

Names: Rohan Mehta, Ammar Alquran, Frederick Javier

Project Report: Recipe Realm

Project Introduction:

Our project aims to develop a web application for managing recipes. This application will allow users to create, view, edit, and delete recipes, as well as organize them into categories. Users will also be able to search for recipes based on various criteria such as ingredients, cuisine, and tags.

Objective:

The main objective of this project is to provide a user-friendly platform for individuals to store and access their favorite recipes. By offering features such as categorization, search functionality, and easy recipe management, our application aims to simplify the process of organizing and finding recipes.

Authentication and User Management:

The application incorporates a login and authentication system to ensure secure access to user accounts. Upon registration, users create unique credentials which are securely stored in the database. Authentication is implemented using JSON Web Tokens (JWT), providing a secure mechanism for user verification and session management. Users can securely log in and log out of their accounts, with passwords securely hashed for storage.

Random Recipe Generation:

One of the key features of the application is the ability to generate random recipes based on user preferences. This feature adds an element of discovery and spontaneity to the user experience, encouraging exploration of new dishes and culinary ideas.

Following and Unfollowing Users:

The application includes a social networking aspect where users can follow and unfollow other users. This functionality allows users to connect with each other, follow their favorite chefs or food enthusiasts, and stay updated on their latest recipes and culinary creations. Users can easily manage their list of followed users, enabling a personalized experience tailored to their interests.

CRUD Operations for Recipes:

The application supports all basic CRUD (Create, Read, Update, Delete) operations for recipes.

Users can create new recipes, providing detailed instructions, ingredient lists, and images to share their culinary creations with the community. They can also browse, search, and view recipes shared by other users. Additionally, users have the ability to edit or delete their own recipes as needed, ensuring full control over their culinary content. This comprehensive set of operations lets users actively participate in recipe sharing and community engagement, bringing a dynamic culinary environment within the application.

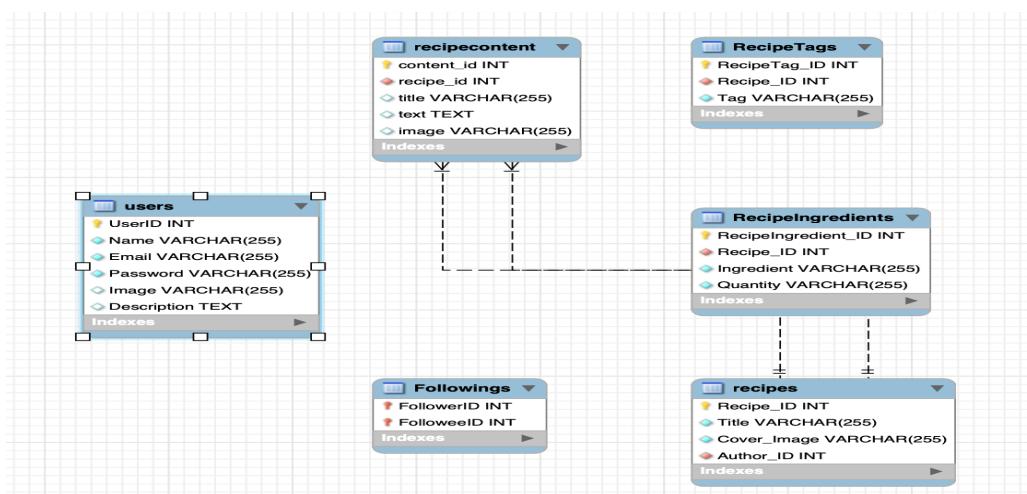
These features collectively enhance the user experience, providing a seamless and interactive platform for discovering, sharing, and enjoying culinary delights.

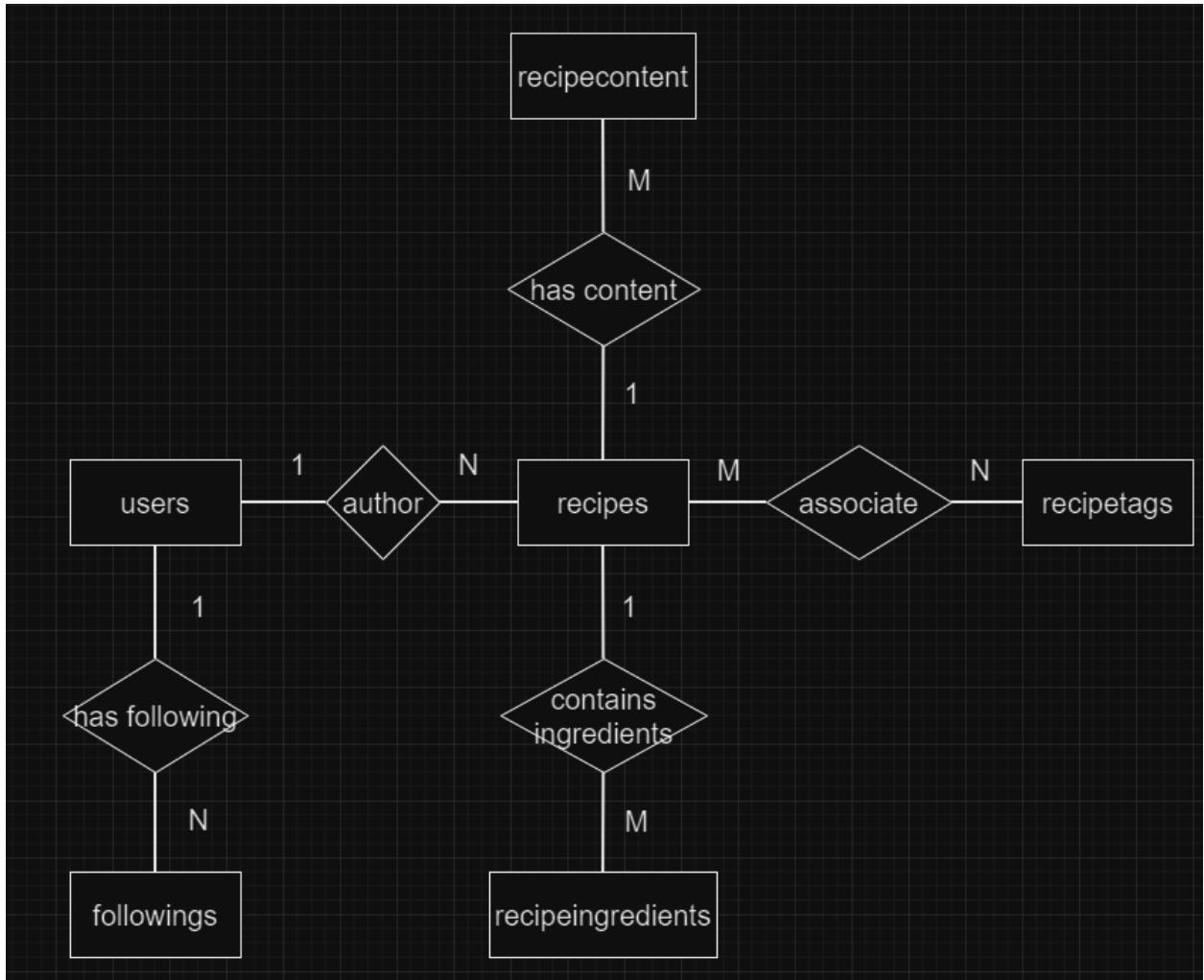
Project High-Level Design:

The project follows a client-server architecture. The client-side is developed using React.js for the user interface, while the server-side is built with Node.js and Express.js to handle requests and interact with the SQL database. The application uses a relational database schema to store recipe data and user information.

Database Design:

The database design includes tables for recipes, categories, users, and their relationships. Below is the Entity-Relationship (ER) Diagram illustrating the database schema:





(note: N = any number including 0, M = many not including 0.)

Normalization of Tables:

The tables in the database are normalized to minimize redundancy and ensure data integrity. Each table represents a single entity, and relationships between entities are established using foreign keys.

To determine whether the tables are in 3NF, we can evaluate each table's attributes and their dependencies on the primary keys.

followings table:

- The primary key consists of FollowerID and FolloweeID, which uniquely identify a follower-followee relationship.
- There are no transitive dependencies since both attributes are atomic and directly depend on the primary key.

recipecontent table:

- The primary key is content_id.
- Attributes title, text, and image are dependent on content_id, which is the primary key.
- There are no transitive dependencies since each attribute directly depends on the primary key.

recipeingredients table:

- The primary key is RecipeIngredient_ID.
- Attributes Ingredient and Quantity are dependent on RecipeIngredient_ID, which is the primary key.
- There are no transitive dependencies since each attribute directly depends on the primary key.

recipes table:

- The primary key is Recipe_ID.
- Attributes Title, Cover_Image, and Author_ID are dependent on Recipe_ID, which is the primary key.
- There are no transitive dependencies since each attribute directly depends on the primary key.

recipetags table:

- The primary key is RecipeTag_ID.
- Attributes Tag is dependent on RecipeTag_ID, which is the primary key.
- There are no transitive dependencies since each attribute directly depends on the primary key.

users table:

- The primary key is UserID.
- Attributes Name, Email, Password, Image, and Description are dependent on UserID, which is the primary key.

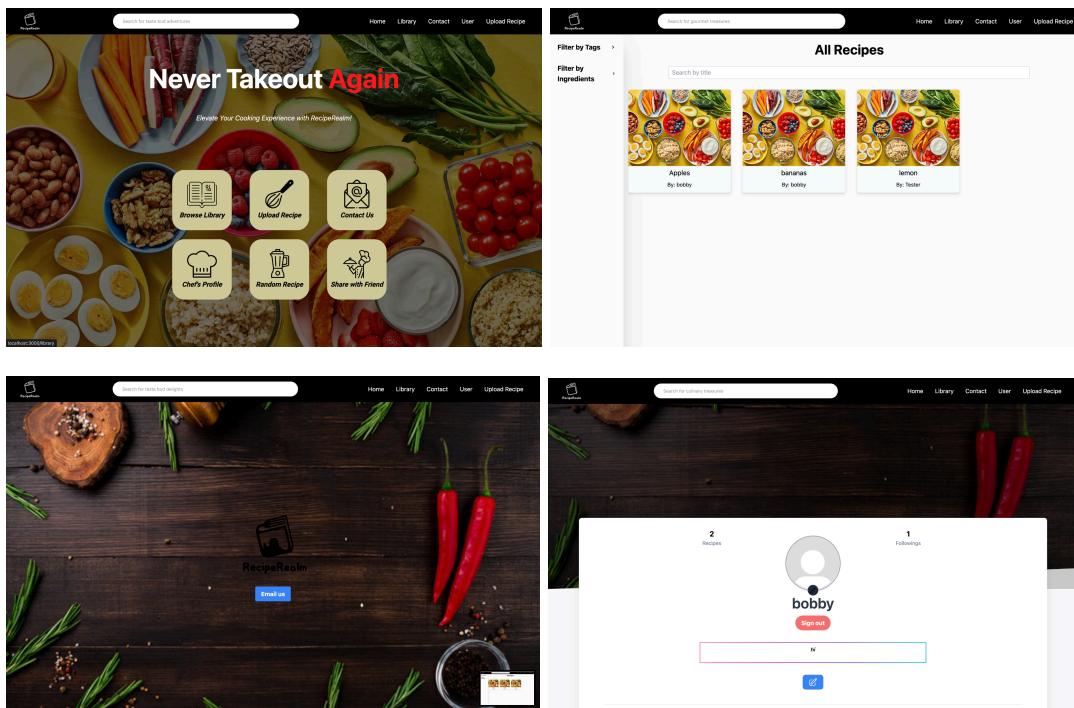
- There are no transitive dependencies since each attribute directly depends on the primary key.

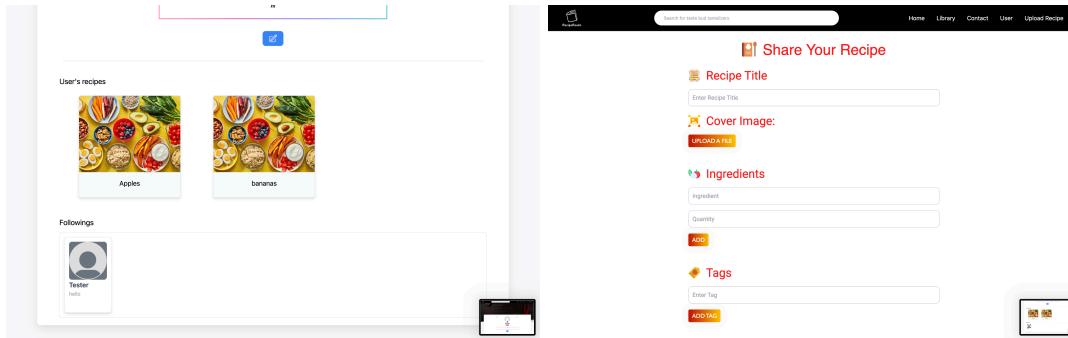
In conclusion, all the given tables are in Third Normal Form (3NF) because they satisfy the conditions of being in 2NF and have no transitive dependencies.

Results:

The web application has been successfully developed and tested, but has not been deployed to the public yet. Users can perform CRUD operations on recipes, manage categories, and search for recipes based on various criteria. The application provides a smooth user experience and meets the functional requirements outlined in the project specifications.

Screenshots:





Contribution/Work Done by Each Team Member:

[Rohan Mehta]: Developed front-end components, developed back-end server-side logic, helped design and implement the database schema

[Frederick Javier]: High level design of entities/tables, ER diagram, report/documentation, helped with database design

[Ammar Alquran]: Front end functionality, database schema construction, lead presenter, report/documentation with reference list formatting

References/Additional Sources:

Front-End Design/Styling:

User interface and styling written together by me (Rohan Mehta) and a contributor Edmund from the following repository since we both had similar project ideas that we needed to implement. Some differences in front-end since the schema and back-end are entirely different.

GitHub Repository of Source: <https://github.com/henrynvn09/chop-recipes-web/>

Full Stack Project Tutorial | Build a Recipe App for your Portfolio using React Node & PostgreSQL: https://www.youtube.com/watch?v=5wwaQ4GiSNU&ab_channel=ChrisBlakely