

UniRecipe

[UniRecipe heroku app](#)

[UniRecipe Git Repo](#)



Introduction

As a student, the motivation to cook big complicated recipes is very rare, so students tend to cook quick and easy recipes they find on social media. So when i saw a Kaggle dataset containing 230 000 recipes i decided to use it and take out the recipes tagged under 15-minutes-or-less. I decided to use only 300 recipes to reduce the load time of the project.

I decided to try and create a web app that can perform the basic CRUD operations on the dataset and that allows users to add their own recipes.

System Overview

- ### Web Pages

For my web app pages i choose to have the index page with a small description of the web app and a link to a fun fact. I then choose to present all the recipes in the dataset in table format so that the user could go through the list and choose a random recipe. The user can then view/edit or delete a recipe from the list, however this was a mistake because it is possible to edit and delete even if the user is not logged in. I then created a page where people can search the dataset, and click to open the choosen recipe in a new tab. I also created a page where a logged in user can create a recipe so it can be added to the dataset. For the login and register pages i choose a basic design.

- ### Database

For this project i used a database with the 300 recipes i took from the kaggle dataset and created 2 collections. One collection was the recipes list indexed by mongodb and the other collection is the user collection listing all the users registered to the web app.

- ### Authentication

For a user to register to the web app, they need to provide a name, email and a password. I choose to require a name so that i can welcome the user by its name on the pages header. The email and the password are used to authenticate the user and the password is encrypted by bcrypt before it's stored in the database. The user then provides its email and password to login to the web app.

Scalability and Security

The web app is currently scalable because it allows users to add their own recipes but it is not secure because currently any user can edit and delete any recipe. In order to make the web app secure and properly scalable, i would need to restrict the editing and deletion of a recipe to the user who created them and a administrator of the website.

Conclusion

I think i created a successfull web app that can perform CRUD operations, allows to search the database using keywords and also allows people to register, login and add their own recipes to the database. I am happy with what i created but if i had more time i would improve the project by adding the option to save a recipe to a user favorites page, i would also add the option to filter the recipe list by tags using checkboxes and i would also add a fun button in the index page that would choose a random recipe.