

**Project Title:** Recipe Sharing API

**Group name:** Dambill

**Group Members:** Erica Ravanera

Jennelyn Encarnacion

Rheabel Pineda

Lyka Murillo

## **Overview of the project**

The Recipe Sharing API project involves the development of a RESTful API to enable users to share their favorite recipes. This project will be accessible to everyone, offering a rich and diverse collection of culinary creations. Users can add new recipes to the system, each including essential details such as a title, a list of ingredients, cooking instructions, and optional tags or categories to help organize the recipes (e.g., Breakfast, Lunch, Dinner, and Dessert). Additionally, users will have the ability to view, edit, and delete their recipes, ensuring a dynamic and user-driven experience.

## **Purpose**

The purpose of the Recipe Sharing API project is to provide a platform where users can share their favorite recipes with others. The project aims to build a RESTful API that allows users to create, manage, view, edit, and delete recipes, including essential details such as titles, ingredients, cooking instructions, and optional tags or categories. This platform encourages collaboration and community engagement by facilitating the exchange of diverse recipes among users.

## **Features**

- **User Authentication:** Allow users to create accounts and log in to manage their recipes.
- **Retrieve Recipes by Category:** Retrieve all recipes for breakfast, lunch, dinner, and dessert.
- **Retrieve a Specific Recipe:** Retrieve a specific recipe by its name or ID.
- **Add a New Recipe:** Allow users to add a new recipe to the system.
- **Modify an Existing Recipe:** Allow users to modify an existing recipe.
- **Delete a Recipe:** Allow users to delete a recipe.

## Technical implementation details

**Backend:** The API is built using Node.js with Express.js, providing a robust and scalable server-side architecture.

**Authentication:** JSON Web Tokens (JWT) are implemented for secure user authentication, protecting user data and ensuring only authorized access.

**Deployment:** The API is deployed on Postman, allowing for easy access and scalability to accommodate a growing user base.

## Challenges faced during the project

Developing the Recipe Sharing API posed several challenges throughout the project. Firstly, unstable internet connectivity intermittently disrupted development and testing processes, leading to delays and frustrations. Secondly, the team encountered difficulties due to a lack of prior knowledge about APIs, requiring extensive research and learning to understand API design principles and best practices. Additionally, errors surfaced during the coding phase, necessitating thorough debugging and problem-solving efforts. Moreover, time management presented a challenge, as balancing the various project tasks, including coding, testing, and documentation, proved to be demanding. Lastly, creating comprehensive documentation for the API was a daunting task, as it required clear and concise explanations of the API functionality, endpoints, and usage instructions. Despite these challenges, the team persevered, learning valuable lessons and enhancing their skills in software development and project management.

## Demonstration using Postman

- GET /api/recipes: Retrieve all recipes.
- GET /api/recipes/:category: Retrieve recipes by category. Replace `:category` with the desired category (e.g., `/api/recipes/main`).
- GET /api/recipes/name/:recipeName: Retrieve a recipe by name. Replace `:recipeName` with the desired recipe name (e.g., `/api/recipes/name/Pasta`).
- GET /api/recipes/id/:recipeId: Retrieve a recipe by ID. Replace `:recipeId` with the desired recipe ID (e.g., `/api/recipes/id/123`).
- POST /api/recipes: Add a new recipe. Send a JSON object representing the new recipe in the request body.
- PUT /api/recipes/:recipeId: Edit an existing recipe. Replace `:recipeId` with the ID of the recipe to edit. Send a JSON object representing the updated recipe in the request body.

- **DELETE /api/recipes/:recipeId:** Delete a recipe by ID. Replace ``:recipeId`` with the ID of the recipe to delete.

## Design Choices

1. **Programming Language:** Node.js - Offers non-blocking I/O, making it efficient for handling concurrent requests in a recipe sharing application.
2. **Framework:** Express.js - Provides a minimal and flexible framework for building web applications and APIs. It's easy to use and has a large community.
3. **Authentication:** Utilize JWT (JSON Web Tokens) for secure user authentication. This ensures that only authorized users can create, update, or delete recipes.

## Unique Feature

1. **User Authentication:** Implement JWT (JSON Web Tokens) for secure user authentication, ensuring that only authorized users can perform actions like creating, updating, or deleting recipes. User authentication enhances the security and privacy of the API. By allowing users to create accounts, log in, and authenticate their actions, the API ensures that recipe-related operations are performed by authorized individuals, adding a layer of control and personalization to the user experience.