

## **CCE 224 :: Database System Sessional**

### **Sessional Project Report**

**Project Title: Diet and Nutrition Support System**

**Submission Date: Saturday, June 14, 2025**

#### **Submitted to:**

Prof. Dr. Md Samsuzzaman

Professor,

Department of Computer and Communication

Engineering,

Patuakhali Science and Technology University

#### **Submitted by:**

Abul Basar

ID: 2102036

Reg: 10163

---

## **Contents**

1. Introduction .....	3
-----------------------	---

2. Objective .....	3
3. Technology .....	3
4. Database Characteristics .....	3
4.1. Schema Diagram .....	4
4.2. E-R Diagram .....	5
4.2.1. Without Attributes .....	5
4.2.2. With All Attributes .....	6
4.3. Gantt Chart .....	6
5. Database Implementation .....	7
5.1. DDL .....	7
5.1.1. Database Creation .....	7
5.1.2. Table Creation .....	7
5.1.2.1. Users Table .....	7
5.1.2.2. Nutrition Plans Table .....	8
5.1.2.3. Feedback Table .....	8
5.1.2.4. Messages Table .....	9
5.1.3. Triggers .....	10
5.1.4. Views .....	11
5.2. DML (SQL Queries) .....	11
5.2.1. Authentication .....	11
5.2.2. User Profile .....	12
5.2.3. Nutrition Plans .....	13
5.2.4. Feedback .....	14
5.2.5. Messaging .....	15
6. Limitations .....	16
7. Conclusion .....	16
8. References .....	17

---

## 1. Introduction

The Diet and Nutrition Support System is an online platform designed to assist users in managing their dietary needs and maintaining a healthy lifestyle. It provides features such as user registration, personalized nutrition plans, feedback submission, and a messaging system for user-admin communication. This system aims to support individuals and healthcare providers in tracking and improving dietary habits.

## 2. Objective

- To develop a platform for creating and managing personalized nutrition plans.
- To enable users to track their dietary progress interactively.
- To assist healthcare professionals in providing dietary advice efficiently.
- To facilitate communication between users and administrators via a messaging system.
- To allow users to provide feedback for continuous improvement.

## 3. Technology

Layer	Technology
Frontend	HTML, CSS, JavaScript
Backend	PHP
Database	MySQL

Layer	Technology
Authentication	Session Storage
Hosting	Localhost, InfinityFree
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 data redundancy.
- Implements security through user authentication and parameterized queries.
- Uses PHP PDO for database interactions, supporting multiple databases.
- Employs triggers to automatically update user activity metrics.

### 4.1. Schema Diagram

*(Note: Insert a diagram here showing tables like users, nutrition\_plans, feedback, and messages with relationships. You can create this using a tool like Lucidchart or draw.io.)*

### 4.2. E-R Diagram

#### 4.2.1. Without Attributes

*(Insert a simple E-R diagram showing entities: Users, Nutrition Plans, Feedback, Messages, with relationships like "has," "submits," and "sends.")*

#### 4.2.2. 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), feedback\_text, created\_at
- **Messages:** ID (PK), sender\_id (FK), receiver\_id (FK), message\_text, sent\_at

#### 4.3. Gantt Chart

*(Insert a Gantt chart showing tasks like Database Design, Frontend Development, Backend Integration, and Testing over a 4-week period.)*

### 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

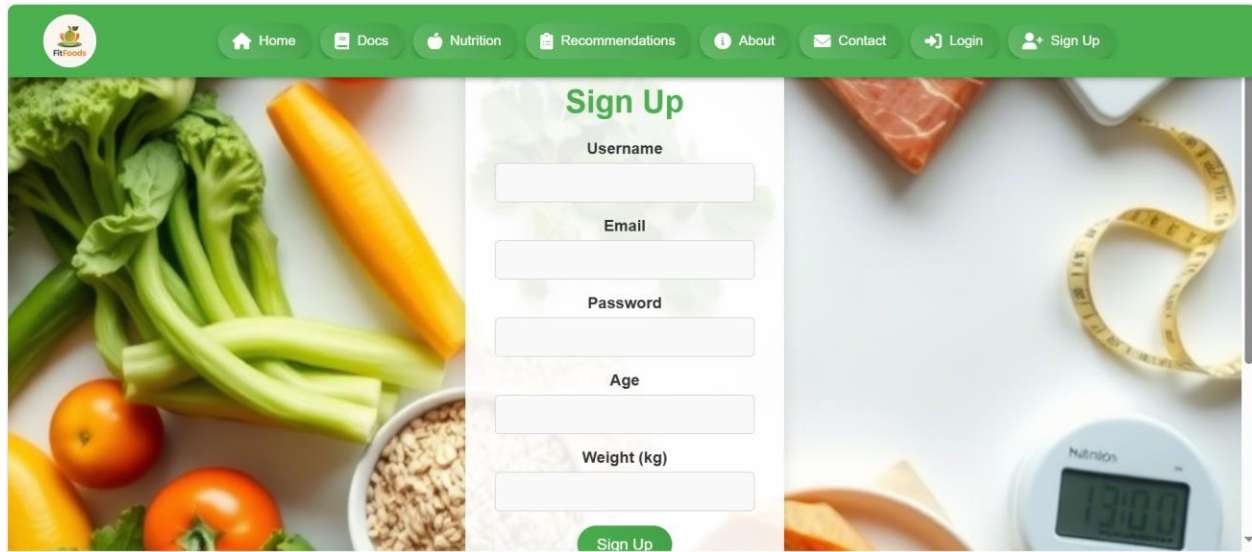
```

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
);

```

		id	username	email	password	age	w
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	1	noushad	noushad@gmail.com	\$2y\$10\$6avH.IljDIFvrro8QQJbqebtgPvhoq2sVPLajf2RvOB...	22	
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	2	Abul Basar	ug2102036@cse.pstu.ac.bd	\$2y\$10\$yrdyN6L5aokw6LxY0dCvwwQmDBBpUQGKJNL2sCp/ahV...	22	
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	3	sabbir	sabbir@gmail.com	\$2y\$10\$hGA5SQArRqHY1A0TiefvHuXcSe.z5jNPrk7rd7xuuOC...	23	
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	4	Muhibullah	muhibullah@gmail.com	\$2y\$10\$//5SphM3WS4wOW670WuADOW6W1iOYfgom1DsvA31sEE...	23	
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	5	nazmul	nazmul@gmail.com	\$2y\$10\$pz.PWDs8/9lQmuTacq3EsuflmmE9qF9vTB69MoSdmYb...	22	
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	6	Tanjil	tanjil@gmail.com	\$2y\$10\$ceftnDvVqnLiKb5UEGh0.iPGWdY4ydy4CxZeFwUzjX...	21	
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	7	Tanvir	tanvir@gmail.com	\$2y\$10\$fPaK6TTFvUIFO7.nD2fiyuU8QHsBBQmyEsP9QuwyUKZ...	22	
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	8	Shahed	shahed@gmail.com	\$2y\$10\$OI3KmmumVtopMUne0b5xEI.MSkMHIEJ97dMDnipDaOiY...	23	





#### 5.1.2.2. Nutrition Plans Table

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

			id	user_id	weight	height	age	gender	activity_level	bmi	bmr	total_calories	protein	calcium	vitami
<div><div></div><div><div>Edit</div><div>Copy</div><div>Delete</div></div></div>	1	1	60	168	21	male	NULL	21.2585	1550	NULL	72	1000			
<div><div></div><div><div>Edit</div><div>Copy</div><div>Delete</div></div></div>	2	2	59	167.6	21	male	NULL	21.0041	1537	NULL	70.8	1000			
<div><div></div><div><div>Edit</div><div>Copy</div><div>Delete</div></div></div>	3	1	55	168	22	male	NULL	19.487	1495	NULL	66	1000			
<div><div></div><div><div>Edit</div><div>Copy</div><div>Delete</div></div></div>	4	3	65	167	23	male	NULL	23.3067	1583	NULL	78	1000			
<div><div></div><div><div>Edit</div><div>Copy</div><div>Delete</div></div></div>	5	3	65	167	23	male	NULL	23.3067	1583	NULL	78	1000			
<div><div></div><div><div>Edit</div><div>Copy</div><div>Delete</div></div></div>	6	4	57	167.7	23	male	NULL	20.2679	1508	NULL	68.4	1000			
<div><div></div><div><div>Edit</div><div>Copy</div><div>Delete</div></div></div>	7	5	60	168	22	male	NULL	21.2585	1545	NULL	72	1000			
<div><div></div><div><div>Edit</div><div>Copy</div><div>Delete</div></div></div>	8	5	65	168	23	male	NULL	23.03	1590	NULL	78	1000			
<div><div></div><div><div>Edit</div><div>Copy</div><div>Delete</div></div></div>	9	5	70	168	24	male	NULL	24.8016	1635	NULL	84	1000			
<div><div></div><div><div>Edit</div><div>Copy</div><div>Delete</div></div></div>	10	6	60	167	21	male	NULL	21.5139	1543	NULL	72	1000			

## Health Condition

**User ID:** 1

**Weight:** 60 kg

**Height:** 167.7 cm

**Age:** 22

**Gender:** Male

**Activity Level:** Sedentary

**BMI:** 21.3

**Health Status:** Normal weight

**BMR (Basal Metabolic Rate):** 1543.1 kcal/day

**Total Daily Calories (TDEE):** 1851.8 kcal/day

**Recommendation:** Great job! Maintain a balanced diet to stay in this healthy range.



## 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
Water	2100	ml

### 5.1.2.3. Feedback Table

```
DROP TABLE IF EXISTS feedback;
CREATE TABLE feedback (
  ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  user_id INT,
  feedback_text TEXT,
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(ID)
);
```

<input type="checkbox"/>	Edit	Copy	Delete	3	6	2	hello	NULL	2025-05-26 02:49:05	2025-05-27 01:29:59
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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
<input type="checkbox"/>	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

#### 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 NOT NULL,
  message_text TEXT,
  sent_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (sender_id) REFERENCES users(ID),
  FOREIGN KEY (receiver_id) REFERENCES users(ID)
);

```

<div><div><div></div><div></div><div></div></div></div>		id	user_id	message	created_at		
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	1	6	hi	2025-05-26 02:45:03
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	2	6	hello	2025-05-26 02:59:09
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	3	2	how to improve my diet activity?	2025-05-27 00:21:24
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	4	7	Abul Basar: Read this documentation and recommenda...	2025-05-27 01:58:50
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	5	8	how control my diet rutting	2025-06-04 11:25:01

#### 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;
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;
```

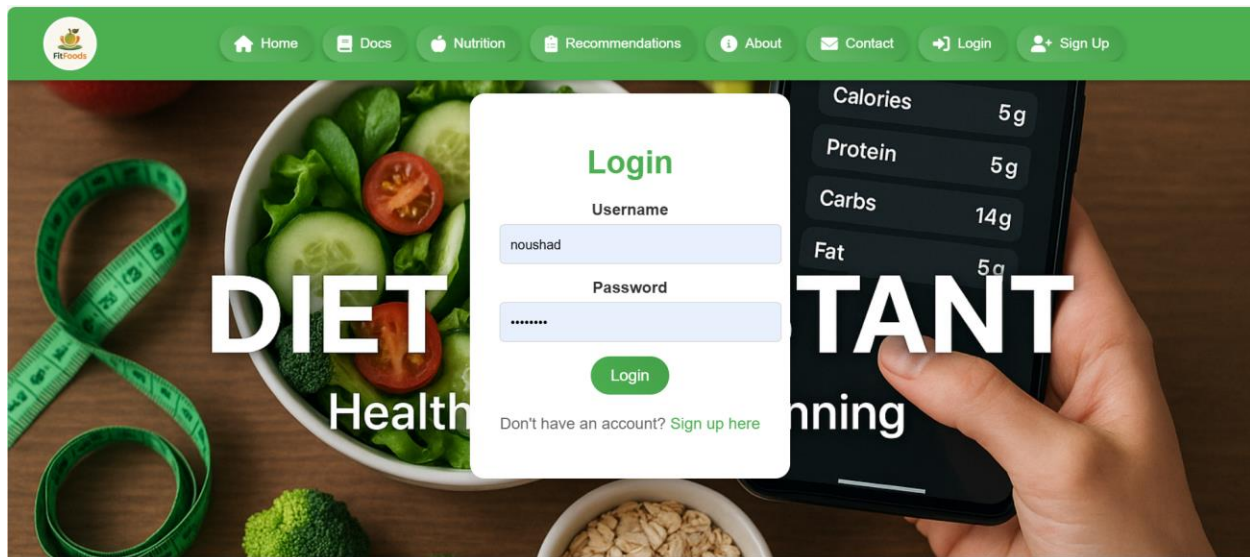
## 5.2. DML (SQL Queries)

### 5.2.1. Authentication

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

*-- User Login*

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



-- Check User Existence

```
SELECT ID FROM users WHERE username = :username;
```

## Output:

```
ID | username | password
1 | noushad | $2y$12$hashedpass...
```

### 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 | noushad | noushad@edu.com | Rakib | Hasan | user | 1 |
2025-06-01 10:00:00
```

### 5.2.3. Nutrition Plans

```
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...
```

#### 5.2.4. Feedback

```
INSERT INTO feedback (user_id, feedback_text)
VALUES ((SELECT ID FROM users WHERE username = :username),
:feedback_text);
```

### Output:

```
ID | user_id | feedback_text | created_at
1 | 1      | Great app!    | 2025-06-16 11:00:00
```

#### 5.2.5. Messaging

```
INSERT INTO messages (sender_id, receiver_id, message_text)
VALUES ((SELECT ID FROM users WHERE username = :sender),
(SELECT ID FROM users WHERE username = :receiver), :message);
```

### Output:

```
ID | sender_id | receiver_id | message_text | sent_at
1 | 1         | 2          | Hi, need help! | 2025-06-16 11:30:00
```

## **6. Limitations**

- Hosted on free servers, which may affect performance.
- Manual review of nutrition plans by admins is required.
- Limited mobile responsiveness in certain sections.

## **7. Conclusion**

The Diet and Nutrition Support System provides a valuable tool for users to manage their dietary needs effectively. It supports both individuals and professionals with an intuitive interface and robust database management, with potential for future automation and enhancements.

## **8. References**

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