

# Meal Planning Nutrition System

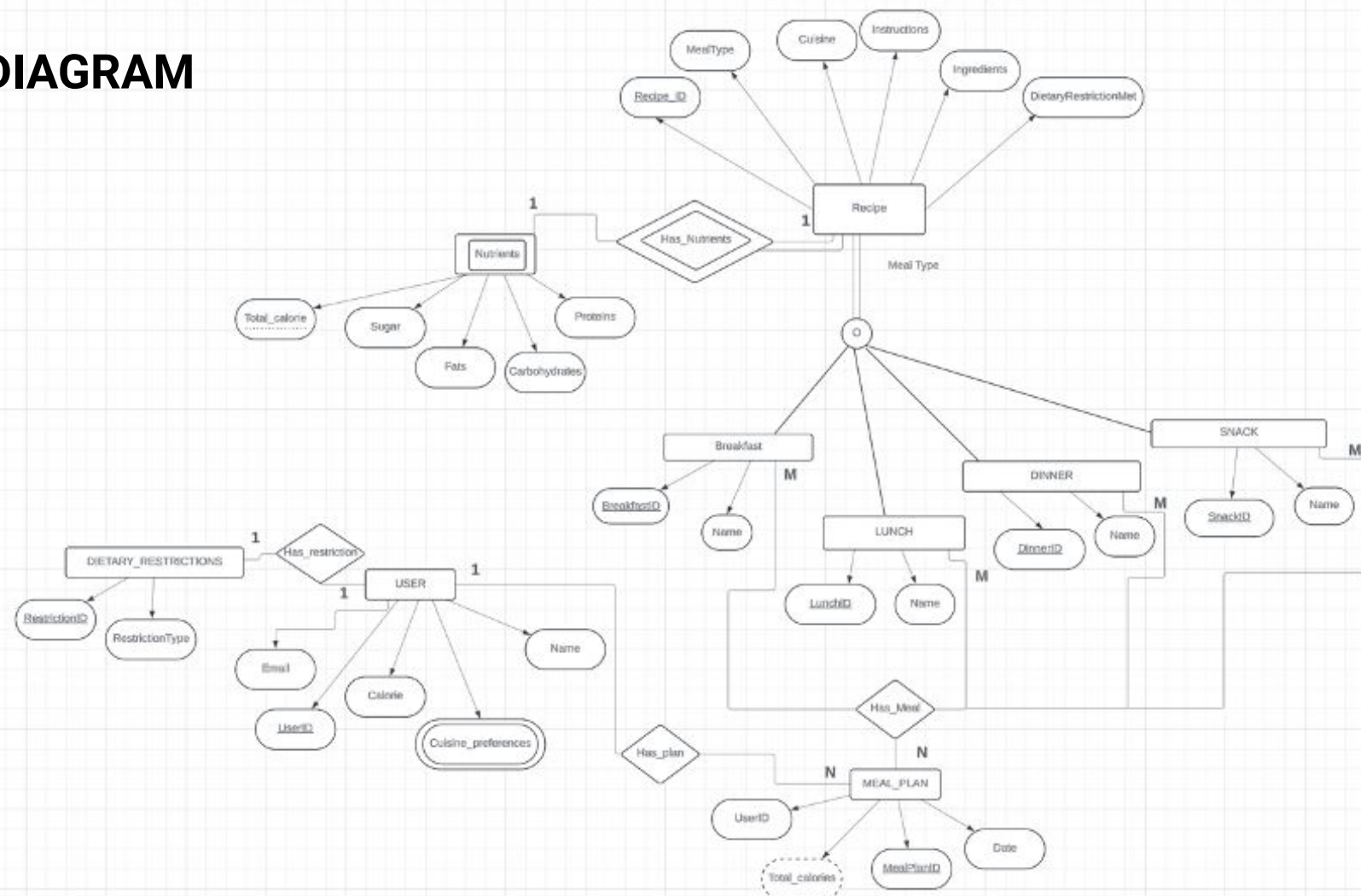
Group Members: Muhammad Siddique, Mudit Upadhyay,  
Anirudh Nemmani, Syed Kabir, Vaishnavi Karaguppi, Senthen  
Velmurugan, Arvinth Kumar Kalainathan

# Introduction

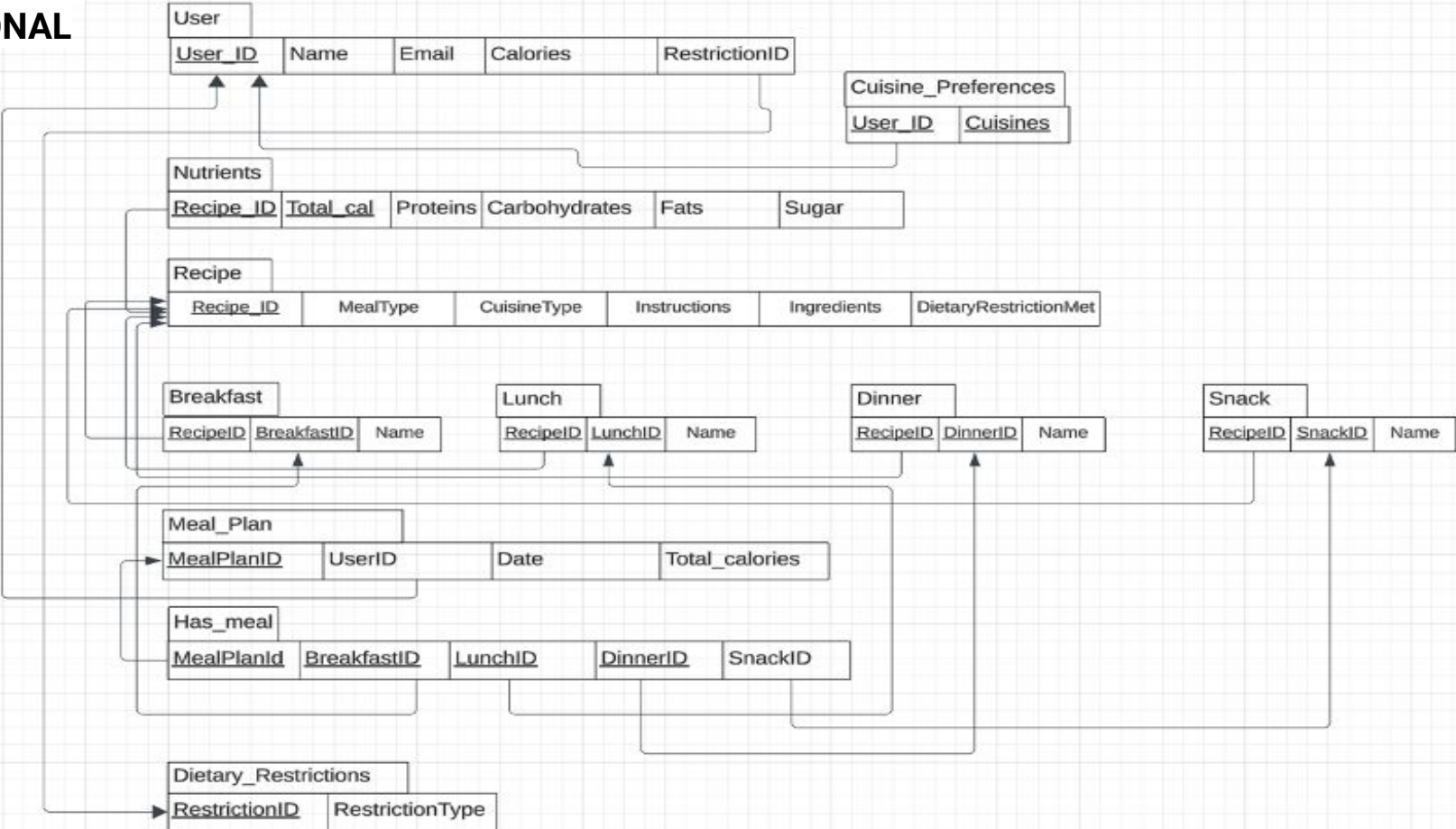


- **Purpose and Problem Addressed:** The project aims to simplify daily meal preparation by creating personalized meal plans tailored to users' dietary preferences and restrictions.
- **Database Design:** The app's database is designed to store user profiles, meal plans, recipes, and nutritional information.
- **Data Security:** The app emphasizes data privacy and protection, ensuring that sensitive user information such as calorie intake, dietary restrictions, and meal preferences is securely stored and handled.
- **Real-World Impact:** The app meets the growing demand for personalized, secure, and accessible nutritional planning.

# EER DIAGRAM



NORMALIZED  
RELATIONAL  
MODEL



# Live Demo/Database Operations SS



# Regular Query - Application

## HealthyMeals Dashboard

Successfully queried Users

Query   Insert   Delete   Update   Quit

### Query Data

Regular Table Query

Users

User_ID	Name	Email	Calories	RestrictionID
1	John Doe	john.doe@example.com	2000	1
2	Jane Smith	jane.smith@example.com	1500	2
3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	Emily Johnson	emily.johnson@example.com	1600	5
6	David Lee	david.lee@example.com	2200	6
7	Anna Gupta	anna.gupta@example.com	1900	7
8	Paul Martin	paul.martin@example.com	2100	8
9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	Carlos Ramirez	carlos.ramirez@example.com	2000	10

# Regular Query - Database

SQLQuery1.sql - TI...IRAMISU\Cake (58))\*

```
SELECT * FROM Users WHERE Calories > 2000;
```

100 %

Results Messages

	User_ID	Name	Email	Calories	RestrictionID
1	6	David Lee	david.lee@example.com	2200	6
2	8	Paul Martin	paul.martin@example.com	2100	8

# Special Query - Application

## HealthyMeals Dashboard

Successfully executed meal-calories query

Query   Insert   Delete   Update   Quit

### Query Data

Meal Calories Analysis



Execute Query

Recipe_ID	MealType	CuisineType	Calories	Proteins	Carbohydrates	Fats
2	Dinner	Italian	800	20.30	90.10	30.50
3	Lunch	Indian	650	25.20	70.50	15.80
9	Breakfast	French	600	12.30	55.50	35.00
7	Lunch	Mexican	550	15.80	65.30	20.20
5	Snack	American	500	5.20	50.50	25.30
6	Dinner	Indian	450	12.50	60.00	10.00
1	Lunch	Indian	400	10.50	60.30	8.20
8	Snack	Italian	350	8.50	40.00	10.50
4	Breakfast	American	300	20.00	2.00	25.00



# Insert - Application - Table Before

## Before

### HealthyMeals Dashboard

Successfully queried Users

Query Insert Delete Update Quit

#### Query Data

Regular Table Query

Users

User_ID	Name	Email	Calories	RestrictionID
1	John Doe	john.doe@example.com	2000	1
2	Jane Smith	jane.smith@example.com	1500	2
3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	Emily Johnson	emily.johnson@example.com	1600	5
6	David Lee	david.lee@example.com	2200	6
7	Anna Gupta	anna.gupta@example.com	1900	7
8	Paul Martin	paul.martin@example.com	2100	8
9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	Carlos Ramirez	carlos.ramirez@example.com	2000	10

# Insert - Application - Insert Page

## HealthyMeals Dashboard

Successfully inserted into Users

Query

Insert

Delete

Update

Quit

Insert Data

Users

12

Anirudh Nemmani

Ani@test.com

2200

3

Insert Data

# Insert - Application - Table After

After

## HealthyMeals Dashboard

Successfully queried Users

Query

Insert

Delete

Update

Quit

Query Data

Regular Table Query

Users

User_ID	Name	Email	Calories	RestrictionID
1	John Doe	john.doe@example.com	2000	1
2	Jane Smith	jane.smith@example.com	1500	2
3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	Emily Johnson	emily.johnson@example.com	1600	5
6	David Lee	david.lee@example.com	2200	6
7	Anna Gupta	anna.gupta@example.com	1900	7
8	Paul Martin	paul.martin@example.com	2100	8
9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	Carlos Ramirez	carlos.ramirez@example.com	2000	10
12	Anirudh Nemmani	Ani@test.com	2200	3

# Insert - Database

SQLQuery1.sql - TI...IRAMISU\Cake (58))\* -> X

```
INSERT INTO Users (User_ID, Name, Email, Calories, RestrictionID)
VALUES (11, 'Mike Anderson', 'mike.anderson@example.com', 2500, 1);
SELECT * FROM Users;
```

100 %

Results Messages

	User_ID	Name	Email	Calories	RestrictionID
1	1	John Doe	john.doe@example.com	2000	1
2	2	Jane Smith	jane.smith@example.com	1500	2
3	3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	5	Emily Johnson	emily.johnson@example.com	1600	5
6	6	David Lee	david.lee@example.com	2200	6
7	7	Anna Gupta	anna.gupta@example.com	1900	7
8	8	Paul Martin	paul.martin@example.com	2100	8
9	9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	10	Carlos Ramirez	carlos.ramirez@example.com	2000	10
11	11	Mike Anderson	mike.anderson@example.com	2500	1

# Update - Application - Table Before

## HealthyMeals Dashboard

Successfully queried Meal\_Plan

Query   Insert   Delete   Update   Quit

### Query Data

Regular Table Query ▼

Meal\_Plan ▼

MealPlanID	Date	UserID	Total_calories
1	2024-10-15T05:00:00.000Z	1	2050
2	2024-10-16T05:00:00.000Z	2	1500
3	2024-10-17T05:00:00.000Z	3	1800
4	2024-10-18T05:00:00.000Z	4	1700
5	2024-10-19T05:00:00.000Z	5	1600
6	2024-10-20T05:00:00.000Z	6	2200
7	2024-10-21T05:00:00.000Z	7	1900
8	2024-10-22T05:00:00.000Z	8	2100
9	2024-10-23T05:00:00.000Z	9	1400
10	2024-10-24T05:00:00.000Z	10	2000

# Update - Application - Update Page

## HealthyMeals Dashboard

Successfully queried Meal\_Plan

Query Insert Delete Update Quit

### Update Data

Meal\_Plan

MealPlanID	Date	UserID	Total_calories	Actions
1	2024-10-15T05:00:00.000Z	1	3000	Update
2	2024-10-16T05:00:00.000Z	2	1500	Update
3	2024-10-17T05:00:00.000Z	3	1800	Update
4	2024-10-18T05:00:00.000Z	4	1700	Update
5	2024-10-19T05:00:00.000Z	5	1600	Update
6	2024-10-20T05:00:00.000Z	6	2200	Update
7	2024-10-21T05:00:00.000Z	7	1900	Update
8	2024-10-22T05:00:00.000Z	8	2100	Update
9	2024-10-23T05:00:00.000Z	9	1400	Update
10	2024-10-24T05:00:00.000Z	10	2000	Update

# Update - Application - After update

## HealthyMeals Dashboard

Successfully queried Meal\_Plan

Query   Insert   Delete   Update   Quit

### Query Data

Regular Table Query

Meal\_Plan

MealPlanID	Date	UserID	Total_calories
1	2024-10-15T05:00:00.000Z	1	3000
2	2024-10-16T05:00:00.000Z	2	1500
3	2024-10-17T05:00:00.000Z	3	1800
4	2024-10-18T05:00:00.000Z	4	1700
5	2024-10-19T05:00:00.000Z	5	1600
6	2024-10-20T05:00:00.000Z	6	2200
7	2024-10-21T05:00:00.000Z	7	1900
8	2024-10-22T05:00:00.000Z	8	2100
9	2024-10-23T05:00:00.000Z	9	1400
10	2024-10-24T05:00:00.000Z	10	2000

# Update - Database

SQLQuery1.sql - Tl...IRAMISU\Cake (58))\* -p X

```
UPDATE Users
SET Calories = 1800
WHERE User_ID = 2;
SELECT * FROM Users;
```

100 %

Results Messages

	User_ID	Name	Email	Calories	RestrictionID
1	1	John Doe	john.doe@example.com	2000	1
2	2	Jane Smith	jane.smith@example.com	1800	2
3	3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	5	Emily Johnson	emily.johnson@example.com	1600	5
6	6	David Lee	david.lee@example.com	2200	6
7	7	Anna Gupta	anna.gupta@example.com	1900	7
8	8	Paul Martin	paul.martin@example.com	2100	8
9	9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	10	Carlos Ramirez	carlos.ramirez@example.com	2000	10



# Delete - Application - User table before

## HealthyMeals Dashboard

Successfully queried Users

Query

Insert

Delete

Update

Quit

Query Data

Regular Table Query

Users

User_ID	Name	Email	Calories	RestrictionID
1	John Doe	john.doe@example.com	2000	1
2	Jane Smith	jane.smith@example.com	1500	2
3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	Emily Johnson	emily.johnson@example.com	1600	5
6	David Lee	david.lee@example.com	2200	6
7	Anna Gupta	anna.gupta@example.com	1900	7
8	Paul Martin	paul.martin@example.com	2100	8
9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	Carlos Ramirez	carlos.ramirez@example.com	2000	10
12	Anirudh Nemmani	Ani@test.com	2200	3
50	Muhammad Siddique	muhammad@test.com	1800	3

# Delete - Application - Delete Page - Delete User Muhammad

## HealthyMeals Dashboard

Successfully queried Users

Query

Insert

Delete

Update

Quit

Delete Data

Select Table

User_ID	Name	Email	Calories	RestrictionID	Actions
1	John Doe	john.doe@example.com	2000	1	Delete
2	Jane Smith	jane.smith@example.com	1500	2	Delete
3	Ahmed Ali	ahmed.ali@example.com	1800	3	Delete
4	Sarah Cohen	sarah.cohen@example.com	1700	4	Delete
5	Emily Johnson	emily.johnson@example.com	1600	5	Delete
6	David Lee	david.lee@example.com	2200	6	Delete
7	Anna Gupta	anna.gupta@example.com	1900	7	Delete
8	Paul Martin	paul.martin@example.com	2100	8	Delete
9	Olivia Taylor	olivia.taylor@example.com	1400	9	Delete
10	Carlos Ramirez	carlos.ramirez@example.com	2000	10	Delete
12	Anirudh Nemmani	Ani@test.com	2200	3	Delete
50	Muhammad Siddique	muhammad@test.com	1800	3	Delete

# Delete - Application - User Table after

## HealthyMeals Dashboard

Successfully queried Users

Query

Insert

Delete

Update

Quit

Query Data

Regular Table Query

Users

User_ID	Name	Email	Calories	RestrictionID
1	John Doe	john.doe@example.com	2000	1
2	Jane Smith	jane.smith@example.com	1500	2
3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	Emily Johnson	emily.johnson@example.com	1600	5
6	David Lee	david.lee@example.com	2200	6
7	Anna Gupta	anna.gupta@example.com	1900	7
8	Paul Martin	paul.martin@example.com	2100	8
9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	Carlos Ramirez	carlos.ramirez@example.com	2000	10
12	Anirudh Nemmani	Ani@test.com	2200	3

# Delete - Database

```
SQLQuery1.sql - TI...IRAMISU\Cake (58))* ✕  
INSERT INTO Users (User_ID, Name, Email, Calories, RestrictionID)  
VALUES (11, 'Mike Anderson', 'mike.anderson@example.com', 2500, 1);  
SELECT * FROM Users;
```

100 %

Results Messages

	User_ID	Name	Email	Calories	RestrictionID
1	1	John Doe	john.doe@example.com	2000	1
2	2	Jane Smith	jane.smith@example.com	1500	2
3	3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	5	Emily Johnson	emily.johnson@example.com	1600	5
6	6	David Lee	david.lee@example.com	2200	6
7	7	Anna Gupta	anna.gupta@example.com	1900	7
8	8	Paul Martin	paul.martin@example.com	2100	8
9	9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	10	Carlos Ramirez	carlos.ramirez@example.com	2000	10
11	11	Mike Anderson	mike.anderson@example.com	2500	1

```
SQLQuery1.sql - TI...IRAMISU\Cake (58))* ✕  
DELETE FROM Users WHERE User_ID = 11;  
SELECT * FROM Users;
```

100 %

Results Messages

	User_ID	Name	Email	Calories	RestrictionID
1	1	John Doe	john.doe@example.com	2000	1
2	2	Jane Smith	jane.smith@example.com	1500	2
3	3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	5	Emily Johnson	emily.johnson@example.com	1600	5
6	6	David Lee	david.lee@example.com	2200	6
7	7	Anna Gupta	anna.gupta@example.com	1900	7
8	8	Paul Martin	paul.martin@example.com	2100	8
9	9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	10	Carlos Ramirez	carlos.ramirez@example.com	2000	10

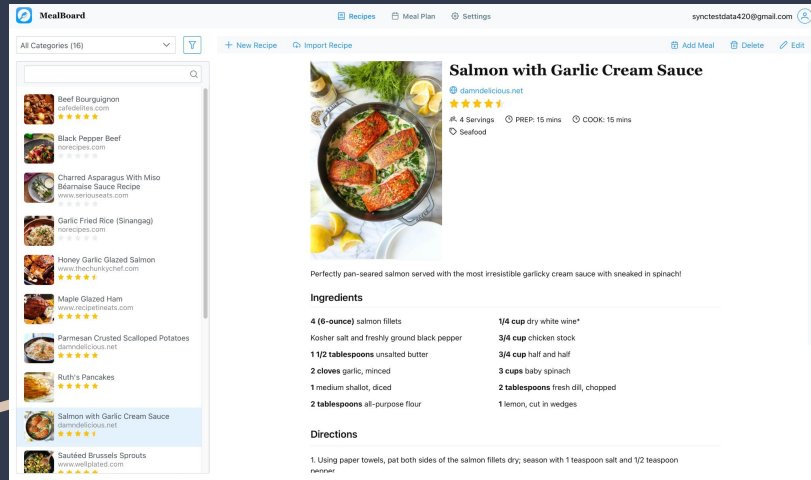
## View - DailyMealPlan

```
4 CREATE VIEW DailyMealPlan
5 AS SELECT
6     MP.MealPlanID,
7     MP.Date,
8     U.Name AS User_Name,
9     U.Calories AS User_Calorie_Goal,
10    B.Name AS Breakfast,
11    L.Name AS Lunch,
12    D.Name AS Dinner,
13    S.Name AS Snack,
14    MP.Total_calories AS Total_MealPlan_Calories
15 FROM
16     Meal_Plan MP
17 JOIN
18     Users U ON MP.UserID = U.User_ID
19 LEFT JOIN
20     Has_meal HM ON MP.MealPlanID = HM.MealPlanId
21 LEFT JOIN
22     Breakfast B ON HM.BreakfastID = B.BreakfastID
23 LEFT JOIN
24     Lunch L ON HM.LunchID = L.LunchID
25 LEFT JOIN
26     Dinner D ON HM.DinnerID = D.DinnerID
27 LEFT JOIN
28     Snack S ON HM.SnackID = S.SnackID;
29 SELECT * FROM DailyMealPlan;
```

## View - DailyMealPlan - Result

[illegible]

# Comparison with Similar Work

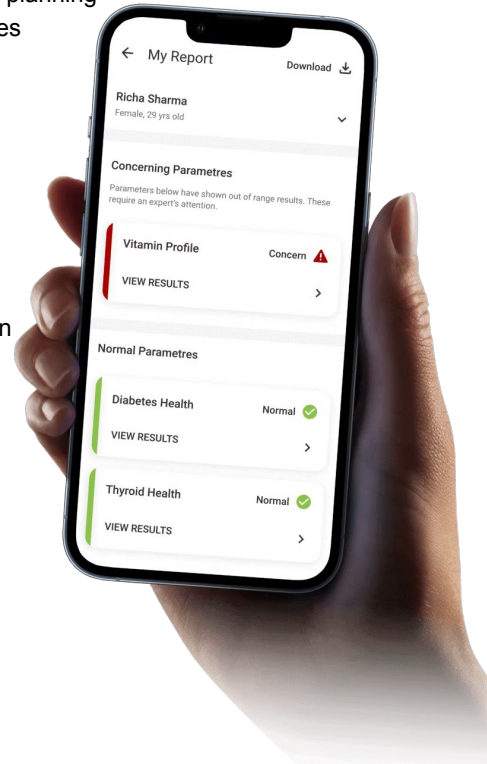


## HealthifyMe

- Similarities:
  - Healthy recipe database
  - Dietary restriction handling
- Differences:
  - Our focus on automated meal planning
  - Enhanced data privacy features

## MealBoard

- Similarities:
  - Meal scheduling functionality
  - Recipe storage
- Differences:
  - Our automated personalization
  - Advanced nutritional tracking



# Conclusion & Future Work

## Achievements

- Secure database design
- Personalized meal planning
- Dietary restriction management

## Future Enhancements

- Machine learning for better recommendations
- Mobile application development
- Integration with grocery delivery services
- Social sharing features
- Recipe rating system



# Questions?

