CPSC 304 Project Cover Page

Milestone #: 4

Date: Mar 31,2024

Group Number: 52

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
George Song	27473412	t9k1b	georgesong97@gmail.com
Jayden Piao	83679589	k7v0w	jaydenpiao@gmail.com
Kohen Lee	12154647	m3b8	Kohenlee1234@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Link to Repository:
https://github.students.cs.ubc.ca/CPSC304-2023W-T2/project_k7v0w_m3b8j_t9k1b

Project Description: Our domain focuses on nutrition and personal meal management. We are creating a web application that allows users to discover recipes, manage their meal plans, and keep track of their grocery needs. Relating to culinary arts,we will provide ingredient combinations, and serving suggestions. In terms of nutrition, we will provide nutritional content and health aspects of recipes such as macronutrient/micronutrient distributions. For meal management, users can plan and organise meals and their grocery shopping to maintain a balanced diet for all types of dietary preferences and restrictions.

Script: The script we will be running is mealprep.sql

Schema changes:
NONE
Final Schema:
Primary keys are underlined, foreign keys are bolded.

- User(userID: int, name: varchar, email: varchar, username: varchar)
- Recipe(recipeID: int, name: varchar, category: varchar, instructions: varchar)
- Category(name)
- VegetarianRecipe(recipeID: int)
- LowCalorieRecipe(recipeID: int)
- SugarFreeRecipe(recipeID: int)
- Ingredient(name: varchar, isVegetarian: BOOLEAN)
- nutritionInfoRecipe(calories: int, sugar: int, proteinContent: int, fatContent: int, carbContent: int, **recipeID**: int)
- nutritionInfoIngredient(calories: int, sugar: int, proteinContent: int, fatContent: int, carbContent: int, **ingredientName**:varchar)
- MealPlan(mealPlanID: int, name: varchar, date: date, **userID**: int)
- ShoppingList(shoppingListID,mealPlanID,recipeID,ingredientName)
- Rating(ratingID: int, score: int, **userID: int, recipeID: int**)
- Review(reviewID: int, date: date, message: varchar, **userID: int, recipeID:** int)
- saves(**userID: int, recipeID: int**)
- recipeContains**(recipeID: int, ingredientName: varchar**)
- mealPlanContains(**mealPlanID: int, recipeID:int**)
- listContains(**shoppingListID:int, ingredientName: varcha**r)

A list of all SQL queries used and where it can be found in the code (i.e., file name
and line number(s)). For SQL query requirements, check the rubric listed on Canvas for Milestone 4.

SQL QUERIES USED:

Insert: Insert meal plan for user
INSERT INTO MealPlan (name, date, userID) VALUES (?, ?, ?)

Location: Services/mealplan.service.ts: line 33
Delete: Delete meal plan
DELETE FROM MealPlan WHERE mealPlanID = ?

Location:Services/mealplan.service.ts: line 59
Update: Edit meal plan name
UPDATE MealPlan SET name = ? WHERE mealPlanID = ?

Location: Services/mealplan.service.ts: line 79
Selection: get meal plans based on userid
SELECT * FROM meal plan WHERE userID = ?

Location: Services/mealplan.service.ts: line 14
Projection: **get nutrition info from recipe**
SELECT recipeID, calories, proteinContent, sugar, fatContent, carbContent
FROM nutritionInfoRecipe
WHERE recipeID = ?;

Location:Services/recipe.service.ts line 133
Join: get reviews
SELECT *
FROM Review
JOIN Recipe ON Review.recipeID = Recipe.recipeID
WHERE Review.recipeID = ?;

Location: Location:Services/recipe.service.ts line 115
Aggregation with Group by: get recipe ratings for each recipe, group by rating
SELECT r.recipeID, r.name, COUNT(*) AS rating_count, ra.score
FROM recipe AS r
JOIN rating AS ra ON r.recipeID = ra.recipeID
WHERE r.recipeID = ?
GROUP BY r.recipeID, r.name, ra.score;

Location: services/recipe.service.ts line 94

Aggregation with Having: Search that can apply filters, and min rating having score >= ?

```
SELECT r.*, AVG(ra.score) AS avgRating
FROM recipe r
JOIN SugarFreeRecipe sf ON r.recipeID = sf.recipeID
JOIN LowCalorieRecipe lc ON r.recipeID = lc.recipeID
JOIN VegetarianRecipe v ON r.recipeID = v.recipeID
LEFT JOIN Rating ra ON r.recipeID = ra.recipeID
WHERE (r.name LIKE '%recipeName%' OR '%recipeName%' = '')
  AND (sf.recipeID IS NOT NULL OR '%sugarFree%' = 'false')
  AND (lc.recipeID IS NOT NULL OR '%lowCalorie%' = 'false')
  AND (v.recipeID IS NOT NULL OR '%vegetarian%' = 'false')
GROUP BY r.recipeID
HAVING (AVG(ra.score) >= minRating OR minRating IS NULL);
```

Location: recipe.service.ts Line 153

Nested Aggregation with GROUP BY: Show average rating for each recipe

```
SELECT r.recipeID, r.name, r.instructions, avg_rating.average_rating
FROM recipe AS r
INNER JOIN (
   SELECT recipeID, AVG(score) AS average_rating
   FROM rating
   GROUP BY recipeID
) AS avg_rating
ON r.recipeID = avg_rating.recipeID;
```

Location: services/recipe.service.ts line 73

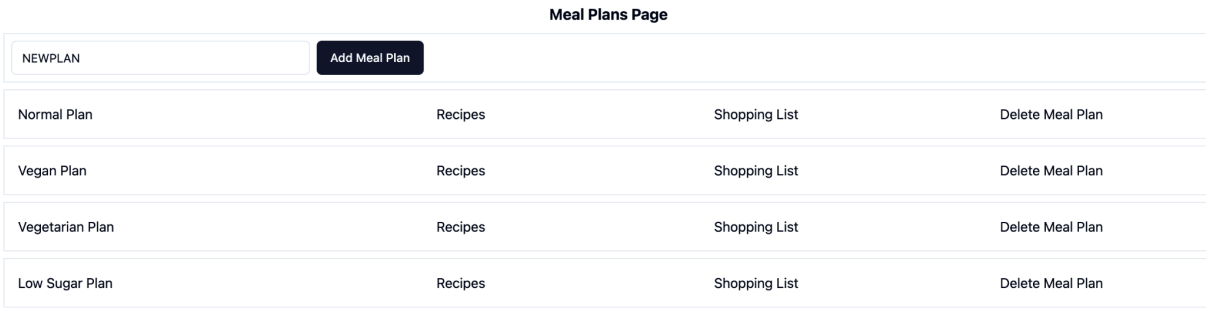
Division: Select users where they have reviewed every recipe

```
SELECT userID, name
        FROM User
        WHERE NOT EXISTS (
          SELECT recipeID
          FROM Recipe
          WHERE NOT EXISTS (
            SELECT *
            FROM Review
            WHERE Review.userID = User.userID
            AND Review.recipeID = Recipe.recipeID
```

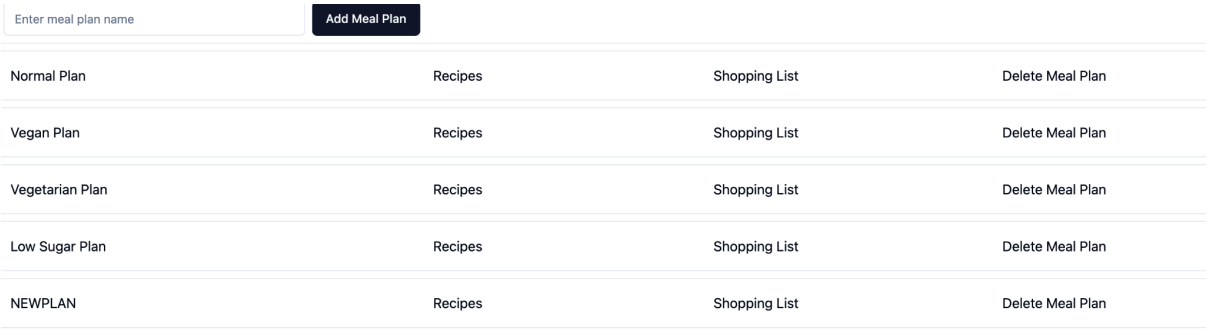
Location: services/user.service.ts line 29

Screenshots demonstrating the functionality of each query using the GUI. We want to see a before/during/after progression of events. For example, the before screenshot would be what data is in the table before you run the query, the during screenshot(s) is how the query is triggered using the GUI, and the after screenshot is what data is in your table afterwards. Please label each set of screenshots with the name of the query it is meant to address (e.g., "Insert Operation").

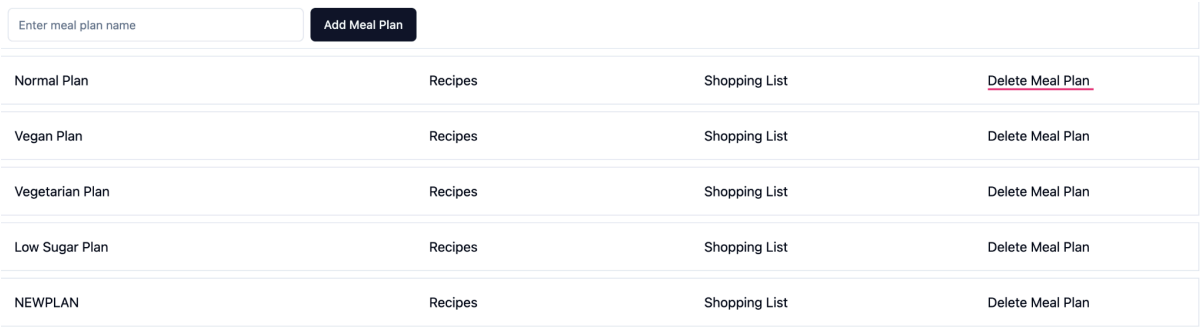
***Note:** the red lines help to indicate what is being pressed, not part of actual web app
Insert: Insert meal plan for user
Before:



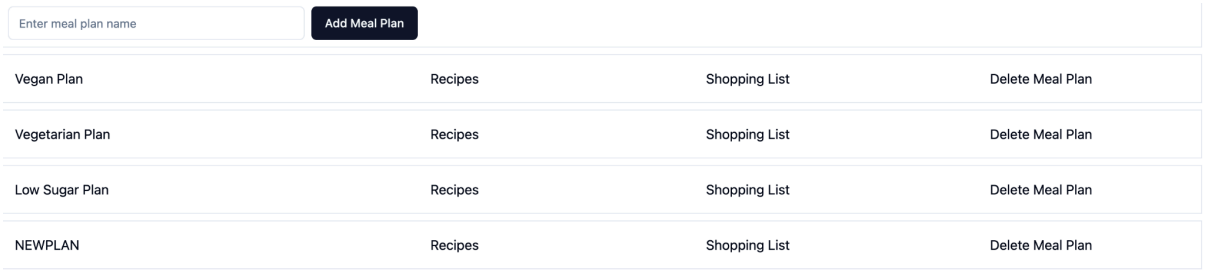
After:



Delete: Delete meal plan
Before:

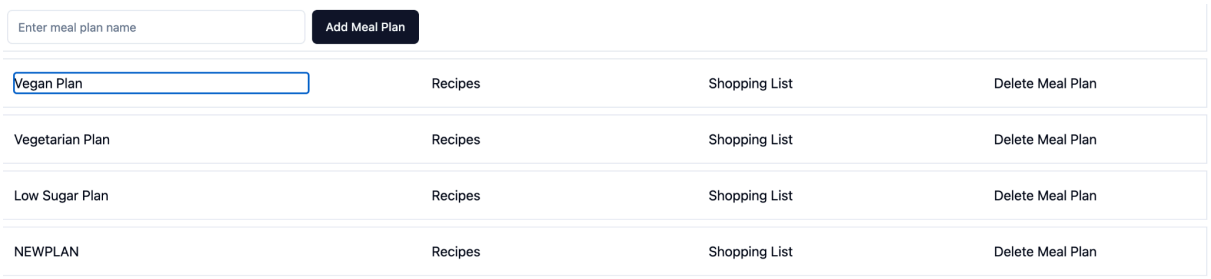


After:

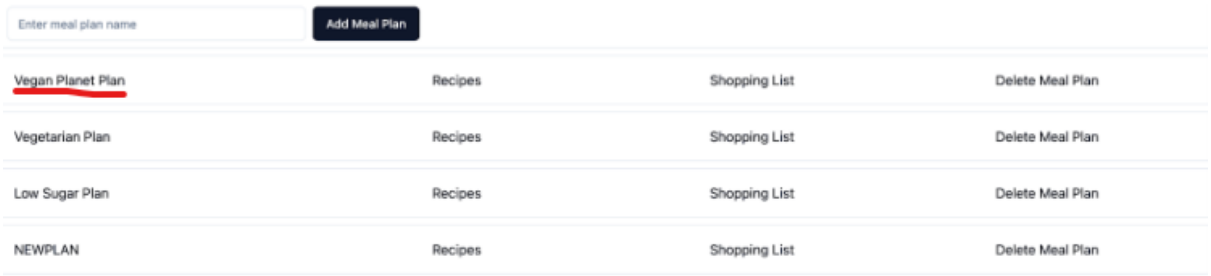


Update: Edit meal plan name

Before:

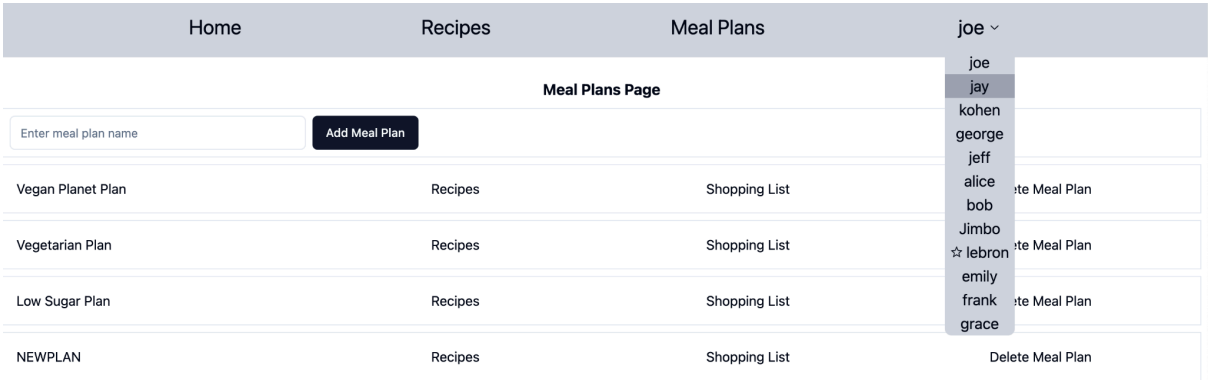


After:



Selection: get meal plans based on userid

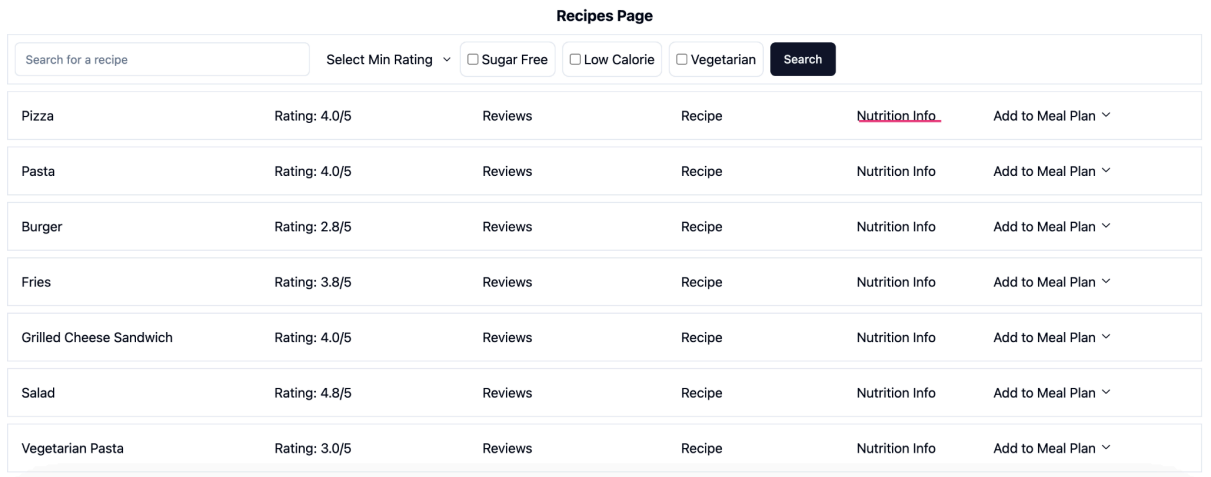
Before:



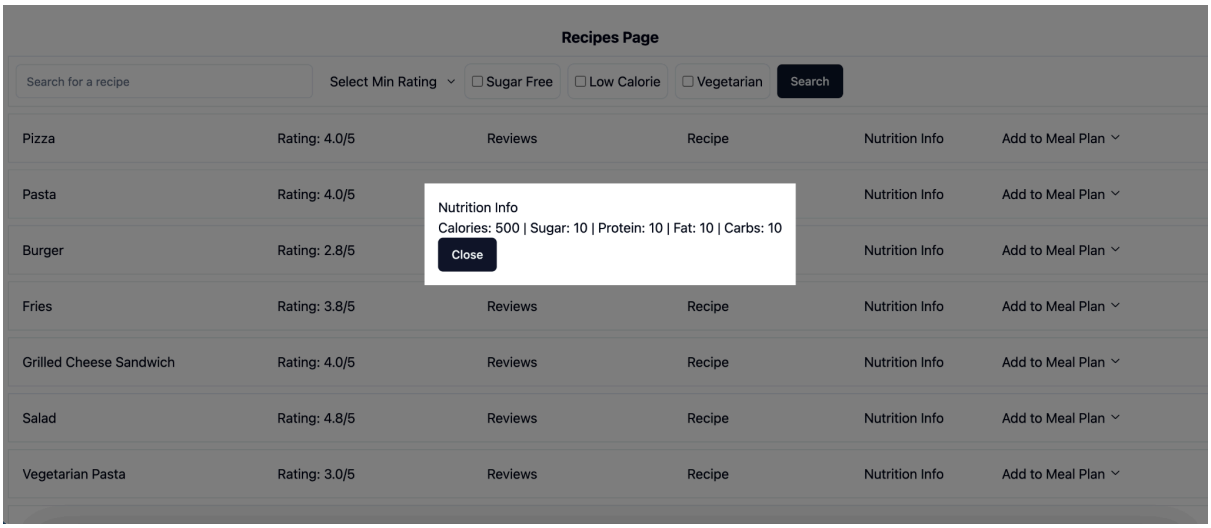
After:

Projection: get nutrition info from recipe

Before:

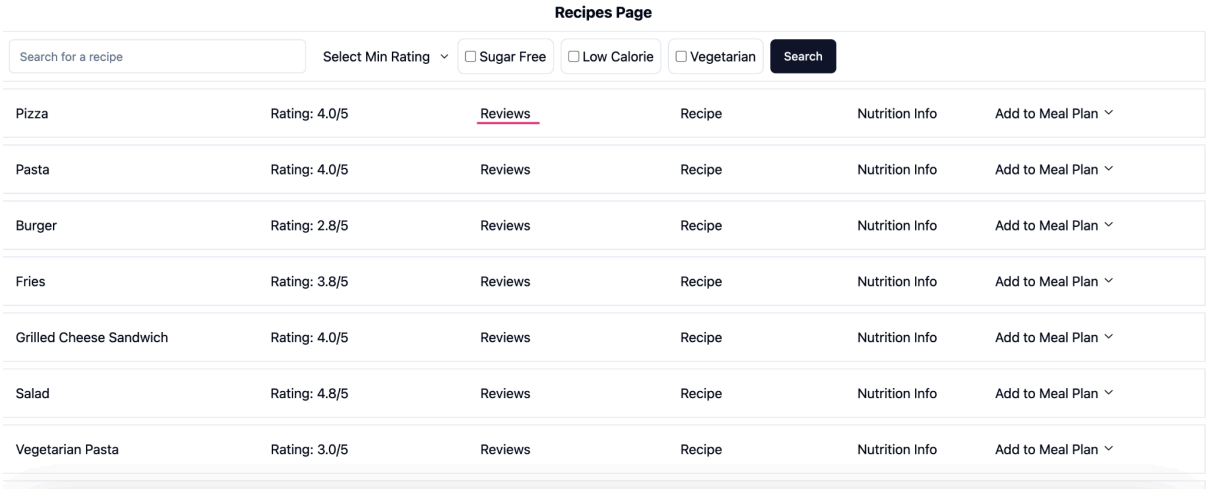


After:

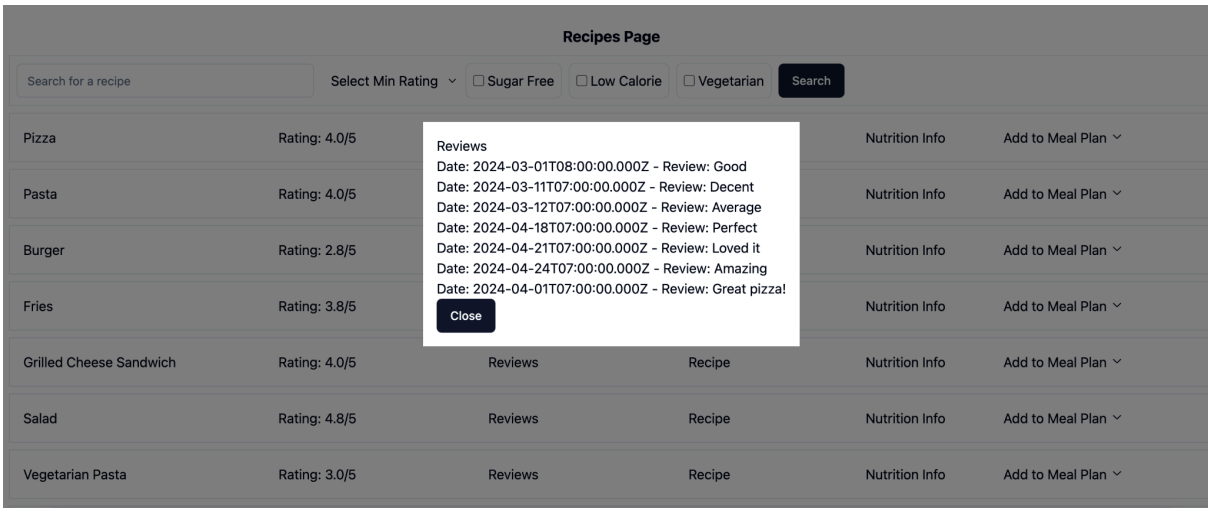


Join: get reviews

Before:

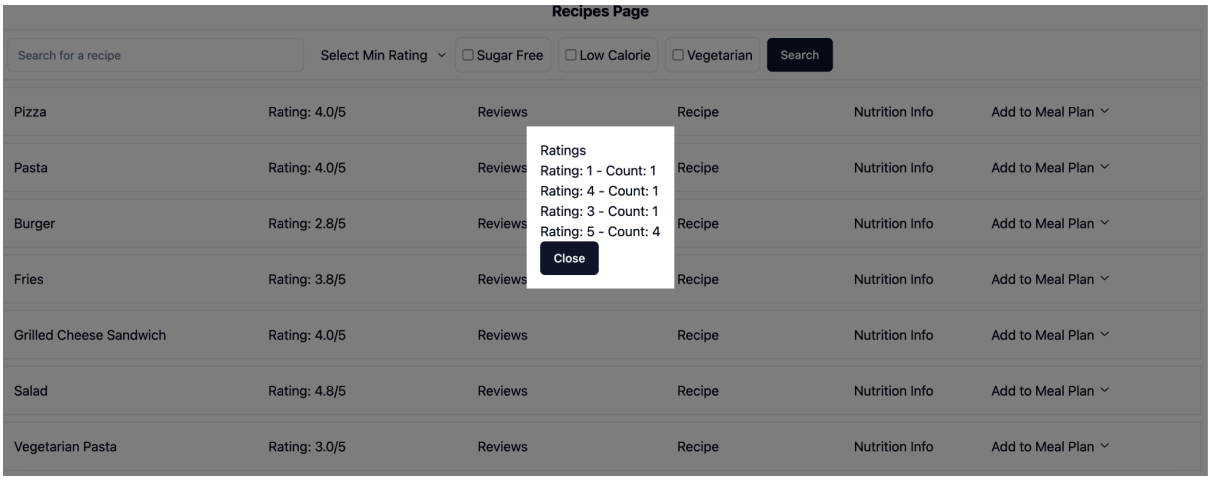


After:



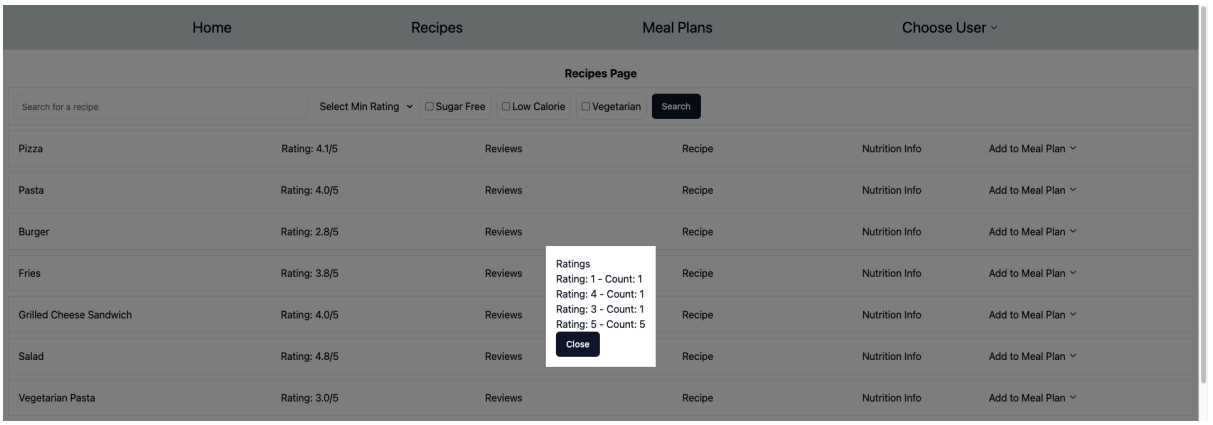
Aggregation with Group by: get recipe ratings, group by rating

Before:



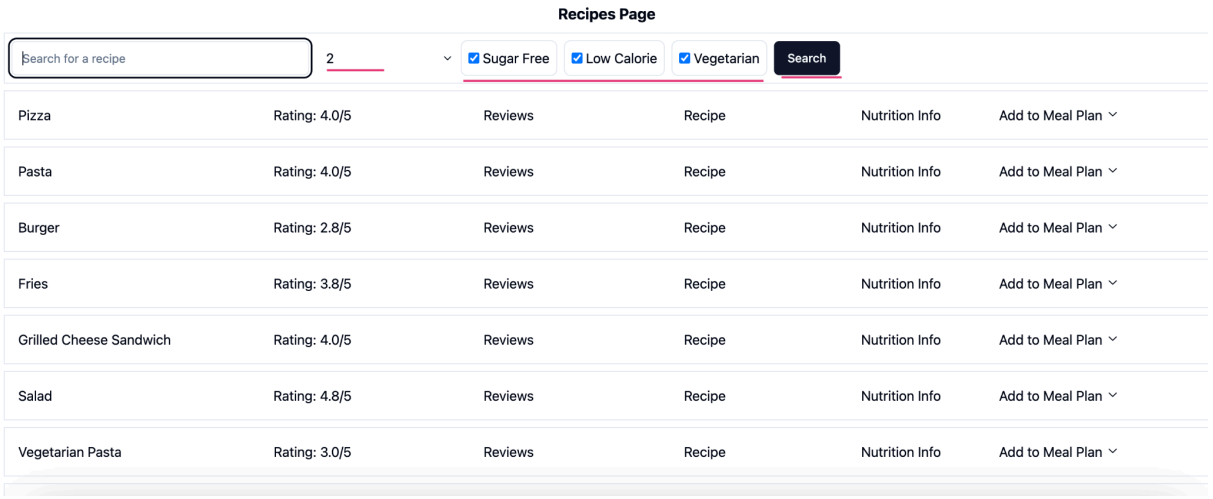
inserted a review with 5 stars

After:

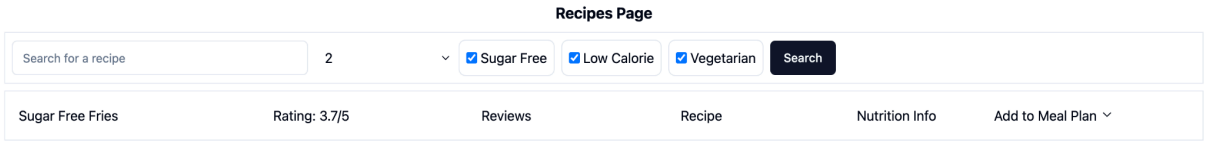


Aggregation with Having: Search that can apply filters, and min rating having score >= ?

Before:

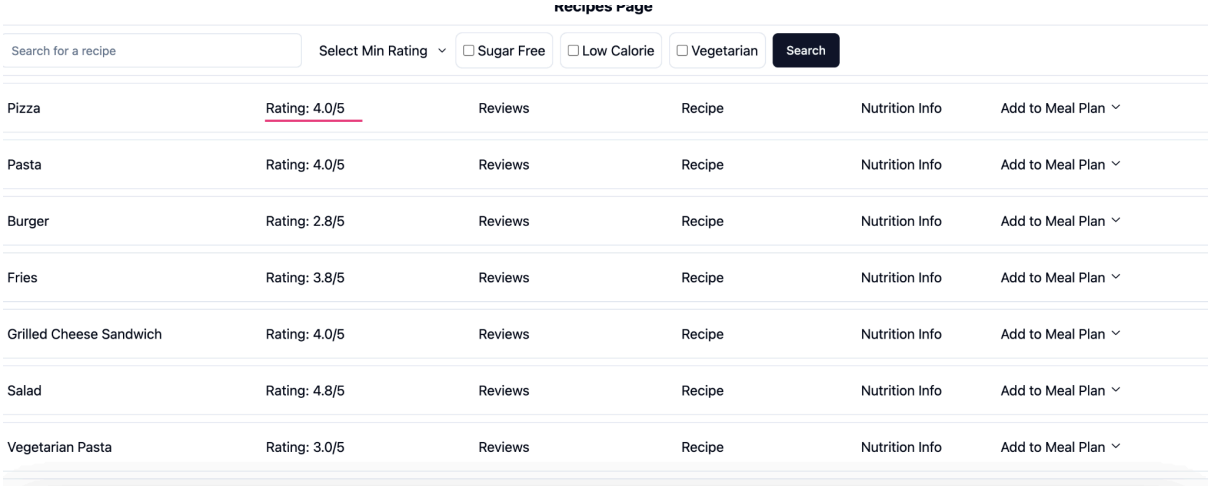


After:



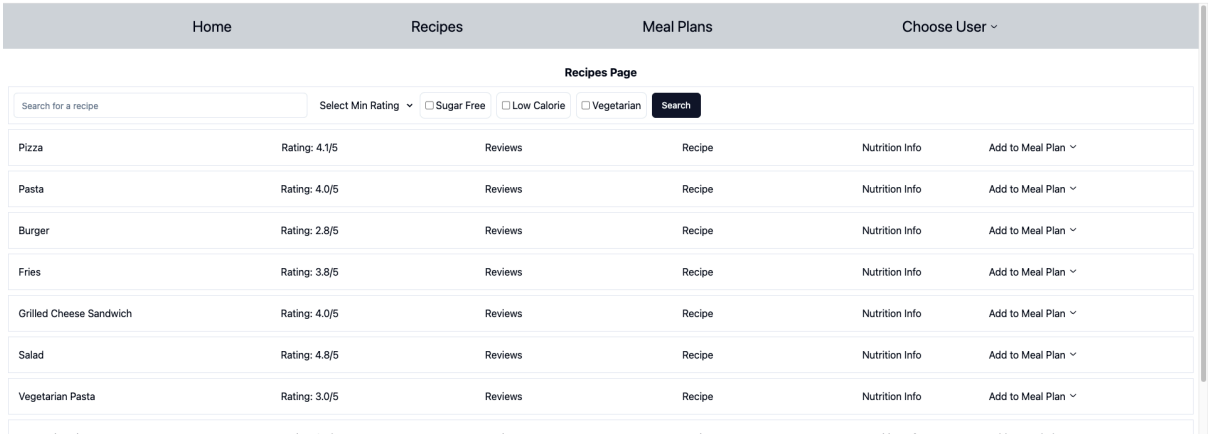
Nested Aggregation with GROUP BY: Show average rating for each recipe

Before:



inserted a review with 5 stars

After:



Division: Select users where they have reviewed every recipe

Triggered on page load in GUI:

Before:

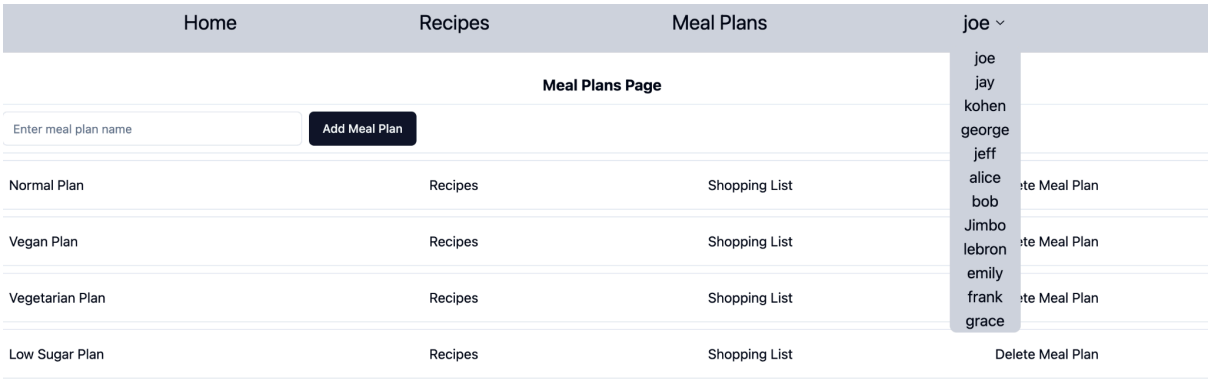
User table:

```
+--------+--------+-------------------+----------+
| userID | name   | email             | username |
+--------+--------+-------------------+----------+
|      1 | Joe    | joe@email.com     | joe      |
|      2 | Jay    | jay@email.com     | jay      |
|      3 | Kohen  | kohen@email.com   | kohen    |
|      4 | George | george@email.com  | george   |
|      5 | Jeff   | jeff@email.com    | jeff     |
|      6 | Alice  | alice@email.com   | alice    |
|      7 | Bob    | bob@email.com     | bob      |
|      8 | Jimbo  | jimbo@email.com   | Jimbo    |
|      9 | Lebron | lebron@email.com  | lebron   |
|     10 | Emily  | emily@email.com   | emily    |
|     11 | Frank  | frank@email.com   | frank    |
|     12 | Grace  | grace@email.com   | grace    |
+--------+--------+-------------------+----------+
12 rows in set (0.01 sec)
```

After query:

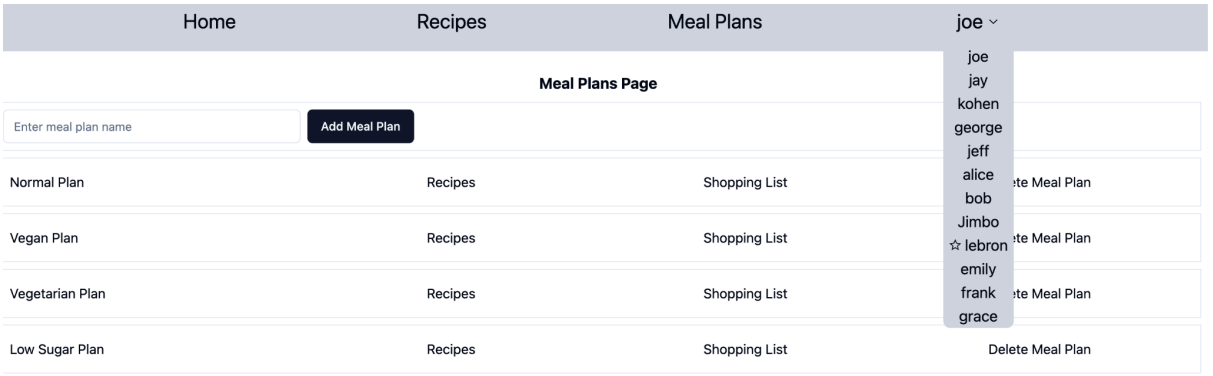
```
mysql> SELECT userID, name
    -> FROM User
    -> WHERE NOT EXISTS (
    ->     SELECT recipeID
    ->     FROM Recipe
    ->     WHERE NOT EXISTS (
    ->         SELECT *
    ->         FROM Review
    ->         WHERE Review.userID = User.userID
    ->         AND Review.recipeID = Recipe.recipeID
    ->     )
    -> );
+--------+--------+
| userID | name   |
+--------+--------+
|      9 | Lebron |
+--------+--------+
1 row in set (0.00 sec)
```

In GUI:
Before:



added review for lebron

After (star denotes super reviewer who has reviewed all recipes):



Schema changes:
NONE
Final Schema:
Primary keys are underlined, foreign keys are bolded.

- User(userID: int, name: varchar, email: varchar, username: varchar)
- Recipe(recipeID: int, name: varchar, category: varchar, instructions: varchar)
- Category(name)
- VegetarianRecipe(recipeID: int)
- LowCalorieRecipe(recipeID: int)
- SugarFreeRecipe(recipeID: int)
- Ingredient(name: varchar, isVegetarian: BOOLEAN)
- nutritionInfoRecipe(calories: int, sugar: int, proteinContent: int, fatContent: int, carbContent: int, **recipeID**: int)
- nutritionInfoIngredient(calories: int, sugar: int, proteinContent: int, fatContent: int, carbContent: int, **ingredientName**:varchar)
- MealPlan(mealPlanID: int, name: varchar, date: date, **userID**: int)
- ShoppingList(shoppingListID,mealPlanID,recipeID,ingredientName)
- Rating(ratingID: int, score: int, **userID: int, recipeID: int**)
- Review(reviewID: int, date: date, message: varchar, **userID: int, recipeID:** int)
- saves(**userID: int, recipeID: int**)
- recipeContains**(recipeID: int, ingredientName: varchar**)
- mealPlanContains(**mealPlanID: int, recipeID:int**)
- listContains(**shoppingListID:int, ingredientName: varcha**r)

SCREENSHOTS OF DATA PRESENT FROM MYSQL

TABLES:

USER:

userID	name	email	username
1	Joe	joe@email.com	joe
2	Jay	jay@email.com	jay
3	Kohen	kohen@email.com	kohen
4	George	george@email.com	george
5	Jeff	jeff@email.com	jeff
6	Alice	alice@email.com	alice
7	Bob	bob@email.com	bob
8	Jimbo	jimbo@email.com	Jimbo
9	Lebron	lebron@email.com	lebron
10	Emily	emily@email.com	emily
11	Frank	frank@email.com	frank
12	Grace	grace@email.com	grace

12 rows in set (0.01 sec)

Ingredient:

name	isVegetarian
Beef	0
Cheese	1
Chicken	0
Dough	1
Lettuce	1
Noodles	1
Olive Oil	1
Pepper	1
Pepperoni	0
Quinoa	1
Rice	1
Salt	1
Tomato	1

13 rows in set (0.00 sec)

Recipe: (selection done without instructions in order to conserve space)

recipeid	name
1	Pizza
2	Pasta
3	Burger
4	Fries
5	Grilled Cheese Sandwich
6	Salad
7	Vegetarian Pasta
8	Vegetarian Pizza
9	Vegetarian Burger
10	Quinoa
11	Low Calorie Pizza
12	Low Calorie Pasta
13	Sugar Free Pizza
14	Sugar Free Pasta
15	Sugar Free Burger
16	Sugar Free Fries
17	Sugar Free Sandwich
18	Tomato Soup
19	Cheese Pizza
20	Chicken Rice

20 rows in set (0.00 sec)

Mealplan:

mealPlanID	name	date	userID
1	Normal Plan	2024-03-01	1
2	Protein Plan	2024-03-02	2
3	Vegetarian Plan	2024-03-03	3
4	Vegan Plan	2024-03-04	4
5	Low Sugar Plan	2024-03-05	5
6	Vegan Plan	2024-04-01	1
7	Vegetarian Plan	2024-04-02	1
8	Low Sugar Plan	2024-04-03	1
9	Emily Plan	2024-04-18	10
10	Frank Plan	2024-04-19	11
11	Grace Plan	2024-04-20	12
12	Lebron Plan	2024-01-01	9

12 rows in set (0.00 sec)

Category:

name	category
Burger	Fast Food
Cheese Pizza	Italian
Chicken Rice	Clean
Fries	Fast Food
Grilled Cheese Sandwich	Sandwich
Low Calorie Pasta	Italian
Low Calorie Pizza	Italian
Pasta	Italian
Pizza	Italian
Quinoa	Grain
Salad	Salad
Sugar Free Burger	Fast Food
Sugar Free Fries	Fast Food
Sugar Free Pasta	Italian
Sugar Free Pizza	Italian
Sugar Free Sandwich	Sandwich
Tomato Soup	Soup
Vegetarian Burger	Fast Food
Vegetarian Pasta	Italian
Vegetarian Pizza	Italian

20 rows in set (0.00 sec)

VegetarianRecipe:

recipeID
4
6
7
8
9
10
16
18
19

9 rows in set (0.01 sec)

LowCalorieRecipe:

recipeID
6
7
8
9
11
12
13
14
15
16
17
18
20

13 rows in set (0.00 sec)

SugarFreeRecipe:

recipeID
10
13
14
15
16
17

6 rows in set (0.01 sec)

NutritionInfoRecipe:

recipeID	calories	sugar	proteinContent	fatContent	carbContent
1	500	10	10	10	10
2	500	20	20	20	20
3	300	30	30	30	30
4	400	40	40	40	40
5	500	50	50	50	50
6	250	15	5	5	30
7	350	10	12	15	40
8	400	20	10	20	45
9	450	25	15	25	50
10	150	0	8	2	25
11	280	5	10	10	30
12	220	4	9	8	35
13	300	0	11	12	40
14	180	0	7	5	20
15	350	0	20	15	30
16	200	0	2	10	25
17	250	0	9	8	28
18	100	2	3	1	15
19	350	5	15	10	40
20	400	10	30	15	100

20 rows in set (0.00 sec)

NutritionInfoIngredient:

ingredientName	calories	sugar	proteinContent	fatContent	carbContent
Beef	400	40	40	40	40
Cheese	100	1	7	8	2
Chicken	500	50	50	50	50
Dough	100	2	2	2	100
Lettuce	1	0	0	0	10
Noodles	200	20	20	20	20
Olive Oil	100	2	0	50	0
Pepper	1	2	1	0	9
Pepperoni	200	5	10	50	2
Quinoa	100	0	2	0	50
Rice	100	10	10	10	10
Salt	300	30	30	30	30
Tomato	20	3	1	0	5

13 rows in set (0.00 sec)

RecipeContains:

recipeID	ingredientName
2	Beef
3	Beef
1	Cheese
1	Dough
2	Noodles
1	Olive Oil
1	Pepperoni
1	Salt
1	Tomato

9 rows in set (0.00 sec)

Mealplancontains:

mealPlanID	recipeID
1	1
1	2
2	3
3	6
6	6
7	8
4	10
6	10
5	11
8	13
8	14
8	15
12	20

13 rows in set (0.00 sec)

ShoppingList:

shoppingListID	mealPlanID
1	1
2	1
3	2
4	3
5	4
6	5
7	6
8	6
9	7
10	8
11	8
12	8
13	12
14	12

14 rows in set (0.00 sec)

listContains:

shoppingListID	ingredientName
2	Beef
12	Beef
2	Chicken
1	Dough
6	Lettuce
1	Noodles
4	Noodles
11	Noodles
1	Olive Oil
1	Pepperoni
6	Quinoa
3	Rice
1	Salt
1	Tomato
7	Tomato
9	Tomato
10	Tomato

17 rows in set (0.00 sec)

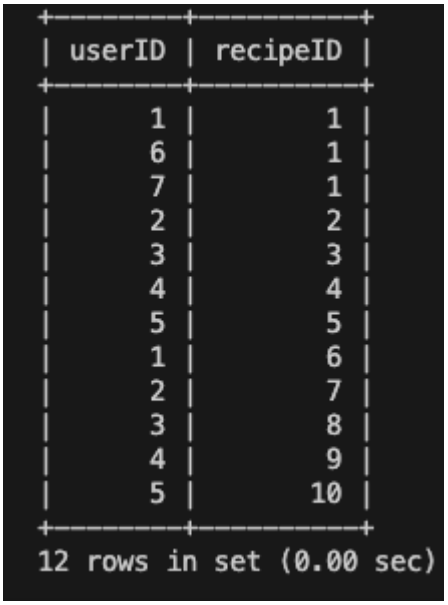
Rating:

ratingID	score	userID	recipeID
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	5	1	6
7	4	2	7
8	3	3	8
9	2	4	9
10	1	5	10
11	4	6	1
12	3	7	1
13	5	6	2
14	4	7	2
16	2	8	3
17	5	6	4
18	4	7	4
20	2	8	5
21	5	6	6
22	4	7	6
24	2	8	7
25	5	6	8
26	4	7	8
28	2	8	9
29	5	6	10
30	4	7	10
32	2	8	11
33	5	6	12
34	4	7	12
36	2	8	13
37	5	6	14
38	4	7	14
40	2	8	15
41	5	6	16
42	4	7	16
44	2	8	17
45	5	6	18
46	4	7	18
48	2	8	19
49	5	10	1
50	4	10	2
51	3	10	3
52	5	11	1
53	4	11	2
54	3	11	3
55	5	12	1
56	4	12	2
57	3	12	3
58	5	9	1
59	5	9	2
60	3	9	3
61	2	9	4
62	5	9	5
63	5	9	6
64	3	9	7
65	2	9	8
66	5	9	9

REVIEW:

```
mysql> select * from review;
+----------+------------+----------------------------------------+--------+----------+
| reviewID | date       | message                                | userID | recipeID |
+----------+------------+----------------------------------------+--------+----------+
|        1 | 2024-03-01 | Good                                   |      1 |        1 |
|        2 | 2024-03-02 | Bad                                    |      2 |        2 |
|        3 | 2024-03-03 | Amazing                                |      3 |        3 |
|        4 | 2024-03-04 | Terrible                               |      4 |        4 |
|        5 | 2024-03-05 | Spectacular                            |      5 |        5 |
|        6 | 2024-03-06 | Okay                                   |      1 |        6 |
|        7 | 2024-03-07 | Not bad                                |      2 |        7 |
|        8 | 2024-03-08 | Great                                  |      3 |        8 |
|        9 | 2024-03-09 | Disappointing                          |      4 |        9 |
|       10 | 2024-03-10 | Outstanding                            |      5 |       10 |
|       11 | 2024-03-11 | Decent                                 |      6 |        1 |
|       12 | 2024-03-12 | Average                                |      7 |        1 |
|       13 | 2024-03-13 | Loved it                               |      6 |        2 |
|       14 | 2024-03-14 | Pretty good                            |      7 |        2 |
|       16 | 2024-03-16 | Could be better                        |      8 |        3 |
|       17 | 2024-03-17 | Delicious                              |      6 |        4 |
|       18 | 2024-03-18 | Tasty                                  |      7 |        4 |
|       20 | 2024-03-20 | Meh                                    |      8 |        5 |
|       21 | 2024-03-21 | Fantastic                              |      6 |        6 |
|       22 | 2024-03-22 | Good choice                            |      7 |        6 |
|       24 | 2024-03-24 | Not bad                                |      8 |        7 |
|       25 | 2024-03-25 | Great!                                 |      6 |        8 |
|       26 | 2024-03-26 | Yummy                                  |      7 |        8 |
|       28 | 2024-03-28 | Disliked it                            |      8 |        9 |
|       29 | 2024-03-29 | Healthy                                |      6 |       10 |
|       30 | 2024-03-30 | Nutritious                             |      7 |       10 |
|       32 | 2024-04-01 | Too light                              |      8 |       11 |
|       33 | 2024-04-02 | No sugar, great                        |      6 |       12 |
|       34 | 2024-04-03 | Miss the sugar                         |      7 |       12 |
|       36 | 2024-04-05 | Lacking flavor                         |      8 |       13 |
|       37 | 2024-04-06 | Nice                                   |      6 |       14 |
|       38 | 2024-04-07 | Not for me                             |      7 |       14 |
|       40 | 2024-04-08 | Bland                                  |      8 |       15 |
|       41 | 2024-04-10 | Crispy                                 |      6 |       16 |
|       42 | 2024-04-11 | Too crispy                             |      7 |       16 |
|       44 | 2024-04-13 | Too plain                              |      8 |       17 |
|       45 | 2024-04-11 | Refreshing                             |      6 |       18 |
|       46 | 2024-04-14 | Lacks depth                            |      7 |       18 |
|       48 | 2024-04-20 | Too cheesy                             |      8 |       19 |
|       49 | 2024-04-18 | Perfect                                |     10 |        1 |
|       50 | 2024-04-19 | Great taste                            |     10 |        2 |
|       51 | 2024-04-20 | Good                                   |     10 |        3 |
|       52 | 2024-04-21 | Loved it                               |     11 |        1 |
|       53 | 2024-04-22 | Nice flavor                            |     11 |        2 |
|       54 | 2024-04-23 | Decent                                 |     11 |        3 |
|       55 | 2024-04-24 | Amazing                                |     12 |        1 |
|       56 | 2024-04-25 | Tasty                                  |     12 |        2 |
|       57 | 2024-04-26 | Not bad                                |     12 |        3 |
|       58 | 2024-04-01 | Great pizza!                           |      9 |        1 |
|       59 | 2024-04-02 | Pasta was delicious.                   |      9 |        2 |
|       60 | 2024-04-03 | Burger could be better.                |      9 |        3 |
|       61 | 2024-04-04 | Fries were too salty.                  |      9 |        4 |
|       62 | 2024-04-05 | Loved the grilled cheese sandwich.     |      9 |        5 |
|       63 | 2024-04-06 | Salad was fresh and tasty.             |      9 |        6 |
|       64 | 2024-04-07 | Vegetarian pasta was okay.             |      9 |        7 |
|       65 | 2024-04-08 | Vegetarian pizza lacked flavor.        |      9 |        8 |
|       66 | 2024-04-09 | Vegetarian burger was excellent.       |      9 |        9 |
|       67 | 2024-04-10 | Quinoa was cooked perfectly.           |      9 |       10 |
```

Saves:



userID	recipeID
1	1
6	1
7	1
2	2
3	3
4	4
5	5
1	6
2	7
3	8
4	9
5	10

12 rows in set (0.00 sec)