Project Report

COMP9900-W16A-FIFA

02.08.2021

Suharsh Nagasampagi - z5087281
z5087281@ad.unsw.edu.au - Scrum Master / Full Stack Support

Eu Shaun Lim - z5156345
z5156345@ad.unsw.edu.au - Recommendation System Engineer

Nehal Yatham - z5274224
z5274224@ad.unsw.edu.au - Frontend Developer

Joel Braganza - z5283268
z5283268@ad.unsw.edu.au - Backend Developer

Table of Contents

Abstract 3

1. Background 4

2. Architecture 6

2.1 Software Architecture Design 6

2.2 Overall System 7

3. Data Model 8

3.1 Data Preparation 8

3.2 Database Architecture 9

3.3 Tables 10

3.3.1 Users 10

3.3.2 Recipes 10

3.3.3 Ingredients 11

3.3.4 Quantities 11

3.3.5 Steps 11

3.3.6 Likes 11

3.3.7 Subscribers 11

3.3.8 Payments 12

3.3.9 Comments 12

3.3.10 Flagged Comments 12

4. System Functionalities 13

4.1 User Authentication 13

4.1.1 Register New User 13

4.1.2 User Login 14

4.1.3 Profile 15

4.2 Recipe 16

4.2.1 Recipe Creator 16

4.2.1.1 Create recipe 16

4.2.1.2 Edit recipe 18

4.2.1.3 Delete recipe 19

4.2.2 Recipe Explorer 20

4.2.2.1 View Recipe 20

4.2.2.2 Like recipe 22
4.2.2.3 Comment on recipe 22
4.2.2.4 Subscribe to Creator 23
4.2.2.5 News feed 24
4.3 Search 25
4.3.1 Ingredients Search 25
4.3.2 Method Search 25
4.3.3 Meal-Type Search 26
4.3.4 Recipe Name Search 26
4.3.5 Combination Search 27
4.4 Recommender System 27
4.4.1 Viewing Recipes 27
5. Novelty Features 30
5.1 Basics Page 30
5.2 Interactive Recipe 32
5.3 Personalized Feed 33
6. Third-Party Functionalities 34
6.1 Licensing 34
6.2 Front-End 35
6.3 Back-End 35
6.4 Database System 35
6.5 Recommender System 35
7. Implementation Challenges 36
7.1 Front-End and Back End 36
7.2 Recommender System 36
8. Future Work 38
9. User Documentation / Manual 39
References 40

Abstract

The need for a comprehensive cooking instructional tool has always been ubiquitous, and unfortunately the existing solutions simply don't hit the mark. With the changes COVID-19 has brought about in our day to day life, we need an all inclusive system now more than ever. More and more people have to rely on their cooking not only for survival, but as an outlet for their creativity as well. The current lockdown in NSW and other states across Australia has seen a spike in social media posts of people using the comfort of their kitchens as consolation for being stuck indoors. Whether it be to perfect a household classic, cater for special dietary constraints or to try out a new creative dish, there is no doubt that the need for a cooking instructional tool is incontestable. Cooking can be a very important skill to learn and practice. Our goal has been to build a web application that caters to a whole range of individuals at different levels of cooking proficiency, ranging from beginners looking to cook a dinner to home cooks aspiring to widen their repertoire. Our app allows contributors who are either seasoned chefs or experienced home cooks, to upload recipes, get feedback on their work in the form of likes and comments, and reach an audience of people who like their videos and would consistently like to see more content. The other side of this system would have explorers who could either be just figuring out how to navigate the kitchen, or intermediately experienced home cooks looking for recipe inspiration or to or elevate their cooking. Whatever their level of expertise, our system caters to them with everything from targeted recipes and a basics page to advanced recipes with multiple simultaneous moving parts.

1. Background

To produce a baseline for our project, we analyzed several currently existing solutions in detail, to get an idea of the tools and functionalities currently available in the recipe tutorial space, and how we can compete with them. It is worth noting that since almost all the resources we are competing against are backed by large organizations and substantial funding, we only compare the technical functionality and its targeted effect. We analysed the websites Kitchen Stories, All Recipes and Taste. Kitchen Stories has a gimmicky user interface and an absent recommender system. While All Recipes had a much better developed recommender system, and more refined user interface elements, it was still lacking in ease and simplicity of use as well as general cohesion. Taste had a coherent design vision, and a well implemented recommender system but still did not provide the creators with any opportunity for feedback from users in the form of likes, comments and a regular audience in the form of subscriptions.

One of two pillars of our project vision are the Creators who are experienced chefs working in restaurants, freelance recipe creators, or experienced home cooks. Our system provides them with the opportunity to upload detailed recipes to our platform for all users to see. They can choose to upload a step-by-step interactive instructional recipe for absolute beginners who can access the entire recipe in a single page, complete with tools like timers and short demonstrational video clips that will guide the user through the process of preparing an entire dish, in a single page. They are rewarded by likes and comments of explorers who will give valuable feedback to creators and subscribe to their pages. This will ensure that their content is reaching people who have liked it in the past and would like to see new content of a similar type.

The second pillar is Explorer centric. Explorers of our system are able to search for recipes by their name, meal type (i.e. whether they are appetizers, entres etc), method of preparation, the ingredients present in a dish as well as a combination of any of these methods. This enables them to find any recipe with ease. Whether they want to satisfy a craving, or just make something with the ingredients and tools at their disposal, this is a convenient and efficient tool. Explorers of our system can convey their opinion by liking a recipe they like, commenting on them, and subscribing to a creator. This will ensure that they see all new content posted by that creator in their newsfeed, as a way to ensure they don't miss out. A recommender system on our application takes all this information into account and recommends recipes to users based on recipes they have liked and creators they have subscribed to. Furthermore, if an intermediate user

progresses in their cooking skills, and believes they should publish a recipe of their own, our system would support this as well.

While there are no regulations and there is no verification process of the recipe, we assume that the peer review aspect of likes and comments will serve as a sufficient metric for assessing credibility of recipes.

2. Architecture

2.1 Software Architecture Design

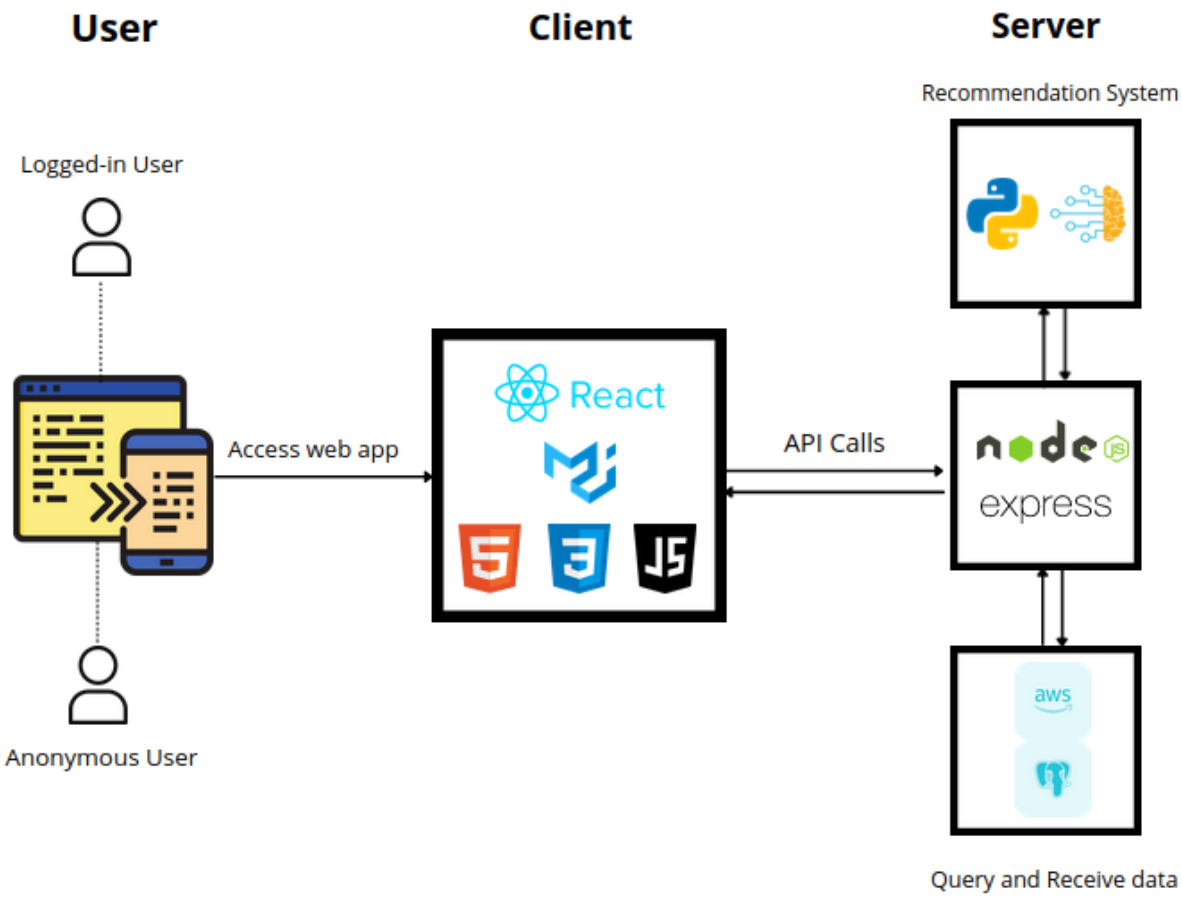


Figure 1. Software Architecture Design

2.2 Overall System

The back-end server consists of the recommendation system, the JavaScript frameworks and the database system. The main back-end server is the JavaScript framework comprising Node.js and Express.js. The database system uses PostgreSQL and is hosted on Amazon RDS while the recommendation system is implemented in Python3. Both the recommendation system and database system will only interact with the JavaScript framework, which will create APIs for the front-end server to make calls from. The front-end system uses React.js as the framework and Material-UI as the design language. The front-end system will serve as the client-facing application and will make API calls to the back-end JavaScript framework to obtain the necessary data. When a user starts the application, they will only interact with the front-end system. Anonymous users will have limited functionality as they will only be able to view recipes on the homepage and the Basics page, while logged-in users will have the full functionality of the application.

3. Data Model

3.1 Data Preparation

This section outlines the preparation of the database architecture. Before creating the actual database, we will first need to look at the logistics. This includes the type of database management system to use, the type of database system to use, and how the database is hosted.

We decided to use a relational SQL database as we prefer a more traditional approach as well as the fact that we have structured data, which is more ideal in relational SQL databases. Among the available relational databases, we decided on using PostgreSQL as it is free, open sourced and compatible with many operating systems, which is important since most of us are using computers with different operating systems. We are all also familiar with PostgreSQL since we have used it in the past before. We decided to host the database on an Amazon RDS server for ease of access. By doing so, any changes made on the database will be reflected immediately across all our local development sites.

In addition, our codebase will be hosted on GitHub. Any changes made on the codebase can be tracked easily and reverted if necessary. We are also able to pull the latest source code on our development sites easily by the use of the command line interface. Our project development will be managed on Jira as we will be able to track our progress easily and adopt the Agile methodology efficiently.

3.2 Database Architecture

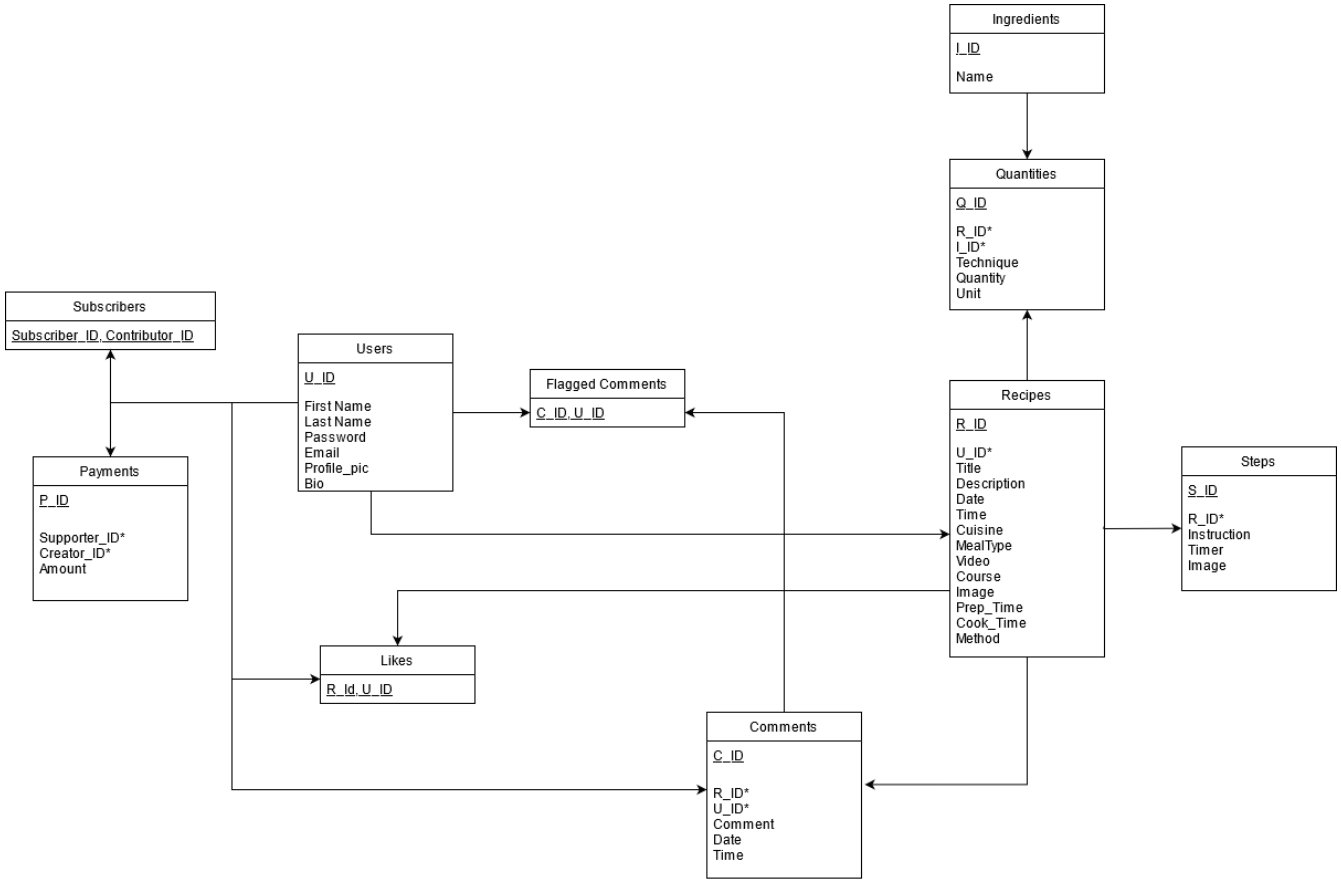


Figure 2. Entity-Relationship (ER) Diagram

3.3 Tables

This section explains the rationale behind the design of the database as shown in the ER diagram above. In the ER diagram above, the column names that are underlined are the primary key or composite key for the table, and the starred column names are the foreign keys for the table. The primary/composite keys are used to uniquely identify each item in the tables, while the foreign keys are primary keys for other tables which are being referenced to.

3.3.1 Users

This table stores the information of all the registered users on MyRecipes. This includes the User ID, first name, last name, email address, password, profile picture and biography. The User ID allows the software to uniquely identify each user. The email address acts as the username. This, along with the password is used to login to the users' account. The profile picture is stored as a URL to the image which will be displayed next to the users' profile. The biography allows each user to uniquely describe themselves in their own manner, whereby other users can then view and maybe make a connection with.

3.3.2 Recipes

This table stores information about the recipes that users have contributed to. This includes the Recipe ID, User ID, title, description, date, time, cuisine, meal type, video, course, image, preparation time, cooking time and method. The title and description contains the recipe title and a short summary of the recipe for recipe explorers to read through before clicking in. The date and time stores when the recipe was created or edited.

Cuisine defines the style of cooking for the recipe, eg. Japanese, Indian, Western, while meal type defines if the meal is suitable for breakfast, lunch, dinner or just a light snack. There is also a video link for each recipe as well as a recipe course page which aims towards beginner cooks. This allows users to follow along with the video as well as teach beginner cooks the basic skills of cooking respectively. The image stores the URL of the recipe image which is displayed next to the recipe.

Lastly, we have preparation time, cooking time and cooking method. This allows explorers to make an estimation for how long their cooking will take if they follow this recipe as well as which cooking appliance they will need.

3.3.3 Ingredients

This table stores all the unique ingredients available. This includes the Ingredients ID and the name of the ingredient. The ingredients are predefined and recipe contributors will be able to select from a list of these ingredients when creating their recipes.

3.3.4 Quantities

This table stores more detailed information about the ingredients for each recipe. This includes the technique, quantity needed and unit measurements. The technique defines how each ingredient is to be prepared. For example, an onion could be sliced or diced. The quantity and unit measurement is simply the quantity needed in the respective unit measurement.

The Quantities table links both the Recipes and Ingredients table together. The rationale behind separating the quantities and ingredient names instead of having an Ingredients table that contains all the information is so that we do not have the same ingredients repeated multiple times with different quantities. With a large enough database, this table would get cluttered and messy quickly.

3.3.5 Steps

This table contains information about the instructions in a recipe. This includes the instructions, image and timer. The instructions are the backbone of a recipe as explorers rely on them to cook the meal. The image is to allow contributors to include a reference for explorers when they are cooking the meal, if they wish. Recipe explorers can validate their cooking at that point by checking the reference picture of the step.

3.3.6 Likes

This table stores the likes made by a recipe explorer towards recipes. Recipe explorers can like a recipe which they find interesting, and the recipe contributor will be notified if someone has liked a recipe they contributed to.

3.3.7 Subscribers

This table stores the subscriptions made by a recipe explorer towards a recipe contributor. If an explorer feels that a certain recipe contributor has created a few great recipes and would like to know whenever they have posted new recipes, they can subscribe to that contributor so that any new recipes will be shown on the feed of the explorer.

3.3.8 Payments

This table stores any payments made by a recipe explorer towards a recipe contributor. Recipe explorers are allowed to make payments or donations to recipe contributors if they so desire. The money made by contributors can act as a motivation to publish more recipes, or even make a living out of it.

3.3.9 Comments

This table stores information about comments made by recipe explorers towards a recipe. This includes the comment itself, the date and time of when the comment was made. Recipe contributors can then read these comments which may be words of motivation, positive feedback or even negative feedback.

3.3.10 Flagged Comments

This table stores information about comments that were flagged. If any malicious comments were made, users can flag these comments which will get reviewed by the relevant authorities.

4. System Functionalities

This section outlines all the basic functionalities that have been implemented in our system along with the relevant project objectives these functionalities are associated with.

4.1 User Authentication

Relevant Objective:

I. *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.*

4.1.1 Register New User

When a new user has just entered the application, they are given the option to either browse the application without an account, or register an account with the application. Having an account gives them access to all the features of the application, while not registering for an account means that they are only allowed to browse the available recipes.

In order to register for an account, they will have to input their first name, last name, email address and password. A confirmation of the password will also be required to ensure they do not mistype the password and be locked out of their account in the future.

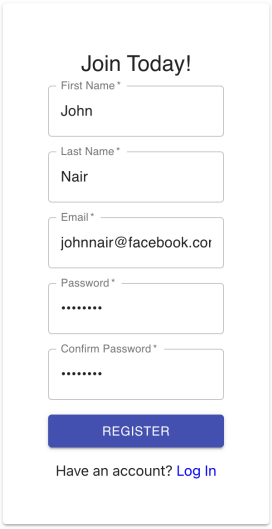


Figure 3. Register New User

Once a user creates a new account by providing the information specified above, they are automatically logged-in, and can see a list of some current popular recipes on the platform.

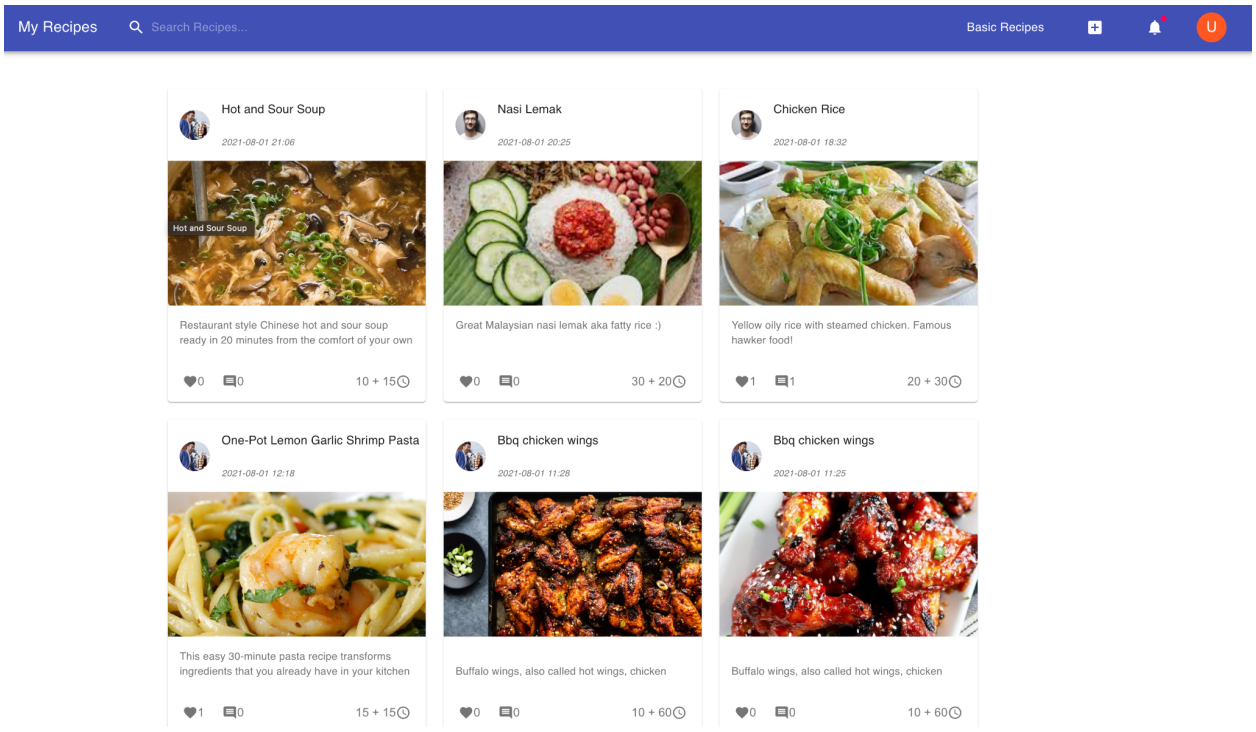


Figure 4. User Home Feed

4.1.2 User Login

Once a user has registered for an account, they can then login into their account, where they have access to more functionalities of the application.

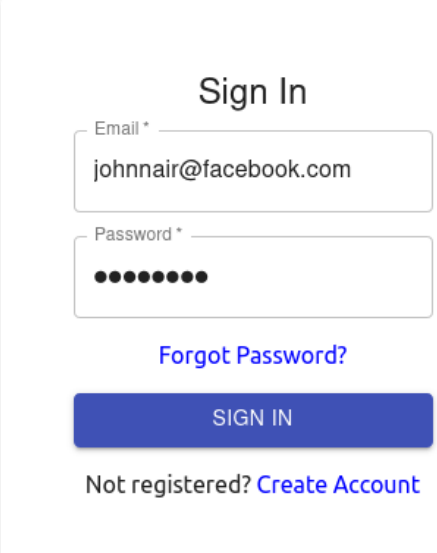


Figure 5. User Login

4.1.3 Profile

Once a user already has an account, they can maintain a profile for themselves, which shows the recipes they have posted, profile picture, biography, contact details, total likes made by other users on their posts as well as a list of subscribers. They can also edit their account details by changing their name, email address, password, biography and profile picture.

Figure 6. User View Profile

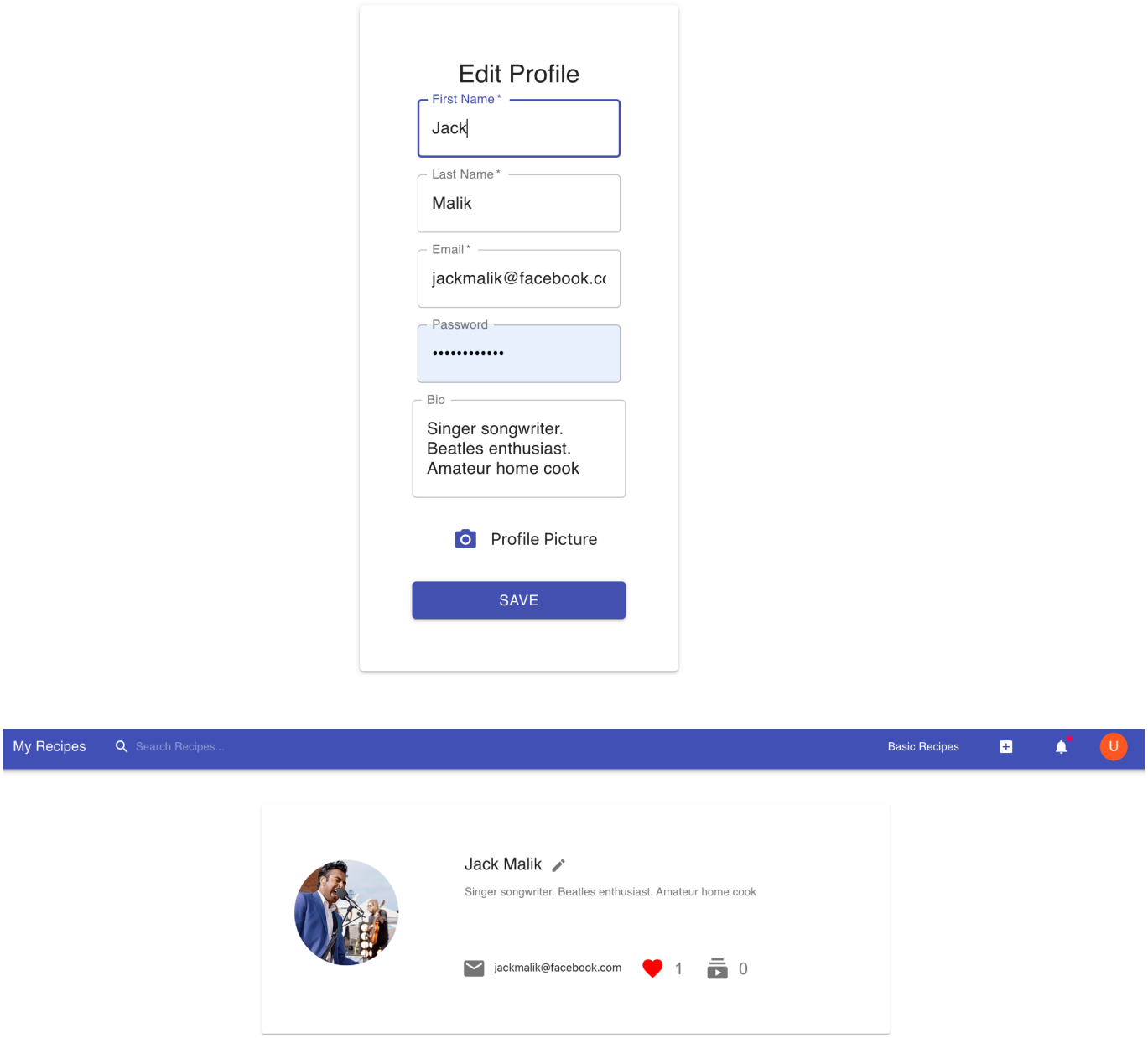


Figure 8. User Edit Profile

4.2 Recipe

Relevant Objective:

II. *The MyRecipes platform must allow contributors to maintain a set of their 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.*

III. *Recipe explorers on the platform must also be able to convey their opinion for any recipe by either liking and/or commenting on it.*

IV. *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.*

V. *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 recipe news feed), be able to see new recipes that are added or have been recently updated by that contributor.*

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

4.2.1 Recipe Creator

4.2.1.1 Create recipe

A user can become a recipe contributor by posting new recipes on the application. They can do so by clicking on the “+” icon on the App Bar. When creating the recipe, they can enter details of the recipe such as the title, cuisine, meal type, prep time, cook time, cooking method and description. They can also upload a photo of the final meal as well as a YouTube video link about the recipe.

They then select the ingredients from a list of ingredients and start to enter the steps needed for the recipe. For each step, contributors can input a timer as well as an image for the step. The timer indicates the time needed to execute the step in real time. Once all the steps are entered, they then click on the “create recipe” button to finalise the creation of the recipe.

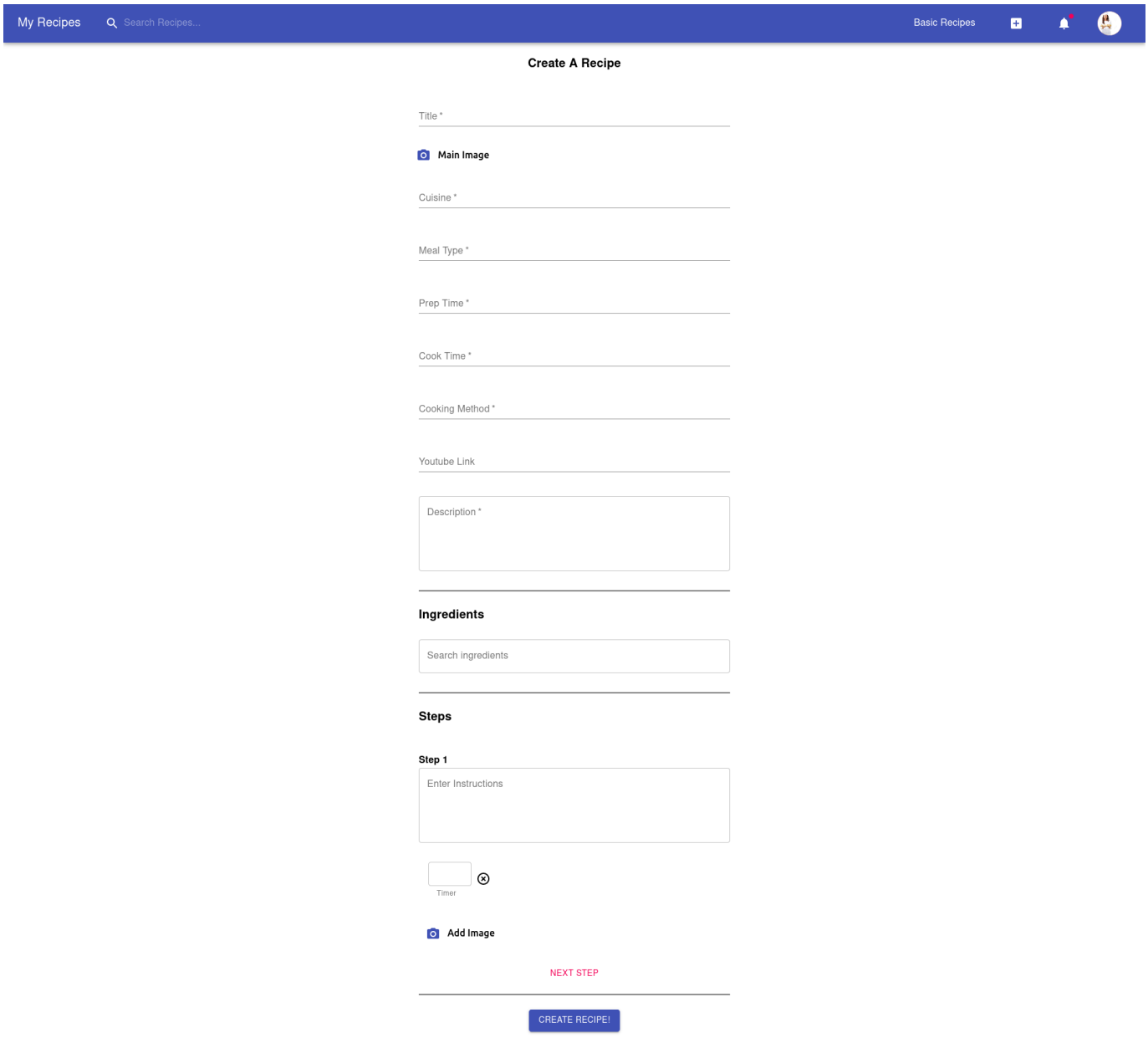


Figure 9. Create Recipe Page

4.2.1.2 Edit recipe

If a recipe contributor would like to make any changes to their recipes, they can do so by going to their own profile and clicking on the “edit” button in the 3 dot icon next to the recipe they want to edit. They can then make any necessary changes to the recipe and finalize the changes by clicking on the “edit” buttons.

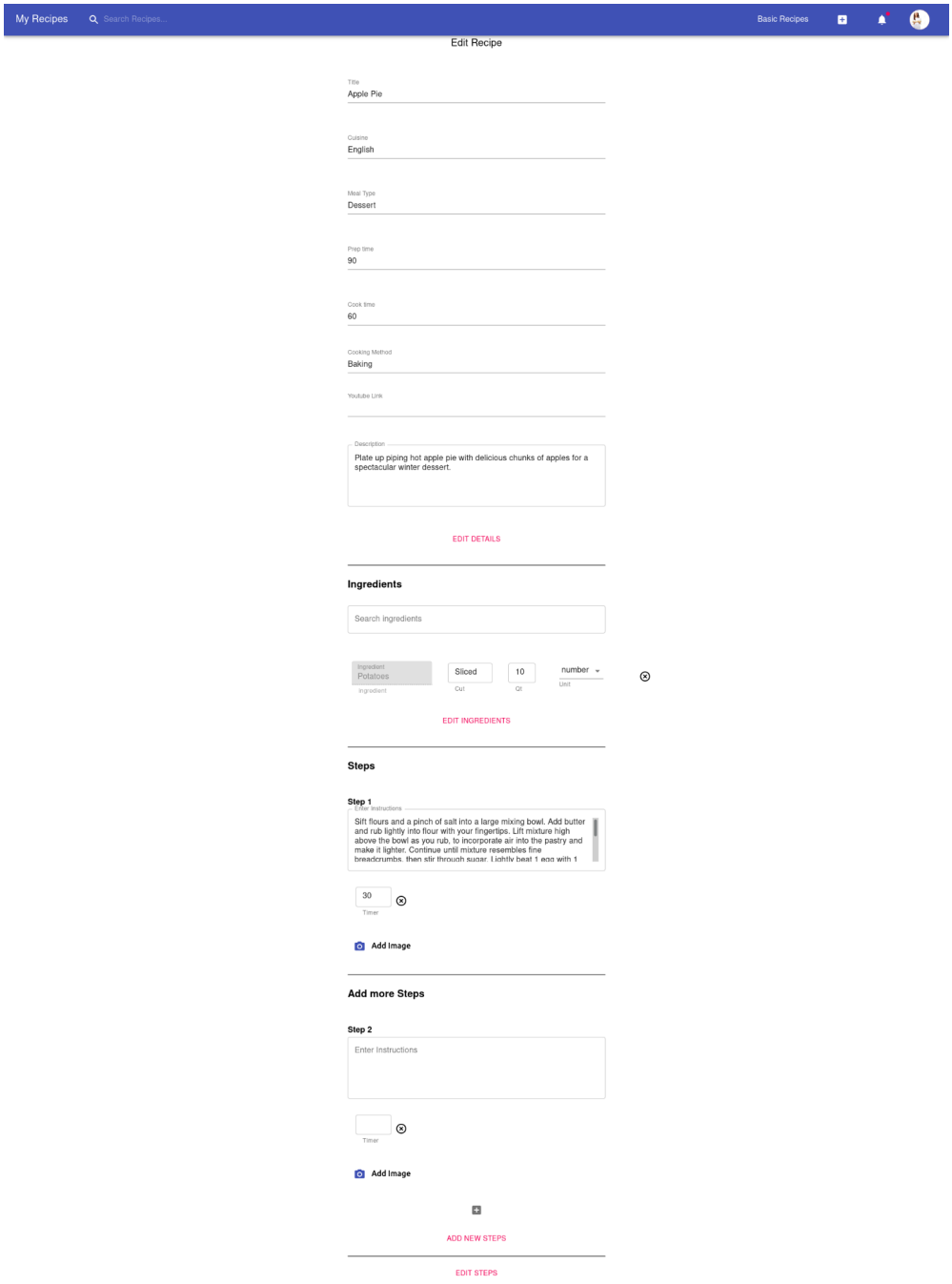


Figure 10. Edit a Created Recipe

4.2.1.3 Delete recipe

If a recipe contributor would like to remove any of their recipes, they can do so by going to their own profile and clicking on the “delete” button in the 3 dot icon next to the recipe they want to remove. Recipe explorers will not be able to look for the deleted recipes anymore.

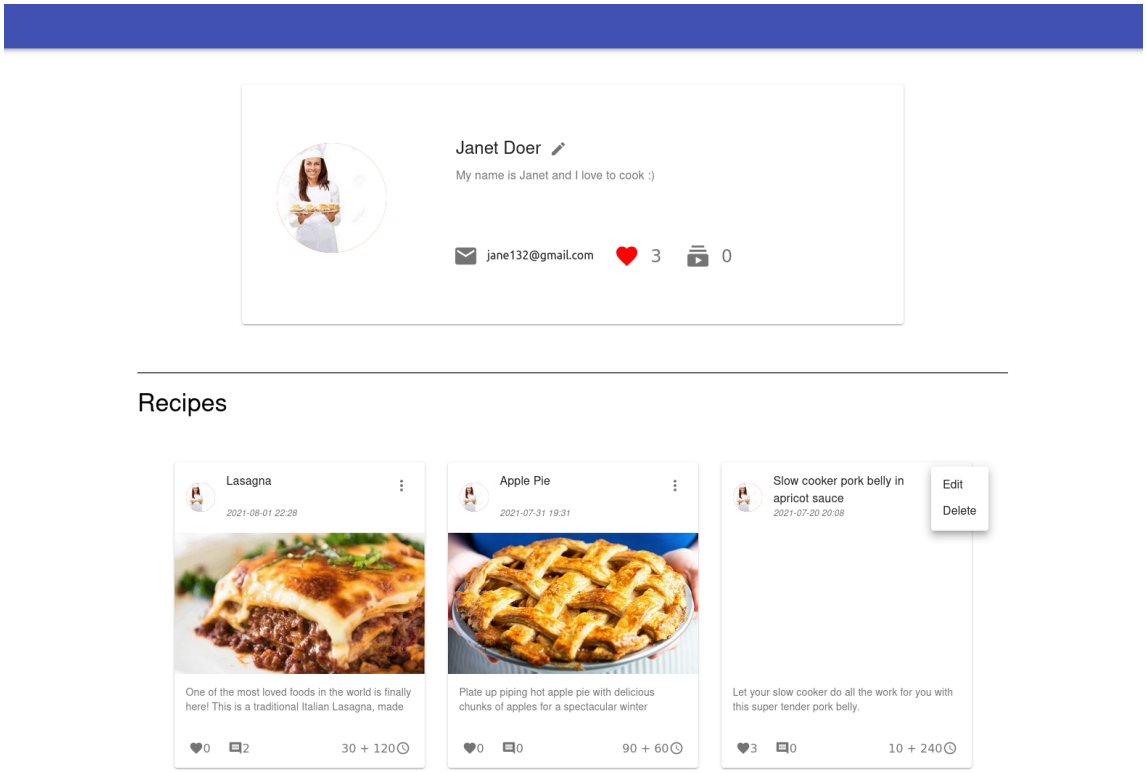


Figure 11. Delete Recipes

4.2.2 Recipe Explorer

4.2.2.1 View Recipe

A recipe explorer can view the full details of the recipe by clicking on the recipe card. This will show the recipe and contributor name, description, cuisine, meal type, prep time, cook time, ingredients, steps, images for each step as well as an “interactive mode” where a timer will start for all the steps in the recipe.

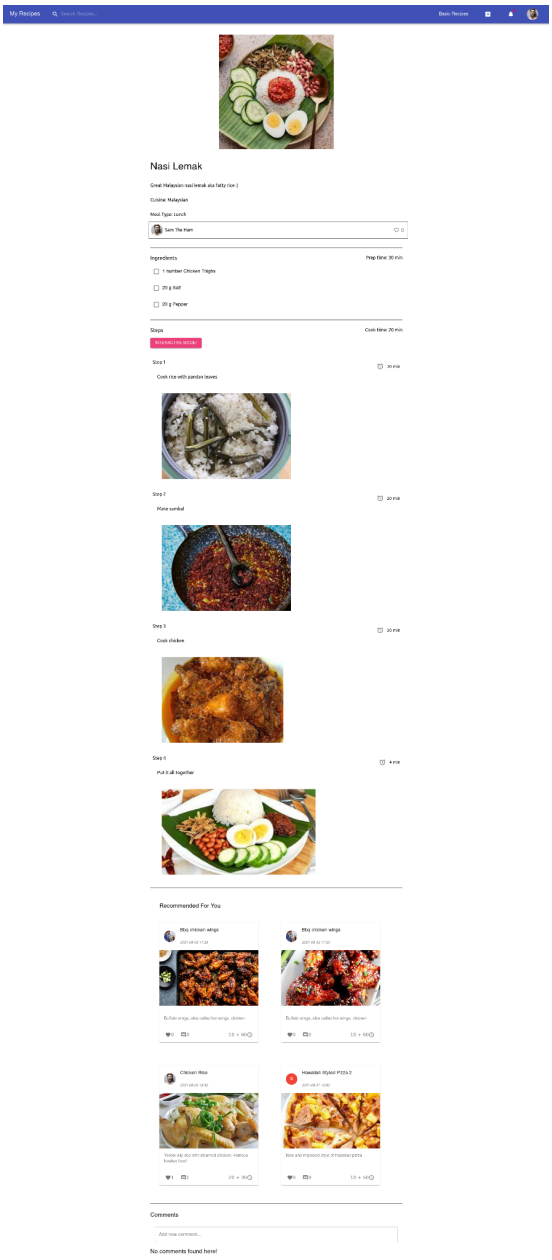


Figure 12. View Full Recipe

4.2.2.2 Like recipe

A recipe explorer can like a recipe by first viewing the recipe and then clicking the “heart” icon next to the recipe contributor’s name.

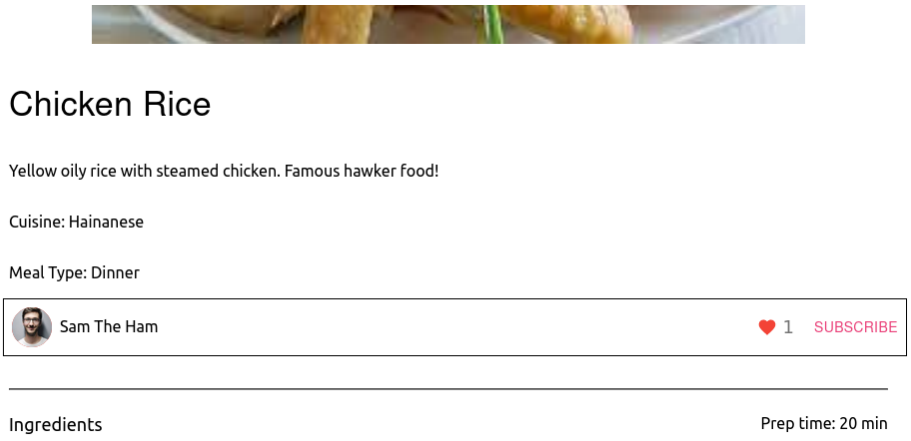


Figure 13. Like Recipe

4.2.2.3 Comment on recipe

A recipe explorer can make comments on a recipe by first viewing the recipe and then scrolling to the comments section where a text box is available to enter comments.

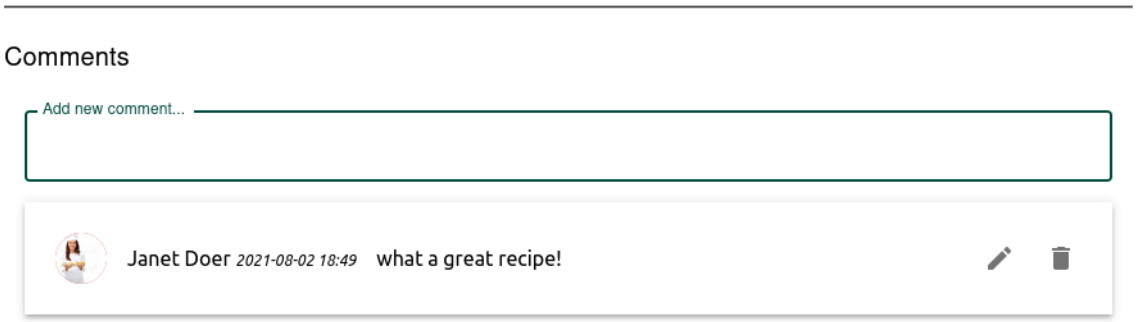


Figure 14. Comment on Recipe

4.2.2.4 Subscribe to Creator

A recipe explorer can subscribe to a recipe contributor by first viewing the recipe and then clicking the “subscribe” button next to the recipe contributor’s name. They can also subscribe through viewing the recipe contributor’s profile and pressing the “subscribe” button there.

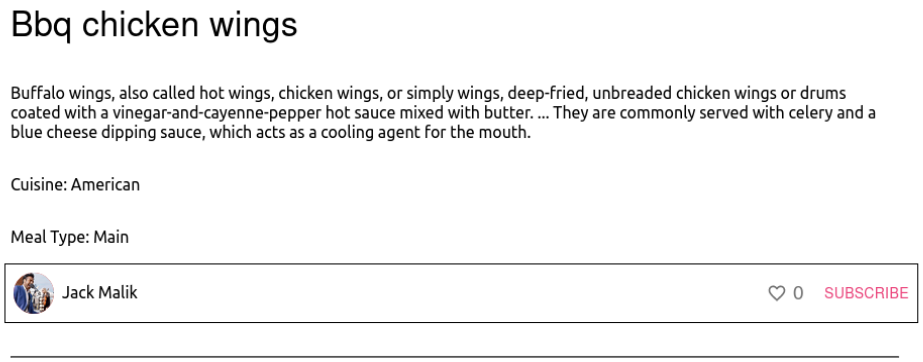


Figure 15. Subscribe to Creator

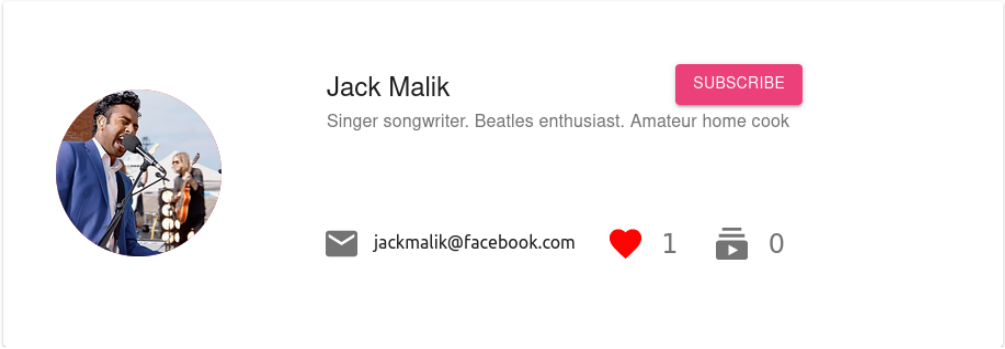
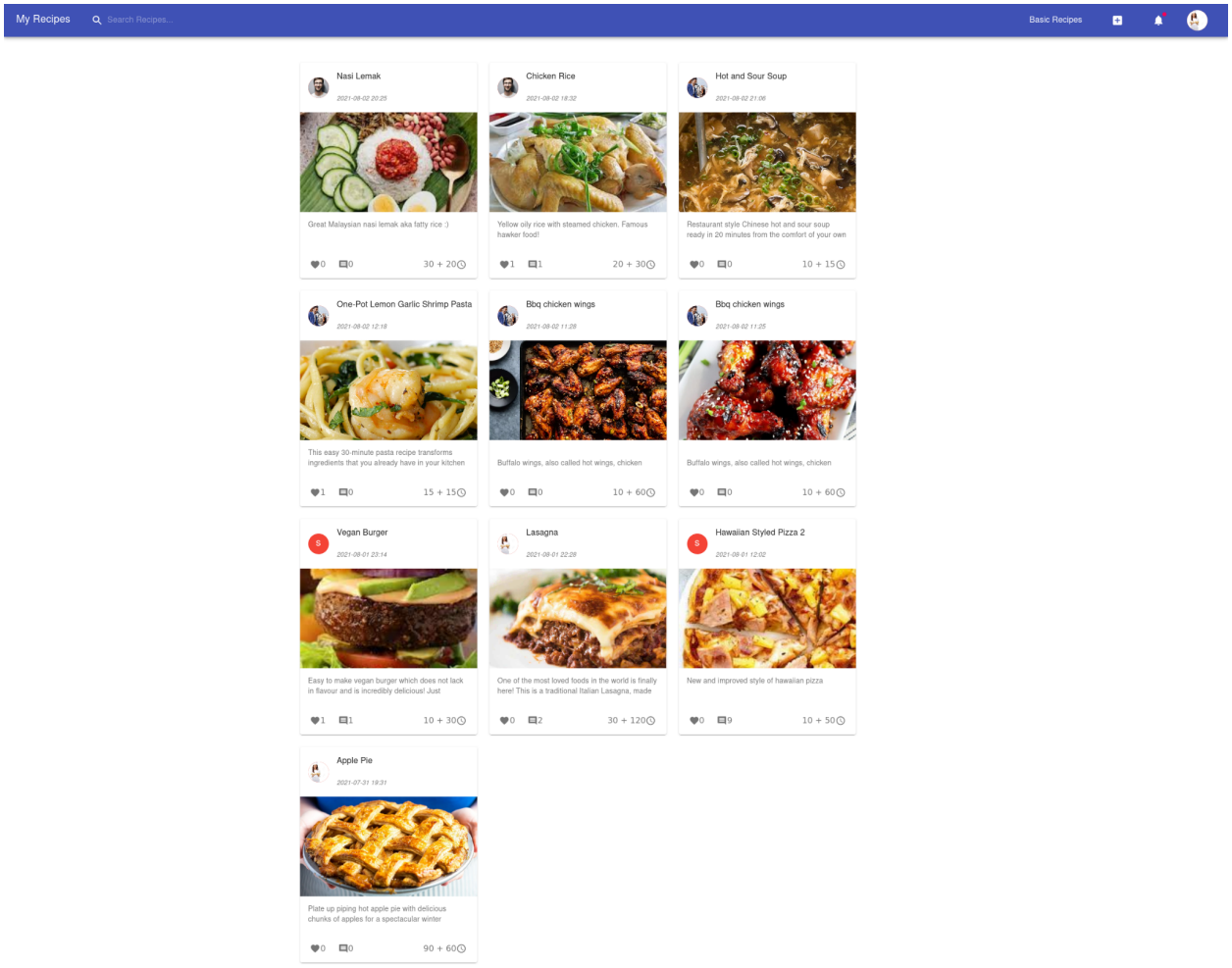


Figure 16. Profile Subscribe

4.2.2.5 News feed

A recipe explorer can go to their feed by clicking on the “My Recipes” button on the top left of the App Bar. This will show recipes made by recipe contributors that they have subscribed to, along with any recipes similar to the recipes that they have liked or commented on. The recipes shown are sorted first by the recipe date, then total likes made by the explorer towards the contributor, and finally the recipe time.



4.3 Search

Relevant Objective:

VII. 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.

4.3.1 Ingredients Search

Recipe explorers can search for recipes using the search bar at the top. By entering the ingredient name as a keyword, the website will return recipes containing those ingredients.

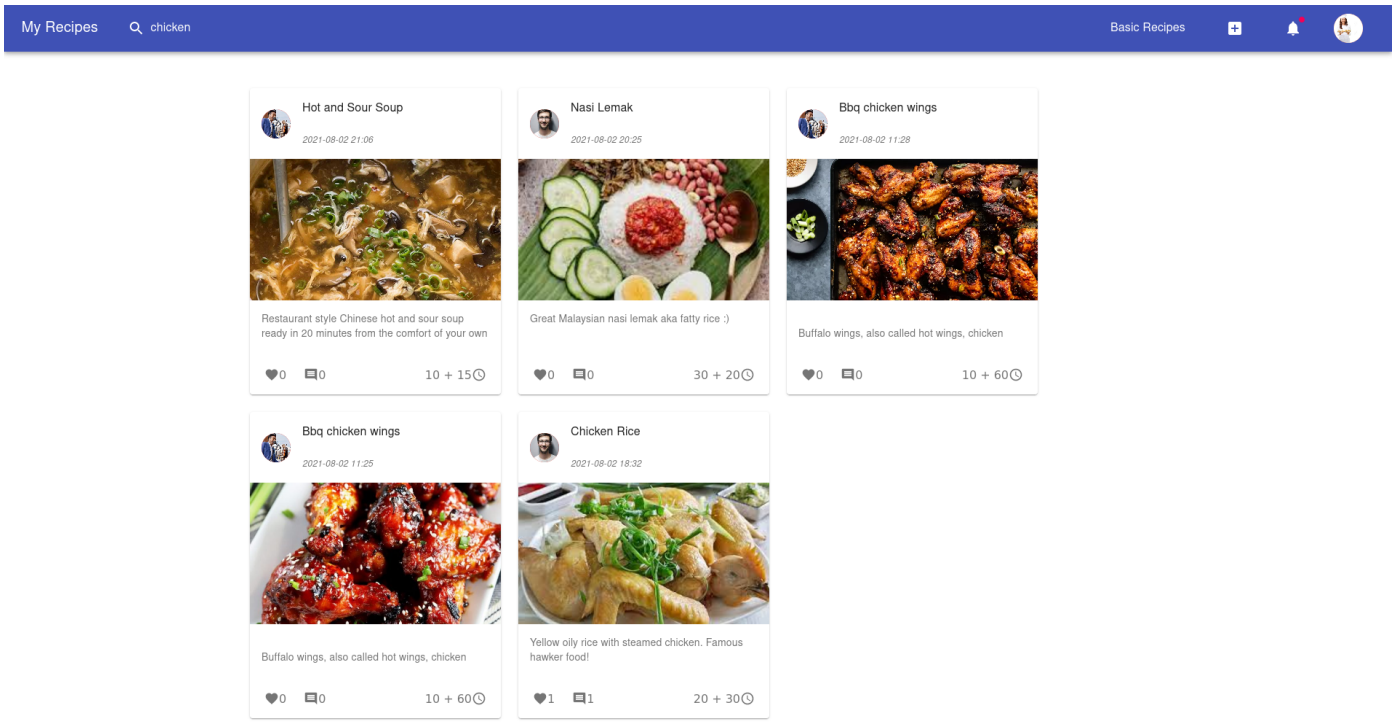


Figure 17. Ingredient Search

4.3.2 Method Search

By entering the cooking method as a keyword, the application will return recipes that use that particular cooking method.

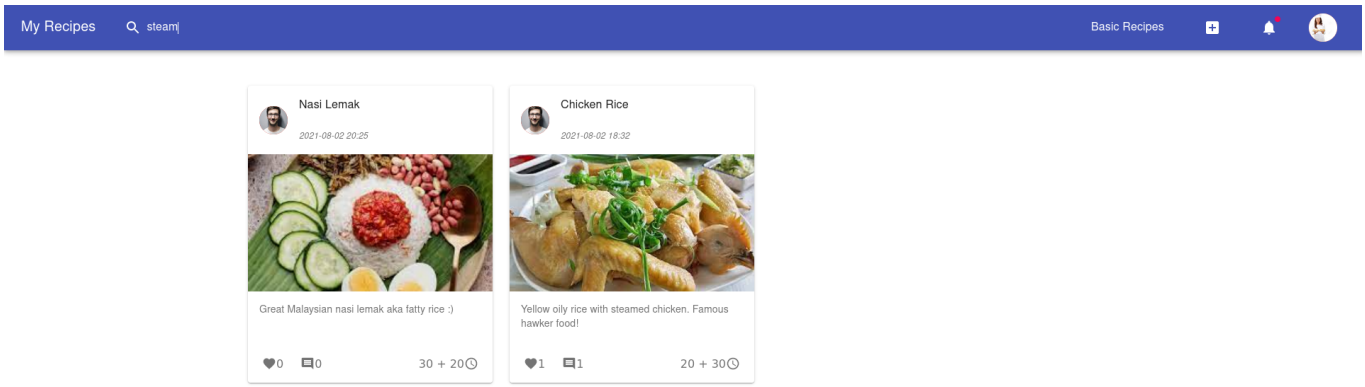


Figure 18. Search by Method

4.3.3 Meal-Type Search

By entering the meal type as a keyword, the application will return recipes with that particular meal type.

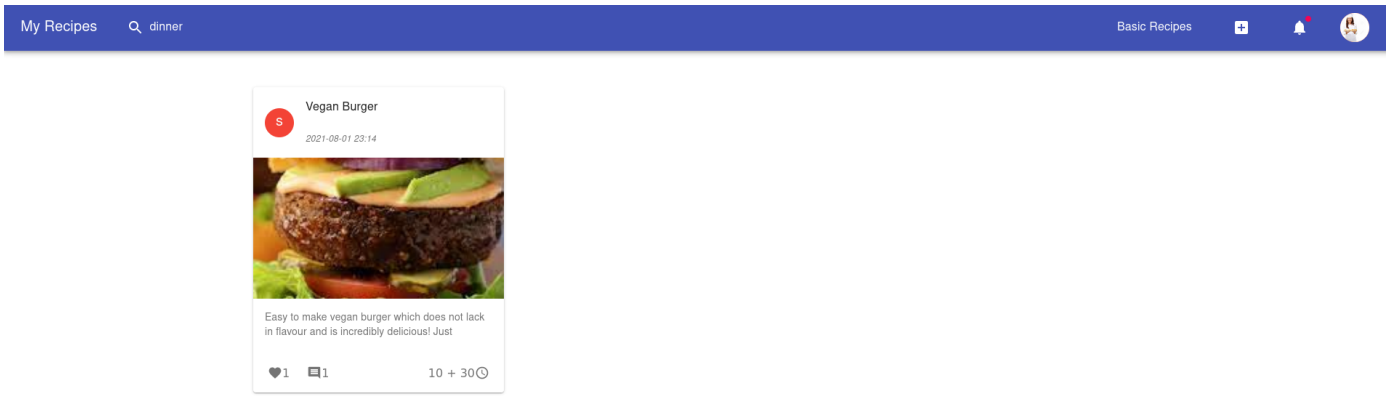


Figure 21. Search by Meal Type

4.3.4 Recipe Name Search

By entering the recipe name as a keyword, the application will return recipes that include those keywords in the name.

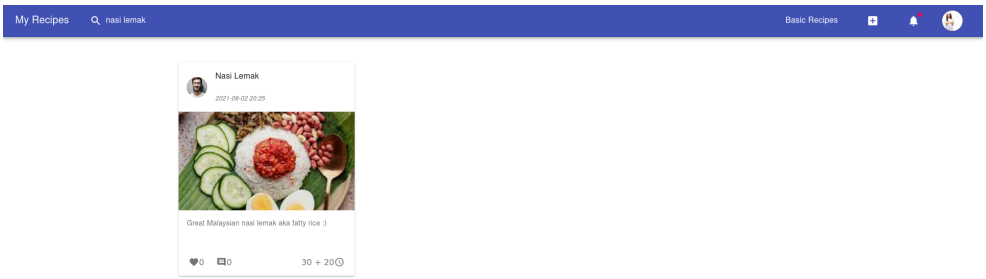


Figure 20. Search by Recipe Name

4.3.5 Combination Search

By entering a combination of ingredient name, method, meal type and recipe name as a keyword, the application will return recipes that are relevant to those keywords.

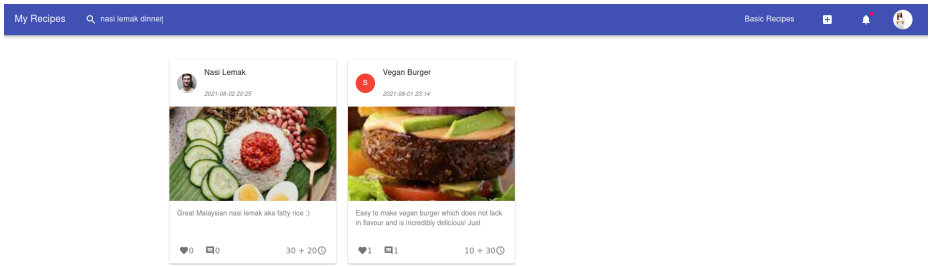


Figure 21. Search by combination

4.4 Recommender System

Relevant Objective:

VIII. 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).

4.4.1 Viewing Recipes

When a recipe explorer is viewing recipes, the application will give recommendations to the user about any similar recipes. This recommendation is made based on the ingredients of the recipe being viewed [1].

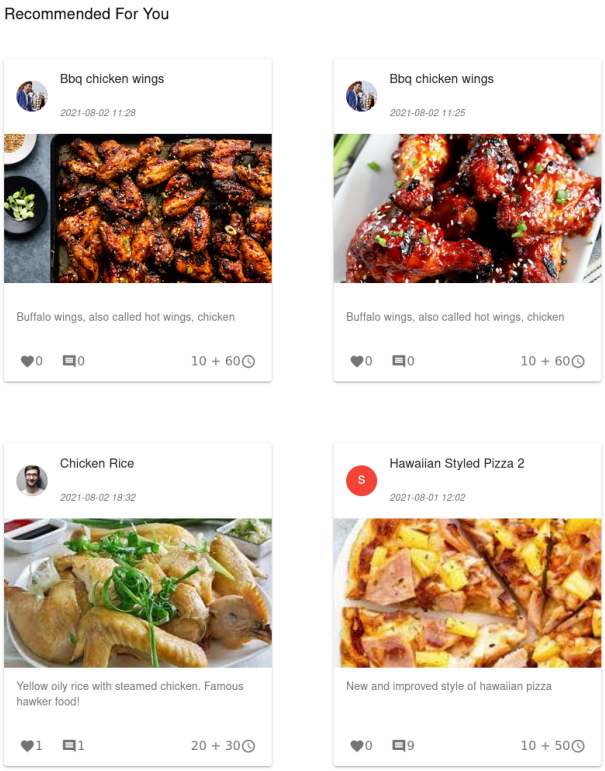


Figure 22. Viewing Recipes

The similarity between the ingredients of other recipes is calculated using the cosine similarity formula [2]:

$$Cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \, \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ is the dot product of the two vectors.

Figure 23. Recommender Algorithm

In short, the cosine similarity calculates the cosine angle between two documents, and the documents with the smallest angle are the most similar to each other. In our case, the documents in question are the list of ingredients that have been normalized by removing any punctuations, converting to lowercase and converting each

word back to its stem word. To speed up the algorithm, these calculations are pre-calculated and updated whenever a new recipe is created or updated. Hence, it does not perform any calculation whenever a recommendation is needed and therefore the user experience is not affected.

5. Novelty Features

5.1 Basics Page

As emphasised earlier, our website is catered towards users of all levels of proficiency. This includes complete novice users who have never set foot inside a kitchen. Such users would need a brief on some essential tips and tricks that would otherwise take months of research and trial and error to accumulate. It is a compilation of all the essential know-how associated with maintaining a home kitchen covering all things from ingredient storage, to knife skills, boiling essential tips and more.

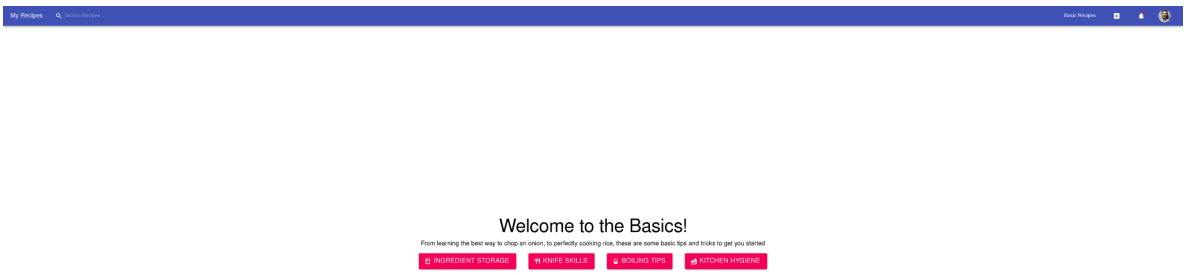


Figure 24. Basics Page

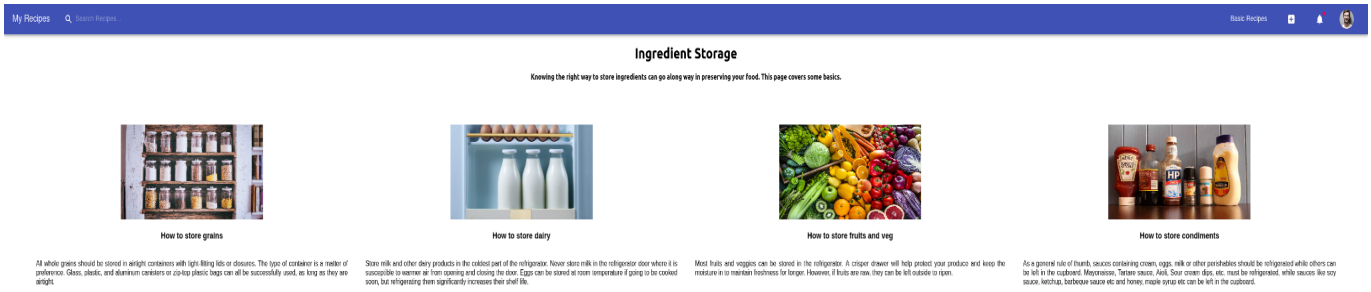


Figure 25. Ingredient Basics

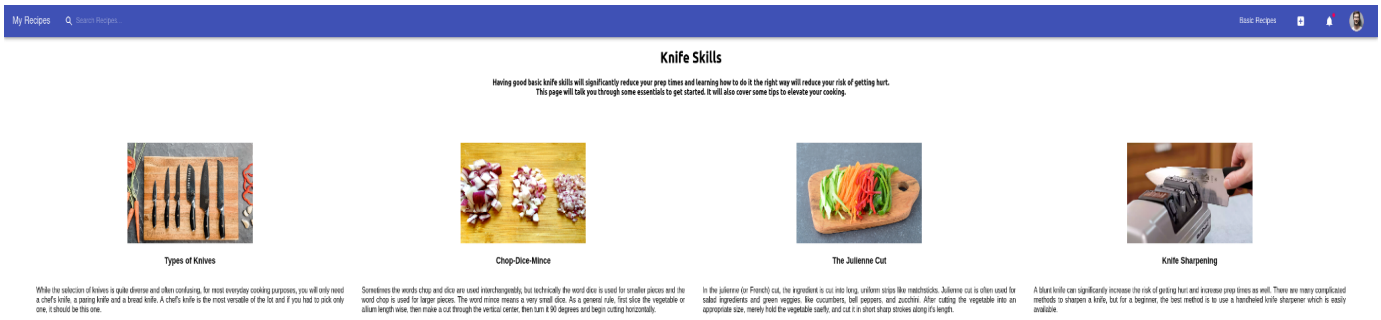


Figure 26. Knife Skills Basics

Figure 27. Boiling Tips Basics

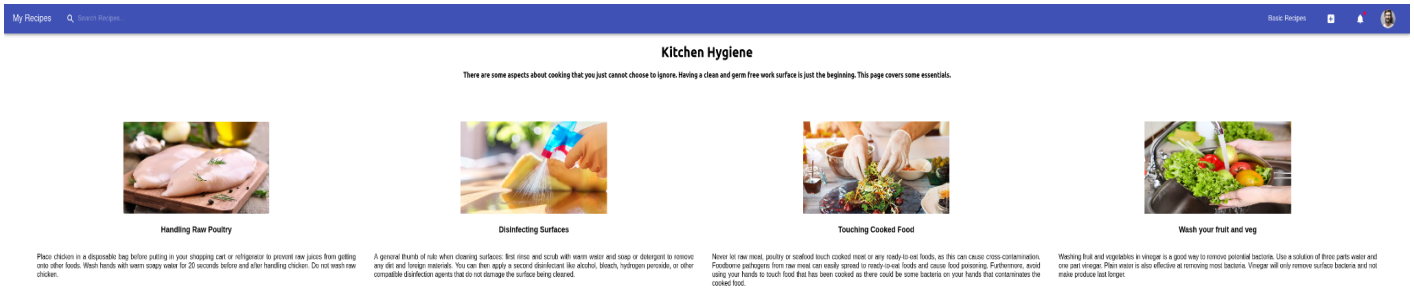


Figure 28. Kitchen Hygiene Basics

5.2 Interactive Recipe

We realize that some user groups might need a little more help than others when it comes to getting started with cooking. In furthering this vision, we include another functionality catered specifically towards beginners and that is the interactive recipe feature.

Some superficial analysis revealed that a large portion of the novice demographic is youth between the ages of 15 to 25. These users are generally more visual users and prefer to get their information in visual media in the form of images and videos. In alignment with this finding, we designed our Interactive Recipe feature. This has all the elements of cooking a recipe, together in a single page from top to bottom. The user would not need to navigate to any other tabs, or look at any other resource till their preparation is complete. This feature is complete with timers to reflect cooking times, images of different stages of the preparation, as well as demonstrational videos for users to follow and replicate. It's an all inclusive instructional tool designed specifically for novice users.

Creators can choose to opt-in to this feature as it might afford them a wider audience, but it is not compulsory. It involves uploading a segmented instructional video as well as images and some additional information, like timer times to make this view more intuitive when compared to other recipes, yet just as detailed.

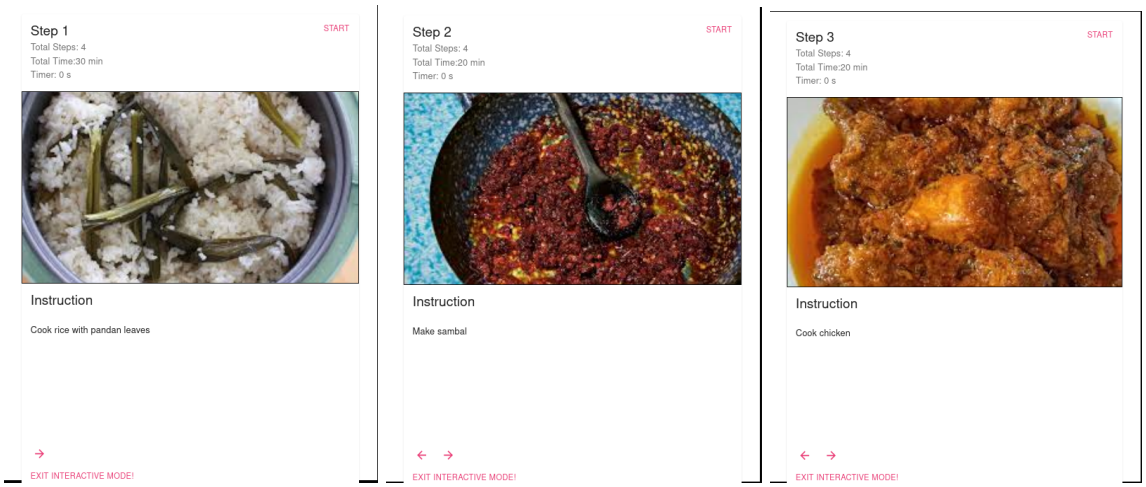
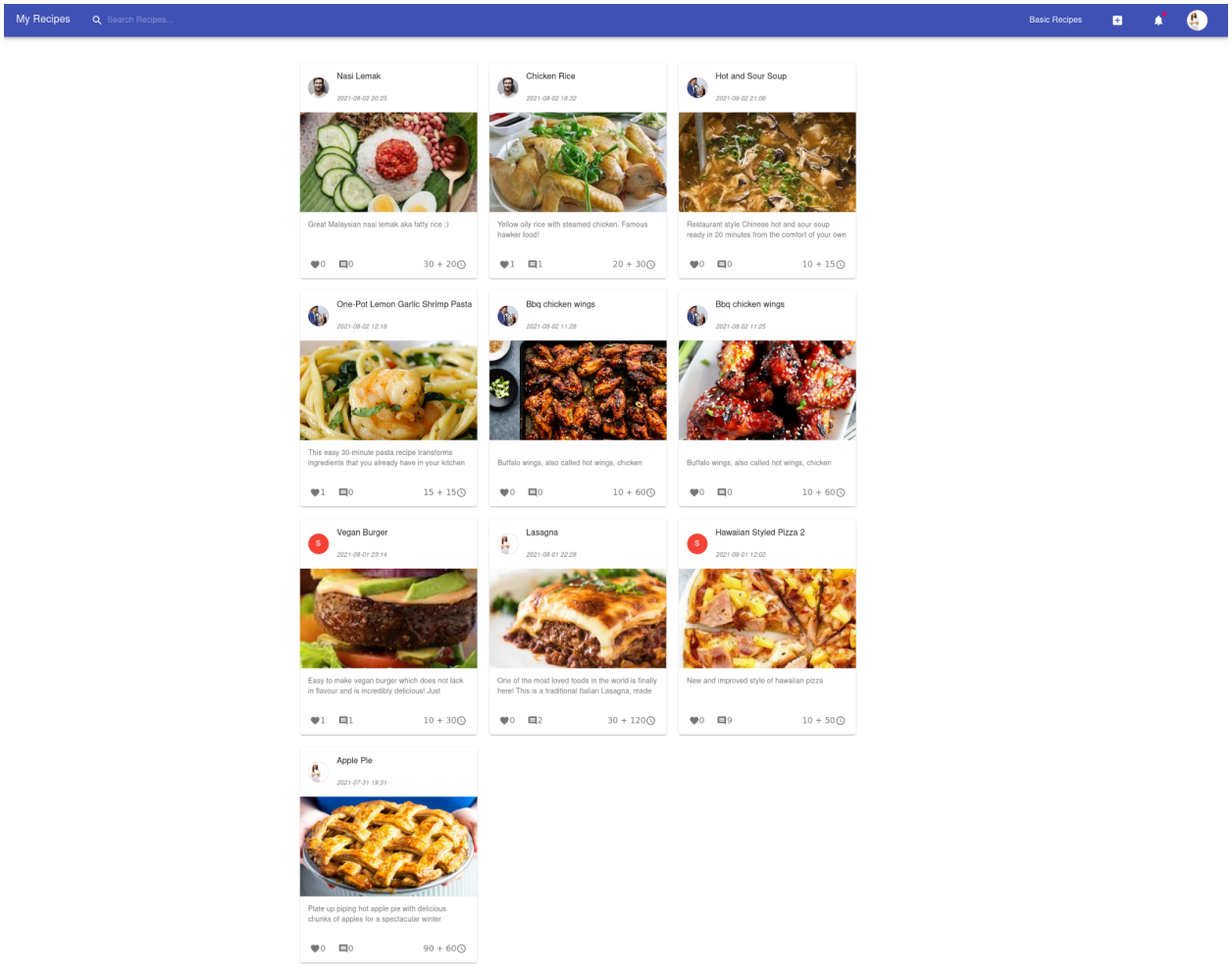


Figure 29. Interactive Recipe

5.3 Personalized Feed

While the project specification mentions to only show recipes made by recipe contributors that the explorers have subscribed to, we took this one step further by giving explorers a more personalized news feed. In addition to the subscriptions made, we also took into account the likes and comments made on recipes by explorers. These recipes are then put through the recommender system to give a list of similar recipes which are also shown on the explorer's news feed.



6. Third-Party Functionalities

6.1 Licensing

All third party functionalities used in this project are free and open-sourced, with the exception of the Amazon RDS server and Amazon S3 service. All JavaScript framework and libraries used in the development of this application falls under the MIT License. Since the licenses of both HTML and CSS depend on the implementer, in our case this would be the Node.js framework, they also fall under the MIT license. PostgreSQL is released under the PostgreSQL License, which is similar to the MIT License. Lastly, Python is dual-licensed under the PSF License Agreement and the Zero-Clause BSD License. Most of the Python libraries used in this project are licensed under the New BSD License, with the exception of the NLTK library, which falls under the Apache 2.0 License.

In layman's terms, the MIT License means that we have full permission to freely use, copy, modify or even distribute copies without any kind of warranty. The Apache 2.0 License is similar as well, with the added protection of patents. The Zero-Clause BSD License is also similar to the MIT License, however the New BSD License adds the condition that we also have to give credit whenever the library is used, as we have done in the codebase when importing the libraries. In terms of the impact these licenses have on the project, there is minimal impact as we have full permission to freely use these free and open-sourced functionalities.

On the other hand, we are using the free tier of the Amazon RDS and S3 services. The free tier Amazon RDS allows for 20 GB of storage, and the free tier Amazon S3 allows 5 GB of storage with 20,000 GET requests and 2,000 PUT requests for one year. For a small scale project like this, this amount of storage is more than enough for our purposes, however in the event that we deploy this application to the public in the future, a paid tier service might be needed as the free tier would not be enough for public usage.

6.2 Front-End

HTML5 and CSS3 were used to implement the front-end system as they are the standard languages in basic web development. HTML5 provides the structure of the web pages including headings, text and images, while CSS3 allows for the presentation of web pages including colours, layouts and fonts. JavaScript, or specifically React.js [8] is used to create the user interface components of the application. The design language used to style the components is Material-UI as it provides a clean interface for the application.

6.3 Back-End

JavaScript was also used in the back-end system to provide a seamless transition between the front-end and back-end during the project development cycle. Node.js was used as it provided a back-end JavaScript runtime environment to execute JavaScript code outside of a browser. In addition, Express.js [10] was used in conjunction with Node.js [9] to create back-end APIs.

6.4 Database System

PostgreSQL [6] was used for the database as it is compatible with many operating systems and simple to use. The database was hosted on an Amazon RDS [3] server for ease of access along with using Amazon S3 [5]. By doing so, any changes made on the database will be reflected immediately across all our local development sites.

6.5 Recommender System

Python3 [7] was used to develop the recommender system as it is widely used in the field of artificial intelligence and machine learning, hence there is a wide array of well-developed and documented libraries available to be used. Among them are pandas and numpy, which allow us to manipulate data efficiently. The NLTK library was used as it provided useful functions for natural language processing such as word stemming, while scikit-learn was used to vectorize and compute the cosine similarity of documents. Lastly, psycopg2 is used to communicate with the PostgreSQL database.

7. Implementation Challenges

Designed the front-end back-end based on n-layer architecture. Basically, it handles separation of concerns. The code is divided into multiple layers so that the code is loosely coupled and as framework agnostic as possible.

7.1 Front-End and Back End

React had a very steep learning curve. None among us was a full fledged developer so we faced a lot of issues with the making of the website from the design aspect to error debugging to the lack of awareness of many libraries which would have made our lives easier. Initially, majority of the time and resources were put into the backend part and this, along with the priorly mentioned inexperience of the front end team, made the front end lag behind a lot when compared to the backend. This led to a series of issues because of the delay in debugging of the APIs in the back end. And thus made us severely lack behind our set schedule.

7.2 Recommender System

In order to provide real-time recommendations and ensure that the user experience is not compromised, the calculations were computed in advance. These calculations are then serialized and stored in a Python3 pickle object. Whenever a recommendation is needed, the application will deserialize the pickle object and extract the relevant data needed. If the calculations were to be made on-the-fly, the time complexity of the algorithm would be $O(n^2)$. For a large enough database, this would take a really long time to provide recommendations and consequently will negatively affect the user experience. Everytime a recipe is created or updated, the calculations will be updated as well. However, this will run in the background and should not affect the user experience in any way or form.

8. Future Work

Due to time constraints, there were a few novelty features that did not manage to be implemented. One such feature is the payments system, where recipe explorers can make payments or donations to recipe contributors if they so desire, similar to a Patreon or YouTube Premium system. Explorers who paid money could then get early access to new recipes and even get personalized feedback on their cooking from the contributors. Another novelty feature that did not make it into the final version is a “courses” feature, where recipe contributors could create cooking classes for beginner explorers

9. User Documentation / Manual

We will be running the system on the Lubuntu 20.4.1 LTS virtual machine image.

First, update the packages and install pip for Python3 on the Lubuntu VM.

```
sudo apt update && sudo apt upgrade -y
sudo apt install python3-pip
```

To install Node.js,

```
sudo apt install npm
```

To download the latest source code from GitHub,

```
git clone
https://github.com/COMP3900-9900-Capstone-Project/capstoneproject-comp9900-w16a-fifa.git
```

Make sure that `config.js` is in `capstoneproject-comp9900-w16a-fifa/server` and `config.ini` is in `capstoneproject-comp9900-w16a-fifa/server/recommender`. These two files contain the login credentials required to connect to the PostgreSQL database.

To install the required Python libraries needed by the recommender system,

```
cd capstoneproject-comp9900-w16a-fifa/server/recommender
pip install -r requirements.txt
```

The nltk Python library needs to download a set of corpus in order to perform word stemming. This will only need to be run once.

```
cd capstoneproject-comp9900-w16a-fifa/server
python3 run.py
```

In separate terminals:

To start the back-end code,

```
cd capstoneproject-comp9900-w16a-fifa/server
npm install
npm run start
```

To start the front-end code,

```
cd capstoneproject-comp9900-w16a-fifa/client
npm install
npm run start
```

References

[1] Horoz, M. (2020, December 31). *Building a Content-Based Food Recommendation Engine*. Medium.https://medium.com/swlh/building-a-content-based-food-recommendation-engine-df2ac7d08129

[2] Prabhakaran, S. (2020, October 11). *Cosine Similarity - Understanding the math and how it works? (with python)*. Machine Learning Plus.https://www.machinelearningplus.com/nlp/cosine-similarity/

[3] *Amazon Relational Database Service (Amazon RDS)* https://aws.amazon.com/rds/

[4] *All Recipes Cooking Website* https://www.allrecipes.com/

[5] *AWS Amazon S3* https://aws.amazon.com/s3/

[6] *PostgreSQL: The World's Most Advanced Open Source Relational Database* https://www.postgresql.org/

[7] *Guido van Rossum at Stichting Mathematisch Centrum (CWI) - 1990s* https://docs.python.org/

[8] *React : A JavaScript library for building user interfaces* https://reactjs.org/

[9] *Node.js is a JavaScript runtime built on Chrome's v8 JavaScript engine* https://nodejs.org/en/

[10] *Express (v4.17.1) Fast, unopinionated, minimalist web framework for Node.js* https://expressjs.com/