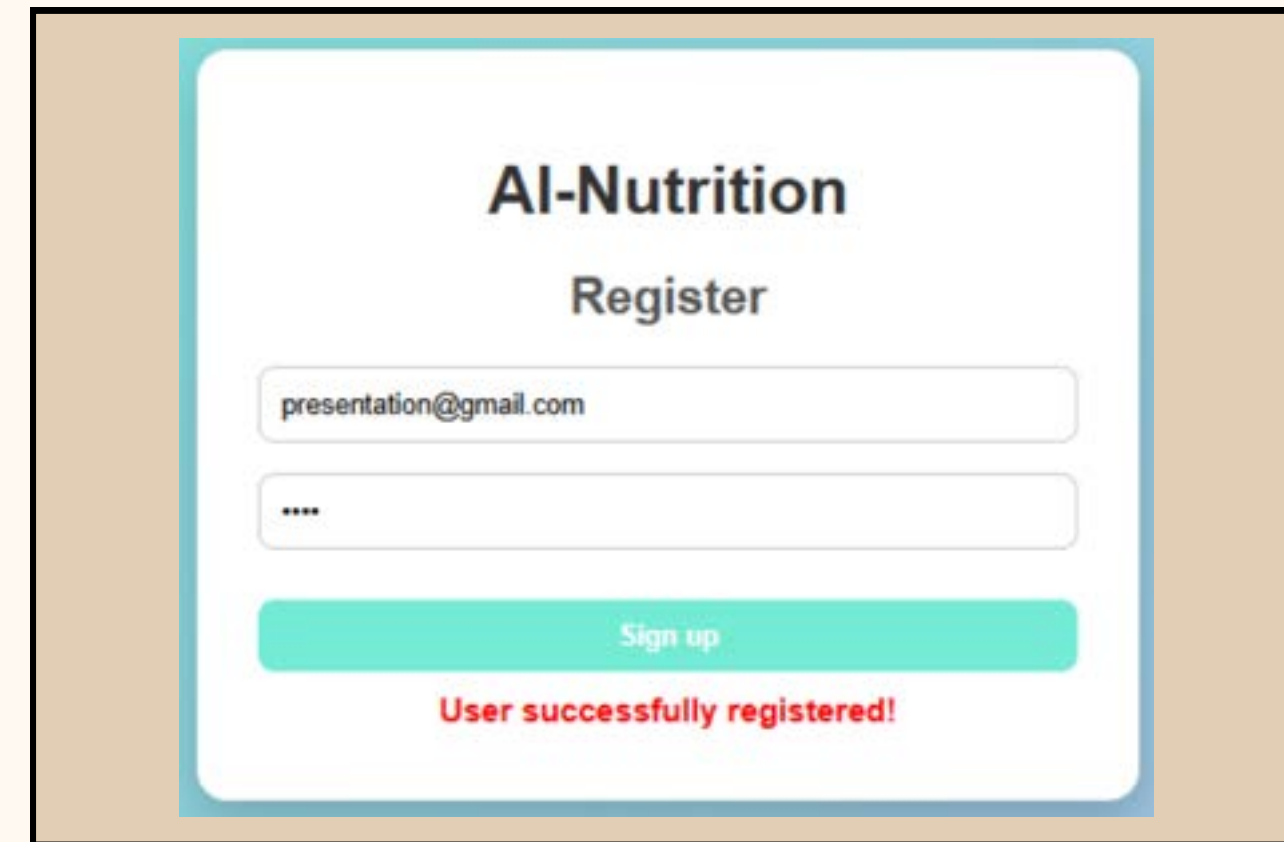
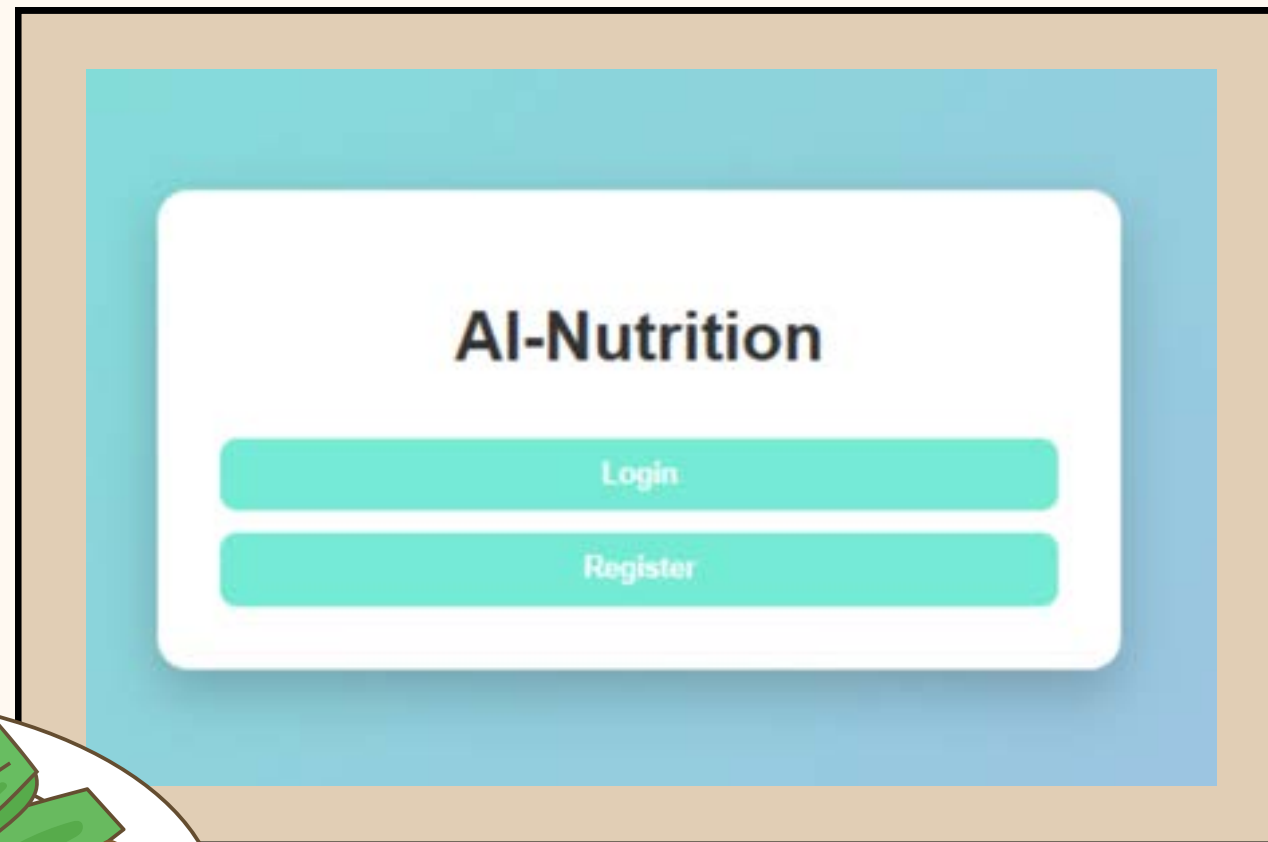
id	email	password
1	rrtrtetre44@gg...	scrypt:32768:8:1\$Neo2VusZzWuCetj8\$383b7543cfba519bec00dc614c5749032692490d10634d3f9acd03b9e16b03e1d358f45cfd80e76f2cf52e6d4d1c13e78168a53a54a
2	fergrgr	scrypt:32768:8:1\$GoOHQOVacEq5Kuj\$a2989acf878c635917144e806a7e66eecf462e560597181eb57440d7bb7700148e41b5ba725e3ec50900a386a8d02e725c1fa4072
3	tgzg	scrypt:32768:8:1\$ijKkoWnR2A6wwi6X\$F75567d49346f0797d99dab16451fd67e2b619e8b3353c223a87932162ababa426d601d5bd818a018d7011360eb17679dfcddd07
4	je fais un test quoi	scrypt:32768:8:1\$JtoY5xKTyRa4cqoF\$5e06c7c23ae400c2ff617b1e65012e7efd4d56cd27ae55f212025c489e15f894267c4c0b14e64045cb411c8279d250fc716e9126b36
5	test@test.com	scrypt:32768:8:1\$bELShaZB4MBTUIDSS\$33db0f88120a16248de4aedd150536091d18c191757a4d4633ba0af7b32741aa3e0bbec7e53c30d57cdfde8b40678439ecea405ec





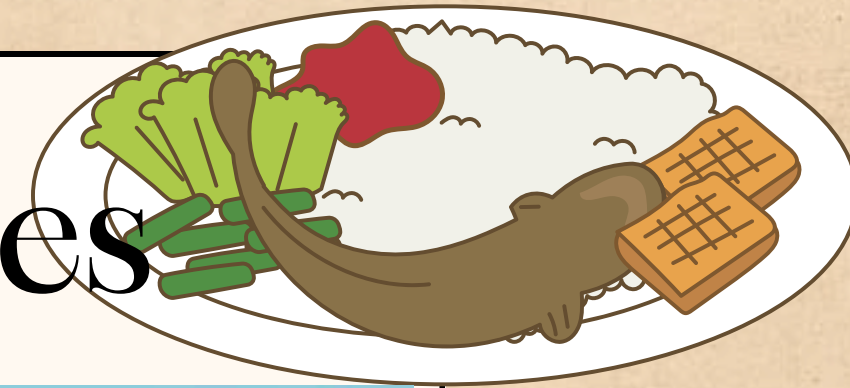
# The Main Page

The first thing a new user is going to come face-to-face with is a Login/Register screen that has been set up with appropriate error messages for different scenarios

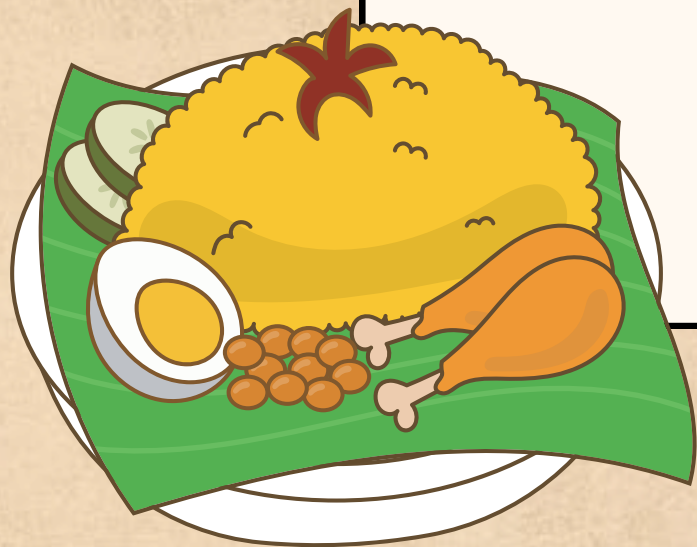




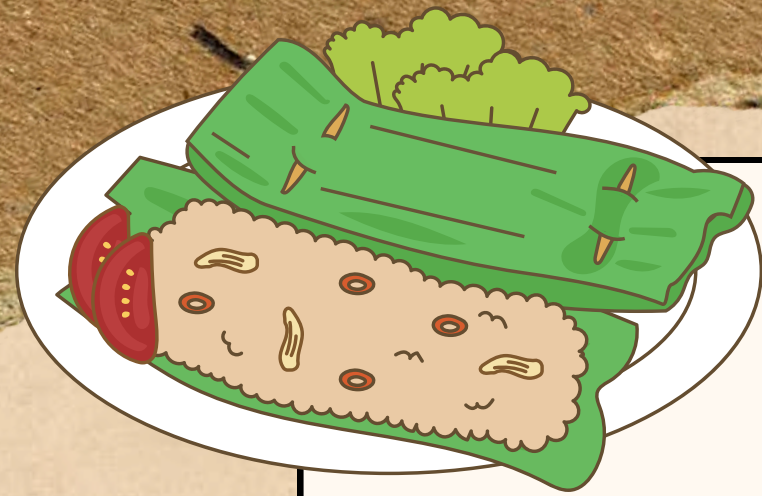
# The Login/Register Use Cases



1. **Successful Registration**
2. **Email Already in Use**
3. **Missing Email or Password in Registration**
4. **Successful Login**
5. **User Not Found**
6. **Incorrect Password**
7. **Missing Email or Password in Login**

A screenshot of a web form titled "AI-Nutrition Register". The form has a light blue border. It contains two input fields: "Email" and "Password". Below these fields is a teal "Sign up" button. At the bottom of the form, there is a red error message that reads "Email and password are required".





# The Chat

Upon successful Login, the Nutrition AI's main chat is visible.

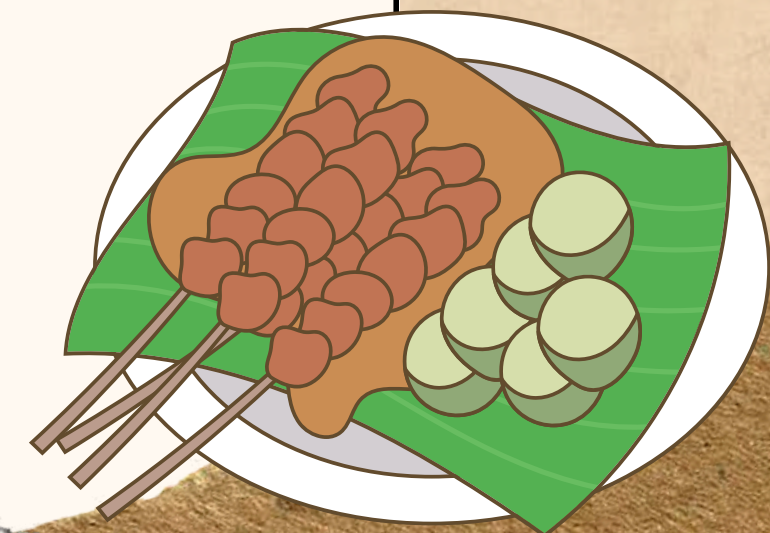
## AI-Nutrition

### Chat with AI

Do you think potatoes are good for me, and in what quantity? I'm 1.92m tall, weigh 95kg, and my goal is to lose fat while gaining muscle. Please give me a short answer

Potatoes can be a healthy part of your diet when consumed in moderation. They are a good source of vitamins, minerals, and carbohydrates. For weight loss and muscle gain, consider limiting serving sizes to about 150-250 grams (cooked) a few times a week, focusing on preparation methods like baking or boiling instead of frying. Combine them with lean proteins and plenty of vegetables for a balanced meal.

Send





# Individual Chats

Each user's chat history and information is unique to them.

what is my name and my favorite color ?

I'm sorry, but I don't have access to personal data about individuals unless it has been shared with me in our conversation. If you'd like to tell me your name and favorite color, I'd be happy to remember it for our chat!

Logging in with  
"test@gmail.com"

what is my name and my favorite color ?

Your name is Fernando, and your favorite color is red.

Logging in with  
"presentation@gmail.com"



# Individual Chats

That is because each account's chat history is stored individually within the database.

The screenshot shows a database query interface with a 'Query' tab and a 'Scratch Pad' tab. The 'Query' tab contains the following SQL query:

```
1
2 SELECT email, chat_history FROM users;
3
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with two columns: 'email' (character varying) and 'chat\_history' (text). The table contains two rows of data:

	email	chat_history
1	test@gmail.com	[]
2	presentation@gmail.c...	[{"role": "user", "content": "salut, sache qeu je m'appelle Fernando et ma couleur pr\u00e9f\u00e9r\u00e9e est le rouge"}, {"role": "assistant", "content": "Salut Fernando ! Ravi de te rencont..."}]

The table is displayed with a red border around the data rows. The interface also includes a toolbar with icons for file operations and a status bar indicating 'Showing rows: 1 to 2' and 'Page No: 1 of 1'.



# Thank You

FOR YOUR ATTENTION

