**ENDPOINTS**

1. POST {{BASE\_URL}}/api/users/register
2. POST {{BASE\_URL}}/api/users/login
3. GET {{BASE\_URL}}/api/users/
4. GET {{BASE\_URL}}/api/users/profile
5. GET {{BASE\_URL}}/api/recipes/
6. GET {{BASE\_URL}}/api/recipes/:id
7. POST {{BASE\_URL}}/api/recipes/
8. PATCH {{BASE\_URL}}/api/recipes/:id
9. DELETE {{BASE\_URL}}/api/recipes/:id
10. POST {{BASE\_URL}}/api/favorites/add
11. GET {{BASE\_URL}}/api/favorites/:id
12. DELETE {{BASE\_URL}}/api/favorites/remove
13. POST {{BASE\_URL}}/api/recipes/submittedby
14. POST {{BASE\_URL}}/api/users/logout

### **1. POST {{BASE\_URL}}/api/users/register**

**Description**: Registers a new user in the system.

**Input**:

* name: (string) The user's full name.
* email: (string) The user's email address.
* password: (string) The user's password.
* confirmPassword: (string) Confirmation of the user's password.
* termsAgreed: (boolean) Whether the user has agreed to the terms and conditions.

**Request Body**:

{

"name": "Jane Smith",

"email": "janesmith@example.com",

"password": "password456",

"confirmPassword": "password456",

"termsAgreed": true

}

**Response**:

**Success (201)**: Returns the newly created user object with basic details (excluding sensitive information like the password).  
  
{

"id": "1234567890",

"name": "Jane Smith",

"email": "janesmith@example.com"

}

**Error (400)**: Returns an error if registration fails (e.g., passwords do not match, missing fields, or email already exists).  
  
{

"error": "Passwords do not match"

}

### 

### **2. POST {{BASE\_URL}}/api/users/login**

**Description**: Authenticates a user and provides a JSON Web Token (JWT) for subsequent requests.

**Input**:

* email: (string) The email address associated with the user account.
* password: (string) The user's password.

**Request Body**:

{

"email": "janesmith@example.com",

"password": "password456"

}

**Response**:

**Success (200)**: Returns a success message along with a token.  
{

"message": "Login successful",

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2Nzg1MDRjM2M5YTc1NGU2M2MyNGY5OGYiLCJpYXQiOjE3MzcwMjI5MzcsImV4cCI6MTczNzAyMzgzN30.BI53xyVUGs8bP-WMxuHbPb6M03LpG-6q7FJ54TxiFgo"

}

**Error (400)**: Returns an error if login fails (e.g., invalid email or password).  
{

"error": "Invalid email or password"

}

### **3. GET {{BASE\_URL}}/api/users/**

**Description**: Fetches a list of all registered users in the system.

**Input**:

* No input is required for this endpoint.

**Response**:

**Success (200)**: Returns an array of user objects. Each object contains:

* + \_id: The unique identifier of the user.
  + name: The name of the user.
  + email: The email address of the user.
  + termsAgreed: A boolean indicating whether the user agreed to the terms.
  + favorites: An array of recipe IDs the user has marked as favorites.
  + \_\_v: A Mongoose internal version key.

**Example Response**:

[

{

"\_id": "677ff12064ba86fd9188b3d5",

"name": "Test User",

"email": "testuser@example.com",

"termsAgreed": true,

"favorites": ["677ffc5d963072ec0eca430a", "6780f868816d4fb4fa15eaa1"],

"\_\_v": 2

},

{

"\_id": "67813fd3f5df5715f842b994",

"name": "John Doe",

"email": "johndoe@example.com",

"termsAgreed": true,

"favorites": ["6780f868816d4fb4fa15eaa1"],

"\_\_v": 3

},

{

"\_id": "678504c3c9a754e63c24f98f",

"name": "Jane Smith",

"email": "janesmith@example.com",

"termsAgreed": true,

"favorites": [],

"\_\_v": 0

}

]

### **4. GET {{BASE\_URL}}/api/users/profile** *—-- Protected route*

**Description**: Fetches the profile of the currently logged-in user. This is a protected route and requires a valid authentication token.

**Input**:

**Headers**:

Authorization: Bearer token (e.g., Bearer <your-auth-token>).

**Responses**:

**Success (200)**:  
Returns the user's profile, including their ID, name, email, and favorites list.  
**Example Response**:  
json  
  
{

"message": "Profile fetched successfully",

"user": {

"id": "67813fd3f5df5715f842b994",

"name": "John Doe",

"email": "johndoe@example.com",

"favorites": [

{

"\_id": "6780f868816d4fb4fa15eaa1",

"name": "Sustainable Pasta Primavera",

"short\_description": "A fresh and vibrant vegetable pasta with minimal environmental impact."

}

]

}

}

**Unauthorized (401)**:  
If the token is invalid, expired, or not provided.  
**Example Responses**:

Invalid Token:  
{

"message": "Invalid or expired token. Please log in again.",

"error": "jwt expired"

}

Missing Token:  
  
{

"message": "Authorization token not provided."

}

### **5. GET {{BASE\_URL}}/api/recipes/**

**Description**: Fetches a list of all available recipes from the database. Each recipe includes its details such as ingredients, instructions, sustainability ratings, and other relevant information.

**Input**:  
No input is required for this endpoint.

**Responses**:

**Success (200)**:  
Returns an array of recipes, with each recipe containing detailed information.  
**Example Response**:  
  
[

{

"sustainability\_ratings": {

"water\_usage": 3,

"energy\_impact": 3,

"waste\_consideration": 4

},

"\_id": "6780f868816d4fb4fa15eaa1",

"name": "Sustainable Pasta Primavera",

"short\_description": "A fresh and vibrant vegetable pasta with minimal environmental impact.",

"ingredients": [

{ "item": "pasta", "quantity": "200g", "\_id": "6780f868816d4fb4fa15eaa2" },

{ "item": "olive oil", "quantity": "2 tablespoons", "\_id": "6780f868816d4fb4fa15eaa3" },

{ "item": "garlic", "quantity": "2 cloves, minced", "\_id": "6780f868816d4fb4fa15eaa4" },

{ "item": "bell pepper", "quantity": "1, chopped", "\_id": "6780f868816d4fb4fa15eaa5" },

{ "item": "zucchini", "quantity": "1, chopped", "\_id": "6780f868816d4fb4fa15eaa6" },

{ "item": "tomatoes", "quantity": "2, chopped", "\_id": "6780f868816d4fb4fa15eaa7" },

{ "item": "spinach", "quantity": "1 cup", "\_id": "6780f868816d4fb4fa15eaa8" },

{ "item": "parmesan cheese", "quantity": "1/4 cup, grated", "\_id": "6780f868816d4fb4fa15eaa9" }

],

"instructions": [

"Cook pasta according to package instructions.",

"In a large skillet, heat olive oil over medium heat.",

"Add garlic and cook until fragrant.",

"Add bell pepper, zucchini, and tomatoes, and sauté for 5 minutes.",

"Add spinach and cook until wilted.",

"Toss the cooked pasta with the vegetables.",

"Top with parmesan cheese and serve."

],

"why\_this\_recipe\_is\_sustainable": [

"Uses locally sourced, seasonal vegetables.",

"Olive oil is a more sustainable fat compared to animal-based oils.",

"The dish is plant-based and uses minimal processed ingredients."

],

"\_\_v": 0

}

]

**No Recipes Found (200)**:  
If no recipes are available, the response will be an empty array.  
**Example Response**:  
[]

**Error (500)**:  
If there is an issue fetching the recipes from the server.

**Example Response**:  
{

"message": "Unable to fetch recipes. Please try again later."

}

### **6. GET {{BASE\_URL}}/api/recipes/:id**

**Description**: Fetches a specific recipe by its unique ID. This endpoint provides detailed information about the recipe, including its sustainability ratings, ingredients, preparation instructions, and more.

**Input**:

**Path Parameter**:

:id - The unique identifier of the recipe to be fetched.

**Example Request**:  
  
GET {{BASE\_URL}}/api/recipes/6780f868816d4fb4fa15eaa1

**Responses**:

**Success (200)**:  
Returns the recipe details.  
**Example Response**:  
  
{

"sustainability\_ratings": {

"water\_usage": 3,

"energy\_impact": 3,

"waste\_consideration": 4

},

"\_id": "6780f868816d4fb4fa15eaa1",

"name": "Sustainable Pasta Primavera",

"short\_description": "A fresh and vibrant vegetable pasta with minimal environmental impact.",

"ingredients": [

{ "item": "pasta", "quantity": "200g", "\_id": "6780f868816d4fb4fa15eaa2" },

{ "item": "olive oil", "quantity": "2 tablespoons", "\_id": "6780f868816d4fb4fa15eaa3" },

{ "item": "garlic", "quantity": "2 cloves, minced", "\_id": "6780f868816d4fb4fa15eaa4" },

{ "item": "bell pepper", "quantity": "1, chopped", "\_id": "6780f868816d4fb4fa15eaa5" },

{ "item": "zucchini", "quantity": "1, chopped", "\_id": "6780f868816d4fb4fa15eaa6" },

{ "item": "tomatoes", "quantity": "2, chopped", "\_id": "6780f868816d4fb4fa15eaa7" },

{ "item": "spinach", "quantity": "1 cup", "\_id": "6780f868816d4fb4fa15eaa8" },

{ "item": "parmesan cheese", "quantity": "1/4 cup, grated", "\_id": "6780f868816d4fb4fa15eaa9" }

],

"instructions": [

"Cook pasta according to package instructions.",

"In a large skillet, heat olive oil over medium heat.",

"Add garlic and cook until fragrant.",

"Add bell pepper, zucchini, and tomatoes, and sauté for 5 minutes.",

"Add spinach and cook until wilted.",

"Toss the cooked pasta with the vegetables.",

"Top with parmesan cheese and serve."

],

"why\_this\_recipe\_is\_sustainable": [

"Uses locally sourced, seasonal vegetables.",

"Olive oil is a more sustainable fat compared to animal-based oils.",

"The dish is plant-based and uses minimal processed ingredients."

],

"\_\_v": 0

}

**Error (404)**:  
If the recipe with the provided ID is not found.  
**Example Response**:  
  
{

"message": "Recipe not found"

}

**Error (500)**:  
If there is an issue with the server or the database.  
**Example Response**:  
  
{

"message": "Unable to fetch recipe. Please try again later."

}

### **7. POST {{BASE\_URL}}/api/recipes/** *—-- Protected route*

**Description**: Creates a new recipe and stores it in the database. This endpoint allows users to submit recipes with detailed information, including ingredients, instructions, and sustainability considerations.

**Input**:

**Request Body** (JSON):  
  
{

"name": "Sustainable Tacos",

"short\_description": "Tasty tacos made with plant-based ingredients and minimal environmental impact.",

"ingredients": [

{ "item": "corn tortillas", "quantity": "6" },

{ "item": "black beans", "quantity": "1 can, drained and rinsed" },

{ "item": "avocado", "quantity": "1, sliced" },

{ "item": "tomato", "quantity": "1, diced" },

{ "item": "red onion", "quantity": "1/2, finely chopped" },

{ "item": "cilantro", "quantity": "1/4 cup, chopped" },

{ "item": "lime", "quantity": "1, cut into wedges" },

{ "item": "taco seasoning", "quantity": "1 tablespoon" },

{ "item": "olive oil", "quantity": "1 tablespoon" }

],

"instructions": [

"Heat olive oil in a skillet over medium heat.",

"Add black beans and taco seasoning, stirring to combine. Cook for 5-7 minutes until heated through.",

"Warm the corn tortillas in a separate skillet or microwave.",

"Assemble the tacos by placing a spoonful of seasoned beans on each tortilla.",

"Top with avocado, tomato, red onion, cilantro, and a squeeze of lime juice.",

"Serve and enjoy!"

],

"why\_this\_recipe\_is\_sustainable": [

"Uses plant-based ingredients, reducing reliance on animal products.",

"Black beans are a great source of protein with minimal environmental impact.",

"Corn tortillas are often made with non-GMO ingredients, reducing pesticide use."

],

"sustainability\_ratings": {

"water\_usage": 4,

"energy\_impact": 3,

"waste\_consideration": 4

}

}

**Response**:

**Success (201)**:  
If the recipe is created successfully.  
**Example Response**:  
  
{

"message": "Recipe created successfully"

}

**Error (400)**:  
If required fields are missing or invalid.  
**Example Response**:  
{

"message": "Validation failed. Please ensure all required fields are provided."

}

**Error (500)**:  
If there is a server error while creating the recipe.  
**Example Response**:  
{

"message": "Unable to create recipe. Please try again later."

}

### **8. PATCH {{BASE\_URL}}/api/recipes/:id** *—-- Protected route*

**Description**: Updates an existing recipe. Only authorized users (e.g., the original submitter or an admin) can update a recipe.

**Input**:

**Path Parameter**:

id (String): The unique ID of the recipe to be updated.

**Request Body** (JSON):  
Partial or full update data. For example:  
  
{

"name": "Updated Sustainable Tacos",

"short\_description": "A new, sustainable twist on the classic Nigerian mashed beans dish."

}

**Response**:

**Success (200)**:  
If the recipe is updated successfully.  
**Example Response**:  
{

"sustainability\_ratings": {

"water\_usage": 4,

"energy\_impact": 3,

"waste\_consideration": 4

},

"\_id": "6788e375c381113e5cca1622",

"name": "Updated Sustainable Tacos",

"short\_description": "A new, sustainable twist on the classic Nigerian mashed beans dish.",

"ingredients": [

{

"item": "corn tortillas",

"quantity": "6",

"\_id": "6788e375c381113e5cca1623"

},

{

"item": "black beans",

"quantity": "1 can, drained and rinsed",

"\_id": "6788e375c381113e5cca1624"

},

{

"item": "avocado",

"quantity": "1, sliced",

"\_id": "6788e375c381113e5cca1625"

},

{

"item": "tomato",

"quantity": "1, diced",

"\_id": "6788e375c381113e5cca1626"

},

{

"item": "red onion",

"quantity": "1/2, finely chopped",

"\_id": "6788e375c381113e5cca1627"

},

{

"item": "cilantro",

"quantity": "1/4 cup, chopped",

"\_id": "6788e375c381113e5cca1628"

},

{

"item": "lime",

"quantity": "1, cut into wedges",

"\_id": "6788e375c381113e5cca1629"

},

{

"item": "taco seasoning",

"quantity": "1 tablespoon",

"\_id": "6788e375c381113e5cca162a"

},

{

"item": "olive oil",

"quantity": "1 tablespoon",

"\_id": "6788e375c381113e5cca162b"

}

],

"instructions": [

"Heat olive oil in a skillet over medium heat.",

"Add black beans and taco seasoning, stirring to combine. Cook for 5-7 minutes until heated through.",

"Warm the corn tortillas in a separate skillet or microwave.",

"Assemble the tacos by placing a spoonful of seasoned beans on each tortilla.",

"Top with avocado, tomato, red onion, cilantro, and a squeeze of lime juice.",

"Serve and enjoy!"

],

"why\_this\_recipe\_is\_sustainable": [

"Uses plant-based ingredients, reducing reliance on animal products.",

"Black beans are a great source of protein with minimal environmental impact.",

"Corn tortillas are often made with non-GMO ingredients, reducing pesticide use."

],

"submittedBy": "67813fd3f5df5715f842b994",

"\_\_v": 0

}

**Error (403)**:  
If the user is not authorized to update the recipe.  
**Example Response**:  
{

"message": "You are not authorized to update this recipe"

}

**Error (404)**:  
If the recipe with the given ID does not exist.  
**Example Response**:  
{

"message": "Recipe not found"

}

**Error (500)**:  
If there is a server error during the update.  
**Example Response**:  
  
{

"message": "Unable to update recipe. Please try again later."

}

### 

### **9. DELETE {{BASE\_URL}}/api/recipes/:id** *—-- Protected route*

**Description**: Deletes a specific recipe by its ID. Only authorized users (e.g., the original submitter or an admin) can delete a recipe.

**Input**:

* **Path Parameter**:

id (String): The unique ID of the recipe to be deleted.

**Response**:

**Success (200)**:  
If the recipe is deleted successfully.  
**Example Response**:  
{

"message": "Recipe deleted successfully"

}

**Error (403)**:  
If the user is not authorized to delete the recipe.  
**Example Response**:  
{

"message": "You are not authorized to delete this recipe"

}

**Error (404)**:  
If the recipe with the given ID does not exist.  
**Example Response**:  
{

"message": "Recipe not found"

}

**Error (500)**:  
If there is a server error during the deletion process.  
**Example Response**:  
{

"message": "Unable to delete recipe. Please try again later."

}

### **10. POST {{BASE\_URL}}/api/favorites/add** *—-- Protected route*

**Description**: Adds a recipe to a user's list of favorite recipes.

**Input**:

* **Request Body** (JSON):
  + userId (String): The ID of the user who wants to add the recipe to their favorites.
  + name (String): The name of the recipe being added.

**Response**:

**Success (200)**:  
If the recipe is successfully added to the user's favorites.  
**Example Response**:  
{

"message": "Recipe added to favorites"

}

**Error (404)**:  
If the user or recipe is not found.  
**Example Response**:  
{

"message": "User or recipe not found"

}

**Error (409)**:  
If the recipe is already in the user's favorites.  
**Example Response**:  
{

"message": "Recipe is already in favorites"

}

**Error (500)**:  
If there is a server error while processing the request.  
**Example Response**:  
{

"message": "Unable to add recipe to favorites. Please try again later."

}

### **11. GET {{BASE\_URL}}/api/favorites/:id** *—-- Protected route*

**Description**: Retrieves a user's list of favorite recipes.

**Input**:

* **URL Parameters**:

id (String): The ID of the user whose favorites are being fetched.

**Response**:

**Success (200)**:  
If the favorites are successfully retrieved for the user.  
**Example Response**:  
[

{

"sustainability\_ratings": {

"water\_usage": 3,

"energy\_impact": 3,

"waste\_consideration": 4

},

"\_id": "6780f868816d4fb4fa15eaa1",

"name": "Sustainable Pasta Primavera",

"short\_description": "A fresh and vibrant vegetable pasta with minimal environmental impact.",

"ingredients": [

{ "item": "pasta", "quantity": "200g", "\_id": "6780f868816d4fb4fa15eaa2" },

{ "item": "olive oil", "quantity": "2 tablespoons", "\_id": "6780f868816d4fb4fa15eaa3" },

{ "item": "garlic", "quantity": "2 cloves, minced", "\_id": "6780f868816d4fb4fa15eaa4" },

{ "item": "bell pepper", "quantity": "1, chopped", "\_id": "6780f868816d4fb4fa15eaa5" },

{ "item": "zucchini", "quantity": "1, chopped", "\_id": "6780f868816d4fb4fa15eaa6" },

{ "item": "tomatoes", "quantity": "2, chopped", "\_id": "6780f868816d4fb4fa15eaa7" },

{ "item": "spinach", "quantity": "1 cup", "\_id": "6780f868816d4fb4fa15eaa8" },

{ "item": "parmesan cheese", "quantity": "1/4 cup, grated", "\_id": "6780f868816d4fb4fa15eaa9" }

],

"instructions": [

"Cook pasta according to package instructions.",

"In a large skillet, heat olive oil over medium heat.",

"Add garlic and cook until fragrant.",

"Add bell pepper, zucchini, and tomatoes, and sauté for 5 minutes.",

"Add spinach and cook until wilted.",

"Toss the cooked pasta with the vegetables.",

"Top with parmesan cheese and serve."

],

"why\_this\_recipe\_is\_sustainable": [

"Uses locally sourced, seasonal vegetables.",

"Olive oil is a more sustainable fat compared to animal-based oils.",

"The dish is plant-based and uses minimal processed ingredients."

],

"\_\_v": 0

}

]

**Error (404)**:  
If the user is not found or has no favorites.  
**Example Response**:  
{

"message": "User not found or no favorites available"

}

**Error (500)**:  
If there is a server error while processing the request.  
**Example Response**:  
{

"message": "Unable to fetch favorites. Please try again later."

}

### **12. DELETE {{BASE\_URL}}/api/favorites/remove**

### *—-- Protected route*

**Description**: Removes a recipe from a user's list of favorites.

**Input**:

* **Body Parameters**:

recipeId (String): The ID of the recipe to be removed from the user's favorites.

**Response**:

**Success (200)**:  
If the recipe is successfully removed from the user's favorites.  
**Example Response**:  
{

"message": "Recipe removed from favorites"

}

**Error (404)**:  
If the recipe ID is not found or not in the user's favorites.  
**Example Response**:  
{

"message": "Recipe not found in favorites"

}

**Error (500)**:  
If there is a server error while processing the request.  
**Example Response**:  
{

"message": "Unable to remove recipe from favorites. Please try again later."

}

### 

### 

### **13. POST {{BASE\_URL}}/api/recipes/submittedby**

*—-- Protected route*

**Description**: Retrieves all recipes submitted by a specific user.

**Input**:

* **Body Parameters**:
  + userId (String): The ID of the user whose submitted recipes are to be fetched.
  + This can be passed either as input or the request can be made without it, where the server assumes the logged-in user's ID.

**Response**:

**Success (200)**:  
If recipes submitted by the user are found.  
**Example Response**:  
[

{

"sustainability\_ratings": {

"water\_usage": 4,

"energy\_impact": 3,

"waste\_consideration": 5

},

"\_id": "6784f218c0f916eeece5495e",

"name": "Sustainable Sushi Rolls",

"short\_description": "Delicious homemade sushi with sustainable ingredients.",

"ingredients": [

{

"item": "sushi rice",

"quantity": "1 cup",

"\_id": "6784f218c0f916eeece5495f"

},

{

"item": "nori sheets",

"quantity": "5 sheets",

"\_id": "6784f218c0f916eeece54960"

},

{

"item": "avocado",

"quantity": "1, sliced",

"\_id": "6784f218c0f916eeece54961"

},

{

"item": "cucumber",

"quantity": "1, julienned",

"\_id": "6784f218c0f916eeece54962"

},

{

"item": "carrot",

"quantity": "1, julienned",

"\_id": "6784f218c0f916eeece54963"

},

{

"item": "tofu",

"quantity": "100g, pan-fried",

"\_id": "6784f218c0f916eeece54964"

},

{

"item": "rice vinegar",

"quantity": "2 tablespoons",

"\_id": "6784f218c0f916eeece54965"

},

{

"item": "soy sauce",

"quantity": "For serving",

"\_id": "6784f218c0f916eeece54966"

}

],

"instructions": [

"Rinse sushi rice thoroughly and cook according to package instructions.",

"Once cooked, season the rice with rice vinegar and let it cool slightly.",

"Lay a sheet of nori on a sushi mat, shiny side down.",

"Spread a thin layer of rice over the nori, leaving a 1-inch border at the top.",

"Arrange avocado, cucumber, carrot, and tofu in a line near the bottom edge of the rice.",

"Using the mat, roll the nori tightly around the fillings to form a log.",

"Seal the roll with a bit of water along the top edge.",

"Repeat with remaining ingredients and slice the rolls into bite-sized pieces.",

"Serve with soy sauce."

],

"why\_this\_recipe\_is\_sustainable": [

"Utilizes plant-based fillings, reducing reliance on seafood.",

"Promotes the use of locally sourced and organic vegetables.",

"Homemade sushi reduces packaging waste compared to store-bought options."

],

"submittedBy": "67813fd3f5df5715f842b994",

"\_\_v": 0

}

]

**Error (404)**:  
If no recipes are found for the specified user.  
**Example Response**:  
{

"message": "No recipes found for this user."

}

**Error (500)**:  
If there is a server error while processing the request.  
**Example Response**:  
{

"message": "Unable to retrieve recipes. Please try again later."

}

### **14. POST {{BASE\_URL}}/api/users/logout** *—-- Protected route*

**Description**: Logs out the current user and invalidates their session.

**Input**:

* **Body Parameters**: No input body required.

**Response**:

**Success (200)**:  
When the user is successfully logged out.  
**Example Response**:  
{

"message": "Logged out successfully"

}

**Error (500)**:  
If there is an issue with the logout process.  
**Example Response**:  
{

"message": "Logout failed. Please try again."

}