COMP9900 REPORT

GROUP TOMCAT

2 August 2021

[Scrum Master]

Patrick Li	Back-end, z5305975	z5305975@ad.unsw.edu.au

[Group Members]

Tingting Peng	Front-end, z5250402	z5250402@ad.unsw.edu.au
Wanchen Zhao	Front-end, z5283975	z5283975@ad.unsw.edu.au
Junlin Pang	Front-end, z5271854	z5271854@ad.unsw.edu.au
Yuhang Yang	Back-end, z5194094	z5194094@ad.unsw.edu.au

Table of Contents

1. Overview

1.1 Overall Introduction

Online recipes platform is becoming the definite first choice for cooking lovers especially during covid lockdowns periods when cooking at home. Team TomCat wish to bring one more choice, a pure and user-friendly platform for people who are looking for new recipes to try or sharing creative ideas.

MyRecipes is a project that aims to provide a platform for people who love cooking to share recipes with each other. It connects people who are willing to share with those who are looking for the recipe they need. On MyRecipes, users are free to post their own recipe for everyone to see it.   In this report, we will refer to the writer of a recipe as a “Contributor”. Alternatively, users can simply browse the trending recipes. The user who is browsing recipes is referred to as an “Explorer”. A person can be both a contributor and an explorer, depending on what he/she is doing.

Users might have different habits/behaviours when they are looking for a recipe. Some might decide “this is the recipe I’ll use” quickly, but many would browse more different recipes before they pick what they like. That is why we want to provide multiple ways for users to discover new recipes: On the homepage, the newest recipes of their subscribed contributor and the trending recipes will be displayed. The search function is always on the top of the page, and the search condition can be further customed. Entering a contributor’s profile, one can see all the recipes of this contributor. Lastly, whenever users have finished reading one recipe, there will be automatic recommendations with similar ingredients. This way the users will always have the next option to go to.

We want to build MyRecipes not only as a platform to present information but also as a community with a lively vibe, where people who love cooking can share their opinions and contributors can be encouraged by positive feedbacks. Thus, MyRecipes also features a feedback system. For each recipe, explorers can support the contributor by clicking a “like” button. And to further express their opinions and feelings, they are free to post comments under the recipe for everyone to see it.

Furthermore, we want to bring users more than just recipes, but also help them with the next steps, for example, shopping for the ingredients. We have specially designed an additional food basket feature which provides a much better food basket functions than other platforms. If a user wants to shop for the ingredients of more than one recipe, most other recipe platforms will only show several individual lists of each recipe. But when there exist the same ingredients in different recipes, it could be a bit annoying when repeated checking what has been bought and what’s not. This puts a burden on the user’s cognitive load, and we believe it would also be simpler if the user can see a summary of all the ingredients of different recipes in one place. On MyRecipes, we organise the ingredient list in 2 different forms, which can be smoothly switched in between. Users can either check ingredients lists for each recipe or view a summary with the same ingredients combined as one line. This way the list will be simpler and more concise, save you the time and energy for double-checking.

The whole project consists of three major parts, which are front-end web application, back-end server, and database. The front-end was implemented with React as the JavaScript library and Material-UI as the UI library. The back-end was implemented in Python 3.7 in Linux with Django framework connected to AWS MySQL and Firebase. The project also used other libraries for special features, which will be covered in the Third-party Library section.

1.2 System Architecture

Team TomCat used the 3-tier architecture pattern, which splits the architecture into three tiers: presentation, application, and data. The entire system was divided into two parts: front-end and back-end.

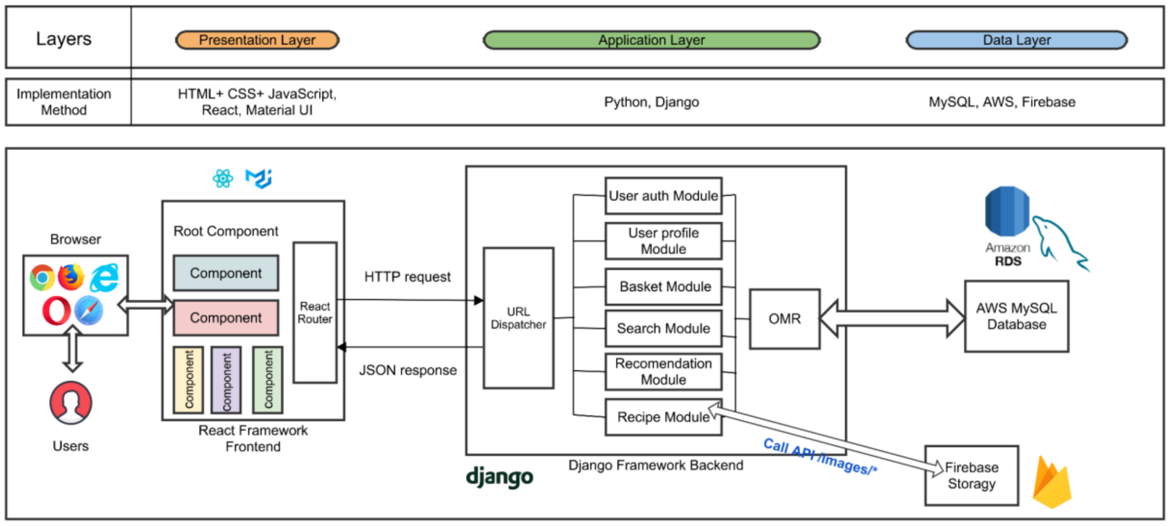


Figure 1.1 System Architecture

1.2.1 Front-end

The front-end is responsible for the presentation tier. The application's top level is occupied by it. It provides the content to web browsers and presents information as a graphical user interface, which users can access directly. It interfaces with other levels in our application by sending data to the application layer via REST API requests.

Our front-end is built with the React.js framework. Instead of a web browser loading completely new pages, it implements a single-page application that interacts with the user by dynamically updating the existing web page with new data from the web server.

1.2.2 Back-end

The application and data tiers are not visible to users, but they support our platform's logic and storage. The back-end encompasses both tiers.

1.2.2.1 Application Tier

The application tier is often referred to as the business tier. By processing the application's logic, it establishes a connection between the presentation layer and the data layer, allowing the presentation and data layers to communicate. We utilised Python for the application tier, with Django as the framework.

Django is an MVT based framework. The Model is the application's logical data structure, which is represented by a database. A model is a class that represents a table or collection in our database, and each of the class's attributes is a field of the table or collection. The term "view" refers to the data that is displayed to the user. It is not so much about the look of the data as it is about the presentation of the data. The view is the primary functional component of the Django design. It is where we write the business logic that will be responsible for requesting and responding to client requests. The template is the component that is used to present HTML pages in a web browser. It has been supplanted by the front end in our Architecture.

We expose application layer Interfaces via a REST API to the presentation tier. The backend accepts HTTP requests from the front-end based on the POST, GET, PUT, and Delete methods, with sending data in json format.

In our views, the URL dispatcher is a mapping between URL path expressions to Python functions. Once the application layer receives the request from the presentation layer, it dispatches the requests to the corresponding module. Each module will deal with the logic and call corresponding services to handle the request.

1.2.2.2 Data Tier

The persistent storage method and the data access layer comprise up the data tier. It allows us to connect to the database to insert, update, remove, and retrieve data from the database using our input data. MySQL was the database we utilised.

We need to fetch and update data from the database at the data access part. Django's Object-Relational Mapper (ORM), which allows us to communicate with our database, is one of its most powerful features. Django's ORM is essentially a pythonic means of writing SQL to query and manipulate our database and return results in a pythonic way.

We've chosen Amazon Web Services MySQL Database as our cloud database provider and Firebase Storage as our cloud storage provider. MySQL is used to store text data in this project. Additionally, we keep user tokens in our database, which eliminates the need for users to continually fill in the same login details. Because the REST API is stateless, we can authenticate the user and retrieve their status by comparing the token in the request to the token in the database.

When a user uploads photographs to the backend, they are automatically stored to the cloud, and our database contains only the image storage URLs.

2. Data Model

2.1 Data Preparation

To best testing our product, some real recipe data were entered manually gathered from websites. 20 users were created, relationships like subscription, like, comment were prepared manually distinctly for each user.

2.2 Database Architecture

AWS MySQL is used to store all text data, and Firebase storage is used for image data. When image is added, it is first uploaded to Firebase and its URL will be generated and stored back to AWS MySQL database. The entity relationship diagram is presented below:

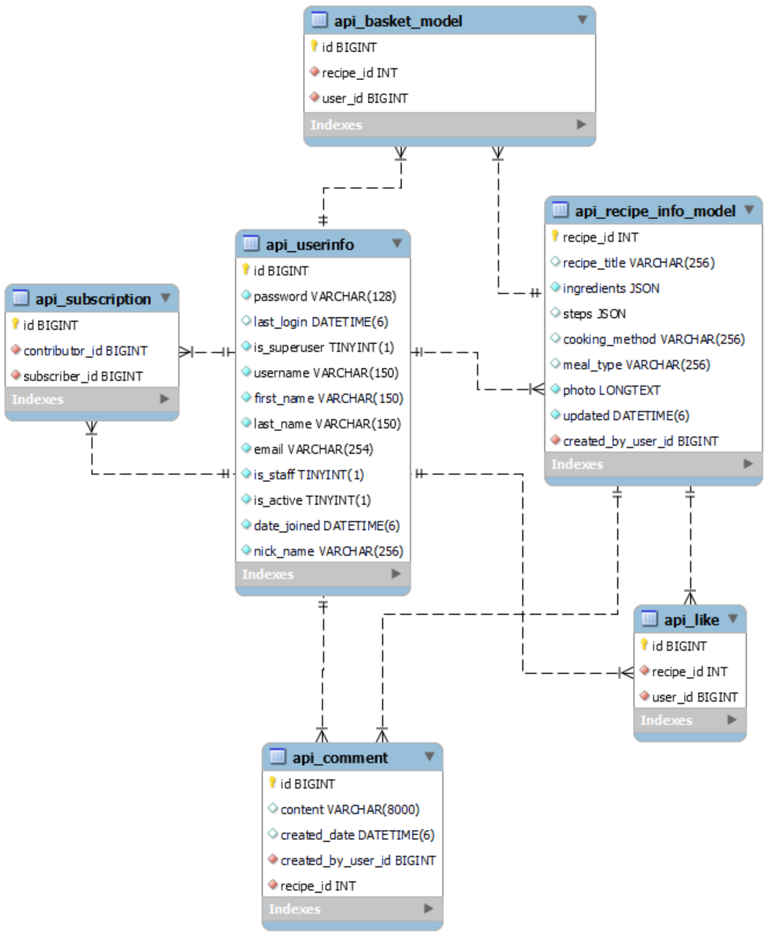


Figure 2.1 MySQL Database Entity Relationship Diagram

2.3 Summary of Data Tables

The design of data tables follows Third Normal Form (3NF) as it avoids data duplication and conflict (Codd, 1971). Each relationship between tables is stored as a new table with foreign keys. The tables created are summarized in this section.

2.3.1 Table “userinfo”

The “userinfo” table stores the user information with id as the primary key. It stores key information encrypted password, email, and nick_name.

2.3.2 Table “recipe_info_model”

The “recipe_info_model” stores recipe information with recipe_id as the primary key. It stores key information recipe_title, ingredients (JSON), steps (JSON), cooking_method, meal-type, photo (URL), created_by_user_id and updated (Datetime). The photo is uploaded to Firebase first then its URL will be stored in MySQL which saves time from encoding photo to text then decoding the text to photo. The user id from who created the recipe will be stored in the data model as a foreign key with “delete_on_cascade” enabled which ensures no data conflict. Updated time is created or will be updated when editing recipe using time in Sydney, Australia.

2.3.3 Table “like”

The “like” table stores the relationship between user and recipe if the user likes the recipe. The user id and recipe are stored in the table as foreign keys. “delete_on_cascade” also enabled to ensure either user or recipe is deleted, their like relationship is also deleted.

2.3.4 Table "subscription"

The "subscription" table stores the relationship between users if one subscribed the other. The user ids are stored in this table as foreign keys, one is contributor_id, the other is subscriber_id with "delete_on_cascade" enabled.

2.3.5 Table "comment"

The "comment" table stores the comment a user gives to a recipe. It stores content, created_date (datetime) and two foreign keys (user id and recipe id) with "delete_on_cascade" enabled.

2.3.6 Table "basket_model"

The "basket" table stores relationship if a user has added a recipe to his basket. The table stores user id and recipe id as foreign keys with "delete_on_cascade" enabled.

3. Project Functionalities

The project objectives are summarized below, and the implementation of the functions are presented in each section. The order of project objectives aligns with the order of MyRecipes's project objectives provided by UNSW on webCMS3.

3.1 Objective – Profile

Contributors must be able to create and maintain a profile for themselves, which includes their username, contact details (email address), and a list of recipes that this contributor has published, and to which MyRecipes users can navigate to.

3.1.1 User Authentication

Users need to log in to attain authentication to experience functionalities like Profiles Management, Recipes Creation, and Search. Thus, we classify user's login status when they try to view a recipe details or a contributor profile. We route unauthenticated users to our Sign-Up Page with a pop-up reminding message. In addition to passive routing, unauthenticated users can also actively direct to the registration or login page with relative buttons on the Navigation Bar. After a user logged in, on our Navigation Bar, in addition to MyRecipes Logo, the original login and registration buttons will be replaced by three icon buttons, corresponding to the three core functionalities of MyRecipes: Recipes Creation, Food Basket and Profile Management.

3.1.1.1 Sign Up

Whether it's passive redirection or actively clicking the registration button at the right of Navigation Bar, a new user can successfully enter our Sign-Up Page to sign up for a new account.

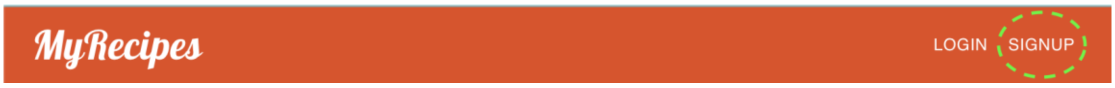


Figure 3.1.1 Sign Up Location

During the new user registration, he/she needs to provide a username, email address, password, confirmation password and nickname. Where the username is a unique identity of each user, it is not allowed to be duplicated or modified. The email address is verified if it is legal (which contains @). The nickname will be the user's most important display information in the future. For security consideration, passwords need to contain 8-16 characters or numbers and should be consistent with confirmed passwords. Any entry field or field that does not satisfy the specification at the submission would trigger a relevant pop-up message. Once registered successfully, the user will view a Welcome Page.

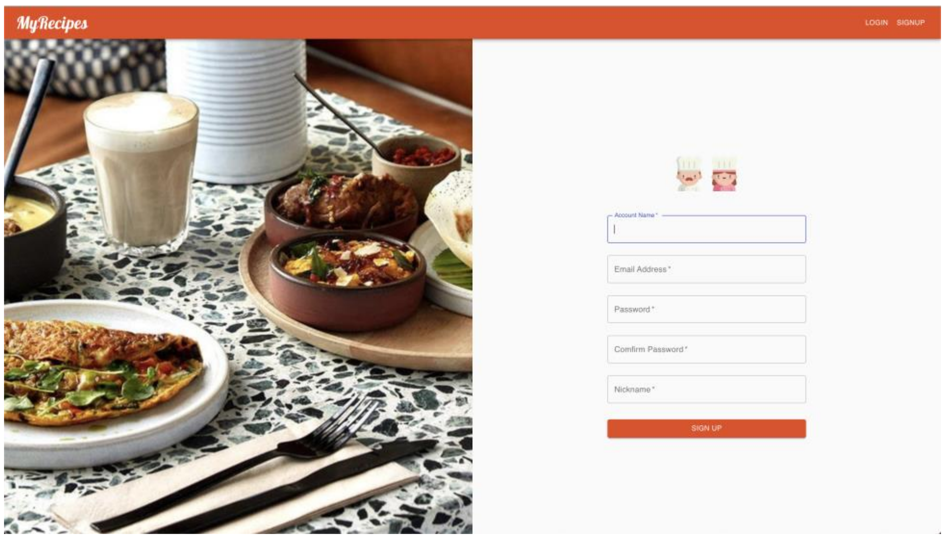


Figure 3.1.2 Sign Up Page

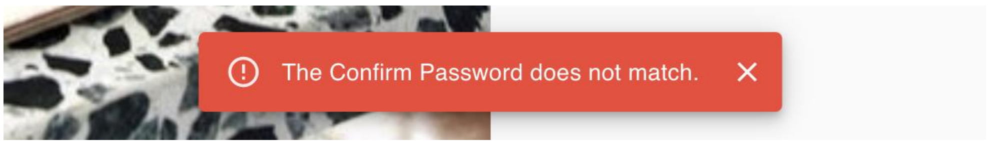


Figure 3.1.3 Confirm Password Not Matching Warning

3.1.1.2 Login

Once users have successfully registered their account, they can enter the Login Page by clicking the login button in the Navigation Bar and login via the username and password they used to register. Once the user has been authenticated by our server backend, they will receive an authentication Token, which allows users to enjoy all the functionalities provided by MyRecipes until they log out. We maintain a Token replica locally, and the next time users open our website, they won't need to log in again. This Token replica will be removed when the user logs out. Once logged in successfully, the user will see a Welcome Page.

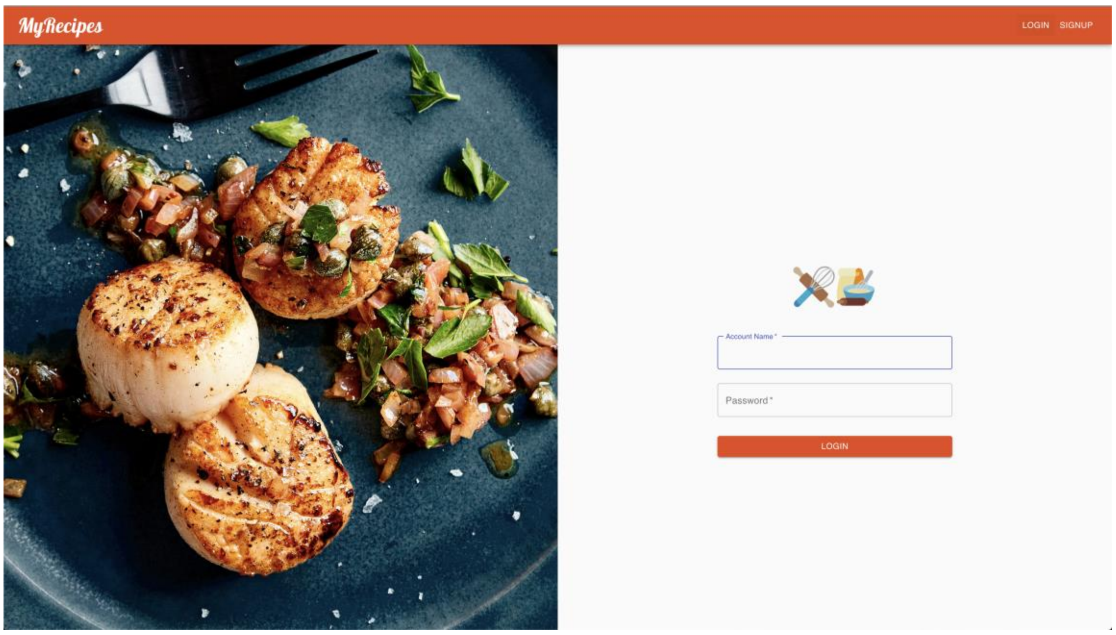


Figure 3.1.4 Login Page

3.1.1.3 Welcome Page

In order to attract more users to join our community and allow new users to attain a better experience of the services we provide, we set up a Welcome Page to display our website's high-quality and popular recipes. On the one hand, it helps users better understand the services provided by our website (Recipes Publishment, Subscription of Recipes, etc.), on the other hand, it also shows the content of our website. When the user has not logged in or registered, clicking action on the Welcome Page will lead to the Sign-Up Page. After login successfully, the user will return to the Welcome Page, and they are free to explore the popular recipes we show up on the Welcome Page.

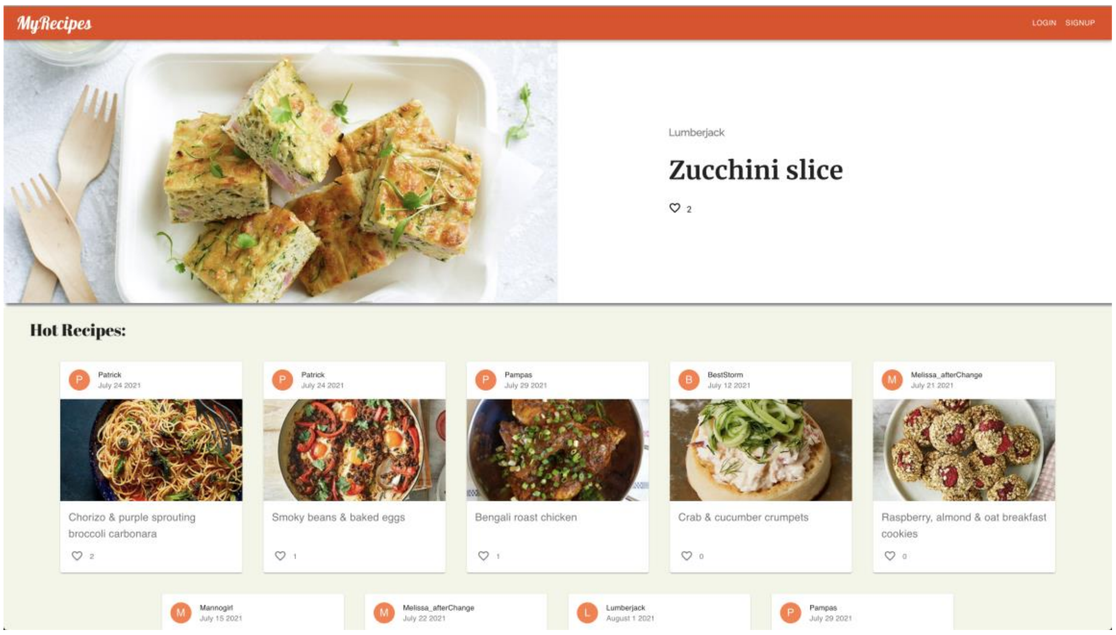


Figure 3.1.5 Welcom Page

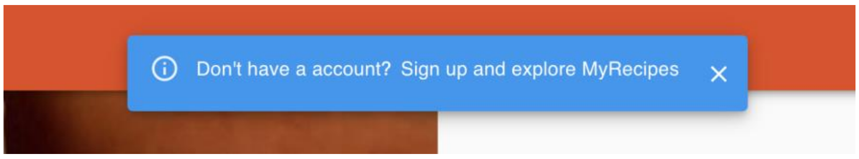

Don't have a account? Sign up and explore MyRecipes

Figure 3.1.6 Welcome Page Warning – Sign Up

3.1.2 Profile Page

After user login, the system would allow user to view and manage their personal information as well as a list of recipes which has been published for all explorers to access.

3.1.2.1 Profile Detail

This information can be accessed through the profile page by clicking on the top right Profile icon of the menu bar. On this page, users' personal information will be listed, including username, contact details (email address), number of followers, number of followings and their avatar.

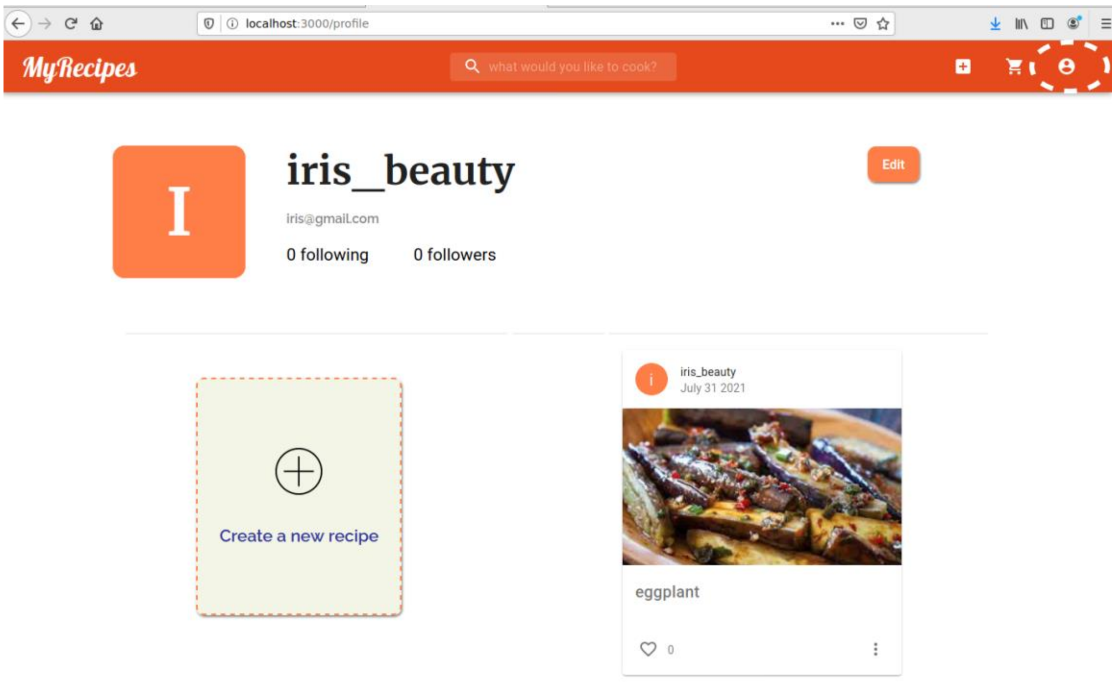
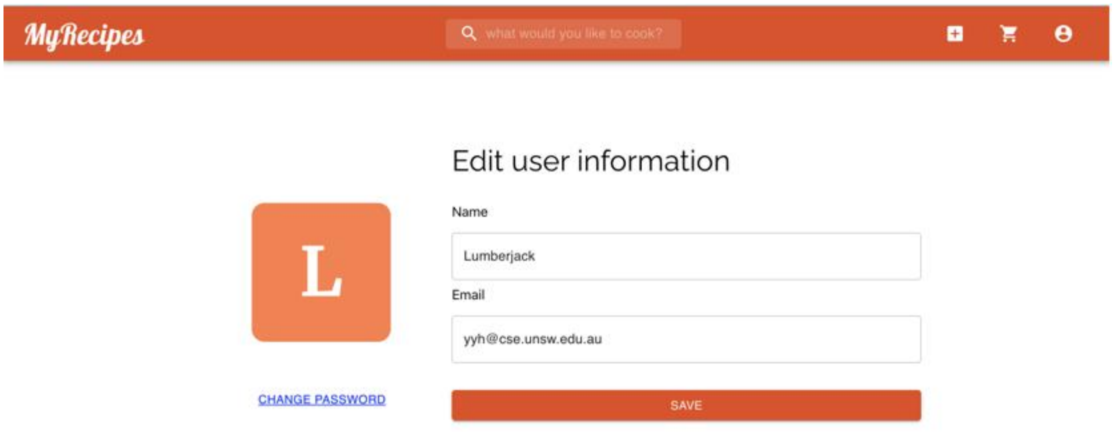


Figure 3.1.7 User Own Profile

3.1.2.2 Edit Profile Information

Users have permission to change their own username and email on their profile page by clicking the 'edit' button. The information of name and email has been prefilled within the text box. The changes will be saved and updated on the profile page after clicking the save button.



Name: Lumberjack

Email: yyh@cse.unsw.edu.au

Figure 3.1.8 Edit User Information Page

3.1.2.3 Edit Password

Users can also change their password on the 'editing user information' page by clicking the 'change password' under the avatar. Users need to input the old password, new password and confirm the new password. There are several validations for the password setting, including new password and confirm new password which need to be the same, and no empty item is allowed. Also, if the old password does not match with the record, the error alert will be raised. After that, if all inputs are valid, by clicking the submit button, a new password will be saved to the system and updated, the profile page will be renavigated to.

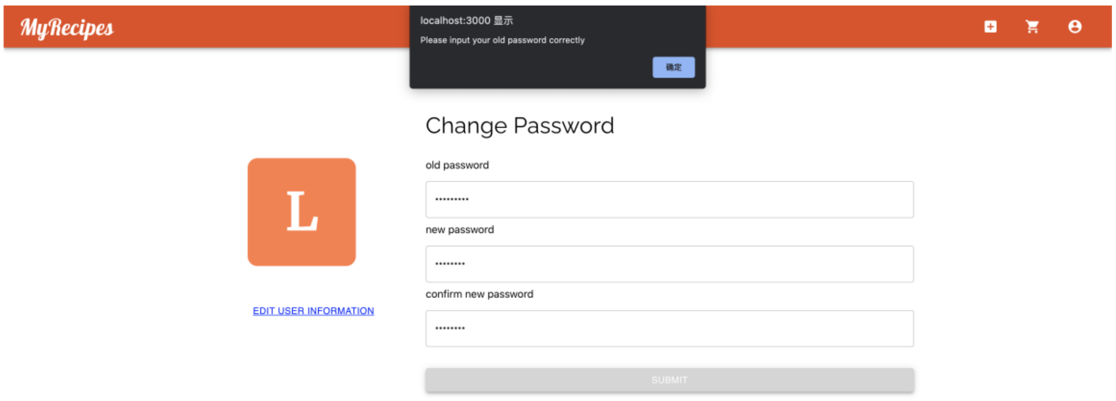


Figure 3.1.9

3.1.2.4 Other’s Profile

In addition, when users browse some attractive dishes on the welcome page, they would like to view the contributor’s personal information or some other similar type dishes from the same contributor. Hence users could click on the avatar on the recipe card to visit different users’ profile pages.

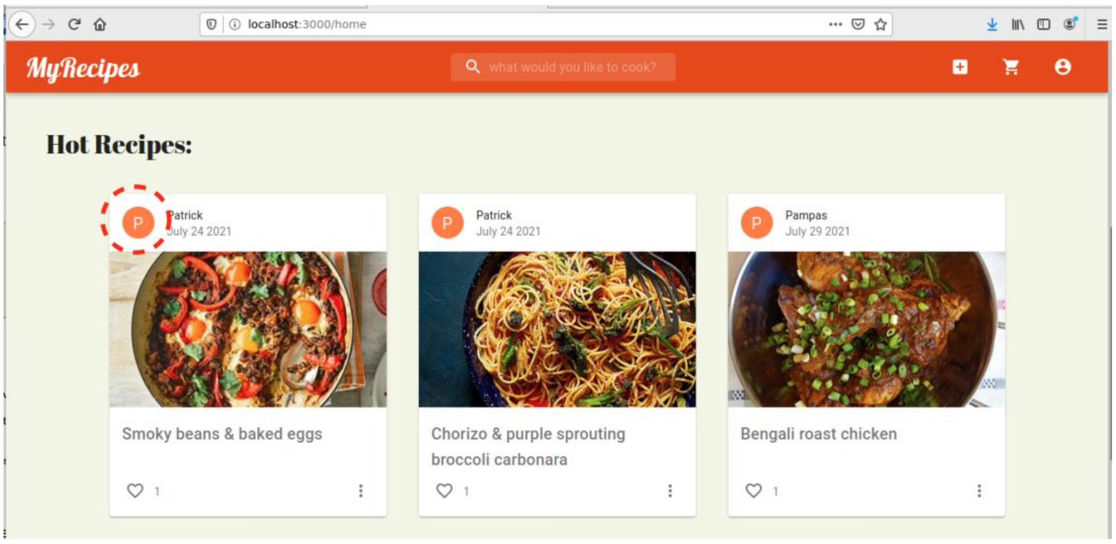


Figure 3.1.10 Welcome Page Showing Recipe Cards

On this page, corresponding recipes created by the contributor will show on the profile page. Besides that, users' personal information has also been posted, including number of followers, number of followings, the name of the user and users' email address.

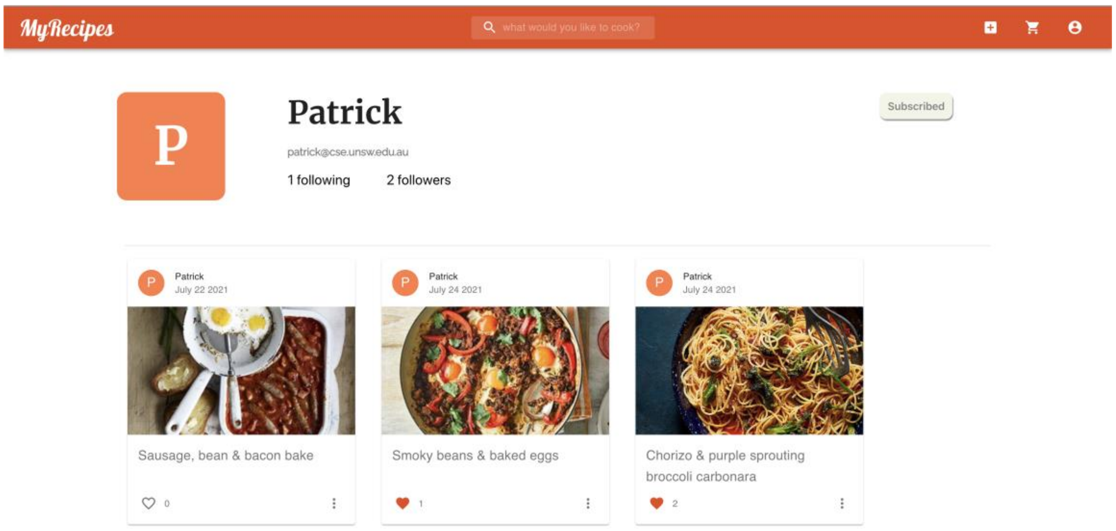


Figure 3.1.11 Other's Profile

3.2 Objective – Maintain Recipe

The MyRecipes platform must allow contributors to maintain a set of their own recipes, (create/edit/delete recipes), that all platform explorers can look through, with each recipe requiring a name, ingredients, method, meal-type(s) and a photo. After users log in to the MyRecipe website and browser content, users could manage their own recipes.

3.2.1 Create Recipe

If users want to create their own recipe, they have two options, one is clicking on the 'create' button from their profile page. Users could also create recipes by clicking the top right 'create' icon through the menu bar.

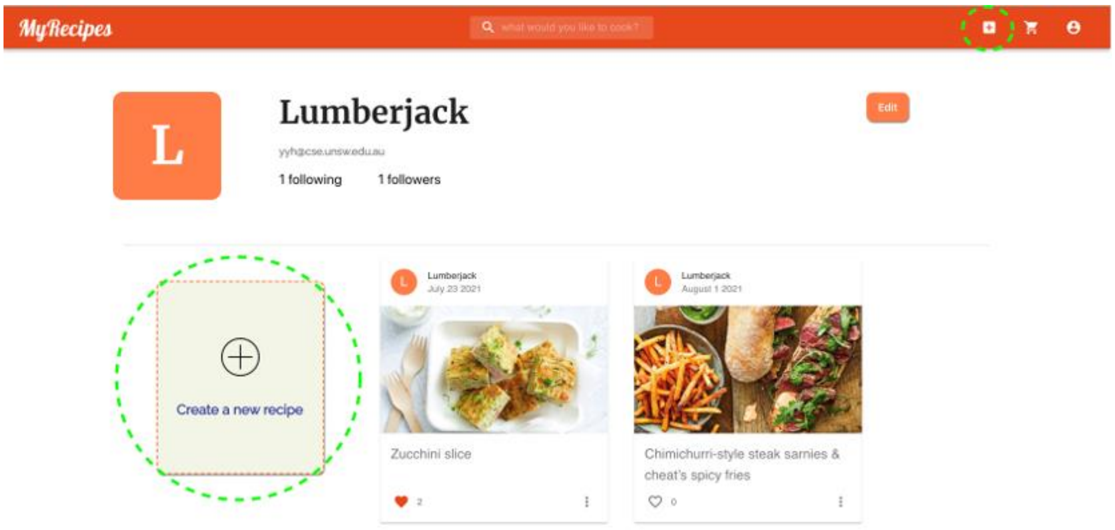


Figure 3.2.1 Create Recipe from Profile

In order to create a valuable and informative recipe, the following information is needed, including recipe name, dishes' image, meal type, cooking method, ingredients, and step descriptions. Specifically, dishes Image could help explorers to have an idea about how it looks like, also the image will become this recipe' visual representative after it has been posted.

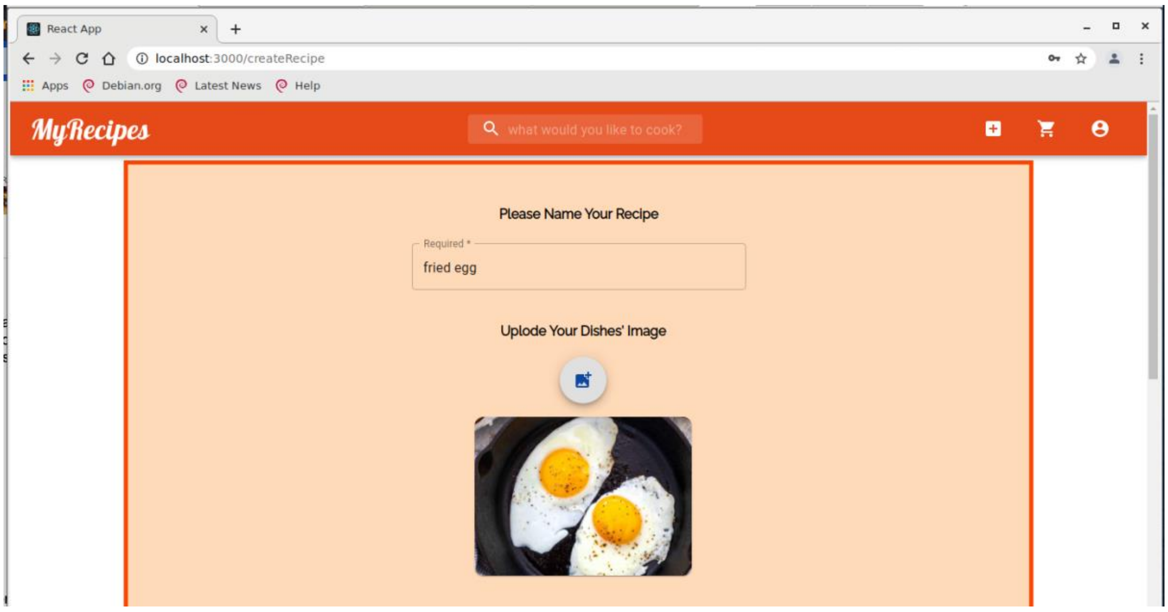


Figure 3.2.2 Create Recipe Page – title & photo

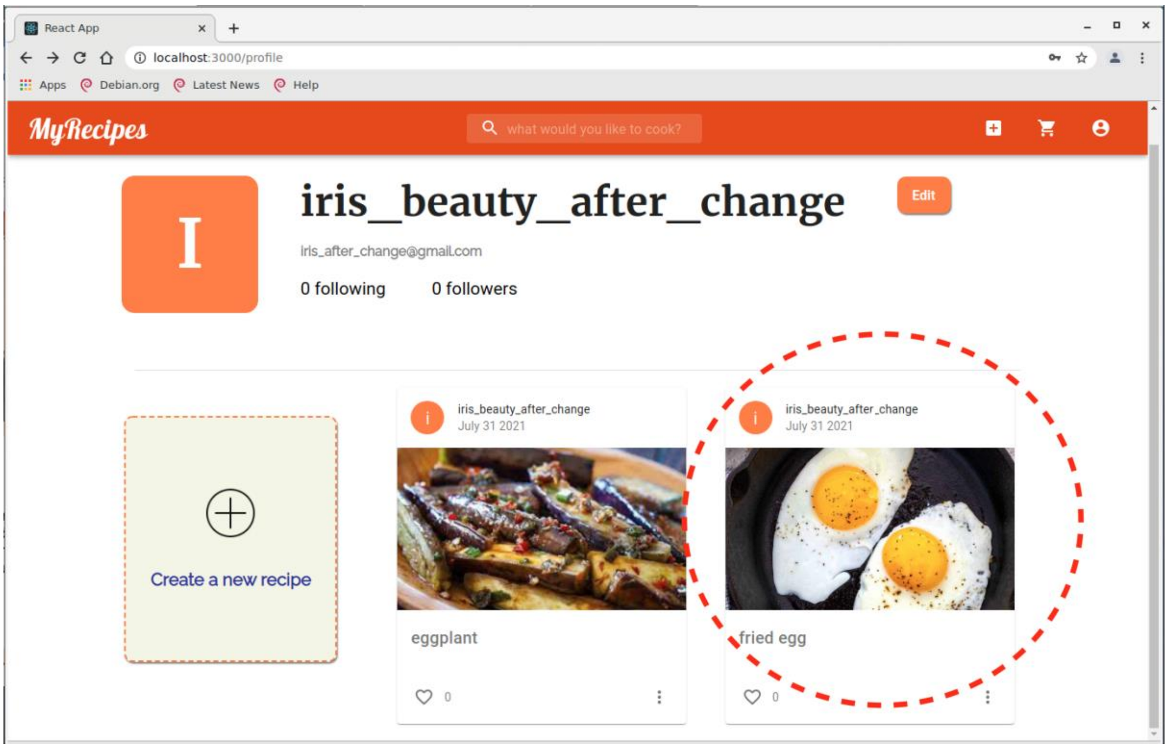
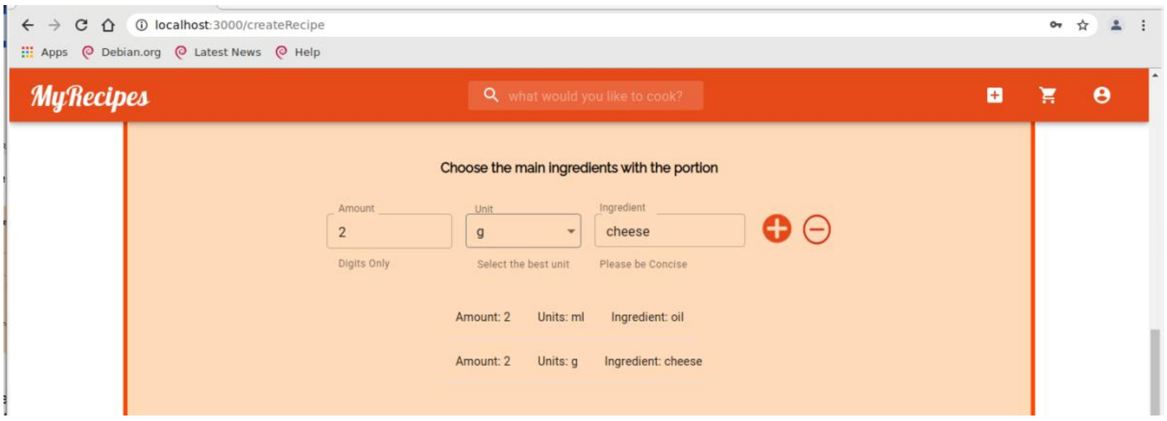


Figure 3.2.3 Recipe Photo from Profile

For the meal type, six categories have been set, such as main, breakfast, desert, starter, snack and beverage. In addition, the cooking method also contains six types, for example steaming, frying, roasting, boiling, smoking and baking. In terms of ingredients, users could add all ingredients related to the dish. They could specify the amount, which is two decimal digits only; units, which include cup, g, ml, oz, spoons and the concise ingredients as well. After users' edition, detailed ingredients will be listed beneath the ingredient box as a preview.



Choose the main ingredients with the portion

Amount	Unit	Ingredient
2	g	cheese
Digits Only	Select the best unit	Please be Concise

Amount: 2 Units: ml Ingredient: oil

Amount: 2 Units: g Ingredient: cheese

Figure 3.2.4 Create Recipe Page - ingredients

Users can add steps by clicking the 'plus' button and they could delete or recall the unwanted step by pressing the delete button. Similarly, the detailed steps will also show under the step container for users' double check before saving and submitting.

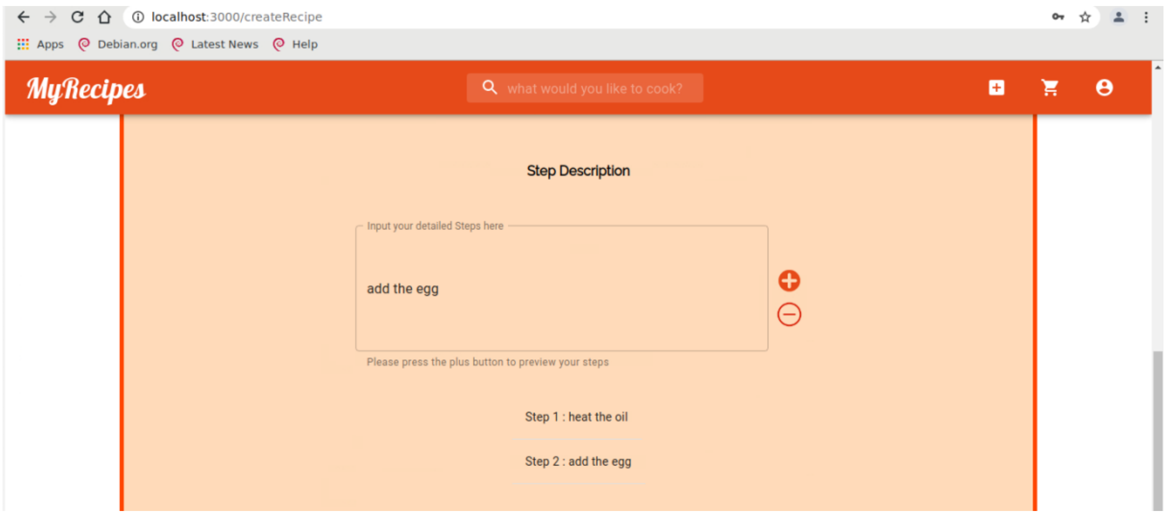
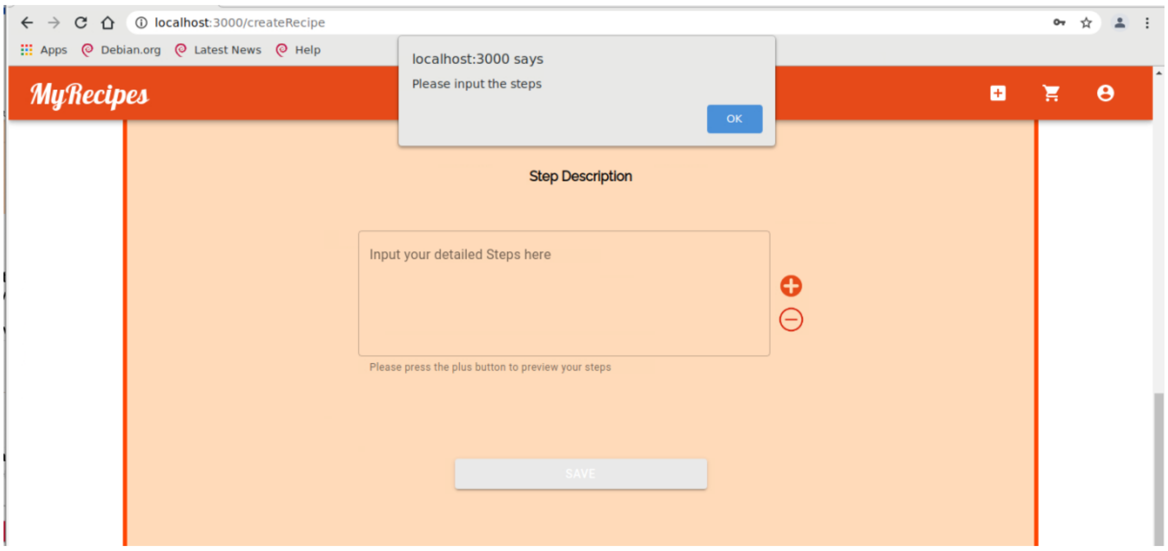

localhost:3000/createRecipe
Apps
Debian.org
Latest News
Help
MyRecipes
what would you like to cook?
Step Description
Input your detailed Steps here
add the egg
Please press the plus button to preview your steps
Step 1 : heat the oil
Step 2 : add the egg

Figure 3.2.5 Create Recipe Page – Step Description

Followed by that, after pressing the save button, the program could check if all items have been filled, otherwise there will be an alert claiming different types of error information. Then, if all requirements have been met, it will redirect users to the profile page to view their post.


localhost:3000 says

Please input the steps

OK
localhost:3000/createRecipe
Apps
Debian.org
Latest News
Help
MyRecipes
Step Description
Input your detailed Steps here
Please press the plus button to preview your steps
SAVE

Figure 3.2.6 Create Recipe Page – Step Description Empty Alert

3.2.2 Edit Recipe

After users post recipes, they might find mistakes or errors from their previous recipes, or they have a better or improved version of recipes, then they could edit their recipe to update.

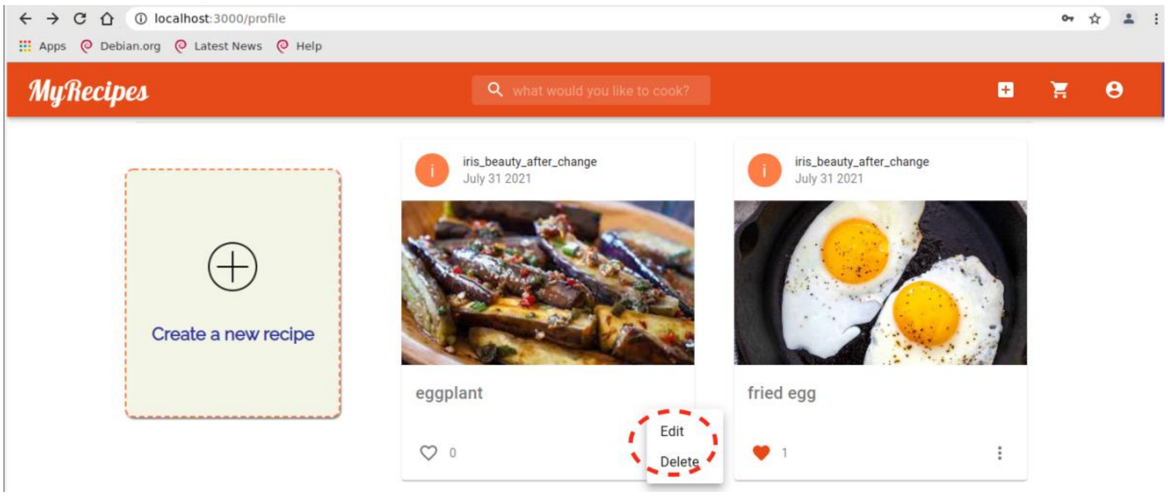


Figure 3.2.7 Edit/Delete Recipe from Profile

For editing recipes, all information has been already prefilled which provides convenience for users to edit instead of recalling and restating it again. All categories of creating recipes have remained within the editing page. Users could rename their recipe; reupload the dish image; reselect their meal type and cooking method; add or delete ingredients and step descriptions as well. Then after clicking the save button, information has been updated and can be checked by viewing the recipe detail page.

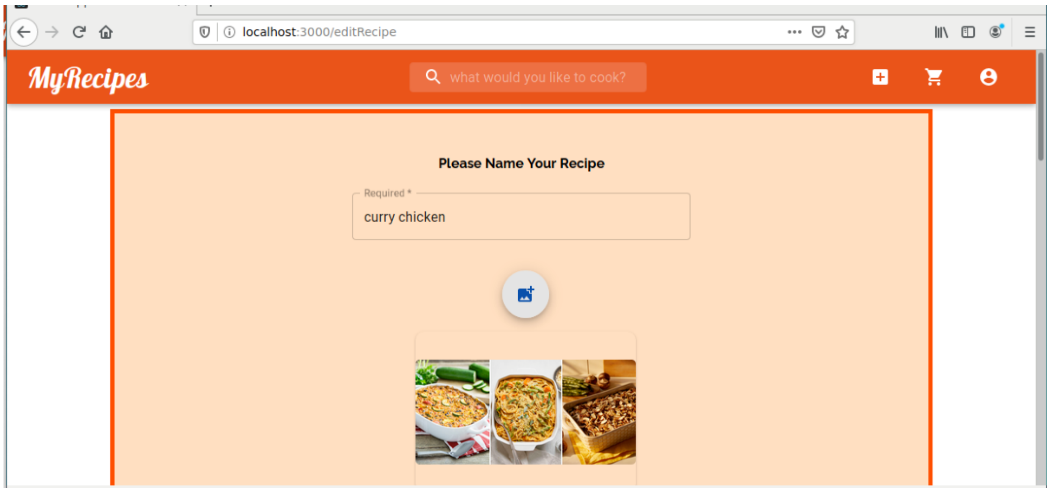


Figure 3.2.8 Edit Recipe Page - Prefilled

3.2.3 Delete Recipe

MyRecipe website also provides users options to delete their recipe if they do not want to keep it anymore. Click the button delete from Figure 3.2.7, then the system would automatically refresh and show the rest of the recipes under users' name.

3.2.4 View Recipe

MyRecipe provides opportunities for users to view their recipe as positive feedback after they create, edit or delete recipes. Users could view all recipes they have posted on profile page with short and concise introductions. In the MyRecipe website, the format of recipe cards is in one same format to maintain the consistency. It shows the creator's name, creation time, dish title and number of likes. Users could change the content on the recipe card by editing their recipe. In addition, clicking the recipe card will redirect users to the detailed recipe information page.

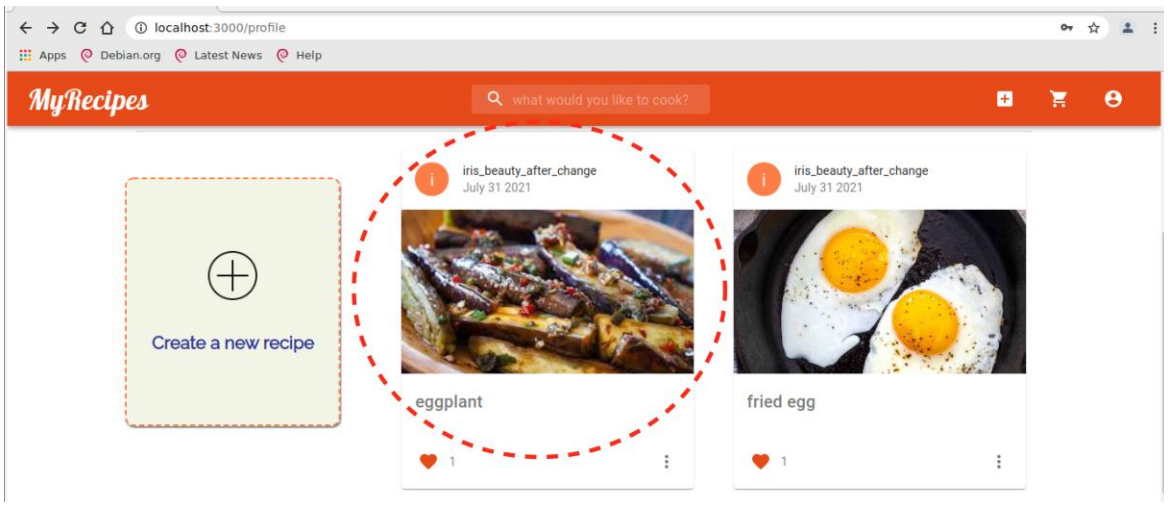


Figure 3.2.9 Recipe Card from Profile

3.3 Objective – Convey Opinion

Recipe explorers on the platform must also be able to convey their opinion for any recipe by either liking and/or commenting on it (note: the total number of likes that an explorer has for a given contributor's profile refers to: the total number of recipes belonging to the given contributor that this explorer has liked).

3.3.1 Like a Recipe

There is a heart-shaped button on both the bottom of each recipe card and on the recipe detail page beside the recipe name on the right. Users can click it and express their love for this recipe. If a user has already liked this recipe before, the heart will be pre-filled after the page is loaded. Re-clicking a filled heart button will cancel the like. There's also a number beside the button indicating how many people have liked it.

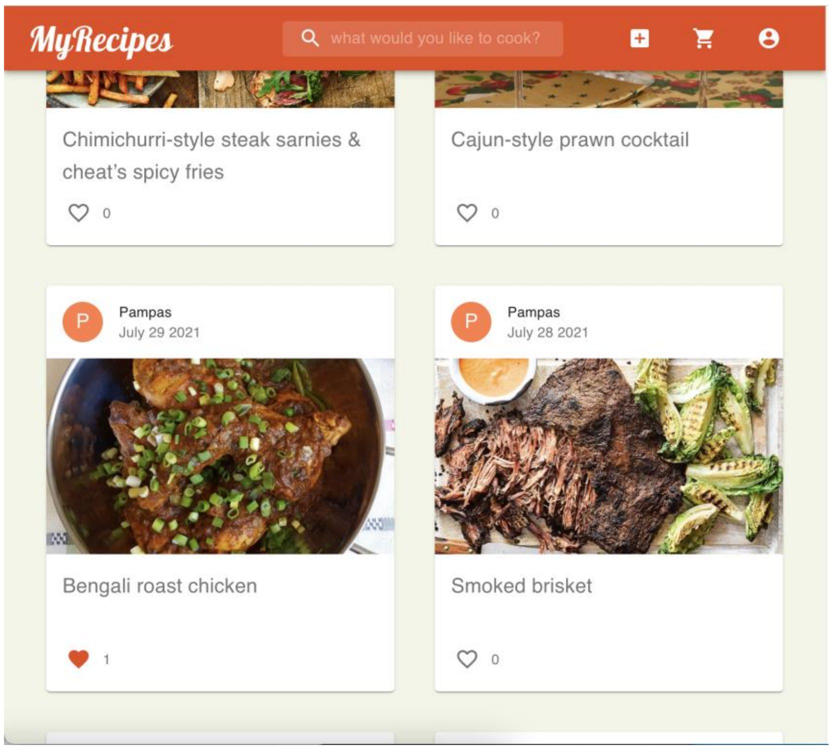

MyRecipes
what would you like to cook?
Chimichurri-style steak sarnies & cheat's spicy fries
0
Cajun-style prawn cocktail
0
P
Pampas
July 29 2021
Bengali roast chicken
1
P
Pampas
July 28 2021
Smoked brisket
0

Figure 3.3.1 Like a Recipe from a Recipe Card

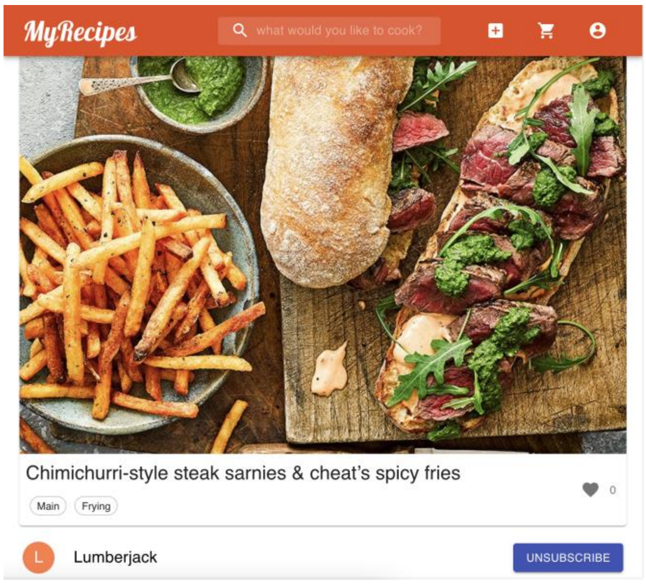
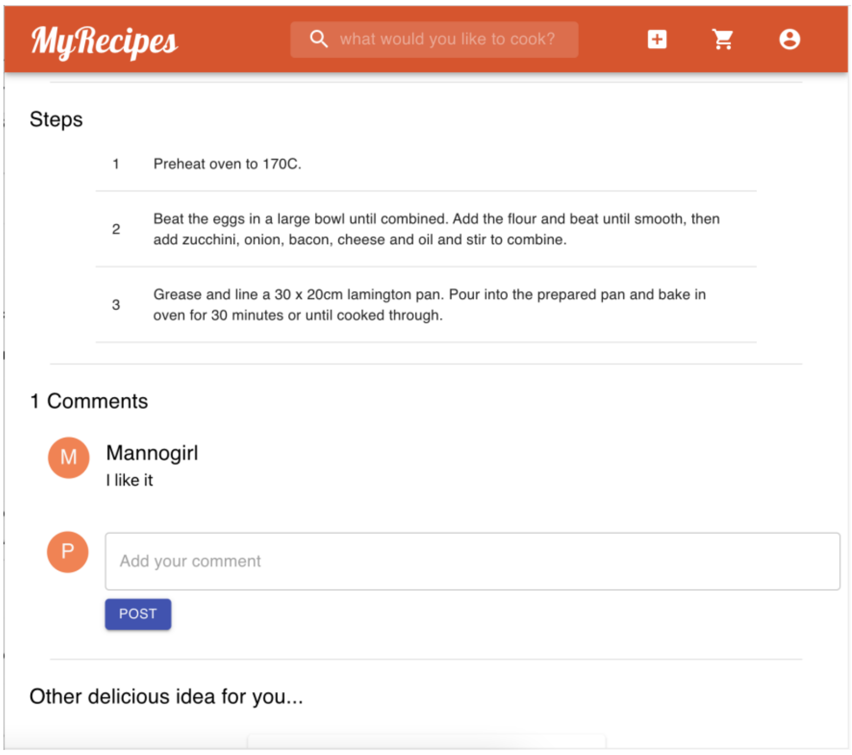

MyRecipes
what would you like to cook?
Chimichurri-style steak sarnies & cheat's spicy fries
Main
Frying
0
L
Lumberjack
UNSUBSCRIBE

Figure 3.3.2 Like a Recipe from Recipe Detail Page

3.3.2 Recipe Comments

On the recipe detail page below the recipe information, there is a comment area where users can see all comments on this recipe. They can also post their own comment by putting the texts in the input area and click the post button. Then their comment will show at the bottom of the comment list immediately.



Steps

1 Preheat oven to 170C.

2 Beat the eggs in a large bowl until combined. Add the flour and beat until smooth, then add zucchini, onion, bacon, cheese and oil and stir to combine.

3 Grease and line a 30 x 20cm lamington pan. Pour into the prepared pan and bake in oven for 30 minutes or until cooked through.

1 Comments

M Mannogirl
I like it

P

Other delicious idea for you...

Figure 3.3.3 Recipe Comments

3.4 Objective – Search

MyRecipes must also provide its explorers with the ability to find recipes they are interested in based on any combination of the following: ingredients, method, meal-type(s), recipe name; where a resulting list of entries should show a summary that includes the name, and photo thumbnail for each recipe that they can navigate to the details from.

3.4.1 Search Page

Once logged in, users can see a Search Bar in the center of the Navigation Bar. Users can input recipe title they are interested in searching for. Once the search is successfully submitted, the user is redirected to the Search Results Page.

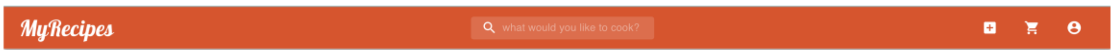


Figure 3.4.1 Search from Navigation Bar

In the Search Results Page, we provide more fields for a wider range of searches. The search fields include:

- Recipe Title: A text input box for providing keywords to the backend with a fuzzy search.
- Ingredient: A text input box which can accept multiple ingredients separated by spaces as a combination.
- Method: A drop-down menu where users can choose the way they cook.
- meal type: A drop-down menu where users can choose their preferred meal type.

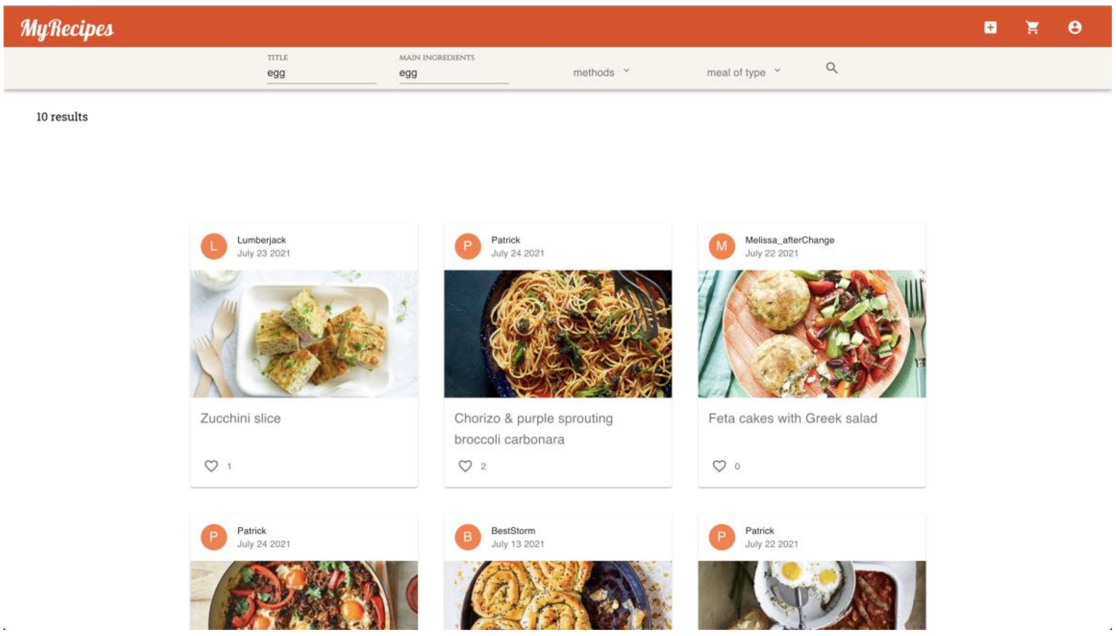


Figure 3.4.2 Search Result Page

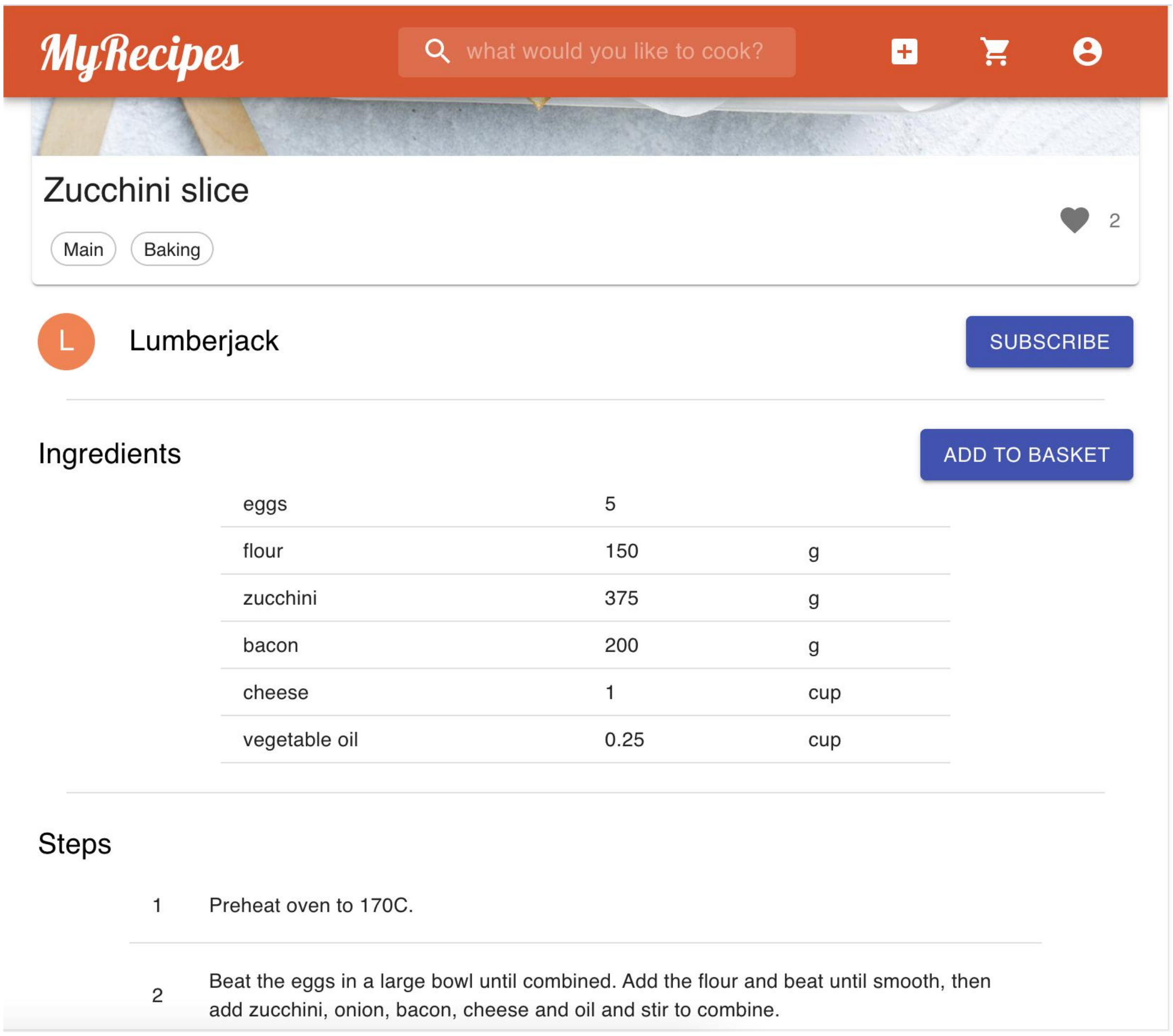
When the user starts searching, the front-end posts the data of these four fields to the back-end together. The back-end will compare the words in each field to their respective fields in the database, returning the result that contains the words or where the similarity is greater than a certain degree (further discussed in challenge section). Meanwhile, the similarity of each field will be taken to calculate a weighted average where similarities from the four factors (title, ingredient, cooking method, meal-type) are normalised. Then the recipes are sorted based on a weighted average and present as recipe cards

3.5 Objective – Recipe Detail

Explorers must be able to navigate to the full details of any given recipe they find, where these full details include: the recipe name, ingredients, method, meal type(s), a photo for the recipe, the number of likes for the recipe, and recipe comments.

3.5.1 Recipe Information

The recipe detail page will show all the information about a recipe. On the first screen of the page, users will see the picture of the dish and the recipe name, followed by the name of the contributor. The ingredients and steps are shown in lists. On the top-right corner of the ingredients list, there's an "Add to basket" button. By clicking this button, users can put all ingredients of this recipe into their private food basket. (Refer to basket feature discussed in later section) Once added, this button will become "Remove from basket" and users can undo the operation by re-clicking.



eggs	5	
flour	150	g
zucchini	375	g
bacon	200	g
cheese	1	cup
vegetable oil	0.25	cup

1 Preheat oven to 170C.

2 Beat the eggs in a large bowl until combined. Add the flour and beat until smooth, then add zucchini, onion, bacon, cheese and oil and stir to combine.

Figure 3.5.1 Detailed Recipe Information

3.6 Objective – Subscription

Explorers must also be able to subscribe/unsubscribe to any contributors on MyRecipes, and when an explorer is subscribed to a given contributor, that explorer should, (on their personal recipe newsfeed), be able to see new recipes that are added or have been recently updated by that contributor.

3.6.1 Subscribe and Unsubscribe a Contributor

On other's profile, the "Subscribe" button beside the contributor's name allows the viewer to subscribe to this user and receive his/her recipes in their news feed in the future. Similarly, to the "like" button, if the viewer has already subscribed to this contributor, the button text will become "Unsubscribe", and re-clicking the button can cancel the subscription.

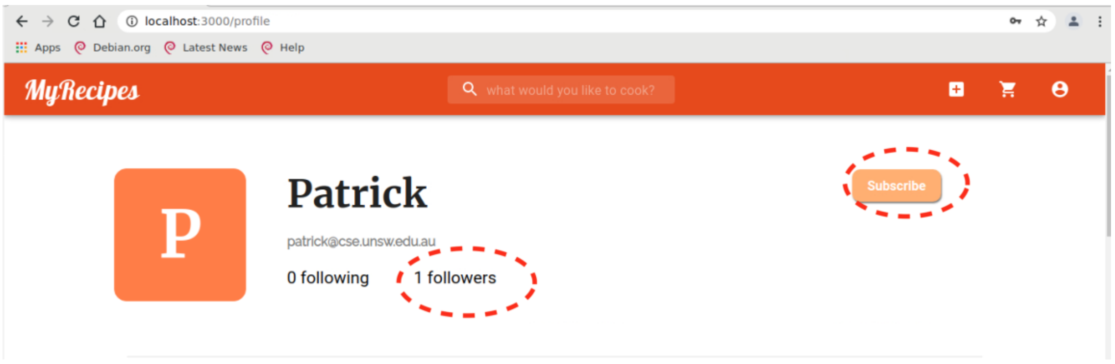


Figure 3.6.1 Subscribe from Other's Profile

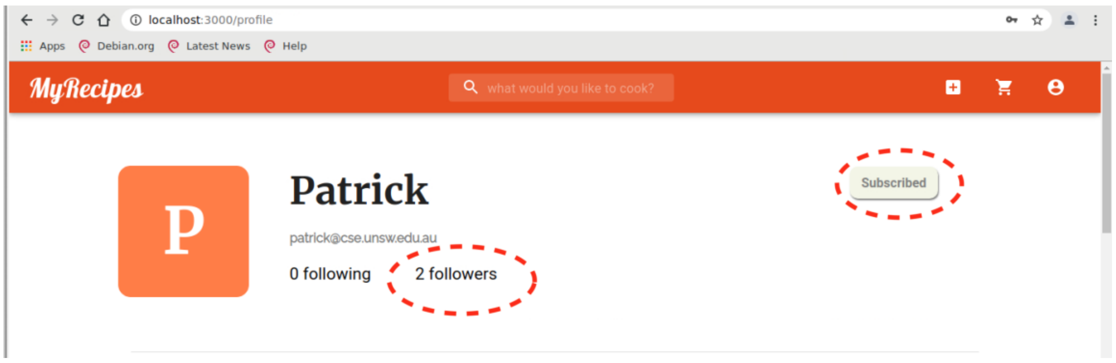


Figure 3.6.2 Unsubscribe from Other's Profile

3.7 Objective – Newsfeed

A given explorer's personal recipe news feed must be sorted based on the recipe creation/update date excluding time component (most recent to oldest), then by the total number of likes that this explorer has for the recipe's contributor profile (highest to lowest count), and then by the recipe creation/update time (most recent to oldest).

3.7.1 NewsFeed Page

Once the user has subscribed some contributors, click MyRecipe Logo on the left side of the Navigation Bar to go to the Newsfeed Page to see all subscribed contributors' recipe thumbnails. The recipes are sorted based on the recipe creation/update date excluding time component (most recent to oldest), then by the total number of likes that this explorer has for the recipe's contributor profile (highest to lowest count), and then by the recipe creation/update time (most recent to oldest)

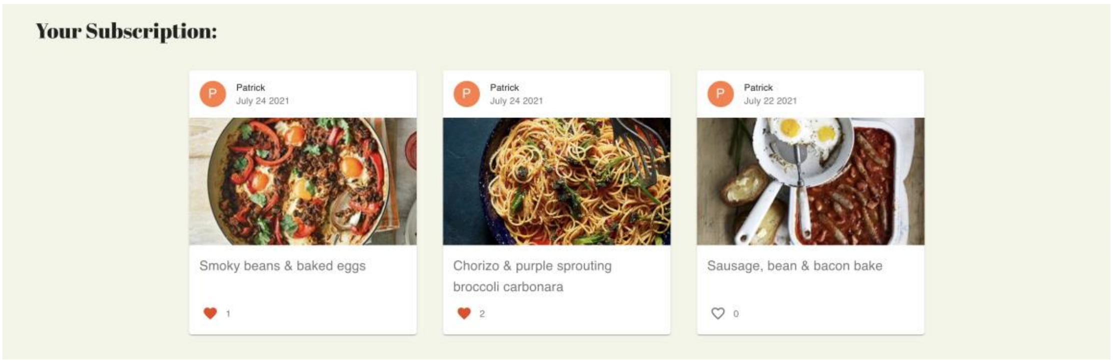


Figure 3.7.1 Newsfeed Page

3.8 Objective – Recommendation

MyRecipes must provide recommendations for recipes that are similar to a given reference recipe based on that reference recipe's ingredients, with recommendations to be ordered based on how close each result is to the reference recipe's ingredients (design a metric to represent closeness between

2 sets of ingredients, and sort results from closest match to reference recipe ingredients to furthest match; and exclude any recipes from results that have no ingredients in common to the reference recipe).

3.8.1 Recommendation based on Jaccard Similarity

At the very bottom of the recipe detail page, there are some recommended recipes with similar ingredients if there's any. The recommendations are ordered by Jaccard similarity between two recipes based on their ingredient. While a request to get details from a specific recipe is sent to back-end, back-end will also return a list of recipes based on similar ingredient compared to the specific recipe. When comparing two recipes, their ingredients are stores into two sets. The similarity is calculated by the size of intersection divided by the size of union. When determine the size, fuzzy match is applied to recognise two words which have minimal difference as one word, an example would be "banana" and "banala".

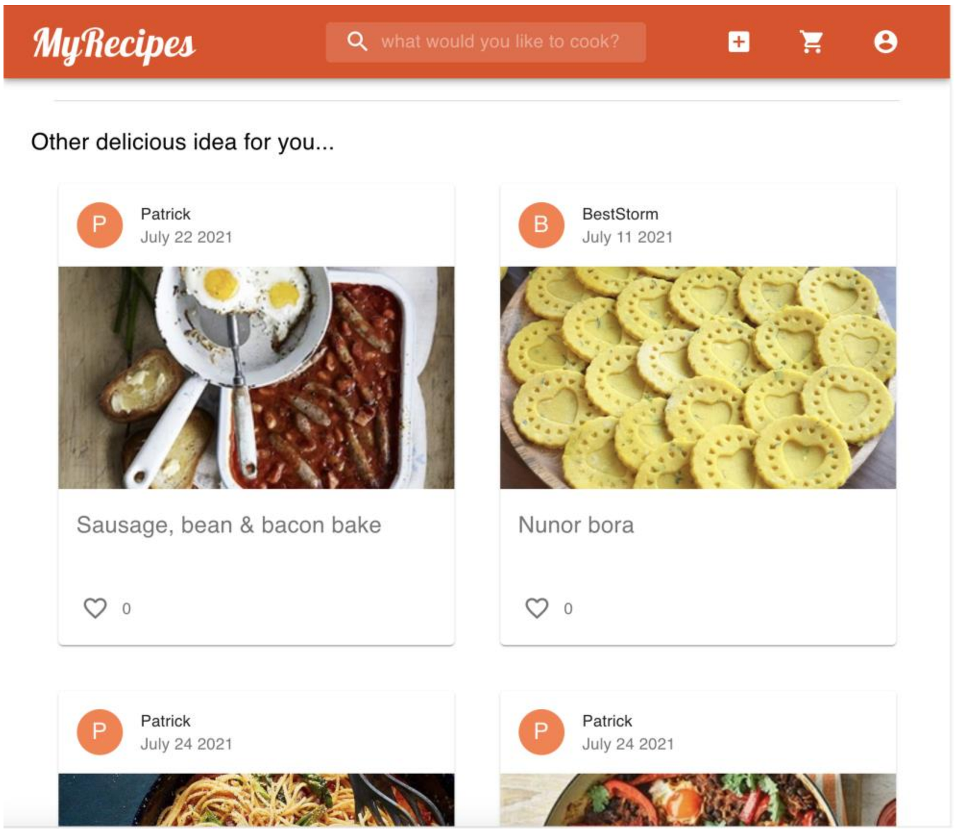


Figure 3.8.1 Recommendation based on ingredients

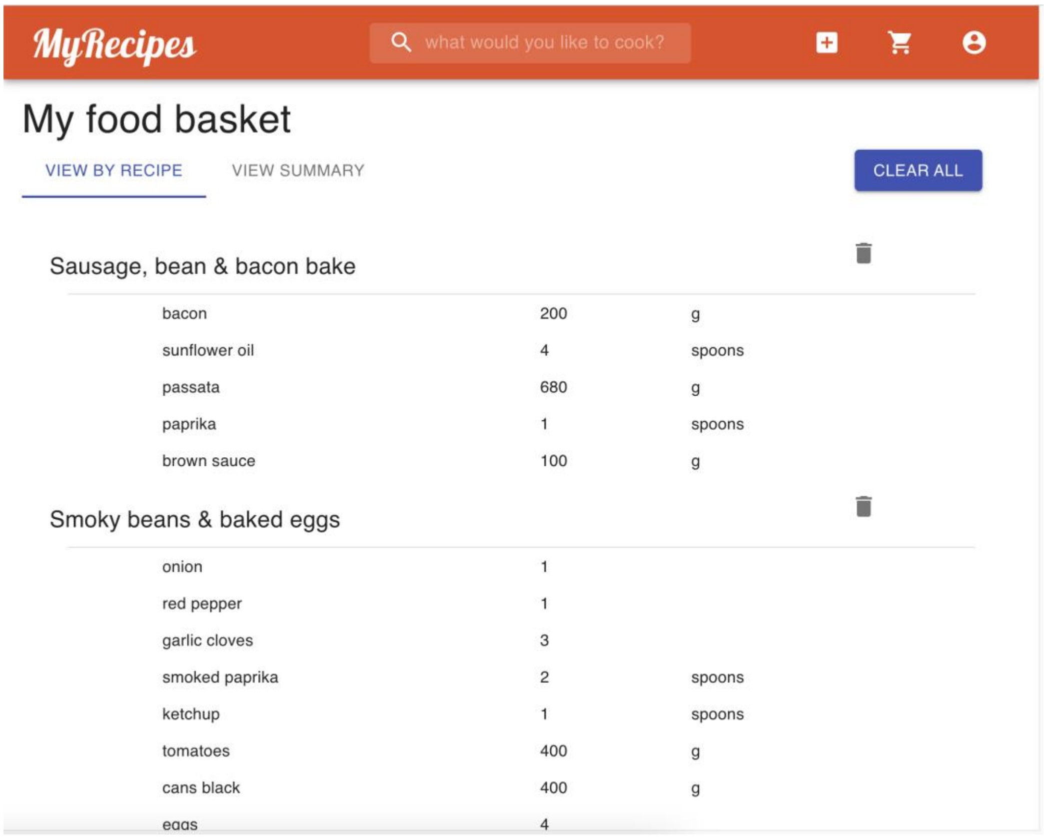
3.9 Objective – Novel function – Food Basket

3.9.1 Food Basket

The Food basket is a novel feature we designed on top of project objective. It is designed to help users to gather all the ingredients of the recipes they would like to try in one place. The ingredients were added to the food basket when users are browsing the recipe on the detail page. The food basket can be used as a shopping list and provide 2 types of views. Users can choose by clicking on the tabs “View by recipe” or “View Summary”. In either of the 2 views, users can delete everything in the basket by clicking the “Clear all” button.

3.9.1.1 View by Recipe

In this view, the ingredients of each recipe are listed under the corresponding recipe name. There’s one “delete” icon beside each recipe name which can be used to specifically delete ingredients of this recipe.



My food basket

Sausage, bean & bacon bake

bacon	200	g
sunflower oil	4	spoons
passata	680	g
paprika	1	spoons
brown sauce	100	g

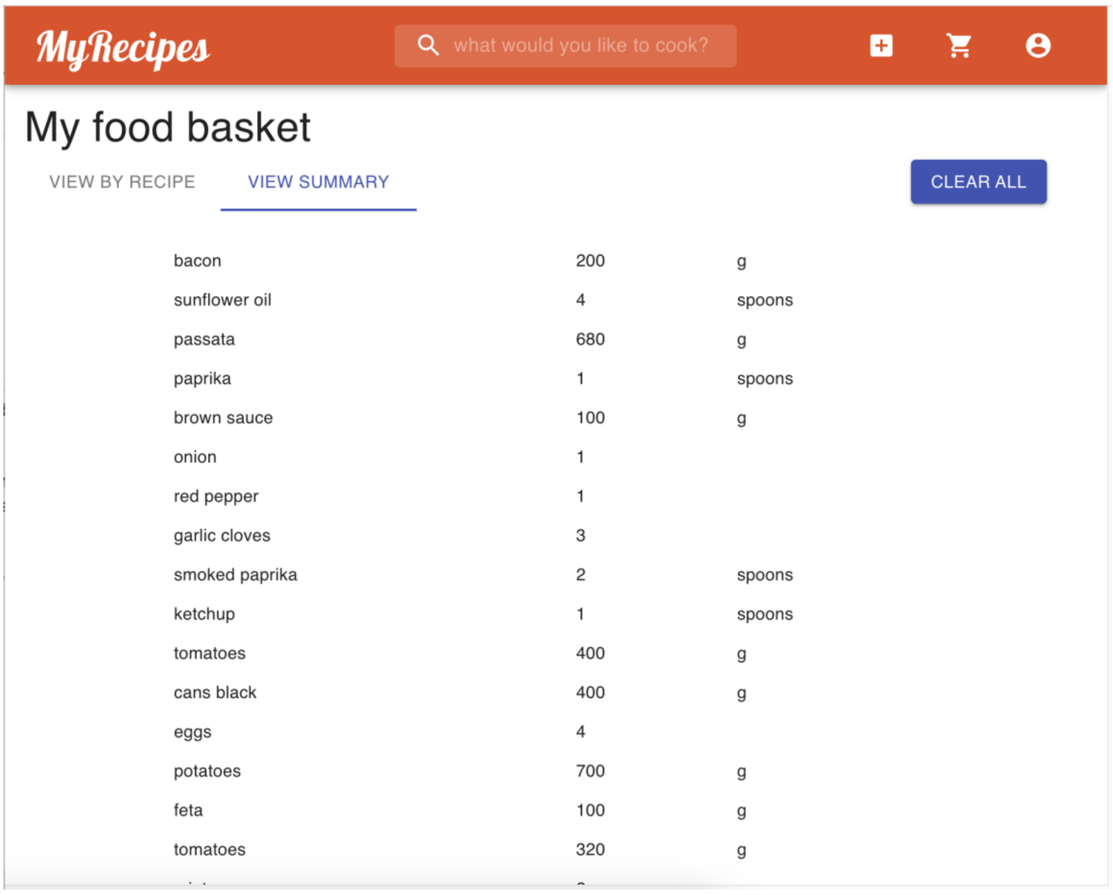
Smoky beans & baked eggs

onion	1	
red pepper	1	
garlic cloves	3	
smoked paprika	2	spoons
ketchup	1	spoons
tomatoes	400	g
cans black	400	g
eggs	4	

Figure 3.9.1 Food Basket from Different Recipes

3.9.1.2 View Summary

The summary view will show an integrated list of all ingredients from all the recipes add to the basket, and the quantities of the same ingredients will be added together.



My food basket

bacon	200	g
sunflower oil	4	spoons
passata	680	g
paprika	1	spoons
brown sauce	100	g
onion	1	
red pepper	1	
garlic cloves	3	
smoked paprika	2	spoons
ketchup	1	spoons
tomatoes	400	g
cans black	400	g
eggs	4	
potatoes	700	g
feta	100	g
tomatoes	320	g

Figure 3.9.2 Food Basket Summary

4 Third-Party Functionalities

4.1 Front-end

4.1.1 ReactJs

ReactJS, an open-source JavaScript library developed by Facebook, is designed to render complex web pages faster. ReactJs has outstanding rendering performance which makes it the most popular front-end framework today. ReactJs breaks down a web page into several simpler components while speeding up the rendering of entire pages by simply re-rendering update components in a way that not only raises the reusing rate of the code but also reduces code duplication (ReactJs, 2021).

ReactJs designs a JSX syntax to help JavaScript better compiles HTML's label syntax and enables browsers to transform code to render components faster. One of the most important concepts of ReactJS is virtual DOM. HTML tags are mounted on the virtual DOM before they are actually mounted to the real DOM. Only the changed or updated component would re-trigger rendering. Compared to traditional JavaScript, the overhead of re-rendering the entire page is greatly reduced.

ReactJs is a powerful front-end framework, which also means that it needs a certain cost of learning. ReactJS design concepts and concepts need to be followed to maximize its performance advantages.

ReactJs is licensed under the MIT license. " Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction" (Facebook, 2021). Its license has no impact on our project.

4.1.2 Material-UI

Material-UI is a components library that started back in 2014 and combines ReactJS and Material Design. Material-UI has grown into one of the most popular React UI libraries in the world. Various universal components library provided by Material-UI are one of the most important reasons why we use it, which greatly reduces the front-end programming workload. On the other hand, Material-UI components provide good browser compatibility and UI flexibility (mui-org, 2021).

Material-UI is licensed under the MIT license. " Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction" (mui-org, 2021). Its license has no impact on our project.

4.1.3 Google Fonts

Google Fonts provides a powerful font library. Different functionality areas on our websites use different fonts to help users more intuitively divide areas and also enhance the richness of our pages.

Google Fonts under Google open-source licenses. "You can use them in any non-commercial or commercial project." (Google, 2021). Its license has no impact on our project.

4.1.4 Font Awesome

Font Awesome is a website that offers partial free icons. We use some icon buttons on the page, and icons can sometimes replace the text to guide the users better. And the beautiful icons further beautify our web page.

Font Awesome Free icon download under the CC BY 4.0 license. "You are free to share (copy and redistribute the material in any medium or format) and adapt (remix, transform, and build upon the material for any purpose, even commercially)." (fontawesome, 2021). Its license has no impact on our project.

4.1.5 Flaticon

Flaticon is a website that provides free icons which adds more icons Font Awesome didn't have.

Flaticon is licensed under Flaticons License Agreement. "Things you can do: Use in any personal or commercial work, Unlimited use, Royalty free, each purchase includes licensed use for 1 User, you may modify or manipulate the icons."(Flaticon, 2021) Its license has no impact on our project since it is free.

4.2 Backend

4.2.1 Django

A "web framework" is a software program that abstracts away many of the common issues associated with website development, such as connecting to a database, managing security, and managing user accounts. Nowadays, most developers rely on web frameworks rather than constructing a website from the ground up.

Django is a free high-level and open-source web framework based in Python. Django was published in 2005 and has been in active development ever since. Today, it is one of the most widely used web frameworks available, as well as being sufficiently adaptable to be a suitable fit for projects (Django, 2021). It Requires Python >=3.6.

Django adopted Python's "batteries-included" approach and comes with support for standard web development tasks such as database models, URL routing, and support for different database backends. This approach enables us to concentrate on the features that make a web application unique, rather than recreating the wheel every time we need to implement conventional, secure web application functionality.

Django has been licensed under Berkeley Software Distribution (BSD-3-Clause) since 2005 and it is an open-source project which is granted free of charge to any person obtaining a copy of the software and its associated documentation (Open Source Initiative, 2021). It’s an open-source license and places little constraints on future behaviour. This enables BSD code to stay Open Source or to be integrated into commercial solutions as the requirements of a project change. It has no impact on our project.

4.2.2 Django REST Framework

Django REST Framework (DRF) is a highly configurable and robust toolkit for developing Web APIs. The Web-browsable API is a significant usability improvement for us. Authentication policies, including OAuth1a and OAuth2 packages. Serialization that is compatible with both ORM-based and non-ORM-based data sources. Fully customizable down to the smallest detail - simply utilise standard function-based views if projects don't require the more complex features. It has extensive documentation and an active community.

The Django Rest Framework simplifies the process of utilising the Django Server as a REST API. While we can create a restful API using standard Django, the Django Rest Framework simplifies the process. We can quickly list and build views with DRF and do authentication.

Django REST Framework is distributed under the BSD license (Open Source Initiative, 2021). It's an open-source license and places little constraints on future behaviour. This enables BSD code to stay Open Source or to be integrated into commercial solutions as the requirements of a project change. Its license has no impact on our project.

4.2.3 Django-rest-knox

Knox is a simple-to-use authentication system for the Django REST framework. The goal is to enable typical patterns in REST-based applications with minimal additional work; and to ensure that connections stay safe.
Knox authentication is token-based, comparable to DRF's built-in Token Authentication. Knox, on the other hand, gives a single token per login view call, allowing each client to have its own token that is erased on the server when the client logs out. Knox tokens are only stored in a hash format for added security like a password. Even if the database was compromised in some way, an attacker would be unable to log in using the compromised credentials. This stops an attacker with a token from gaining unfettered access to an account if the database is hacked (McMahon, 2015). Knox tokens can be programmed to expire in the app's settings.

Django-rest-knox is licensed under the MIT license. "Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction" (Open Source Initiative, 2021). It's an open-source license which is permissive. We have permission to use, change, publish, develop something with, and sell the code, as long as we retain the licence and copyright text from the source. It has no impact on our project.

4.2.4 Firebase Storage

Cloud Storage for Firebase is a highly scalable, simple-to-use, and cost-effective object storage service. The Firebase SDKs for Cloud Storage integrate Google security into file uploads and downloads, regardless of network quality. (Firebase, 2021). The front-end components can load images faster than AWS RDS when using Firebase. Additionally, storing photos as urls rather than encoded text speeds up the database.

Firebase is under the Firebase Paid Services Agreement and Google APIs Terms of Service. It allows us to use Python to manipulate Firebase on any application, whether it's commercial or personal, regardless of charge or free Firebase payment tier, as long as we adhere to the terms of service. We are using the Firebase Spark Plan which is free for limited usages. The free plan gives us 1 GiB of data storage (Firebase, 2021). If our file storage capacity is less than 1 GiB, this has no effect; otherwise, we must pay for additional storage space.

4.2.5 Pyrebase

Pyrebase is a lightweight Python library that wraps the Firebase API. Pyrebase was developed for Python 3 and would not function properly in Python 2 (Childs-Maidment, 2020). Pyrebase is a Python interface to the REST API provided by Firebase. Pyrebase enables us to effortlessly upload and download images to Firebase from our website.

Pyrebase is licensed under the MIT license. "Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction" (Open Source Initiative, 2021). It's an open-source license which is permissive. We have permission to use, change, publish, develop something with, and sell the code, as long as we retain the licence and copyright text from the source. It has no impact on our project.

4.2.6 django-cors-headers

It is a Django App that adds Cross-Origin Resource Sharing (CORS) headers to responses. This allows in-browser requests to the Django application from other origins. It Requires Python >=3.6.
By default, the same-origin security policy prohibits certain "cross-domain" requests, most notably Ajax requests. CORS defines a mechanism by which a browser and a server can communicate to assess whether a cross-origin request is safe to authorise. It provides more flexibility and freedom than strictly same-origin requests but is safer than simply permitting all cross-origin queries.
django-cors-headers is licensed under the MIT license. "Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction" (Open Source Initiative, 2021). It's an open-source license which is permissive. We have permission to use, change, publish, develop something with, and sell the code, as long as we retain the licence and copyright text from the source. It has no impact on our project.

4.2.6 PyMySQL

PyMySQL is a pure-Python MySQL client library based on PEP 249 of the Python standard library. The majority of public APIs are compatible with

MySQLdb and mysqlclient. PyMySQL is compatible with MySQL 5.5 and MariaDB 5.5 and above.
PyMySQL is licensed under the MIT license. "Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction" (Open Source Initiative, 2021). It's an open-source license which is permissive. We have permission to use, change, publish, develop something with, and sell the code, as long as we retain the licence and copyright text from the source. It has no impact on our project.

4.2.8 Amazon RDS

Amazon Relational Database Service (Amazon RDS) is a web service that simplifies the process of deploying, operating, and scaling a relational database in the Amazon Web Services (AWS) Cloud (AWS, 2021). It provides scalable capacity for an industry-standard relational database at a low cost and performs common database administration duties. Our cloud database is powered by Amazon RDS.
Amazon RDS is under AWS Service Terms and Amazon Software License. It allows us to use Amazon RDS on any application, whether it's commercial or personal, regardless of charge or free RDS tier, as long as we adhere to the terms of service. The AWS Free Tier is available to us for 12 months starting with the date we create our AWS account. It sets to be expired on 2022 May 1. The free tier also gives us 20 GB of DB Storage. When our free usage expires or if our application use exceeds the free usage tiers, we need to pay standard, pay-as-you-go service rates.

5. Implantation Challenge

5.1 Front-end

5.1.1 Improve User Interface

Problem statement:

After implementing the basic frontend structure of the MyRecipe website, how to make the website be more professional and user friendly has become a serious problem.

Solution:

In this case, there are two categories which have been considered. One is about the visual user interface design, and another is about the user operation's logic. First, in order to draw users' attention when they are browsing MyRecipe website, lots of design principles have been applied,

For users' convenience and to make the website easy to learn, the design principle 'creating consistency and using common UI elements' (Zhang, 2020) has been applied. For example, all icons the website applied can be easily recognized and utilized, which provides high readability and clarity.

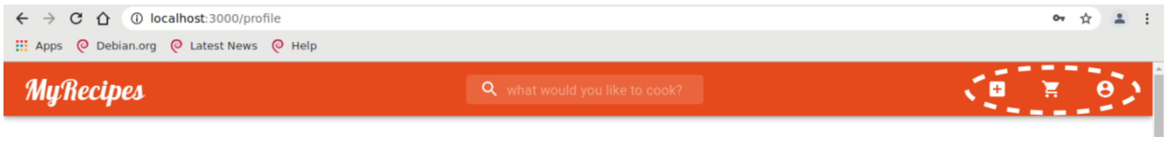
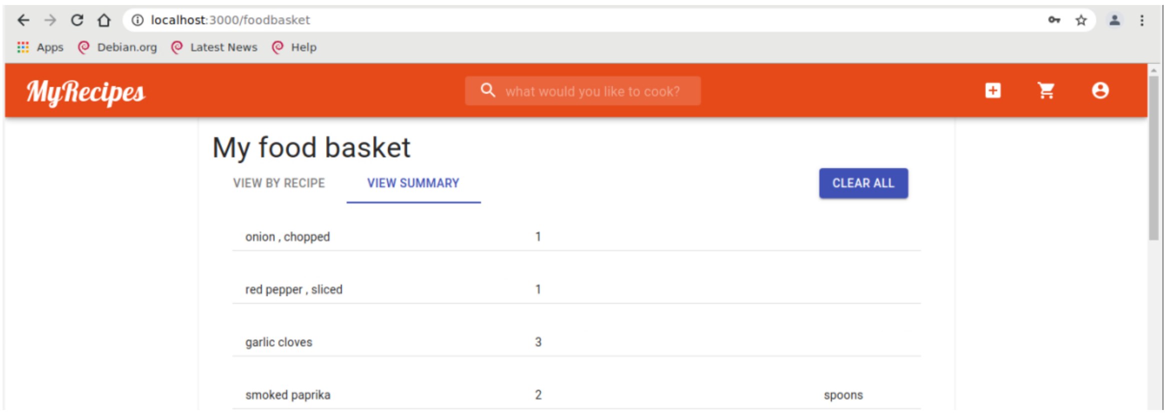


Figure 5.1 Icon in Navigation Bar

In addition, the main theme colour has been set to orange, because through the research, warm colour brings users' emotion with warmth, brightness, passion and happiness (Smashingmagazine, 2010), which has the similar effect when users see recipes and food. In addition, in order to better illustrate the novel function 'basket' of MyRecipe, the design principle 'keep the interface simple' (User Interface Design Basics | Usability, 2020) has been applied. It avoids the unnecessary messages, elements and helps users more concentrate on the functions and information they have.



My food basket

onion , chopped	1	
red pepper , sliced	1	
garlic cloves	3	
smoked paprika	2	spoons

Figure 5.2 User Interface

Besides that, to make the program navigation smoother, the logic behind each operation has been considered carefully. Since all group members are part of the development team, other unrelated users are invited to test and compile the program. Hence from their feedback, the page's loading time, the shorter the better. In addition, after users' operation such as creating and editing their recipe, the most updated page should be shown automatically on their profile page instead of manually refreshing it. So, JavaScript call back component API Use Effect has been added to fetch the updated features automatically.

Therefore, to better satisfy user requirements, buffer time still needs to be minimized (the image storage will be discussed in the backend challenge sector) and then improve program performance.

5.2 Back-end

5.2.1 Photo Storage

Problem statement:

When users create or edit recipes, they can upload images. If these images are posted and kept locally, they are inaccessible to other users. If the image is encoded and then stored in the database, the database must accommodate a very long string. This results in the transmission of large amounts of data during front-end and back-end communication, as well as between the back-end and the database. This results in slower front-end component rendering and slower database insert and query operations.

Solution:
We implemented the cloud storage function for images using firebase storage. When users create a recipe, they have the option of uploading their own local pictures on the front end. The front end first encodes the selected file in 64-bit and then sends it together with other recipe data to the backend. The backend will decode the encoded image and then upload the restored binary image via the firebase API to the specified folder in the firebase storage, naming the image with the user id plus the current time to ensure the image is unique. When the recipe creation information is stored in the database, the image's URL is also stored. Uploading the image to firebase via the backend rather than the frontend avoids inconsistencies in the final file and data when the user creates a recipe but the recipe's creation request to the back end fails, or when the firebase upload fails.

When editing a recipe, if the image is changed, the backend will decode the encoded string transmitted from the frontend and get the image path according to the image's URL. Then the backend will use the firebase API to update the image. The URL will always remain unchanged throughout the database restore. When deleting a recipe, the backend will also delete the accompanying image via the API using the path parsed by the image URL.

5.2.2 SQL Table Design

Problem statement:

The project adopted a MySQL database established with AWS API. Initially, the recipe table is designed to store the number of likes and comments, the user table was designed to include a list of user id he subscribed to, and the basket table stores a list of recipe ids. The initial design had two issues, the first was when modifying one table, another duplicated request must be sent to relevant tables, which in turn may lead to data anomalies. For example, if a recipe is deleted, we had to delete this recipe id from other lists as well, if only one of these requests was executed but the others are not, the database will have a conflict. The second issue was, the initial implementation had much duplication, for example, the user table resulted in an additional column to store a list of user id who he subscribed to. The two issues that occurred in designing database tables caught our attention.

Solution:

"Third Normal Form is a design approach that uses normalising principles to reduce the duplication of data, avoid data anomalies, ensure referential integrity, and simplify data management." (Codd, 1971) A Third Normal Form (3NF) approach was adopted in our database design to solve the issues mentioned above. Instead of storing primary keys from other tables in a list, we created tables for each relationship. We created tables like subscription, comment, and basket_model. For example, like table now contains two foreign keys, user id and recipe id. The foreign keys are designed with a cascade delete. This design approach avoids database conflict, and it also minimised the storage required because there is no duplicated data.

5.2.4 Fuzzy Match

Problem statement:

In functions like search and recommendation based on ingredients, the initial functions were designed to match the content between user input and database instance. Problems occurred when a user had a typo “orenge” creating the recipe, the wrong word was then stored in recipe blocking further matches; when a user enters a typo “banala” to match “banana” from the database; when a user typed “cherries” to match “cherry” from the database. The backend needs a methodology to recognise “banala” as “banana” or the other way around to match these two words.

Solution:

A python built-in package “difflib” was adopted to generate the similarity between strings. According to Rnab, SequenceMatcher from difflib is a tool to generate similarities, and then a cutoff of 0.7 in similarity is able to detect if the twos words just have a typo difference (Rnab, 2019). After a number of experiments, the team was able to reproduce the code and achieved the required performance. For example, the similarities generated by difflib between “cherries” and “cherry”, “banala” and “banana”, “orenge” and “orange” are all above 0.7. When the words are different like “chickens” and “cherries” has 0.625 in similarity, “yogurt” and “cherry” has 0.167 in similarity, where similarities are all below 0.7. By using this package, now we are able to match the two words even if they have typos or different plurals.

6. User Manual

The following instructions will introduce how to operate the MyRecipe platform on the VLAB. After setup, the storage that the project took would be approximately 400 MB. It's recommended to set up backend environment first.

6.1 Back-end Setup

Requirements: Python version 3.7 and above

Please make sure you are under the folder 'capstoneproject-comp9900-t18a-tomcat' with your terminal, i.e. if you run **pwd** you should see something like this: **/your_path_to_this_folder/capstoneproject-comp9900-t18a-tomcat**

1. Run **"cd backend"** in your terminal to enter folder 'backend'
2. For creating isolated virtual python environments, run **"pip3 install virtualenv"** in your terminal to install the virtual environment tool.
3. Run command **"python3 -m virtualenv ."** to set up virtual python environments. Lib will be created in the current directory.
4. Run command **"source bin/activate"** to activate the virtual environment. (To deactivate the virtual environment, you can run **"deactivate"**)
5. While the virtual environment is active, run **"pip3 install -r requirements.txt"** to install required packages into your virtual environment. It may take several minutes to install.
6. Now, you can run the backend server by using **"python3 manage.py runserver 5005"**. This will start development server at (http://127.0.0.1:5005/)

6.2 Frontend Setup

*Make sure the current location is under the folder name '/capstoneproject-comp9900-t18a-tomcat'

1. Input **"cd frontend"** to change the current directory to the frontend.
2. Input **"yarn install"** to set up the 'yarn' environment and install its packages
3. input **"yarn start"** to initiate the frontend environment and the browser will automatically pop up and direct to the address http://localhost:3000/

If yarn install or yarn start fails, look for yarn alternative set up in Section 6.3

6.3 Optional Setup

If solutions could not be found in this section, please contact Patrick at z5305975@ad.unsw.edu.au .

6.3.1 (Optional) Use your own cloud MySQL database

We've set up an online AWS database for you. You can connect to and use the online database directly without changing anything. However, if you want to use your own cloud MySQL database, you can open file **capstoneproject-comp9900-t18a-tomcat/backend/backend/settings.py** and change the following part:

```
DATABASES = {
  'default': {
    'ENGINE': 'django.db.backends.mysql',
    'NAME': 'your_cloud_databasse_schema_name',
    'USER': 'your_user_name',
    'PASSWORD': 'your_password',
    'HOST': 'your_host',
    'PORT': 'your_port_default_is_3306',
   }
}
```

Please make sure the schema before migration is empty.

With terminal open at directory **capstoneproject-comp9900-t18a-**

tomcat/backend/, you can run **“python3 manage.py makemigrations api; python3 manage.py migrate api; python3 manage.py makemigrations; python3 manage.py migrate”** to migrate the database. It may take several minutes.

You can run the backend server by using **“python3 manage.py runserver 5005**”

6.3.2 (Optional) Use your local MySQL database

You can also use your local MySQL database. Just need to change the following in **capstoneproject-comp9900-t18a-tomcat/backend/backend/settings.py**:

```
DATABASES = {
  'default': {
    'ENGINE': 'django.db.backends.mysql',
    'NAME': 'your_local_databasse_schema_name',
    'USER': 'your_user_name',
    'PASSWORD': 'your_password',
    'HOST': localhost’,
    'PORT': 'your_port_default_is_3306',
   }
}
```

Also, please make sure the schema is empty. With terminal open at directory **capstoneproject-comp9900-t18a-tomcat/backend/**, migrating your database by running: **“python3 manage.py makemigrations api; python3 manage.py migrate api; python3 manage.py makemigrations; python3 manage.py migrate”**

You can run the backend server by using **“python3 manage.py runserver 5005”**

6.3.3 (Optional) Use Your Local SQLite Database

Instead of using a MySQL database, you can use SQLite database as well. First you need to at the directory **capstoneproject-comp9900-t18a-**

tomcat/backend/ with your terminal. Then run **touch sqlite3.db** to create a file called **sqlite3.db** under this folder.

After that just need to change the following part in file **capstoneproject-comp9900-t18a-tomcat/backend/backend/settings.py**:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'sqlite3.db',
    }
}
```

You can name the **xxx.db** by yourself, just make sure to change the 'NAME' in setting.py accordingly.

SQLite is the default database for Django. However, it should not be used in production since it is usually slow.

Finally, with terminal open at directory **capstoneproject-comp9900-t18a-tomcat/backend/**, migrating your database by running: **"python3 manage.py makemigrations api; python3 manage.py migrate api; python3 manage.py makemigrations; python3 manage.py migrate"**

You can run the backend server by using **"python3 manage.py runserver 5005"** now.

6.3.4 (Optional) Use Your Own Firebase

You can also use your own firebase, just open file **capstoneproject-comp9900-t18a-tomcat/backend/api/views.py** and change the following part:

```
config = {
    "apiKey": "your_api_key",
    "authDomain": "your_auth_Domain",
    "projectId": "your_project_Id",
    "storageBucket": "your_storage_Bucket",
    "messagingSenderId": "your_api_key",
    "serviceAccount": "myrecipe.json",
    "appId": "your_api_id",
    "measurementId": "your_measurement_Id",
    "databaseURL": ""
}
```

You can find this information at your Online Firebase -> Project settings -> General -> Your apps

Open file **capstoneproject-comp9900-t18a-tomcat/backend/myrecipe.json** and replace it with your generated JSON key to like this:

```
{
    "type": "service_account",
    "project_id": "PROJECT_ID",
    "private_key_id": "SOME_NUMBER",
    "private_key": "-----BEGIN PRIVATE KEY-----\nPRIVATE_KEY\n-----END PRIVATE KEY-----\n",
    "client_email": "SERVICE_ACCOUNT_EMAIL",
    "client_id": "...",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://accounts.google.com/o/oauth2/token",
    "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
    "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/SERVICE_ACCOUNT_EMAIL"
  }
```

6.3.5 (Optional) Alternative to yarn

There is another option, if yarn install does not work, input **"npm install"** followed by inputing "**npm start",** then the yarn equalitvant is implemented.

7. Reference

1. AWS (2021 July 30) https://aws.amazon.com/rds/
2. Childs-Maidment, James. (2021 July 30) https://github.com/thisbejim
3. Codd, E. F. (1971) "Further Normalization of the Data Base Relational Model", p. 34.
4. Django. (2021 July 30) https://www.djangoproject.com/
5. Facebook. (2021 July 30) https://github.com/facebook/react/blob/main/LICENSE
6. Firebase (2021 July 30) https://firebase.google.com/
7. Flaticon. (2021 July 30) https://flaticons.co/license
8. fontawesome. (2021 July 30) https://fontawesome.com/license/free
9. Google. (2021 July 30) https://developers.google.com/fonts/faq
10. McMahon, James. (2021 July 30) https://github.com/James1345
11. mui-org. (2021 July 30). “LICENSE” https://github.com/mui-org/material-ui/blob/next/LICENSE
12. Open Source Initiative. (2021 July 30) https://opensource.org/
13. ReactJs (2021 July 30). “Getting Started” https://reactjs.org/docs/getting-started.html
14. Rnab. (2021 July 30) “Python Programing difflib” https://medium.com/boring-tech/python-programming-the-standard-library-difflib-28ffaf5c1155
15. Smashingmagazine (2010). Smashingmagazine. https://www.smashingmagazine.com/2010/01/color-theory-for-designers-part-1-the-meaning-of-color/
16. User Interface Design Basics | Usability (2020). “Usability.Gov.” https://www.usability.gov/what-and-why/user-interface-design.html
17. Zhang, H. (2020 October 15). 7 Principles of Icon Design - UX Collective. Medium. https://uxdesign.cc/7-principles-of-icon-design-e7187539e4a2