Software Test Report

Overview of Testing Included in this Document

This document includes detailed reports for various testing methodologies employed during the development and validation of the application. These are:

1. Manual Test Report

- Tools Used: Expo and Android Studio
- This section documents the results of manual testing conducted on critical components of the application, including the Meal Plan Page, Recipes Page, and Profile Page.
- The testing focused on verifying functionality, usability, and seamless navigation between components.
- Key functionalities, such as adding recipes, managing meal plans, and updating user profiles, were validated to ensure they meet user requirements and deliver a smooth user experience.

2. API Test Report

- Tools Used: Postman
- This section outlines the testing of backend API endpoints using Postman.
- The API testing ensured that endpoints for user authentication, data retrieval (e.g., recipes, meal plans, pantry items), and data manipulation (e.g., adding, deleting) function as expected.
- Tests included validating response formats, HTTP status codes, and handling of edge cases, such as invalid input or unauthorized access.

3. Unit Test Report

- Tools Used: Jest
- This section includes the results of unit testing conducted on critical frontend components.
- Each component was tested in isolation to validate its behavior, logic, and error handling.
- Tests ensured that user inputs, state updates, and component rendering function correctly across various scenarios.

Manual Test Report:

1. Login Component:

Test Type	Test No	Description	Result	Requirements	Test Steps	Expected Results	Actual Results
Functionality	1	Verify that the user can log in with valid credentials.	Passed	As a user, I need to log in using valid email and password	Enter valid email and password and tap "Sign In." Verify redirection to Homepage	User is logged in successfully and redirected to the Homepage.	Verified successful login and redirection to Homepage.
Functionality	2	Verify login fails with invalid credentials.	Passed	As a user, I should not be able to log in using incorrect credentials.	Enter invalid email/password and tap "Sign In."	Login Fails	Verified Login Fails
Usability	3	Verify error messages for empty email and password fields.	Passed	As a user, I need clear feedback if required fields are empty.	Leave email/password blank and tap "Sign In." Verify appropriate error messages appear for each field.	Error messages "Email is required" and "Password is required" are displayed.	Verified correct error messages appeared.
Usability	4	Verify the skip-login button navigates to the Homepage.	Passed	As a developer, I need to skip the login step during testing.	Tap on the "Skip Login (Dev)" button. Verify that the page redirects correctly to the Homepage.	User is redirected to the Homepage without requiring login.	Navigation worked as expected

Comments:
Requires message to let the user know if invalid/incorrect password or email is used
Needs Reset Password Finctionality

2. Registration Component

Test Type	Test No	Description	Result	Requirements	Test Steps	Expected Results	Actual Results
Functionality	1	Verify that the user can register with valid details.	Partial Pass	As a user, I need to register with valid information	Enter valid details for all fields and tap "Create Account." Verify successful registration and redirection to Login page.	User is registered successfully and redirected to the Login page. .	Verified successful registration but not redirected to Login page. .
Functionality	2	Verify registration fails with invalid email format..	Passed	As a user, I should see an error if the email format is invalid.	Enter an invalid email format and tap "Create Account." Verify appropriate error message displays.	Error message "Please enter a valid email" is displayed.	Verified correct error message appeared.
Functionality	3	Verify error messages for mismatched passwords.	Passed	As a user, I need to confirm my password correctly.	Enter a valid password and a mismatched confirm password. Tap "Create Account." Verify error message displays.	Error message "Passwords do not match" is displayed. | Verified d.	Verified correct error message appeared. .
Usability	4	Verify error messages for empty required fields.	Passed	As a user, I need clear feedback if required fields are empty.	Leave any required field empty and tap "Create Account." Verify appropriate error messages appear for each field.	Error messages for empty fields are displayed.	Verified all Error messages for empty fields are displayed.

3. Homepage Component:

Test Type	Test No	Description	Result	Requirements	Test Steps	Expected Results	Actual Results
Functionality	1	Verify that the user's profile avatar and name are displayed correctly.	Passed	As a user, I need to see my profile avatar and name when logged in.	Log in to the app and navigate to the Homepage. Verify that the avatar and name match the logged-in user.	The user's avatar and name are displayed correctly based on the logged-in profile.	Verified that the user's avatar and name appeared accurately.
Functionality	2	Verify navigation from Homepage to Fridge & Pantry page.	Passed	As a user, I need to navigate to the 1. Fridge & Pantry 2. Meal Plan Page 3. Recipes from the Homepage.	Tap on the 1. Fridge & Pantry 2. Meal Plan Page 3. Recipes button. Verify that the page redirects correctly to the pages	User is redirected to the page without errors.	Navigation worked as expected.

4. Logout Button Component

Test Type	Test No	Description	Result	Requirements	Test Steps	Expected Results	Actual Results
Functionality	1	Verify that the user can Logout Using Logout Button from any page	Passed	As a user, I need to be able to Logout from any page that has the Logout Button	Log in to the app and navigate to any page consisting the logout Button. Click the Logout Button	The user is Logged out and is redirected to the Login Page.	Worked as expected

5. Fridge & Pantry Page:

Test Type	Test No	Description	Result	Requirements	Test Steps	Expected Results	Actual Results
Functionality	1	Verify that fridge/Pantry items are fetched and displayed correctly.	Passed	As a user, I need to see all my fridge items when I open the Fridge and Pantry page.	Open the Fridge & Pantry page. Verify that all fridge/Pantry items are displayed with correct details (name, quantity, calories).	Fridge/Pantry items are displayed correctly with accurate details.	Verified that fridge/Pantry items appeared with correct details.
Functionality	2	Verify that the user can add a new item to the fridge./pantry .	Passed	As a user, I need to add a new item to my fridge/Pantry	Search for an item. Select an item and add a quantity and measure. Verify that the item is added to the fridge/pantry list with correct details.	Item is added to the fridge/pantry with correct details.	Verified that the item was added to the fridge/pantry successfully.
Functionality	3	Verify that the user can delete an item from the fridge/pantru.	Passed	As a user, I need to delete an item from my fridge/pantry	Select an item from the fridge and delete it. Verify that the item is removed from the fridge/pantry list.	Item is removed from the list.	Verified that the item was deleted successfully
Usability	4	Verify that item details are displayed in the modal before adding to	Passed	As a user, I need to see the details of an item before adding it to my fridge or pantry.	Search for an item. Select an item to open the modal. Verify that the modal displays all relevant item	Item details are displayed in the modal correctly.	Verified that the modal displayed correctly

Test Type		Description	Result	Requirements	Test Steps	Expected Results	Actual Results
		fridge/pantry.			details (name, brand, calories, measures, etc.).		
Usability	5	Verify that the search functionality works	Passed	As a user, I need to search for an item	Enter a search query. Verify that the search results display relevant items for both fridge and pantry.	Search results display relevant items from fridge and pantry.	Verified that search results appeared correctly.
Functionality	6	Verify that the user can edit an item from the fridge/pantru.	Failed	As a user, I need to update an item from my fridge/pantry	Select an item from the fridge and edit it. Verify that the item is updated	Item is Updated	Update item functionality not available

Comments:
Could add feedback for duplicate items ie correct handle addItem function
Needs Update Item functionality
Performance can be enhanced by using filters

6. Meal Plan Page

Test Type	Test No	Description	Result	Requirements	Test Steps	Expected Results	Actual Results
Functionality	1	Verify that the calendar displays and allows date selection.	Passed	As a user, I need to see a calendar and select a date to view the meal plan.	Open the Meal Plan page. Verify that the calendar is displayed and that selecting a date redirects to the meal plan for that date.	Calendar displays and date selection works correctly.	Verified that the calendar displayed and date selection worked as expected.

Functionality	2	Verify that meal data is fetched and displayed for the selected date.	Passed	As a user, I need to view my meal data for the selected date.	Select a date from the calendar. Verify that the meal data (breakfast, lunch, dinner) is fetched and displayed for that date.	Meal data for the selected date is displayed correctly.	Verified that meal data was displayed for the selected date.
Functionality	3	Verify that a new meal can be added to a specific meal type.	Passed	As a user, I need to add a new meal to breakfast, lunch, or dinner.	Search for a meal or product. Select the meal and add it to breakfast, lunch, or dinner. Verify that the meal is added to the correct section for the selected date.	Meal is added to the correct section for the selected date.	Verified that the meal was added to the correct section successfully.
Functionality	4	Verify that a meal can be deleted from a specific meal type.	Passed	As a user, I need to delete a meal from breakfast, lunch, or dinner.	Select a meal from breakfast, lunch, or dinner. Delete the meal. Verify that the meal is removed from the correct section for the selected date.	Meal is removed from the correct section for the selected date.	Verified that the meal was deleted from the correct section successfully.
Usability	5	Verify that total calories for the day are calculated	Passed	As a user, I need to see the total calories for all meals on a selected date.	Select a date from the calendar. Verify that the total calories for breakfast,	Total calories are calculated and displayed correctly.	Verified that the total calories were calculated

Test Type	Test No	Description	Result	Requirements	Test Steps	Expected Results	Actual Results
		and displayed.			lunch, and dinner are calculated and displayed correctly.		and displayed correctly.
Usability	6	Verify that the search functionality works for adding new meals.	Passed	As a user, I need to search for meals or products to add to my meal plan.	Enter a search query. Verify that the search results display relevant meals or products that can be added to the meal plan.	Search results display relevant meals or products.	Verified that search results appeared correctly.

7. **Recipes Page:**

Test Type	Test No	Description	Result	Requirements	Test Steps	Expected Results	Actual Results
Functionality	1	Verify that recipes are fetched and displayed correctly.	Passed	As a user, I need to see all my saved recipes when I open the Recipes page.	Open the Recipes page. Verify that all recipes are displayed with correct details (name, calories).	Recipes are displayed correctly with accurate details.	Verified that recipes appeared with correct details.
Functionality	2	Verify that the user can add a new recipe with valid details.	Passed	As a user, I need to add a new recipe with a title, ingredients, and instructions.	Tap on the "+" button. Add a title, ingredients, and instructions. Save the recipe. Verify that the recipe is added to the list.	Recipe is added to the list with correct details.	Verified that the recipe was added successfully.

Functionality	3	Verify that the user can delete a recipe.	Passed	As a user, I need to delete a recipe I no longer need.	Select a recipe. Tap on the "Delete" button. Verify that the recipe is removed from the list.	Recipe is removed from the list.	Verified that the recipe was deleted successfully.
Usability	4	Verify that the search functionality works for adding ingredients to recipe.	Passed	As a user, I need to search for ingredients to add to a recipe.	Enter a search query in the ingredient search bar. Verify that the search results display relevant ingredients that can be added to the recipe.	Search results display relevant ingredients.	Verified that search results appeared correctly.
Usability	6	Verify that total calories are calculated correctly for a recipe.	Passed	As a user, I need to see the total calories for all ingredients in a recipe.	Add ingredients to a recipe. Verify that the total calories are calculated and displayed correctly.	Total calories are calculated and displayed correctly.	Verified that total calories were calculated correctly.
Functionality	7	Verify that the user can Update a recipe.	Fail	As a user, I need to Update a recipe I no longer need.	Select a recipe. Tap on the "Delete" button. Verify that the recipe is Updated	Recipe is updated	Verified that the recipe was updated successfully.

8. Profile Page:

Test Type	Test No	Description	Result	Requirements	Test Steps	Expected Results	Actual Results
Test Type	Test Number	Test Case Description	Results	Requirements to be Tested	Test Steps	Expected Result	Actual Result
Functionality	1	Verify that user profile details are displayed.	Passed	As a user, I need to see my profile details on the Profile page.	Open the Profile page. Verify that the profile details (name, email, avatar) are displayed.	Profile details are displayed correctly.	Verified that profile details appeared correctly.
Functionality	2	Verify that the user can edit their profile.	Passed	As a user, I need to edit my profile details.	Tap on the "Edit Profile" button. Update the profile details. Save the changes. Verify that the updated details are displayed on the Profile page.	Profile details are updated and displayed correctly.	Verified that profile updates were saved and displayed correctly.

API Testing:

Login Component (UserController):
Incorrect Password

```
1  {
2      "email": "nidhidesai4848@gmail.com",
3      "password": "password"
4  }
```

Body   Cookies   Headers (15)   Test Results   🕓   400 Bad Request • 557 ms

Pretty   Raw   Preview   Visualize   JSON ∨   ⇄

```
1  {
2      "error": "Incorrect email or password"
3  }
```

```
1  {
2      "email": "nidhidesai4848",
3      "password": "Athena44"
4  }
```

Body   Cookies   Headers (15)   Test Results   🕓

Pretty   Raw   Preview   Visualize   JSON ∨   ⇄

```
1  {
2      "error": "Incorrect email or password"
3  }
```

Valid User Credentials

```
1  {
2      "email": "nidhidesai4848@gmail.com",
3      "password": "Athena44"
4  }
```

Body   Cookies   Headers (15)   Test Results   🕓   200 OK • 797 ms • 668 B • 🌐 🔒 e.g.

Pretty   Raw   Preview   Visualize   JSON ∨   ⇄   🔗 🗐

```
1  {
2      "firstName": "Nidhi",
3      "lastName": "Desai",
4      "email": "nidhidesai4848@gmail.com",
5      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
          eyJfaWQiOiI2NzUwODA3NjEyOGEyMWFkMWU4ZWU3ZGIiLCJpYXQiOjE3MzQwOTcyNzQsImV4cCI6MTczNDE4MzY3NH0.
          maUEhWR8wGiMyM11xBAbKBiPqm9rXYTKOpaDOVy9W38"
```

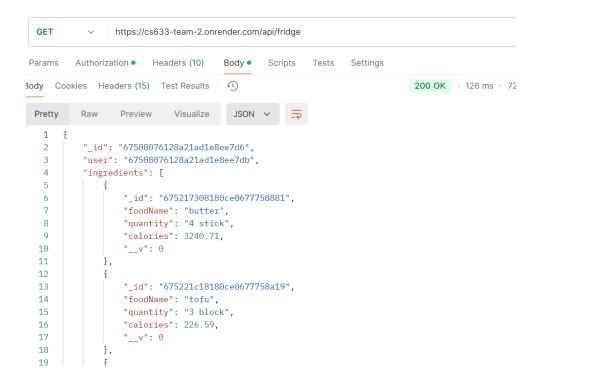**Registration Component (UserController):**
Valid Credentials:

```
1  {
2      "firstName": "Nidhi",
3      "lastName": "Desai",
4      "email": "nidhird@bu.edu",
5      "password": "password123"
6  }
```

Cookies   Headers (15)   Test Results   ⟲   200 OK • 816 ms • 639 B • 🔒 e.g. ⋯

Pretty   Raw   Preview   Visualize   JSON ⌄   ⇄   🔗 ⎘ 🔍

```
1  {
2      "email": "nidhird@bu.edu",
3      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
           eyJfaWQiOiI2NzVjM2FkZWVjNDljZWMwODhmNjJhNDIiLCJpYXQiOjE3MzQwOTc2MzAsImV4cCI6MTczNDE4NDAzMH0.
           OEKwpMq0o-LWl7NDTy6My5y4GOiTF8hKNTfLayrWpwQ"
4  }
```

Invalid:

```
1  {
2      "firstName": "Nidhi",
3      "lastName": "Desai",
4      "email": "nidhird"
5  }
```

Body   Cookies   Headers (15)   Test Results   ⟲   400 Bad Request • 77

Pretty   Raw   Preview   Visualize   JSON ⌄   ⇄

```
1  {
2      "msg": "All fields are required"
3  }
```

Email already exists:

```
1  {
2      "firstName": "Nidhi",
3      "lastName": "Desai",
4      "email": "nidhird",
5      "password": "password123"
6  }
```

Body   Cookies   Headers (15)   Test Results   ⟲   400 Bad Request • 100 ms

Pretty   Raw   Preview   Visualize   JSON ⌄   ⇄

```
1  {
2      "error": "Email is already taken"
3  }
```

Fridge & Pantry (fridgeController & PantryController)
Add ingredient to Users Fridge

```
1  {
2    "foodName": "Cheese",
3    "quantity": "1",
4    "calories": "120"
5  }
```

Body   Cookies   Headers (15)   Test Results   🕐   200 OK · 598 ms · 626 B

Pretty   Raw   Preview   Visualize   JSON ∨   ⇄

```
1  {
2      "_id": "67508076128a21ad1e8ee7d6",
3      "user": "67508076128a21ad1e8ee7db",
4      "ingredients": [
5          "675217308180ce0677758881",
6          "675221c18180ce0677758a19",
7          "6753097e72bf8d2ff25ca1ca",
8          "675c44277e791d862888211a"
9      ],
10     "createdAt": "2024-12-04T16:16:54.778Z",
11     "updatedAt": "2024-12-13T14:26:47.927Z",
```

Postbot
Ctrl Alt P

Get Users Fridge Items:

GET   ∨   https://cs633-team-2.onrender.com/api/fridge

Params   Authorization ●   Headers (10)   Body ●   Scripts   Tests   Settings

Body   Cookies   Headers (15)   Test Results   🕐   200 OK · 126 ms · 72

Pretty   Raw   Preview   Visualize   JSON ∨   ⇄

```
1  {
2      "_id": "67508076128a21ad1e8ee7d6",
3      "user": "67508076128a21ad1e8ee7db",
4      "ingredients": [
5          {
6              "_id": "675217308180ce0677758881",
7              "foodName": "butter",
8              "quantity": "4 stick",
9              "calories": 3240.71,
10             "__v": 0
11         },
12         {
13             "_id": "675221c18180ce0677758a19",
14             "foodName": "tofu",
15             "quantity": "3 block",
16             "calories": 226.59,
17             "__v": 0
18         },
19         {
```

Delete Ingredient:

POST ⌄ https://cs633-team-2.onrender.com/api/fridge/deleteIngredient

Params   Authorization ●   Headers (10)   Body ●   Scripts   Tests   Settings

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL   JSON ⌄

```
1  {
2      "ingredientId":"675217308180ce0677758881"
3  }
```

Body   Cookies   Headers (15)   Test Results   🕒   200 OK

Pretty   Raw   Preview   Visualize   JSON ⌄   ⇥

```
1  {
2      "_id": "67508076128a21ad1e8ee7d6",
3      "user": "67508076128a21ad1e8ee7db",
4      "ingredients": [
5          "675221c18180ce0677758a19",
6          "6753097e72bf8d2ff25ca1ca",
7          "675c44277e791d862888211a"
8      ],
9      "createdAt": "2024-12-04T16:16:54.778Z",
10     "updatedAt": "2024-12-13T17:03:53.197Z",
11     "__v": 21
12 }
```

Postbot
Ctrl Alt P

Get users pantry items:

GET ⌄ https://cs633-team-2.onrender.com/api/pantry

Params   Authorization ●   Headers (10)   Body ●   Scripts   Tests   Settings

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL   JSON ⌄

```
1  {
```

Body   Cookies   Headers (15)   Test Results   🕒   200 OK • 33.78 s • 75

Pretty   Raw   Preview   Visualize   JSON ⌄   ⇥

```
1  {
2      "_id": "67508076128a21ad1e8ee7d9",
3      "user": "67508076128a21ad1e8ee7db",
4      "ingredients": [
5          {
6              "_id": "67521af18180ce067775893b",
7              "foodName": "chocolate chip",
8              "quantity": "10 oz (approx 60 pcs)",
9              "calories": 1360.8,
10             "__v": 0
11         },
12         {
13             "_id": "6753131872bf8d2ff25ca299",
14             "foodName": "flour",
15             "quantity": "1 cup",
```

Add ingredients to users pantry:

POST ∨ https://cs633-team-2.onrender.com/api/pantry/addIngredient

Params   Authorization ●   Headers (10)   **Body** ●   Scripts   Tests   Settings

○ none   ○ form-data   ○ x-www-form-urlencoded   ● raw   ○ binary   ○ GraphQL   **JSON** ∨

```
1  {
2    "foodName": "Pasta",
3    "quantity": "1 box",
4    "calories": "800"
5  }
```

Body   Cookies   Headers (15)   Test Results   ⟲   200 OK   •

Pretty   Raw   Preview   Visualize   JSON ∨   ⇄

```
1  {
2      "_id": "67508076128a21ad1e8ee7d9",
3      "user": "67508076128a21ad1e8ee7db",
4      "ingredients": [
5          "67521af18180ce067775893b",
6          "6753131872bf8d2ff25ca299",
7          "6753132672bf8d2ff25ca29f",
8          "6753133d72bf8d2ff25ca2a5",
9          "675c6592b8a0d86f558dc094"
10     ],
11     "createdAt": "2024-12-04T16:16:54.867Z",
```

┌─────────────────┐
│ Postbot   │
│ Ctrl Alt P   │
└─────────────────┘

Delete item:

POST ∨ https://cs633-team-2.onrender.com/api/pantry/deleteIngredient

Params   Authorization ●   Headers (10)   **Body** ●   Scripts   Tests   Settings

○ none   ○ form-data   ○ x-www-form-urlencoded   ● raw   ○ binary   ○ GraphQL   **JSON** ∨

```
1  {
2      "ingredientId":"67508076128a21ad1e8ee7d9"
3  }
```

Body   Cookies   Headers (15)   Test Results   ⟲   200 OK   •   100 ms   •   62

Pretty   Raw   Preview   Visualize   JSON ∨   ⇄

```
3      "user": "67508076128a21ad1e8ee7db",
4      "ingredients": [
5          "67521af18180ce067775893b",
6          "6753131872bf8d2ff25ca299",
7          "6753132672bf8d2ff25ca29f",
8          "6753133d72bf8d2ff25ca2a5",
9          "675c6592b8a0d86f558dc094"
10     ],
11     "createdAt": "2024-12-04T16:16:54.867Z",
12     "updatedAt": "2024-12-13T17:02:20.204Z",
13     "__v": 12
14 }
```

┌─────────────────┐
│ Postbot   │
│ Ctrl Alt P   │
└─────────────────┘

Recipe Page (recipeController):
Add Recipe

POST ⌄ https://cs633-team-2.onrender.com/api/recipe/   Send ⌄

Params   Authorization ●   Headers (10)   Body ●   Scripts   Tests   Settings   Cookies

Body   Cookies   Headers (15)   Test Results   🕑   200 OK · 278 ms · 819 B

Pretty   Raw   Preview   Visualize   JSON ⌄

1 {
2   "message": "Recipe added",
3   "recipe": {
4   "user": "67508076128a21ad1e8ee7db",
5   "recipeName": "Spaghetti Bolognese",
6   "calories": 530,
7   "ingredients": [
8   "675c674fb8a0d86f558dc0ac",
9   "675c674fb8a0d86f558dc0ae",
10   "675c674fb8a0d86f558dc0b0",
11   "675c674fb8a0d86f558dc0b2",
12   "675c674fb8a0d86f558dc0b4"
13   ],
14   "instructions": "1. Cook spaghetti according to package instructions. 2. In a separate pan,
               sauté onions and garlic until translucent. 3. Add ground beef and cook until browned. 4.
               Pour in tomato sauce and simmer for 10 minutes. 5. Combine spaghetti with the sauce and
               serve.",
15   "_id": "675c674fb8a0d86f558dc0b6",
16   "createdAt": "2024-12-13T16:56:47.9217"

Postbot
Ctrl Alt P

Get user recipes:

GET ⌄ https://cs633-team-2.onrender.com/api/recipe/   Send ⌄

Params   Authorization ●   Headers (10)   Body ●   Scripts   Tests   Settings   Cookies

Body   Cookies   Headers (15)   Test Results   🕑   200 OK · 218 ms · 866 B

Pretty   Raw   Preview   Visualize   JSON ⌄

1 {
2   "recipes": [
3   {
4   "_id": "675c674fb8a0d86f558dc0b6",
5   "user": "67508076128a21ad1e8ee7db",
6   "recipeName": "Spaghetti Bolognese",
7   "calories": 530,
8   "ingredients": [
9   {
10   "_id": "675c674fb8a0d86f558dc0ac",
11   "foodName": "spaghetti",
12   "quantity": "200g",
13   "calories": 200,
14   "__v": 0
15   },
16   {
17   "_id": "675c674fb8a0d86f558dc0a...
18   "foodName": "ground beef",
19   "quantity": "150g",

Postbot
Ctrl Alt P

Delete recipe

DELETE ∨ https://cs633-team-2.onrender.com/api/recipe/675c674fb8a0d86f558dc0b6 **Send** ∨

Params Authorization • Headers (10) Body • Scripts Tests Settings   Cookies

Body Cookies Headers (15) Test Results ⟲   200 OK • 199 ms • 500 B • 🌐 ⟐ ⌨ ∘∘∘

Pretty Raw Preview Visualize | JSON ∨ | ⇥   🔗 ⎘ 🔍

```
1   {
2       "message": "Recipe deleted successfully"
3   }
```

search Controller:
Test searchallItems endpoint

POST ∨ https://cs633-team-2.onrender.com/api/search/all

Params Authorization • Headers (10) Body • Scripts Tests Settings

◯ none ◯ form-data ◯ x-www-form-urlencoded ⦿ raw ◯ binary ◯ GraphQL **JSON** ∨

```
1   {
2       "query": "tofu"
3   }
```

ody Cookies Headers (15) Test Results ⟲   200 OK • 438 ms

Pretty Raw Preview Visualize | JSON ∨ | ⇥

```
1   [
2       {
3           "name": "tofu",
4           "serving_quantity": 0.2,
5           "serving_unit": "block",
6           "image": "https://nix-tag-images.s3.amazonaws.com/850_thumb.jpg",
7           "tag_name": "tofu",
8           "locale": "en_US"
9       },
10      {
11          "name": "tofu pho",
12          "serving_quantity": 1,
```

┌─────────────────┐
│ Postbot   │
│ Ctrl Alt P   │
└─────────────────┘

Search one item:

POST ⌄ https://cs633-team-2.onrender.com/api/search/one

Params   Authorization ●   Headers (10)   Body ●   Scripts   Tests   Settings

○ none   ○ form-data   ○ x-www-form-urlencoded   ● raw   ○ binary   ○ GraphQL   JSON ⌄

```
1  {
2    "query": "tofu"
3  }
```

Body   Cookies   Headers (15)   Test Results   ⟳   200 OK • 321 ms • 94

Pretty   Raw   Preview   Visualize   JSON ⌄

```
1  [
2    {
3      "foodName": "tofu",
4      "brandName": null,
5      "servingQuantity": 0.2,
6      "servingUnit": "block",
7      "servingWeightGrams": 91,
8      "calories": 75.53,
9      "totalFat": 4.79,
10     "saturatedFat": 0.84,
11     "cholesterol": 0,
12     "sodium": 3.64,
```

Postbot
Ctrl Alt P

Unit Testing Using JEST

Registration Page:
1. should display validation errors when fields are empty
2. should call registerUser API on valid input

```
PASS  app/(Authentication)/__tests__/RegistrationScreen.test.tsx
  RegistrationScreen
    √ should display validation errors when fields are empty (171 ms)
    √ should call registerUser API on valid input (407 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        3.425 s
Ran all test suites matching /app\\\(Authentication\)\\__tests__\\RegistrationScreen.test.tsx/i.
○ (base) PS C:\Users\ravir\OneDrive\Desktop\Fridge\CS633-Team-2>
```

Login Page:
1. should display validation errors when fields are empty
2. should call loginUser API on valid input

```
PASS  app/(Authentication)/__tests__/LoginScreen.test.tsx
  LoginScreen
    √ should display validation errors when fields are empty (1361 ms)
    √ should call loginUser API on valid input (394 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        3.732 s, estimated 6 s
Ran all test suites matching /app\\\(Authentication\)\\__tests__\\LoginScreen.test.tsx/i.
(base) PS C:\Users\ravir\OneDrive\Desktop\Fridge\CS633-Team-2>
```

Homepage:
1. should navigate to MealPlan Calendar Page
2. should navigate to Recipes Page
3. should navigate to Fridge & Pantry Page

```
PASS  app/__tests__/homepage.test.tsx
  Homepage
    √ should navigate to MealPlan Calendar Page (553 ms)
    √ should navigate to Recipes Page (7 ms)
    √ should navigate to Fridge & Pantry Page (3 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        1.944 s, estimated 2 s
Ran all test suites matching /app\\__tests__\\homepage.test.tsx/i.
(base) PS C:\Users\ravir\OneDrive\Desktop\Fridge\CS633-Team-2> []
```

Fridge & Pantry
1. renders the fridge and pantry items
2. searches for items and displays search results
3. opens the modal when an item is selected
4. deletes an item from the fridge
5. navigates to different tabs

```
PASS  app/__tests__/Fridge.test.tsx
  Fridge_Pantry Component
    √ renders the fridge and pantry items (594 ms)
    √ searches for items and displays search results (81 ms)
    √ opens the modal when an item is selected (85 ms)
    √ deletes an item from the fridge (68 ms)
    √ navigates to different tabs (58 ms)

Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        2.264 s, estimated 4 s
Ran all test suites matching /app\\__tests__\\Fridge.test.tsx/i.
```

Test Type	Functionality
Test Number	1
Test Case	Verify that users can add an ingredient to the fridge, ensuring it is saved and

Description	displayed accurately in the Fridge_Pantry list.
Results	Passed
Requirements to be Tested	As a user, I need to add ingredients to my fridge, which includes their name and quantity, and see them listed in the pantry view for easy tracking.

Test Steps	Search for the item (rice) and select the item
	Input the ingredient details:
	
	    ● Measure: g
	    ● Quantity: 1000
	
	Tap the Add Item button to save the ingredient.
	Verify that "rice" appears in the pantry list with the correct details.

Expected result	The ingredient "rice" is successfully displayed in the Fridge_Pantry list with the correct details entered by the user.
Actual Result	Verified that the ingredient was added successfully and displayed in the pantry list with all details correct.
Comments	The test passed successfully. As a potential improvement, a feature to edit the ingredients could enhance the user experience.

Test Type	Functionality
Test Number	2
Test Case Description	Verify that adding an ingredient (e.g., "Tofu") that already exists in the fridge updates the quantity of the existing entry instead of creating a duplicate entry.
Results	Failed
Requirements to be Tested	Adding an existing ingredient should update its quantity in the Fridge list. No duplicate entries should appear in the fridge list. Quantity updates should reflect accurately and immediately in the UI.

Test Steps	Search for the item (tofu)
	Add "Tofu" as an ingredient with the following details:
	    - Quantity: "200 "
	    - Measure: "grams"
	Navigate to the fridge list and confirm that "Tofu" is added with the specified quantity.
	Add "Tofu" again with a new quantity:
	    - Quantity: "100 "
	    - Measure: "grams"
	Navigate back to the fridge list to check the status of "Tofu."

Expected result	The quantity of "Tofu" in the fridge should update to reflect the new total quantity (e.g., "300 g" if previously "200 g" and "100 g" were added). No duplicate entry for "Tofu" should be created.
Actual Result	"Tofu" is added as a new entry in the fridge list instead of updating the existing quantity
Comments	Duplicate entries are created when adding an ingredient like "Tofu" again instead of updating the existing quantity. Edit functionality is required