Patuakhali Science and Technology University

Faculty of Computer Science and Engineering

CCE 22 :: Database System Sessional

Sessional Project Report

Project Title: Diet and Nutrition Support System
Submission Date: Friday, June 20, 2025

Submitted to:
Prof. Dr. Md Samsuzzaman
Professor,
Department of Computer and Communication Engineering,
Patuakhali Science and Technology University

Submitted by:
Name: Abul Basar
ID: 2102036
Reg: 10163

Contents

---

1. Introduction

The Diet and Nutrition Support System is an innovative web-based platform designed to help users manage their dietary habits and receive personalized nutrition advice. It offers features such as user registration, customized meal plans, a contact/feedback system, private and public messaging, dietary calculations, and a history tracker for user activities. This system aims to support individuals and healthcare professionals in promoting healthier lifestyles.

2. Objective

- To develop a platform for creating and tracking personalized nutrition plans.
- To provide an interactive interface for monitoring dietary progress.
- To facilitate communication between users and administrators via private and public messages.
- To enable users to submit feedback through a contact page.
- To calculate and display nutritional data for informed decision-making.
- To maintain a history of user activities for progress tracking.

3. Technology

| Layer	Technology
Frontend	HTML, CSS, JavaScript
Backend	PHP
Database	MySQL
Authentication	Session Storage
Hosting	Localhost, Infinity Free
Version Control	Git
CI/CD	GitHub

4. Database Characteristics

- Supports CRUD operations (Create, Read, Update, Delete).
- Ensures data integrity with foreign keys and constraints.
- Applies normalization to reduce redundancy.
- Implements security via user authentication and parameterized queries.
- Uses PHP PDO for database connectivity.
- Employs triggers for automatic updates of user metrics.

4.1. Schema Diagram
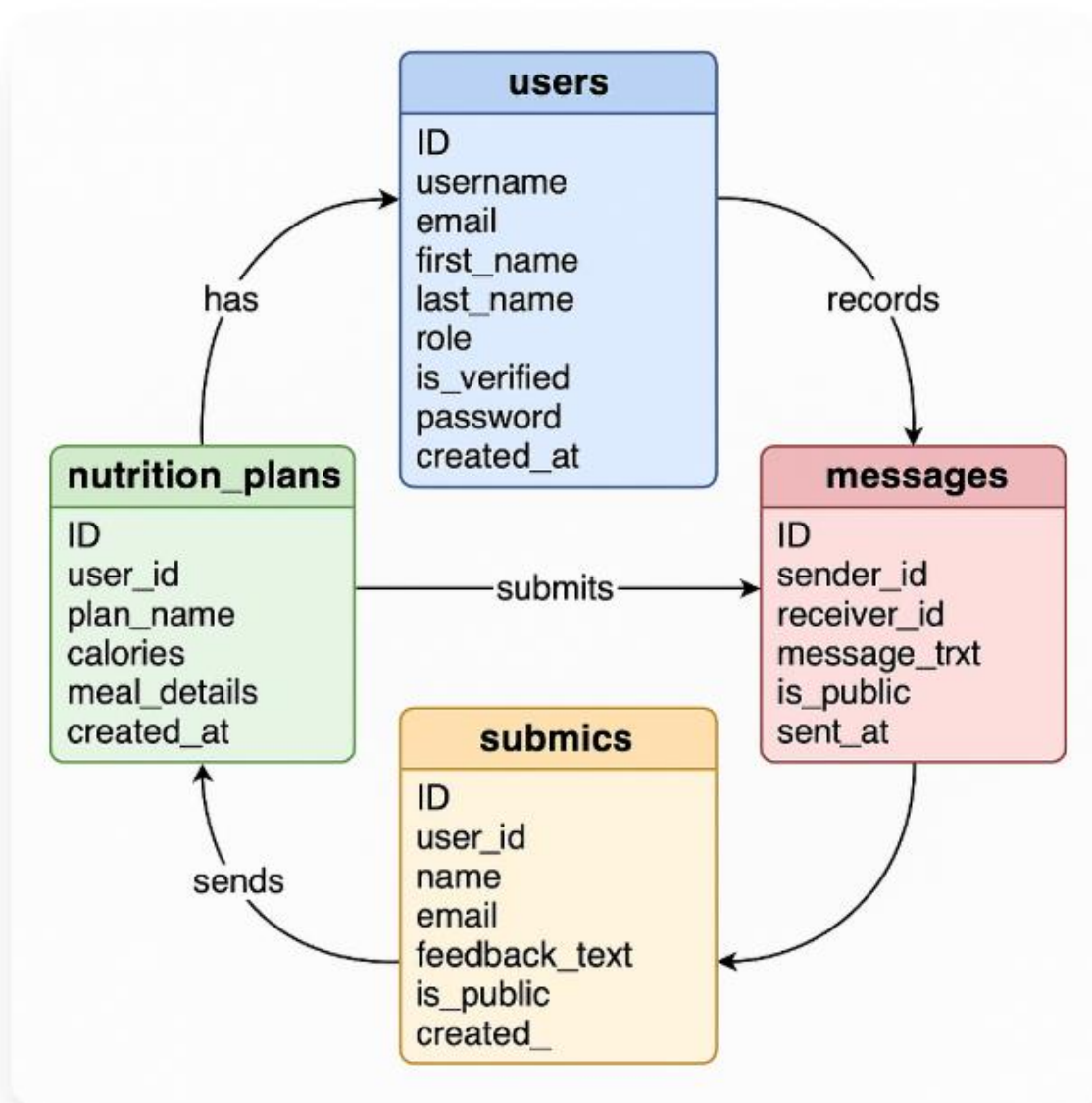
4.2. E-R Diagram
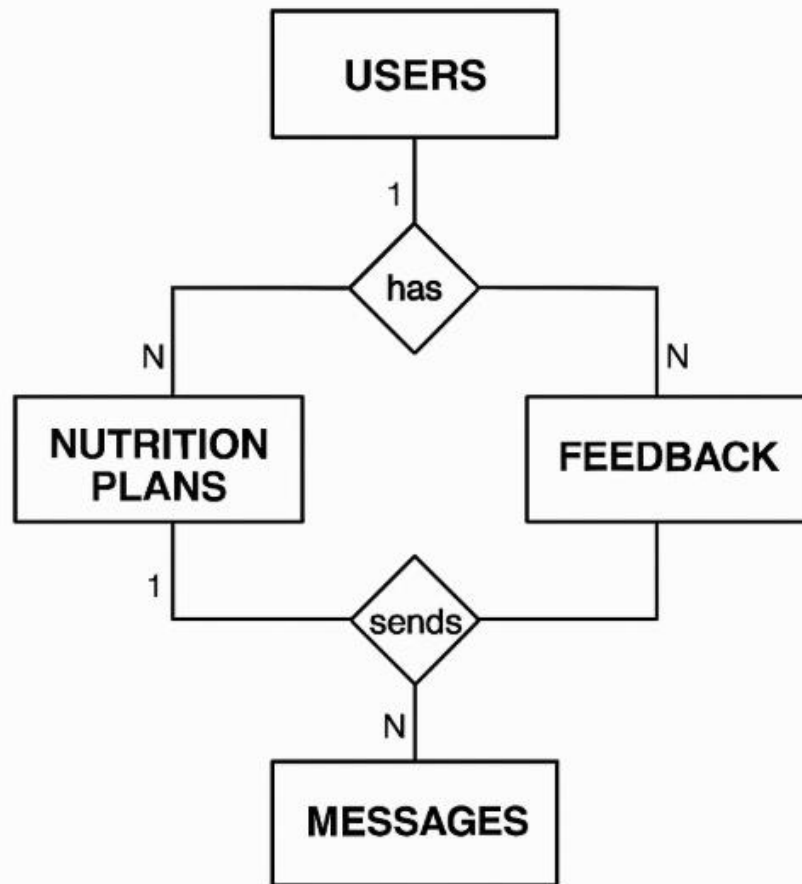
4.2.1. With All Attributes

- **Users**: ID (PK), username, email, first_name, last_name, role, is_verified, password, created_at
- **Nutrition Plans**: ID (PK), user_id (FK), plan_name, calories, meal_details, created_at
- **Feedback**: ID (PK), user_id (FK), name, email, feedback_text, created_at
- **Messages**: ID (PK), sender_id (FK), receiver_id (FK), message_text, is_public, sent_at
- **History**: ID (PK), user_id (FK), action_type, details, created_at

4.3. Gantt Chart
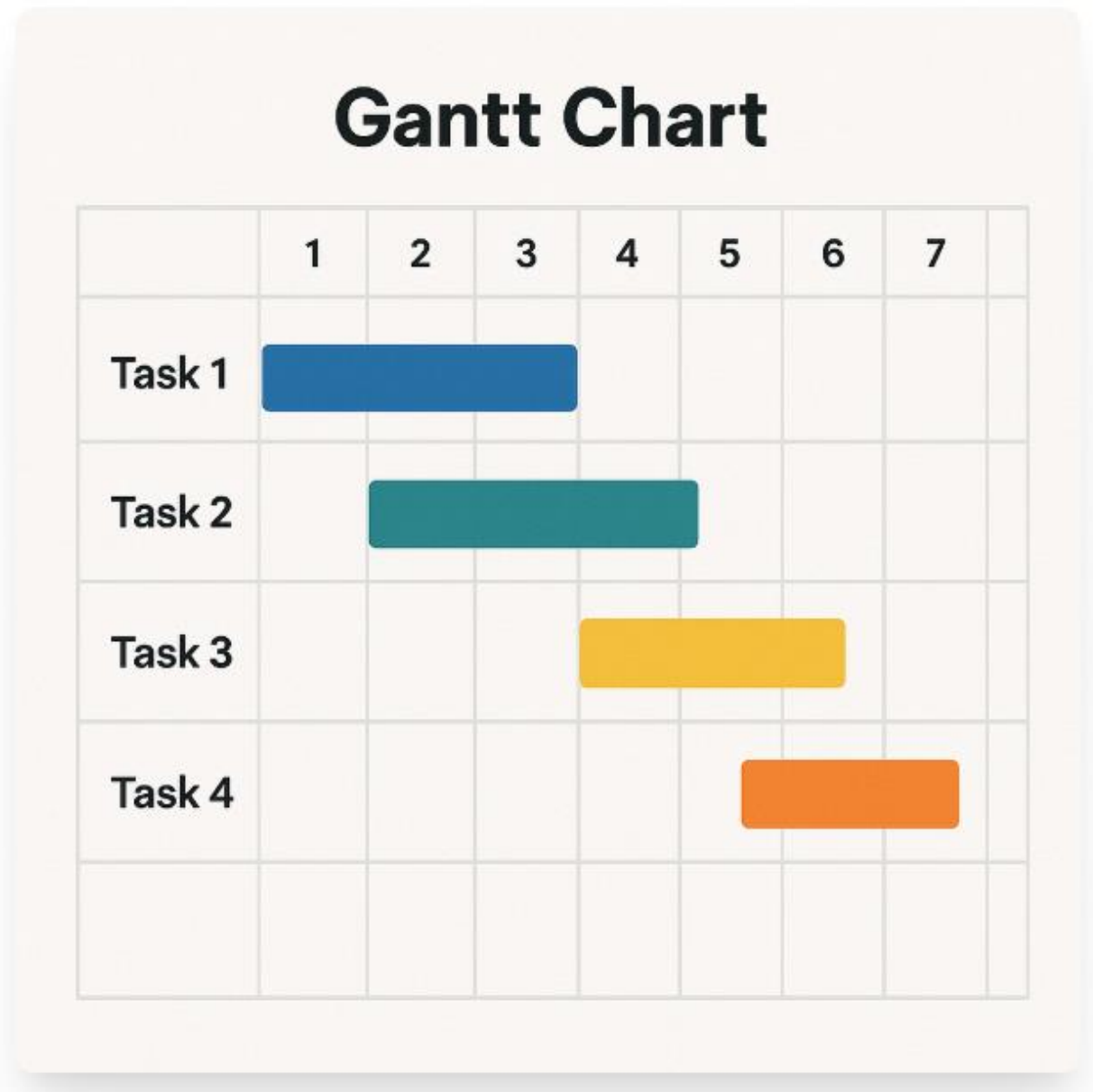

5. Database Implementation

5.1. DDL

5.1.1. Database Creation

```
CREATE DATABASE IF NOT EXISTS diet_support;
USE diet_support;
```

5.1.2. Table Creation

5.1.2.1. Users Table

```sql
DROP TABLE IF EXISTS users;
CREATE TABLE users (
    ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL UNIQUE,
    email VARCHAR(100) NOT NULL,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    role ENUM('admin', 'user') DEFAULT 'user',
    is_verified BOOLEAN DEFAULT FALSE,
    password VARCHAR(255) NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```
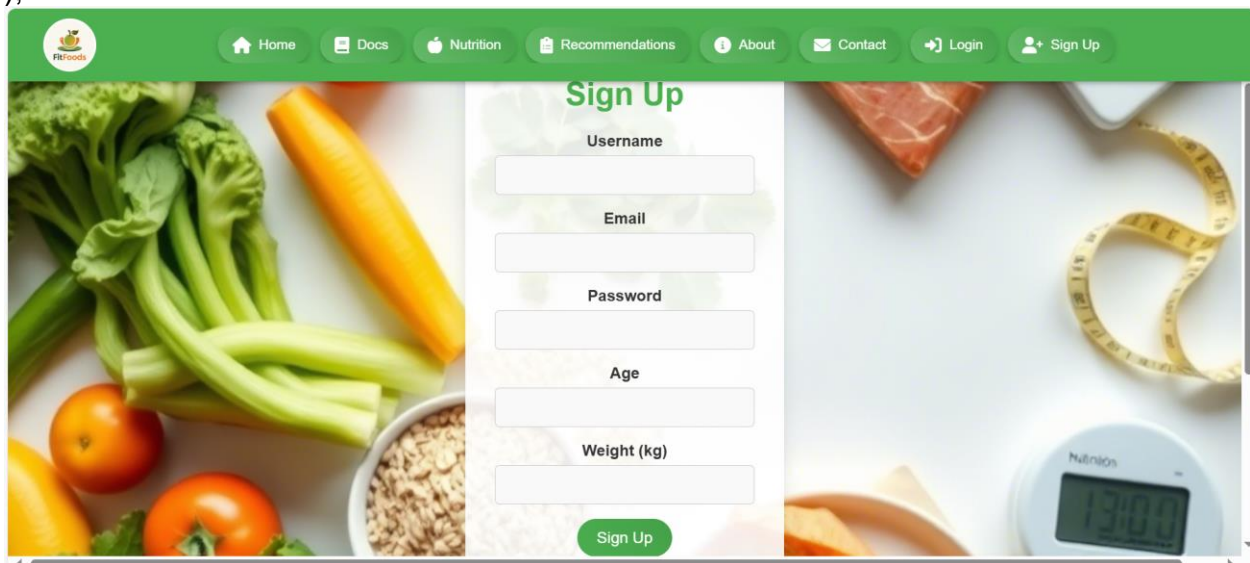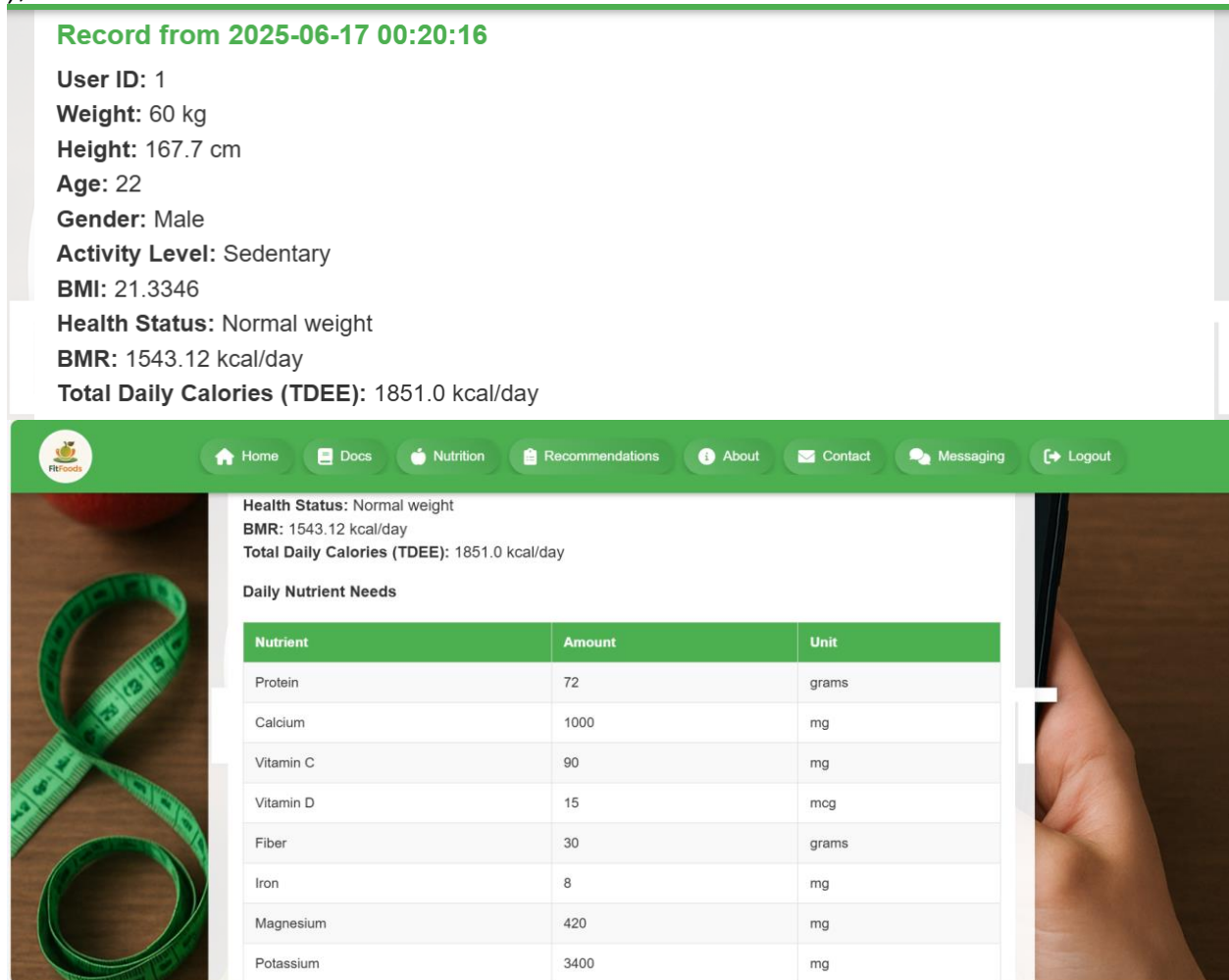

5.1.2.2. Nutrition Plans Table

```sql
DROP TABLE IF EXISTS nutrition_plans;
CREATE TABLE nutrition_plans (
    ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    plan_name VARCHAR(100),
    calories INT,
```

```sql
    meal_details TEXT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(ID)
);
```


Record from 2025-06-17 00:20:16

User ID: 1
Weight: 60 kg
Height: 167.7 cm
Age: 22
Gender: Male
Activity Level: Sedentary
BMI: 21.3346
Health Status: Normal weight
BMR: 1543.12 kcal/day
Total Daily Calories (TDEE): 1851.0 kcal/day

Health Status: Normal weight
BMR: 1543.12 kcal/day
Total Daily Calories (TDEE): 1851.0 kcal/day

Daily Nutrient Needs

Nutrient	Amount	Unit
Protein	72	grams
Calcium	1000	mg
Vitamin C	90	mg
Vitamin D	15	mcg
Fiber	30	grams
Iron	8	mg
Magnesium	420	mg
Potassium	3400	mg

5.1.2.3. Feedback Table

```sql
DROP TABLE IF EXISTS feedback;
CREATE TABLE feedback (
    ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    name VARCHAR(50),
    email VARCHAR(100),
    feedback_text TEXT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
```

```
    FOREIGN KEY (user_id) REFERENCES users(ID)
);
```


5.1.2.4. Messages Table

```
DROP TABLE IF EXISTS messages;
```

```
CREATE TABLE messages (
    ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    sender_id INT NOT NULL,
    receiver_id INT,
    message_text TEXT,
    is_public BOOLEAN DEFAULT FALSE,
    sent_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (sender_id) REFERENCES users(ID),
    FOREIGN KEY (receiver_id) REFERENCES users(ID)
);
```
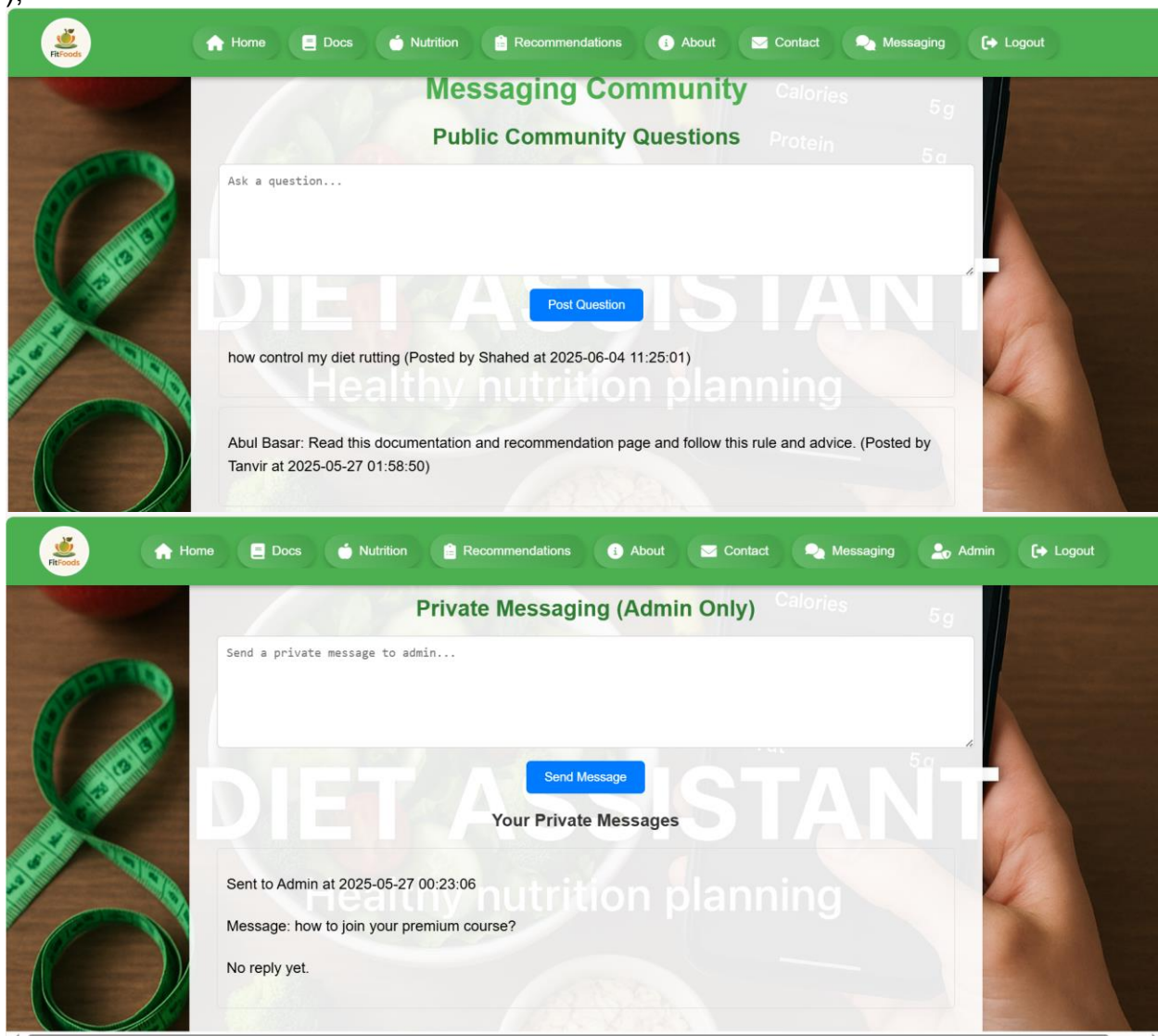

5.1.2.6. Admin Table

```sql
DROP TABLE IF EXISTS admin;
CREATE TABLE admin (
    ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```
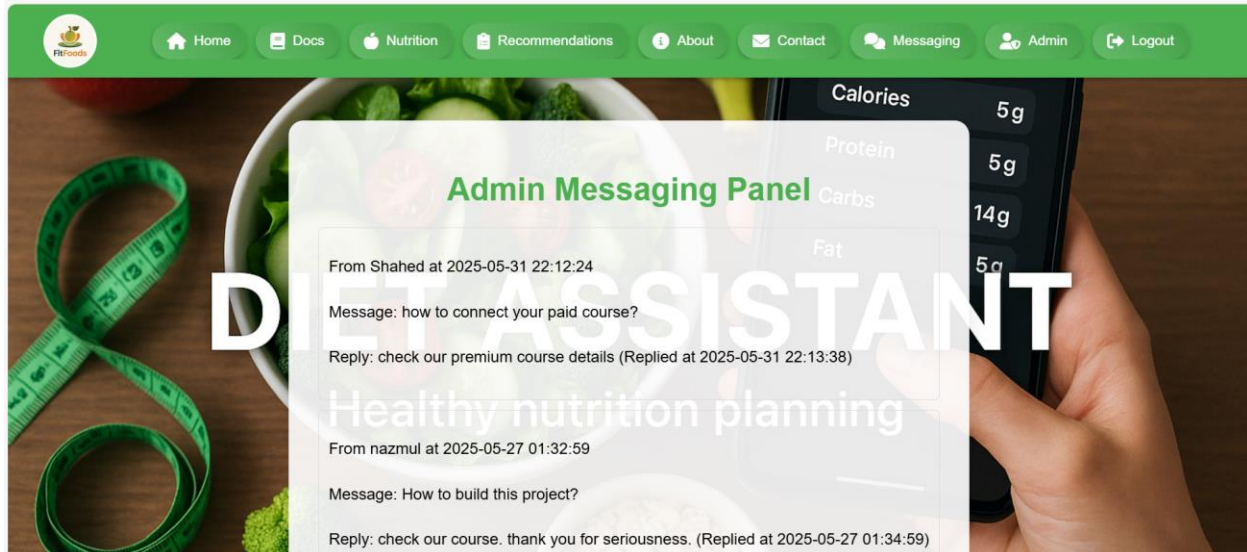

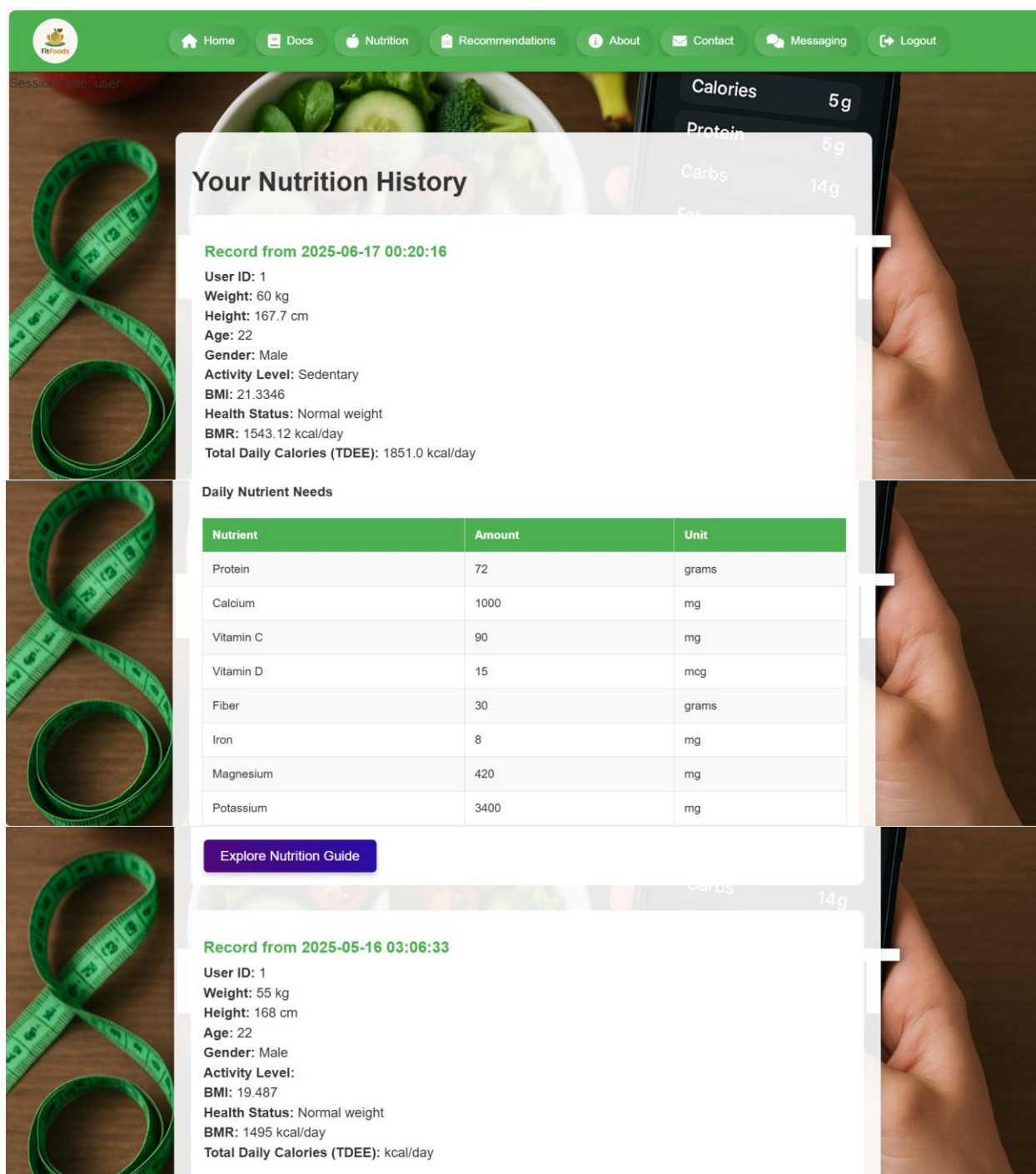## 5.1.2.5. History Table

```sql
DROP TABLE IF EXISTS history;
CREATE TABLE history (
    ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    action_type VARCHAR(50),
    details TEXT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(ID)
);
```

Session role: user

Calories 5g

Protein 5g

Carbs 14g

Your Nutrition History

Record from 2025-06-17 00:20:16

User ID: 1
Weight: 60 kg
Height: 167.7 cm
Age: 22
Gender: Male
Activity Level: Sedentary
BMI: 21.3346
Health Status: Normal weight
BMR: 1543.12 kcal/day
Total Daily Calories (TDEE): 1851.0 kcal/day

Daily Nutrient Needs

Nutrient	Amount	Unit
Protein	72	grams
Calcium	1000	mg
Vitamin C	90	mg
Vitamin D	15	mcg
Fiber	30	grams
Iron	8	mg
Magnesium	420	mg
Potassium	3400	mg

Explore Nutrition Guide

Carbs 14g

Record from 2025-05-16 03:06:33

User ID: 1
Weight: 55 kg
Height: 168 cm
Age: 22
Gender: Male
Activity Level:
BMI: 19.487
Health Status: Normal weight
BMR: 1495 kcal/day
Total Daily Calories (TDEE): kcal/day

Daily Nutrient Needs

Nutrient	Amount	Unit
Protein	66	grams
Calcium	1000	mg
Vitamin C	90	mg
Vitamin D	15	mcg
Fiber	30	grams
Iron	8	mg
Magnesium	420	mg
Potassium	3400	mg

5.1.3. Triggers

```
DELIMITER $$
CREATE TRIGGER after_plan_created
AFTER INSERT ON nutrition_plans
FOR EACH ROW
BEGIN
    UPDATE users
    SET total_plans = COALESCE(total_plans, 0) + 1
    WHERE ID = NEW.user_id;
    INSERT INTO history (user_id, action_type, details)
    VALUES (NEW.user_id, 'plan_created', CONCAT('Plan: ', NEW.plan_name));
END$$
DELIMITER ;
```

5.1.4. Views

```
CREATE VIEW active_users AS
SELECT username, COUNT(nutrition_plans.ID) AS plan_count
FROM users
LEFT JOIN nutrition_plans ON users.ID = nutrition_plans.user_id
GROUP BY username
HAVING plan_count > 0;
```
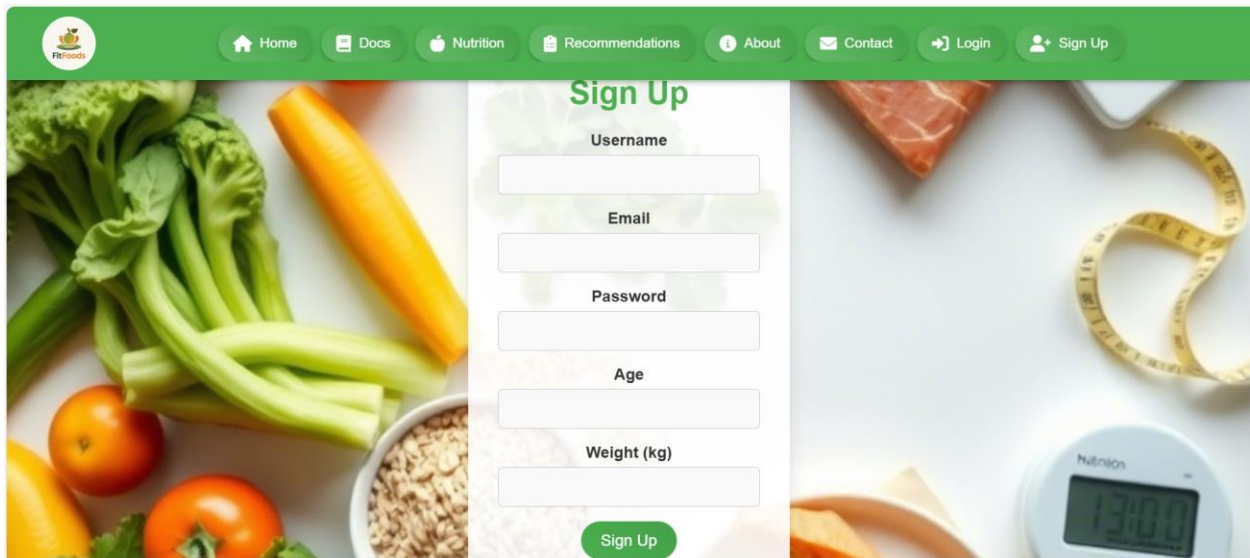
5.2. DML (SQL Queries)

5.2.1. Authentication

INSERT INTO users (username, email, first_name, last_name, password)
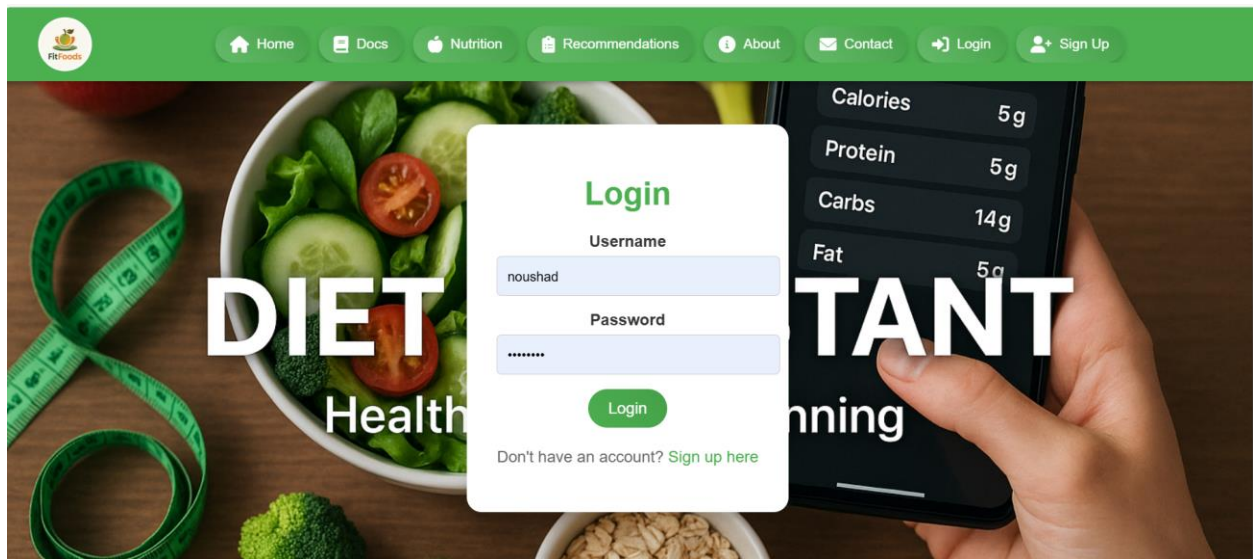VALUES (:username, :email, :first_name, :last_name, :password);

Screen Description: A "Sign Up" form with fields for First Name, Last Name, Username, Email, Password, and a "Register" button.

SELECT ID, username, password FROM users WHERE username = :username;

Screen Description: A "Login" page with Username, Password fields, and a "Login" button.

SELECT ID FROM users WHERE username = :username;

Screen Description: A validation message "Username taken" if the username exists.

5.2.2. User Profile

SELECT * FROM users WHERE username = :username;

Output:

ID | username | email | first_name | last_name | role | is_verified | created_at
1 | rakib | rakib@edu.com | Rakib | Hasan | user | 1 | 2025-06-01
10:00:00

Screen Description: A "Profile" page with editable fields for Name, Email, and a bio.

5.2.3. Nutrition Plans

```sql
SELECT plan_name, calories, meal_details
FROM nutrition_plans
WHERE user_id = (SELECT ID FROM users WHERE username = :username);
```

Output:

```
plan_name    | calories | meal_details
Daily Diet   | 2000     | Breakfast: Oatmeal...
```

Screen Description: A "My Plans" list showing plan names and expandable meal details.

			id	user_id	weight	height	age	gender	activity_level	bmi	bmr	total_calories	protein	calcium	vitar
☐	🖉 Edit	⁊ᵉ Copy ⊖ Delete	1	1	60	168	21	male	*NULL*	21.2585	1550	*NULL*	72	1000	
☐	🖉 Edit	⁊ᵉ Copy ⊖ Delete	2	2	59	167.6	21	male	*NULL*	21.0041	1537	*NULL*	70.8	1000	
☐	🖉 Edit	⁊ᵉ Copy ⊖ Delete	3	1	55	168	22	male	*NULL*	19.487	1495	*NULL*	66	1000	
☐	🖉 Edit	⁊ᵉ Copy ⊖ Delete	4	3	65	167	23	male	*NULL*	23.3067	1583	*NULL*	78	1000	
☐	🖉 Edit	⁊ᵉ Copy ⊖ Delete	5	3	65	167	23	male	*NULL*	23.3067	1583	*NULL*	78	1000	
☐	🖉 Edit	⁊ᵉ Copy ⊖ Delete	6	4	57	167.7	23	male	*NULL*	20.2679	1508	*NULL*	68.4	1000	

■ Console

5.2.4. Feedback (Contact)

```sql
INSERT INTO feedback (user_id, name, email, feedback_text, created_at)
VALUES ((SELECT ID FROM users WHERE username = :username), :name, :email,
:feedback_text, NOW());
```

Output:

```
ID | user_id | name    | email          | feedback_text | created_at
1  | 1       | Rakib   | rakib@edu.com  | Great app!    | 2025-06-20 09:28:00
```

Screen Description: A "Contact Us" form with Name, Email, and Message fields, and a "Send" button.

			id	name	email	weight	age	height	district	country	message	created_at
☐	✏ Edit	┇ Copy	⊖ Delete	1	ABUL BASAR	ug2102036@cse.pstu.ac.bd	59	21	168	narsingdi	Bangladesh	Thank You!!	2025-05-16 02:17:16
☐	✏ Edit	┇ Copy	⊖ Delete	2	nazmul	nazmul@gmail.com	65	23	167.7	gazipur	Bangladesh	How to connect your trainer?	2025-05-20 19:12:59
☐	✏ Edit	┇ Copy	⊖ Delete	16	nazmul	nazmul@gmail.com	65	23	167.7	gazipur	Bangladesh	How to connect your trainer?	2025-05-20 19:47:58

↑ ☐ Check all   *With selected:*   ✏ Edit   ┇ Copy   ⊖ Delete   📊 Export

5.2.5. Messaging (Private/Public)

- **Insert Private Message**

INSERT INTO messages (sender_id, receiver_id, message_text, is_public, sent_at)
VALUES (:sender_id, :receiver_id, :message_text, FALSE, NOW());

- **Fetch Private Messages**

SELECT m.*, u1.username AS sender_name, u2.username AS receiver_name
FROM messages m
JOIN users u1 ON m.sender_id = u1.ID
JOIN users u2 ON m.receiver_id = u2.ID
WHERE (sender_id = :user_id AND receiver_id = :other_id)
  OR (sender_id = :other_id AND receiver_id = :user_id)
ORDER BY sent_at DESC;

Output:

ID | sender_id | receiver_id | message_text | is_public | sent_at
1 | 1 | 2 | Hi, help! | 0 | 2025-06-20 09:30:00

Screen Description: A "Messages" chat window with a list of private conversations.

- **Insert Public Message**

INSERT INTO messages (sender_id, receiver_id, message_text, is_public, sent_at)
VALUES (:sender_id, NULL, :message_text, TRUE, NOW());

- **Fetch Public Messages**

SELECT m.*, u.username AS sender_name
FROM messages m
JOIN users u ON m.sender_id = u.ID

```sql
WHERE m.is_public = TRUE
ORDER BY sent_at DESC;
```

Output:

```
ID | sender_id | receiver_id | message_text    | is_public | sent_at
2  | 1         | NULL        | Healthy tips!   | 1         | 2025-06-20 09:35:00
```

Screen Description: A "Public Feed" section showing all public messages.

	id	user_id	admin_id	message	reply	created_at	updated_at
☐ Edit Copy Delete	1	6	2	hello	NULL	2025-05-26 02:45:49	2025-05-27 01:29:59
☐ Edit Copy Delete	2	6	2	hello	NULL	2025-05-26 02:47:46	2025-05-27 01:29:59
☐ Edit Copy Delete	3	6	2	hello	NULL	2025-05-26 02:49:05	2025-05-27 01:29:59
☐ Edit Copy Delete	4	2	2	how to join your premium course?	NULL	2025-05-27 00:23:06	2025-05-27 01:29:59
☐ Edit Copy Delete	5	1	2	how to purchases your course?	check our premium course details.	2025-05-27 01:10:41	2025-06-04 11:26:06
☐ Edit Copy Delete	6	1	2	how to check your trainer?	visit our website. Thank you for comment.	2025-05-27 01:16:51	2025-05-27 01:31:27
☐ Edit Copy Delete	7	5	2	How to build this project?	check our course. thank you for seriousness.	2025-05-27 01:32:59	2025-05-27 01:34:59
☐ Edit Copy Delete	8	8	2	how to connect your paid course?	check our premium course details	2025-05-31 22:12:24	2025-05-31 22:13:38

↰ ☐ Check all *With selected:* ✏ Edit ⧉ Copy ⊖ Delete ⬇ Export

5.2.6. Calculations

- **Total Calories Consumed**

```sql
SELECT u.username, SUM(np.calories) AS total_calories
FROM users u
LEFT JOIN nutrition_plans np ON u.ID = np.user_id
WHERE u.ID = :user_id
GROUP BY u.username;
```

Output:

```
username | total_calories
rakib    | 4000
```

Screen Description: A "Nutrition Stats" dashboard showing total calories.

- **Average Daily Caloric Intake**

SELECT u.username, AVG(np.calories) AS avg_calories
FROM users u
LEFT JOIN nutrition_plans np ON u.ID = np.user_id
WHERE u.ID = :user_id
GROUP BY u.username;

Output:

username | avg_calories
rakib | 2000

| total_calories	protein	calcium	vitamin_c 1	vitamin_d	fiber	iron	magnesium	potassium	water	created_
NULL	72	1000	90	15	30	8	420	3400	2100	2025-05-1 02:00:02
1832.0	70.8	1000	90	15	30	8	420	3400	2065	2025-06-0 11:21:54
1844.0	72	1000	90	15	30	8	420	3400	2100	2025-06-0 11:18:50
1844.0	72	1000	90	15	30	8	420	3400	2100	2025-05-3 22:04:30
1912.0	78	1000	90	15	30	8	420	3400	2275	2025-05-2 01:49:26
1840.0	70.8	1000	90	15	30	8	420	3400	2065	2025-05-2 03:44:30
2193.0	78	1000	90	15	30	8	420	3400	2275	2025-05-2 01:16:33
1840.0	70.8	1000	90	15	30	8	420	3400	2065	2025-05-2

5.2.7. History Data

INSERT INTO history (user_id, action_type, details)
VALUES (:user_id, 'plan_created', :plan_details);

Output:

```
ID | user_id | action_type  | details         | created_at
1  | 1       | plan_created | Plan: Daily Diet | 2025-06-20 09:40:00
```

SELECT action_type, details, created_at
FROM history
WHERE user_id = :user_id
ORDER BY created_at DESC;

Screen Description: A "History" tab listing actions like "plan_created" with timestamps.
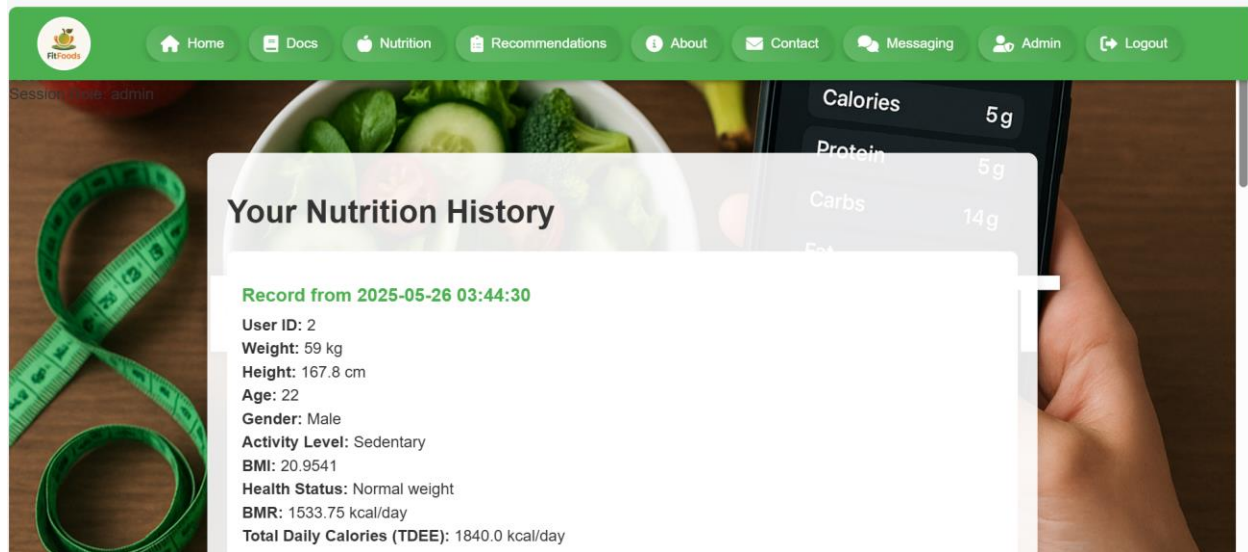
6. Limitations

- Hosted on free servers, impacting performance.
- Manual review of nutrition plans is required.
- Limited mobile responsiveness in some sections.

7. Conclusion

The Diet and Nutrition Support System provides a robust platform for managing dietary needs, supporting users and professionals with interactive features. Its comprehensive database and query system lay a foundation for future enhancements, including automation and mobile optimization.

8. References

- [W3Schools SQL](#)
- [PHP Manual](#)

THE END