Final Report

Meal Recommendation System

COMP9900-9900-M18R-BSoDwarning

Taste Studio

UNSW SYDNEY

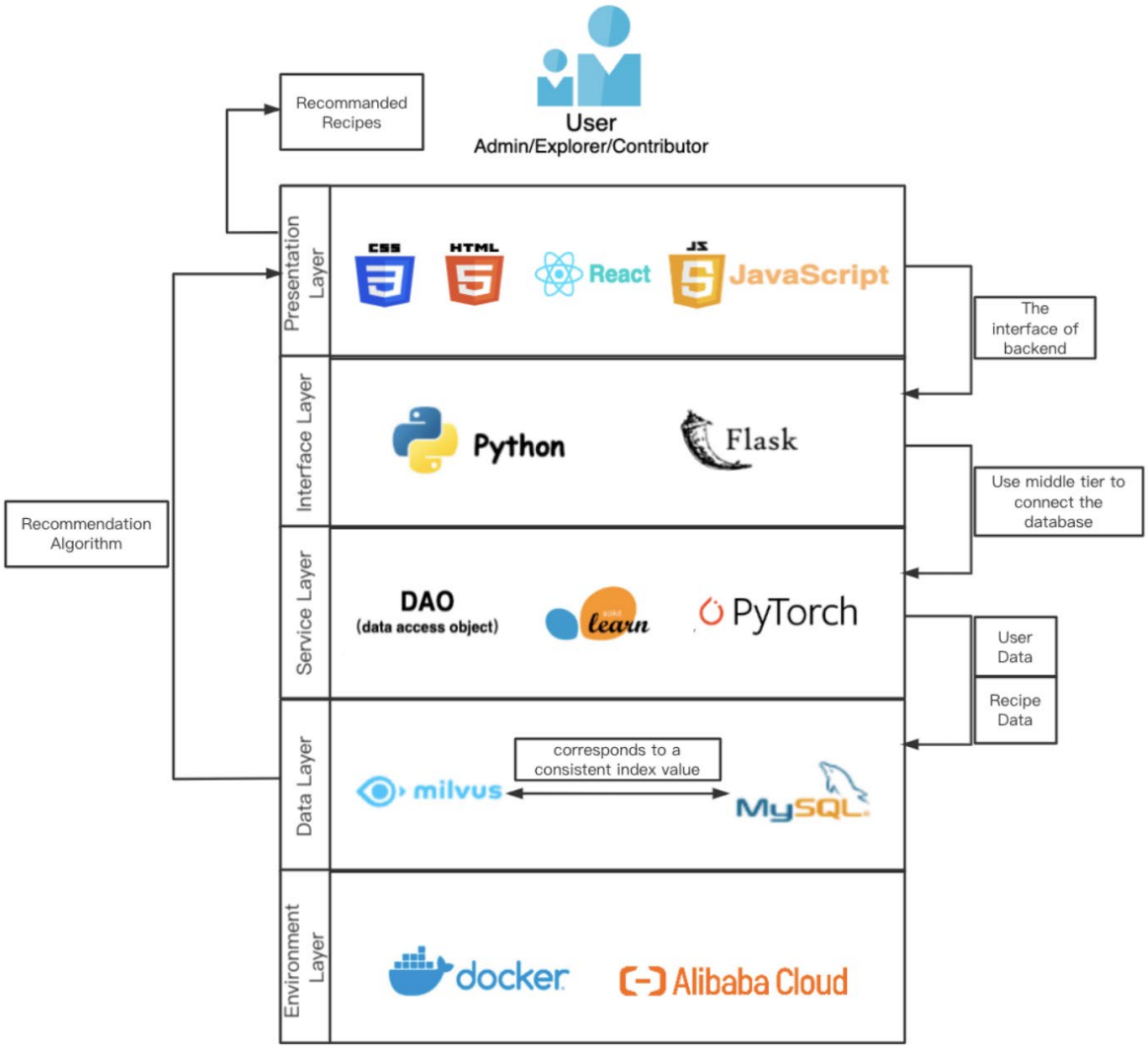
Name	Role	UNSW Email	Student ID
Chong Tong	Full Stack Developer	z5377912@ad.unsw.edu.au	z5377912
Laurence (Dian) Jin	Scrum Master Backend Developer	z5380108@ad.unsw.edu.au	z5380108
Qiushi Li	Frontend Developer	z5376373@ad.unsw.edu.au	z5376373
Jingze Chen	Frontend Developer	z5342303@ad.unsw.edu.au	z5342303
Shengtao Bian	Backend Developer	z5275203@ad.unsw.edu.au	z5275203

Submission Date: 18/Nov/2022

Table of Contents

1. Overview

1.1 System Architecture



1.1.1 Presentation Layer

For the presentation layer, we used React framework to implement the frontend. It is a highly used open-source JavaScript library which helps to create our attractive Taste Studio pages with minimal work and coding. We can develop user interfaces (UIs) that improve application speed. It contributes to modularity which is easier to reuse code with high maintainability.
It provides an abundant JavaScript library which is used to build user interfaces based on UI components. JavaScript libraries give us more flexibility to choose how we want to design the Taste Studio.

The designed pages consist of multiple components, each with its own logic and controls. These components output a small piece of reusable HTML code that can be reused anywhere we need it during the process of developing. Reusable codes help to make the Taste Studio easier to develop and maintain. These components can be nested with other components to allow complex applications to be built using simple blocks.

1.1.2 Interface Layer

In the interface layer, Python plays a crucial role in the implementation and development. And Flask is chosen as the web application framework to interact with the front end. Because the core of the Flask framework is simple, it can also maintain the scalability of functions for our project. When we use Flask to develop Taste Studio, we can add different functions according to our requirements and user stories. A variety of powerful plug-in libraries allow us to develop according to the user stories during the process.

Compared with the Django framework, because of the characteristics of flexibility, lightness and high efficiency, we chose to use the Flask framework to develop our project. Although the Django framework is also powerful and can quickly contribute to the development of websites, since various templates in the Django framework have been pre-packaged, it is difficult for us to design websites according to the requirements during the development process. This limits the application of the Django framework.

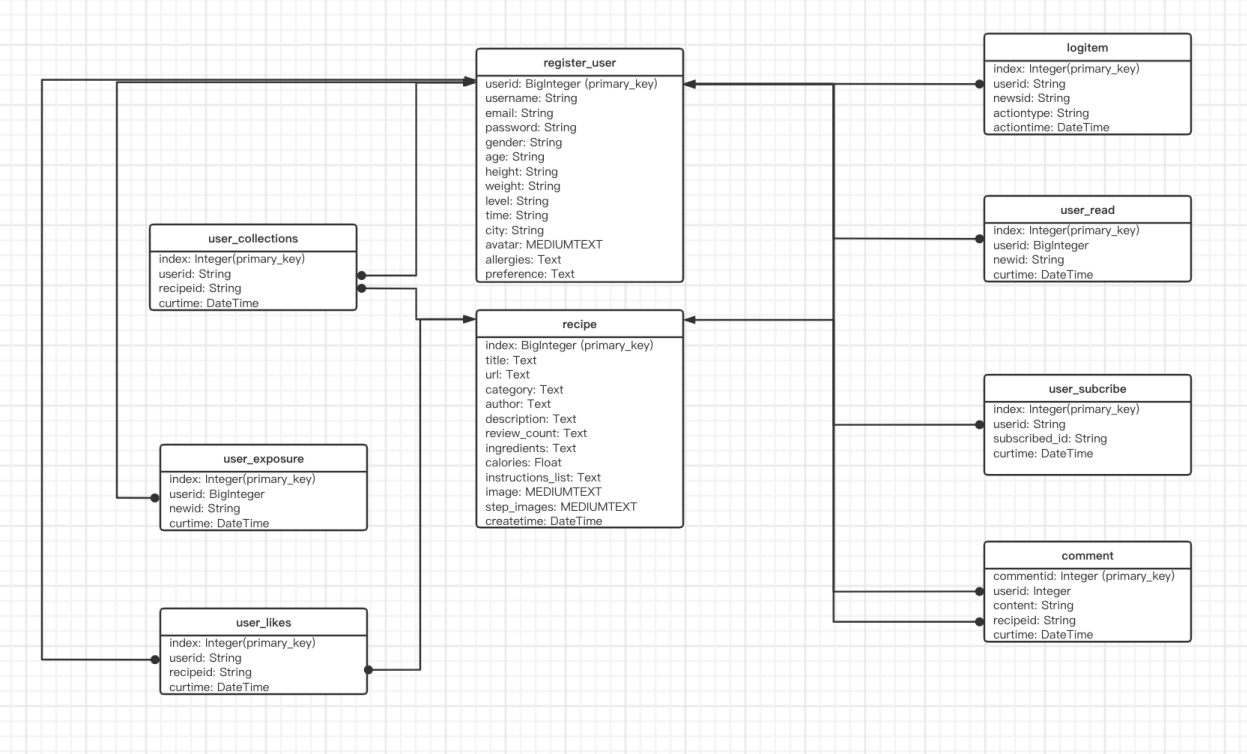
1.1.3 Service Layer

In the service layer, we will use some DAO frameworks (Pymysql, Pymongo etc.) to interact with the database. Recommendation algorithms are addressed to personalize recipe suggestions to users. In addition, Neural search algorithms (Embedding) are used for searching engines (elasticsearch) to facilitate users to retrieve the content they want. This algorithm is mainly implemented by scikit-learn or pytorch.

Through NLP technology, we will extract the ingredient name of each recipe in the database. Through the simCSE model, we will convert the ingredient name into a vector to do similarity search and store these into milvus. Then, this model will return the top six most similar recipe to the frontend. To be noticed, we will find if these six recipes contain the allergic ingredient or does not contain any same ingredient with its recipe. If so, we will not recommend this recipe to the user.

1.1.4 Data Layer

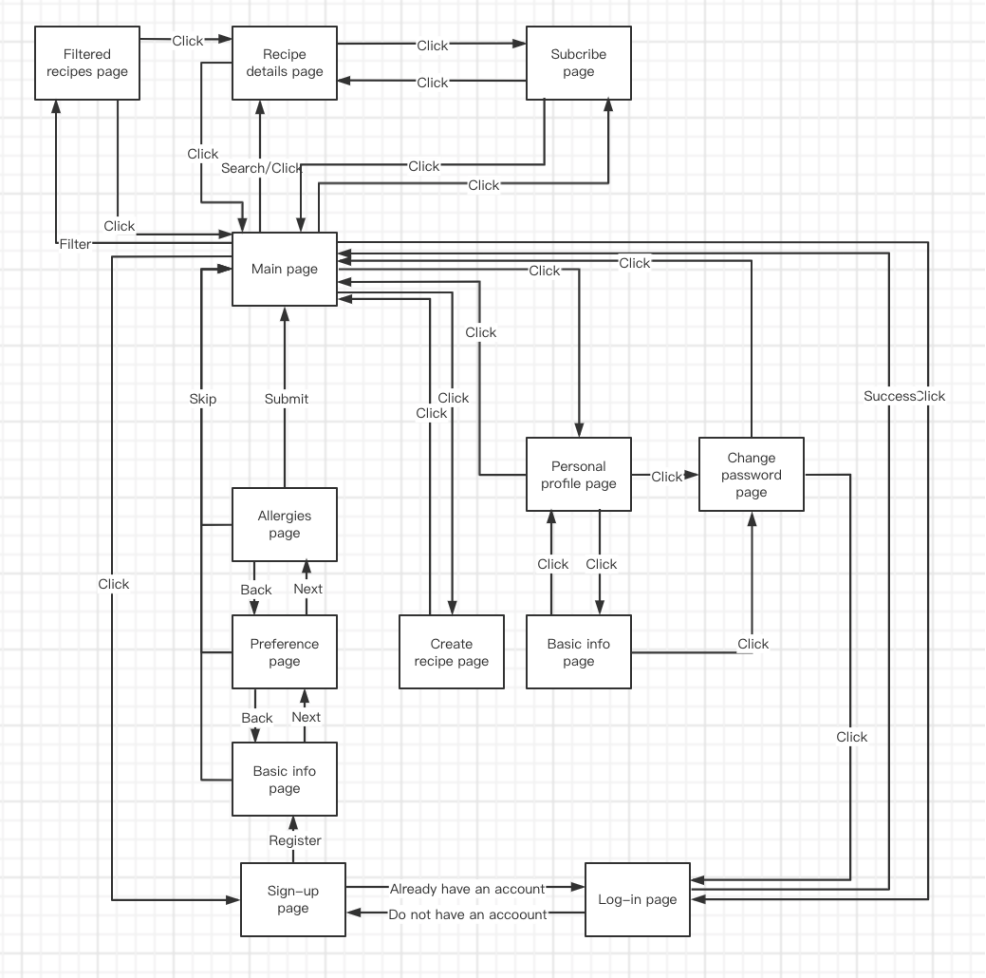
For the database, all the historical data will be saved in mysql. Milvus can help us easily deal with massive unstructured data retrieval in the Taste Studio. It enhances flexibility by providing support for various APIs. Besides, it supports multiple vector index types, such as quantization-based and graph-based indexes, as well as advanced query processing. Milvus uses a log structure to process dynamic vector data which can maintain the efficiency of data insertion and deletion, and support real-time search in the project.



1.1.5 Environment Layer

To improve efficient release and deployment of Taste Studio updates, we used Docker in the environment layer. It also helps to ensure the environment's consistency. From a development point of view, there are three basic steps including build, transport, and run. The container technology isolates an independent running space, does not compete with other applications for system resources, and does not need to consider the interaction between applications. Besides, Docker provides a way to manage the development environment for us which ensures the synchronization of the environment for testing.

1.2 Page Structure



1.3 Project objectives

1. Users are asked to register and they can own persistent data to retain their collection (Novel), preference, watched history and personalized recipe recommendation.
2. Users are able to click the thumbnail to view details of the recipe.
3. The system authorizes users to navigate the recipes that they would like to view based on recipes name and tags.
4. The system needs to recommend the user's other recipes similar to the recipe that the user navigated and avoid the allergic (Novel) one.
5. Users are able to deliver their opinion for all the recipes in terms of liking and making the comment.
6. The system authorizes users to contribute and modify their own recipes with each detailed information.

7. Users are able to have their own profile to view their personal information and recipes published by themselves.
8. Users are able to subscribe and unsubscribe the contributor and they can get the new post from the subscribed contributor.
9. Users can sort the order of posts by the contributor they subscribe to in order of likes and times.
10. Users can search the recipe through image upload in which we will implement it by AI model. (Novel)

User type

- Explorer (EX) - Users who want to find the recipe and are recommended some potential interesting recipes.
- Contributor (CO) - Users who want to contribute and share their recipe to others.
- Registered User (RU) - Users who own persistent data and can be authenticated, and users are compulsory to register.

*Specific user stories contained in **Appendix**.*

2. Functionalities developed

The specific user stories can be checked in Appendix below and functionalities mapping with the objective number and user story code. To be noticed, all the icons will show their name when you put the mouse above them.

2.1 Registration

This part addresses **objective 1** and user stories code **RU1** and **EX11**.

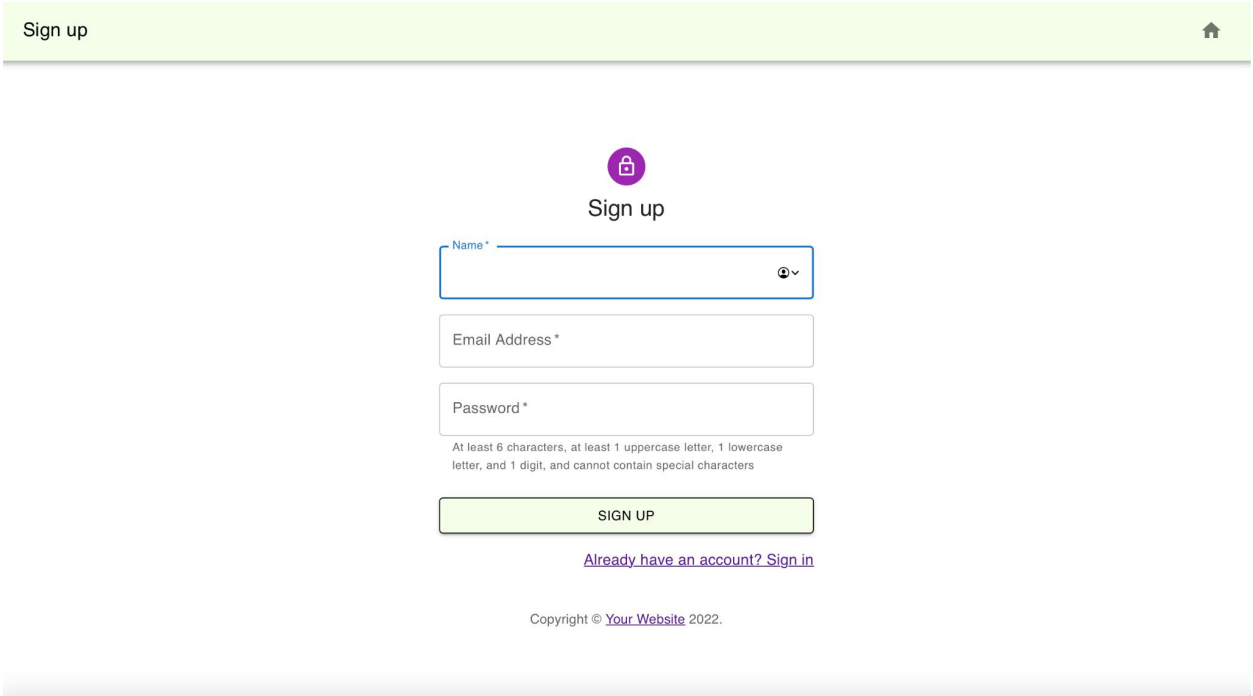


Fig. 2.1.1 Register

This page allows users to sign up with the account, also they get a convenient way to fill their profile and also allergic food (novel) after clicking the sign up page. Also, they can skip the personal information fill.

Condition	**Result**
Email is invalid or existed	Display the relevant message with error and the user needs to retry.
Password • is less than 6 characters • does not contain an upper letter • does not contain number and letter in the same time	Display the relevant message with error and the user needs to retry.

• contain the symbol.	
Click the “SIGN UP” button without the previous errors.	Users will create their own account and go to the page to fill in the private information.
Click the home icon	Back to the main page.
Click “Already have an account”	Navigate to the Sign in page.

After clicking “SIGN UP”

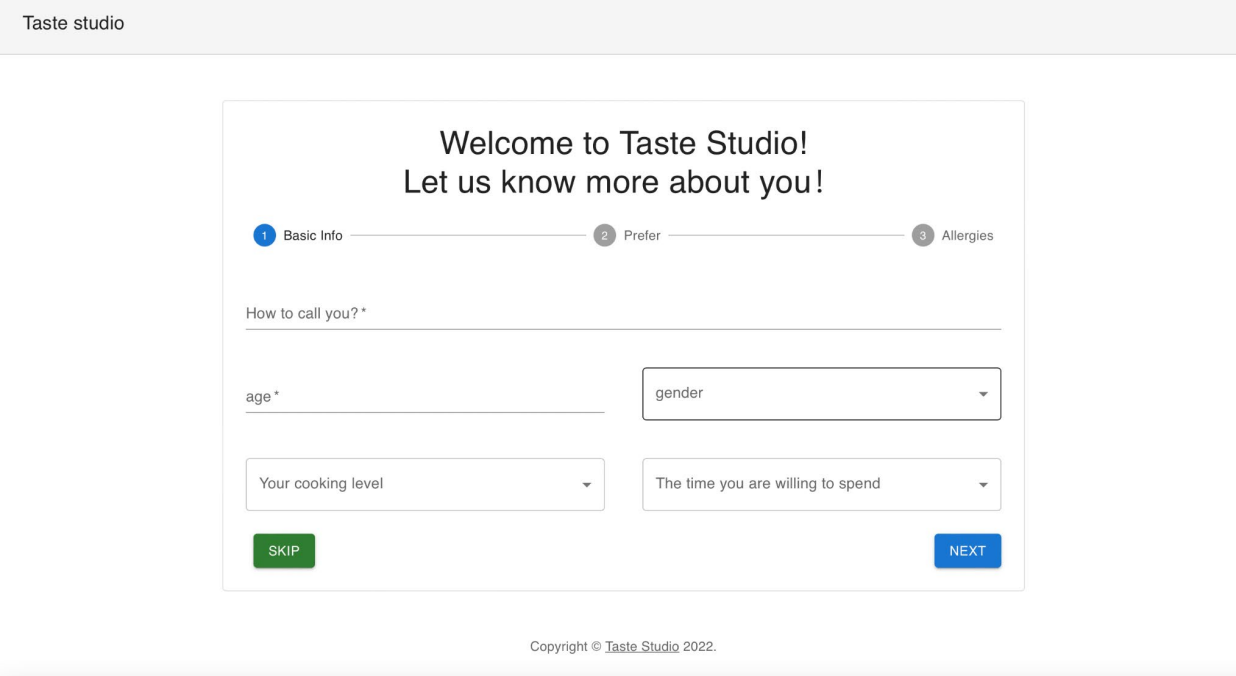


Fig. 2.1.2 Fill the personal information

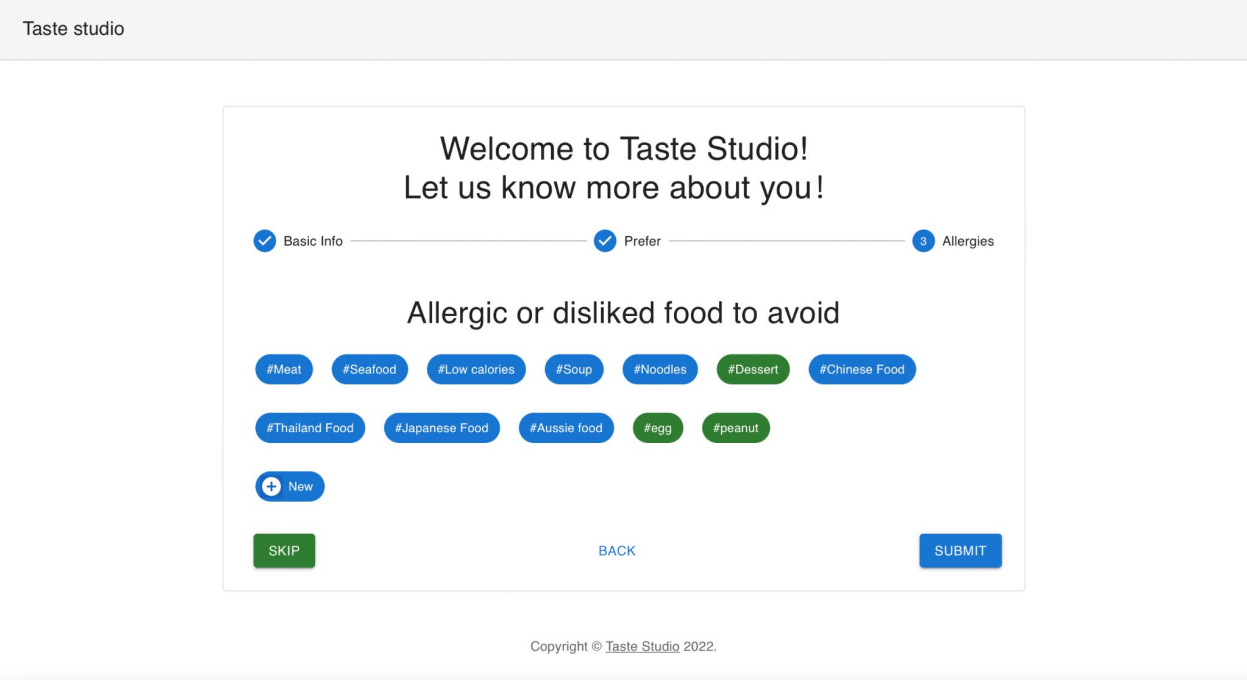


Fig. 2.1.3 Fill the allergic food

After clicking “SKIP” or “SUBMIT”

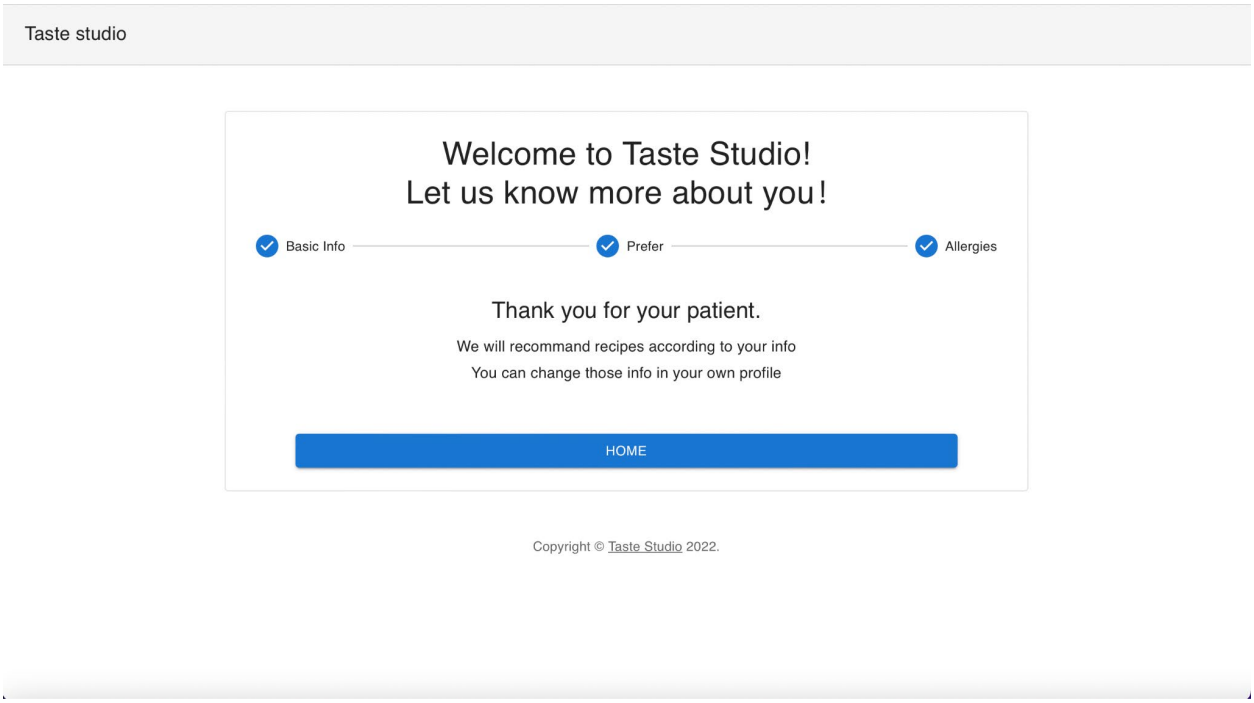


Fig. 2.1.4 Finish the attempt

Condition	**Result**

Click “SKIP”	Finish registration without filling the personal information.
Click “NEW”	Users can add more customized preferred or allergic food.
Click “SUBMIT”	The personal information will be stored in their profile.
Click “HOME”	Back to the main page and they can remain logged in.

2.2 Login

Address **objective 1** with **RU2** and **RU5**

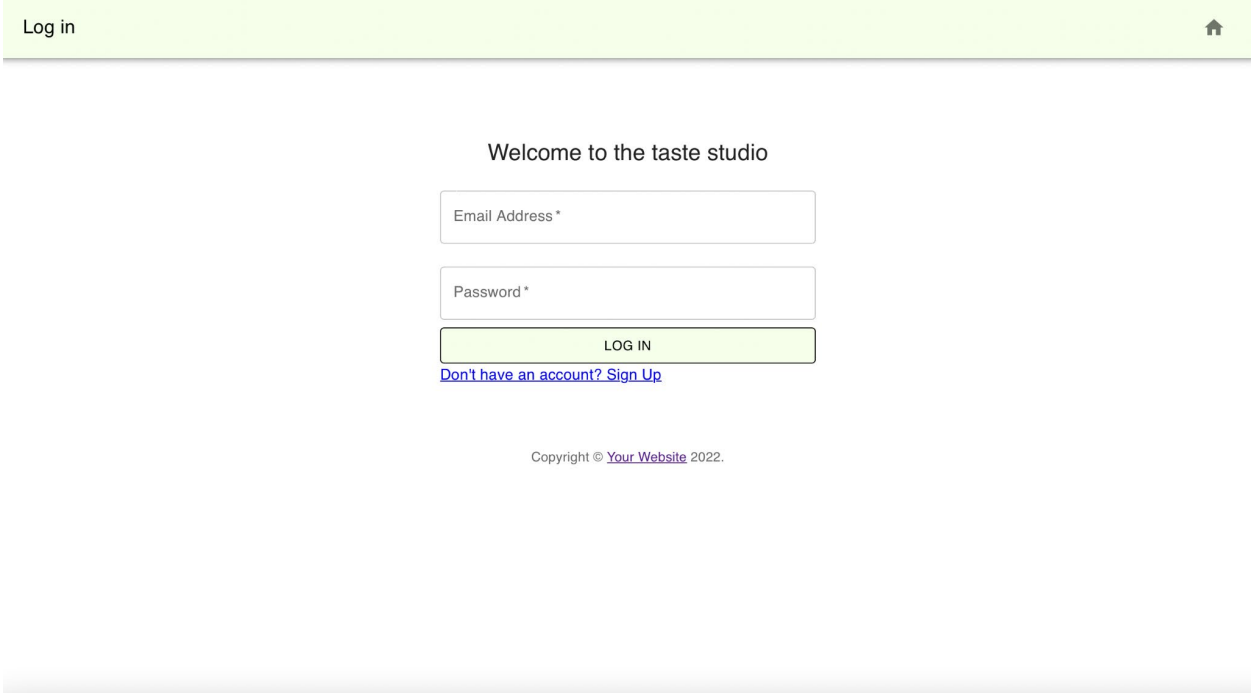


Fig. 2.2 Login

Condition	**Result**
Invalid email or incorrect password.	Display the relevant message with error and the user needs to retry.
Click the blue color sentences underline as “ Don’t have an account? Sign up”.	Navigate to the page of registration
Click “LOG IN” and log in successfully.	Navigate to the home page and display the message of successful login.

Log in the same account in another page	Two websites sync the information with the same account.
Leave it in a long time	Remaining login status.
Click the home icon	Back to the main page.

2.3 Homepage

These pages are to map **objectives 2, 3 & 10** and address **UR3**, **CO1**, **EX1**, **EX3, EX6, EX7, EX10** and **EX12.**

The first page is when the user does not login and exclude functionalities of subscription and create the recipe which is **CO1** and **EX7**. The second page will contain **CO1** and **EX7**.

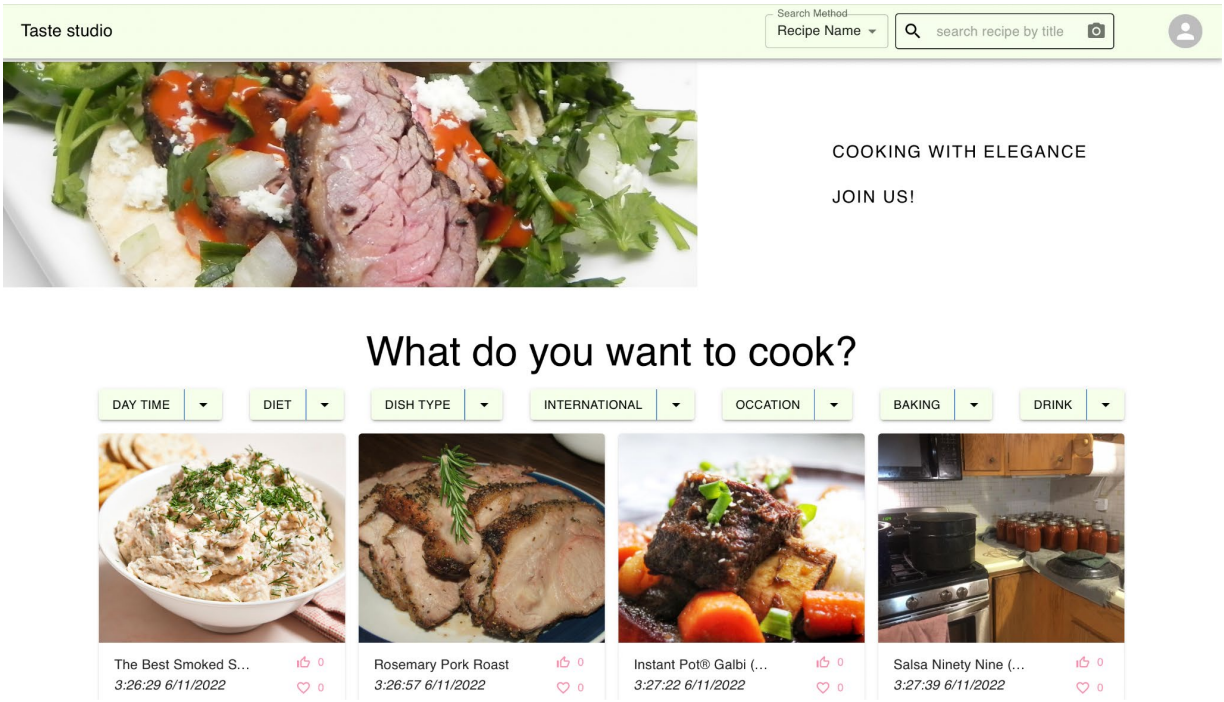


Fig. 2.3.1 Homepage without login

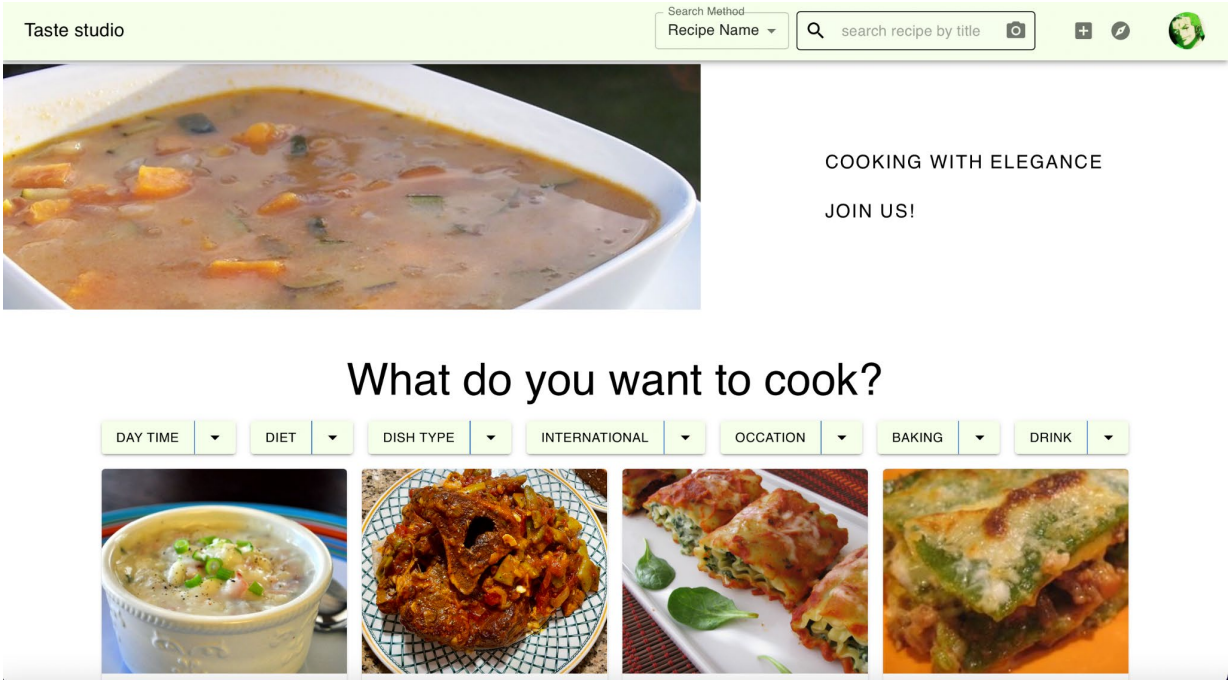


Fig. 2.3.2 Homepage with the user login

That is, for unregistered user they can have several functionalities mapping to user stories, they can:

- Search the recipe by name or ingredient.
- Filter the recipe category from the pop-up box with some given tags.
- Click to like the recipe.
- Click the recipe card to navigate the details of the recipe.
- Navigate to the profile page through the right-top icon.
- Save the recipe to their favorite list. (novel)
- Search the recipe through the image of food. (novel)

For registered user they can:

- Contribute the recipe to the system
- Subscribe other user
- Log out

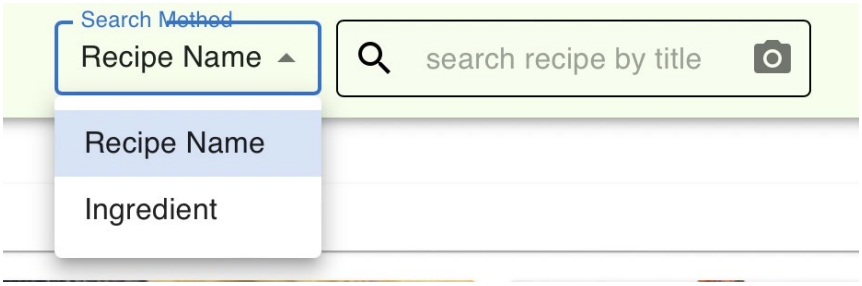


Fig. 2.3.3 Change the search method

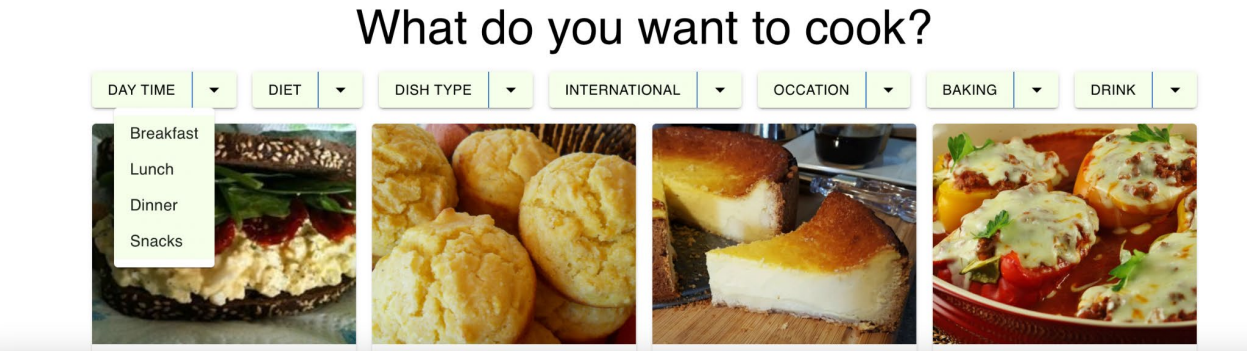


Fig. 2.3.4 Filter the category

Condition	**Result**
Click “Search Method”	Pop out the box of search with Recipe name or ingredient.
Click “Recipe Name” and type something to search	Navigate to the search result page containing many relevant recipes.
Click the pop-up box below like DAY-TIME and click “breakfast”	Navigate to the search result page containing many relevant recipes.
Click the camera icon to upload the image to search.	Navigate to the search result page containing many relevant recipes.
Click the poster or recipe card below	Navigate to the page of recipe detail.
Click the like to recipe with thumb icon	With like, the solid love icon will be displayed on the page and the number of this recipe will increase by 1. Also, users can unlike.
Click love icon	Collect and save the recipe or cancel the collection, if you are not user, you can’t see the collected recipe.
(Registered user) Click the head portrait and click logout	Log out the account and stay in the homepage.
(Registered user) Click the head portrait and click profile	Navigate to the page of the profile.
(Registered user) Click the icon of add to contribute the recipe.	Navigate to the page to contribute the recipe.
(Registered user) Click the icon of compass to see the subscription.	Navigate to the subscription page

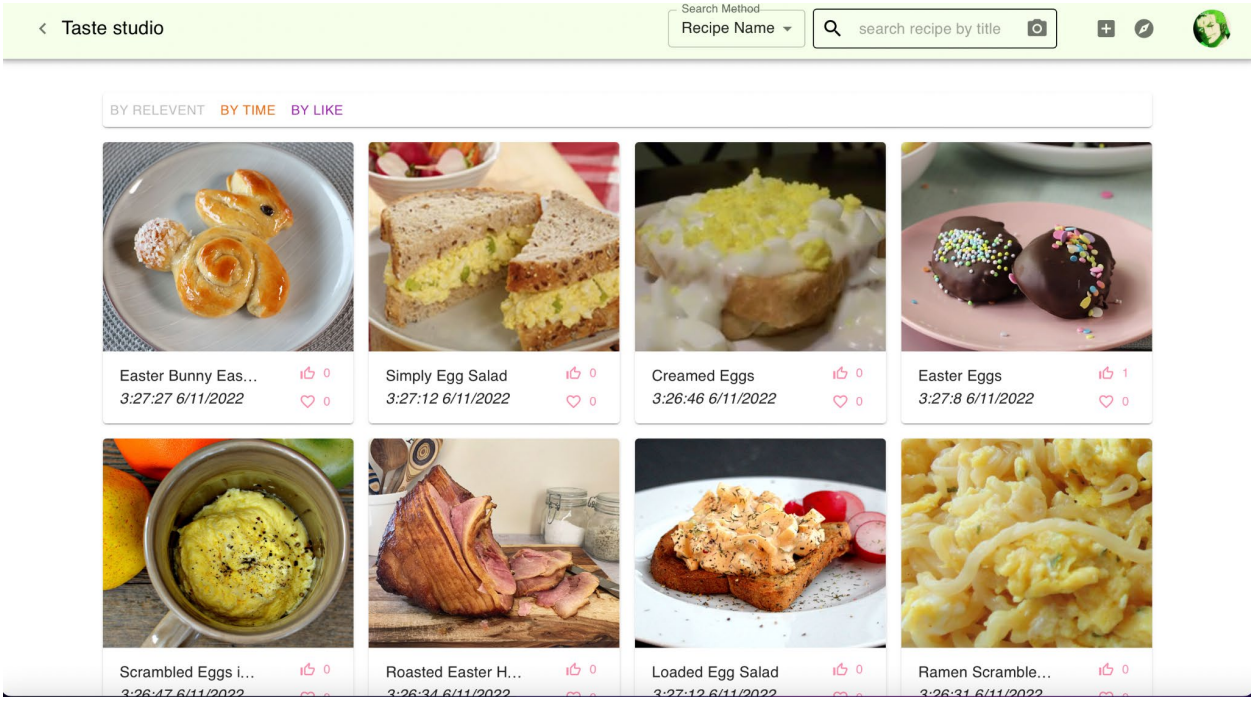


Fig 2.3.5 Search page

Condition	**Result**
Click “By time”	Sort the order of image by the most recent time.
Click “By Like”	Sort the order of image by the number of likes.

2.4 Recipe detail

Address **objectives 4 & 5** with **CO2, EX2, EX4, EX5, EX7, EX9, EX10** and **EX11**

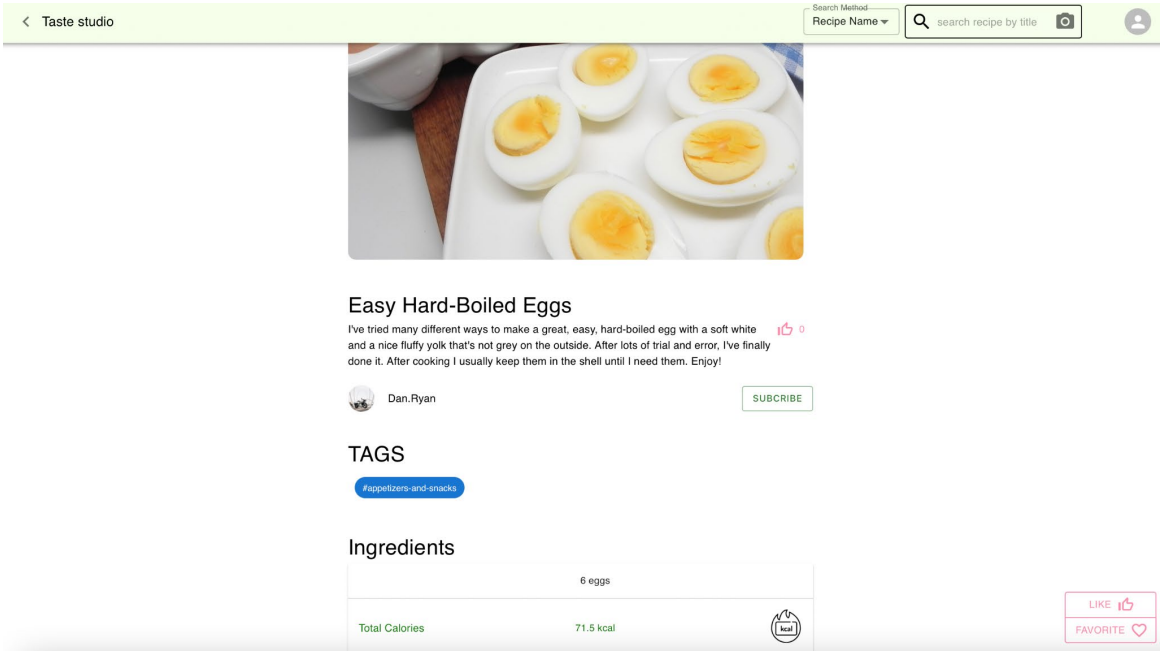
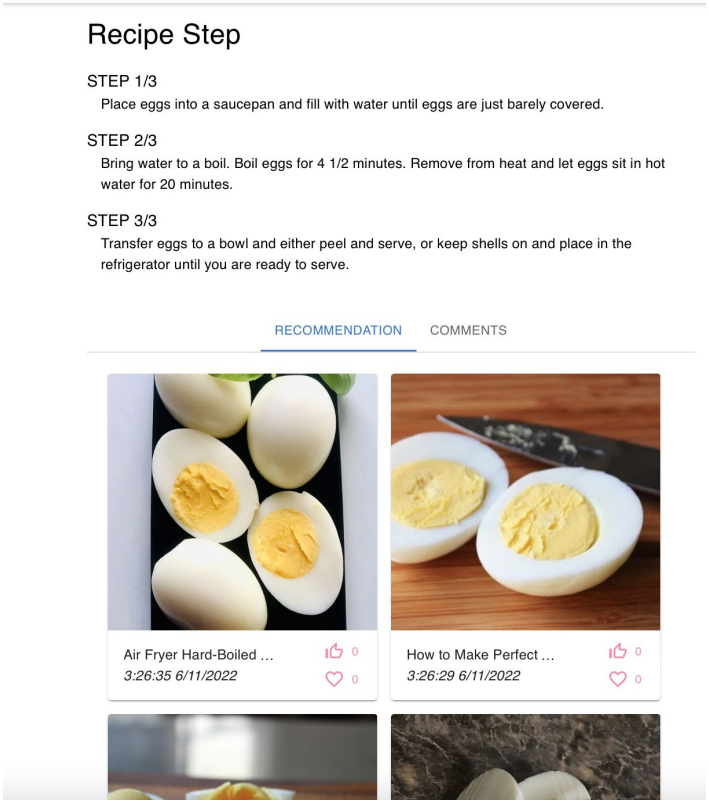


Fig 2.4.1 Recipe detail



This page only occurs when users check their own recipe since they can edit or delete the recipe and like and collect will not appear.

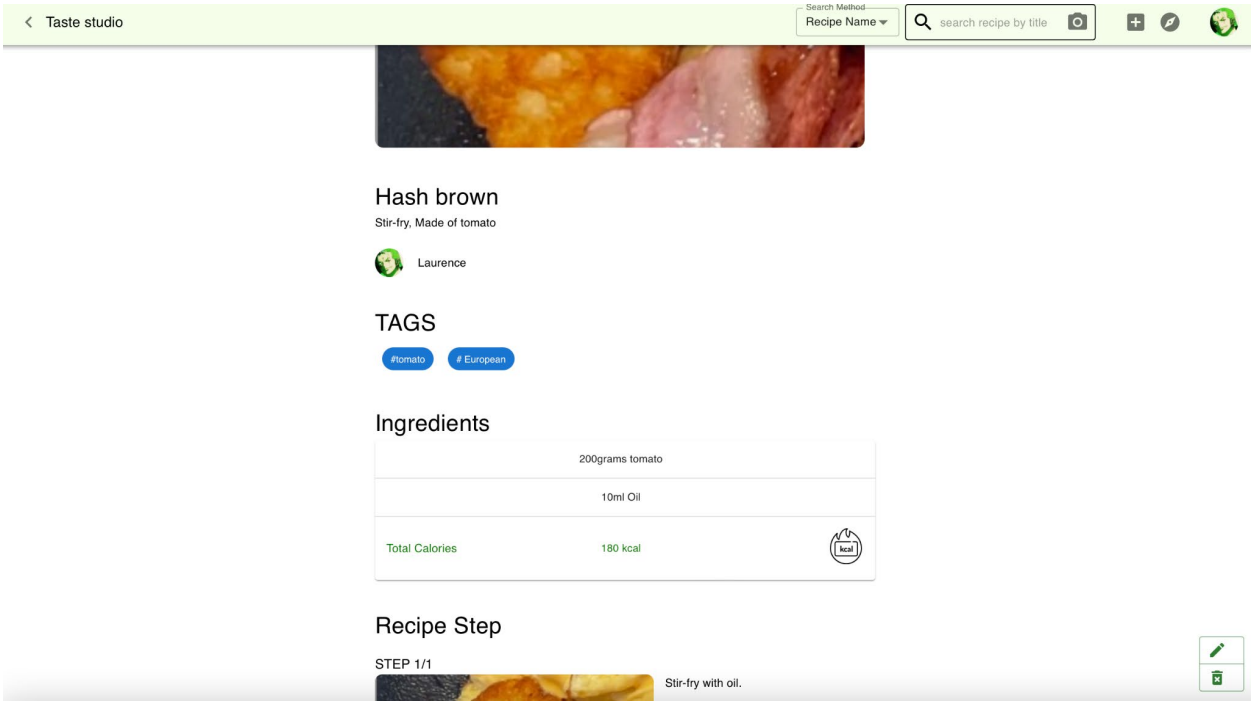


Fig 2.4.2 Edit or delete the recipe

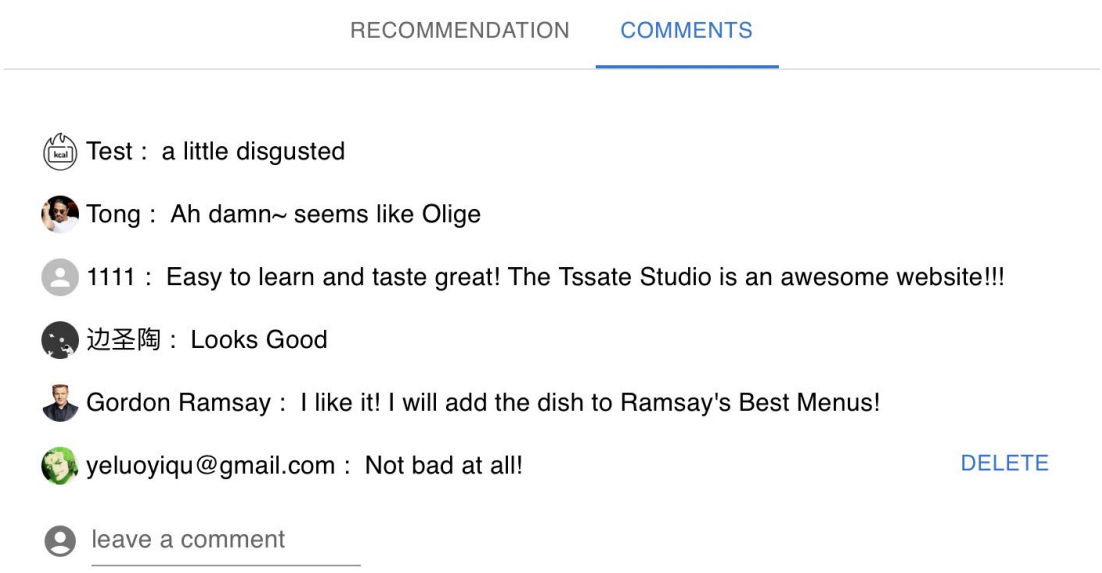


Fig 2.4.3 Leave the comment of the recipe

The page of recipe detail mapping to the following functionalities. Users are able to edit and delete their own recipe. They can view the details of the recipe with all relevant information especially calories (novel). Users will get recommendations from similar names and ingredients

of this recipe in which the system will avoid the recommendation of allergies (novel) and dislikes. Users can like or comment on the recipe. Users can collect and save the recipe (novel). They can go to the subscription page and profile page.

The action related to the result can be seen as:

Condition	**Result**
Click “Subscribe”	Subscribe the user or unsubscribe
Click thumb icon	Like or unlike. Increase the number of like if like
Click love icon	Collect and save the recipe or cancel the collection
Click “Comment”	Change to subpage as comment page
Click “Recommendation”	Change to the default subpage
Type the comment and enter	Post the comment from the bottom
Click “Delete” in comment	Delete your own comment permanently
If it is your recipe, click the pen icon	Navigate to the page to edit the recipe
If it is your recipe, click the trash icon	Delete the recipe permanently and navigate to the homepage

2.5 Profile

Address **objective 7** with **RU4**, **RU6** and **CO3**

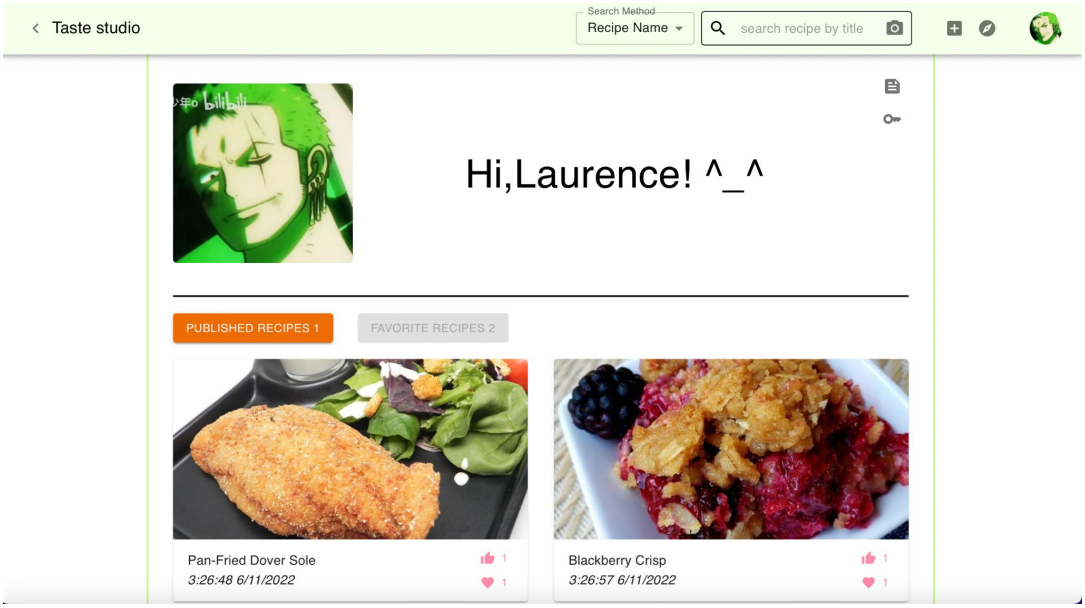


Fig. 2.5.1 Profile page

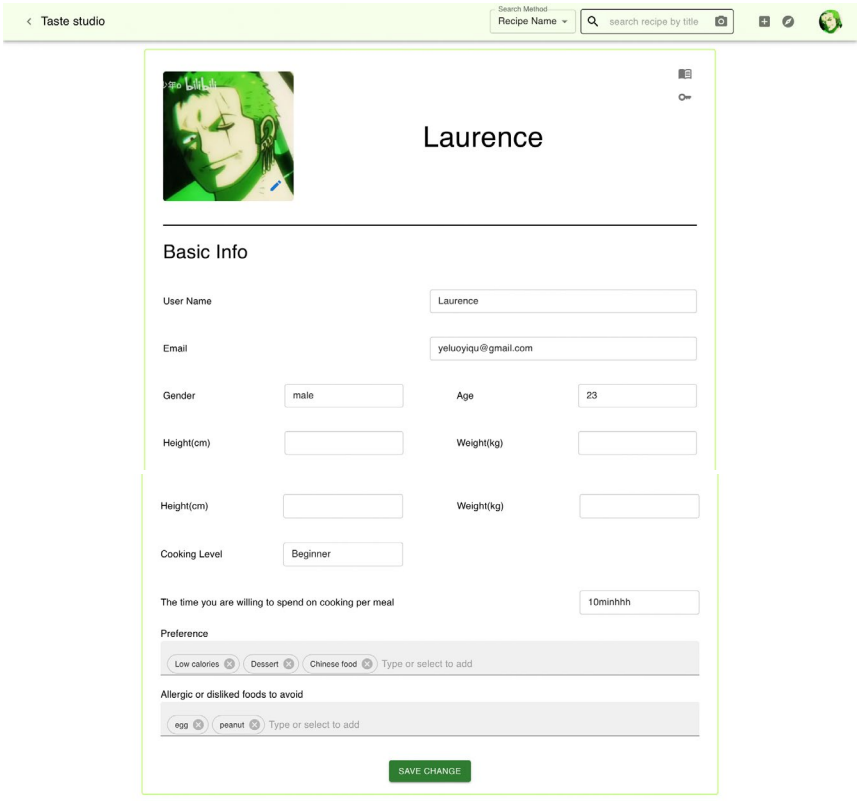


Fig. 2.5.2 Edit the profile

In brief, this page maps to functionalities as users can view their published recipe and collected recipe, and users can modify their profile and reset the password.

The action related to the result can be seen as:

Condition	**Result**
Click the first icon as a one page book in Fig.2.5.1.	Navigate to the page of profile modification in Fig. 2.5.2
Click the second icon as a key	Navigate to the page to reset the password
Click “Favorite” and “Like”	Like and favorite or Unlike and unfavorite
Click “Save change”	Save the modification of the profile
Click the icon of reading book in Fig. 2.5.2	Back to the previous profile page without saving the modification.

2.6 Contribute the recipe

Address **objective 6** with **CO1**

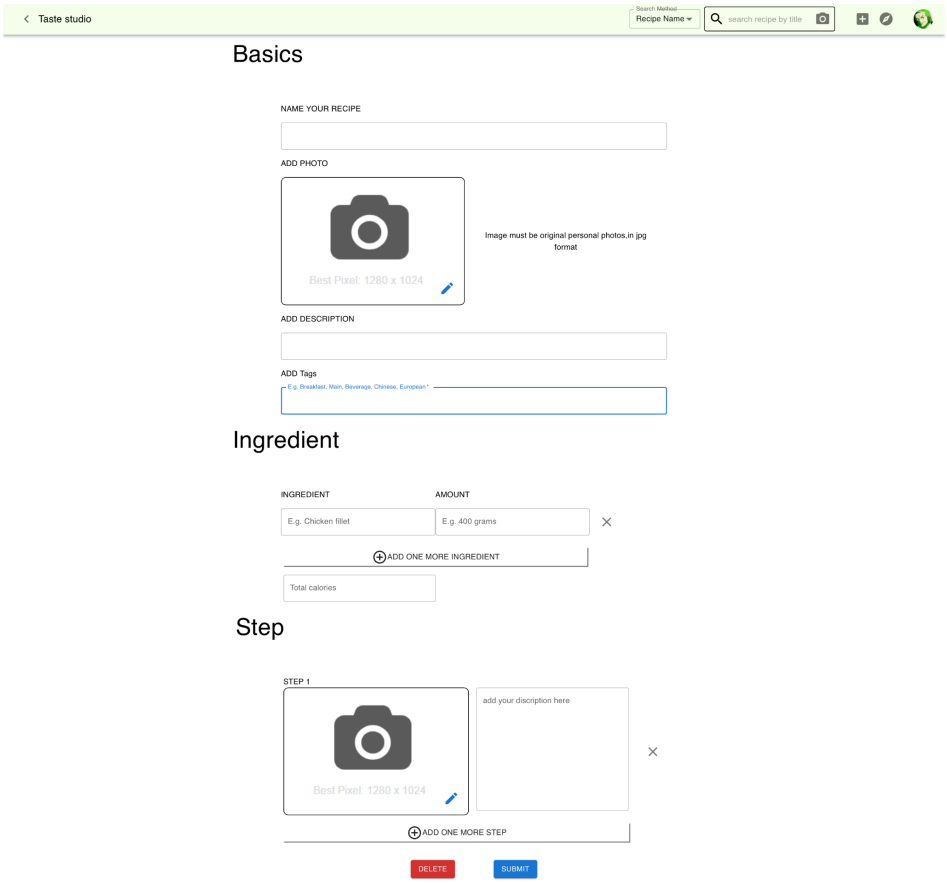


Fig 2.6 Contribute the recipe

Registered users can upload their recipe freely.

Condition	Result
Click “ADD ONE MORE …”	One more line to add ingredient or step
Click delete icon	Remove one line of step or ingredient and it can remove all the terms.
Click “SUBMIT”	Submit the recipe to the system with a message and navigate to the homepage.
Click “DELETE”	Discard the draft recipe and navigate to the homepage.

2.7 Subscription

Address **objective 8 & 9** with **EX7** and **EX8**

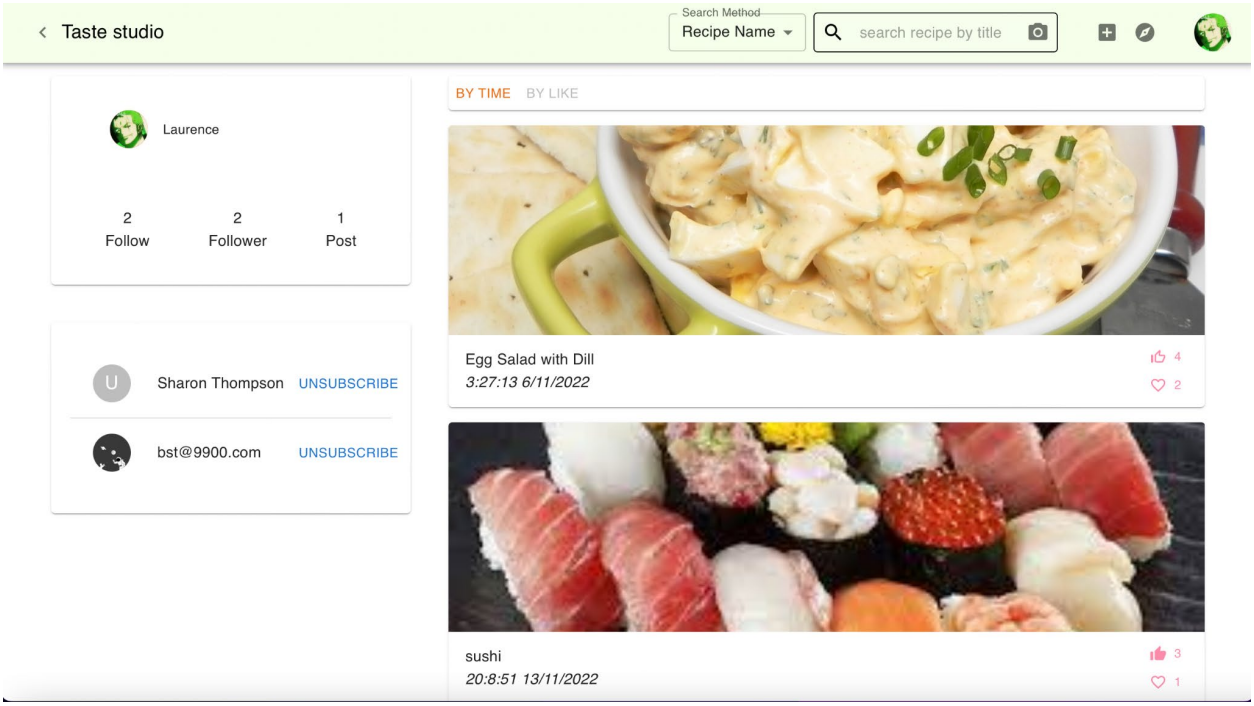


Fig 2.7 Subscription page

This page maps to the functionalities that users can manage their subscribed user and see their post. Also, they can sort the order by the most recent time or the most like freely.

Condition	Result
Click “UNSUBSCRIBE”	That user will be removed and unsubscribe and the page will

	refresh to display there is no user in the subscription.
Click “BY TIME”	Sort the order of recipe posted by the subscribed user with the most recent time.
Click “BY LIKE”	Sort the order of recipes posted by the subscribed user with the most likes.

3. Third-party functionalities

3.1 Frontend

3.1.1 Material UI

During the process of developing frontend pages, we used Material UI which is a library of React UI components that implement Google’s Material Design design principles. (Material UI, 2022) Material-UI can make the color of our pages, as a recipe recommendation system, more vivid and the animation effect more prominent. It implements many practical components, and supports us to modify the component's theme, internal style, color and other customized operations.

3.2 Backend

3.2.1 SQLAlchemy

SQLAlchemy is a powerful Python ORM toolkit and we can use URLs to specify the database to be accessed. Firstly, we need to create an Engine object which implements the encapsulation and scheduling of various database clients. It is the entry of all applications. To use the SQLAlchemy library to operate a database, we need to obtain an Engine object, and all subsequent operations on the database must be performed through this object. After the connection of the database, we can start defining the data model and create the class that maps the data table. The session allows us to operate the database entry through SQLAlchemy. We use the connection management function of the engine to realize the session. Finally, we can realize adding, deleting, modifying and checking the database of Taste Studio. Figure 3.2.1.1 in Appendix is an example of the comment table in the database which can show the process described above.

3.3.2 Milvus

We chose Milvus because of high performance when doing vector searches on large datasets. PyMilvus is a middle tier used in the project. It can directly operate on the object and isolate business logic and data access details. Besides, whether it is a DBS, a local library, or a cloud service, the codes can be applied to various Milvus server states without changing. (Milvus,2022)

3.3.3 pandas

Pandas is a NumPy-based tool which is used to solve data analysis tasks. There is a large number of libraries and some standard data models that provide the tools for efficiently manipulating large data sets. Through the functions and methods we can process data quickly and easily.

4. Implementation challenges

4.1 Frontend

I. Frontend and backend joint debugging

The main challenge that we met in the frontend is frontend and backend joint debugging. There exist plenty of interfaces that need to be designed and tweaked to fit the demand of functional requirements in our system. Meanwhile, we push the progress of backend development whilst the frontend development is going on, that is, we take the strategy of dividing and conquering in this short-term project. In this case, we need to apply the preset mock data of backend in the progress of frontend development so that the interface can demonstrate the temporary content and form.

However, the process of joint debugging with the implemented backend makes a huge change in terms of port, form and template. After the frontend developer learns a session in sprint 1 joint debugging, we hold a retrospective meeting and declare the possible backend data clearly and this optimizes the logic to configure the preset term and shorten the unnecessary time-consuming joint debugging process.

II. Local storage employment

The second big challenge is to upload and make the image search in our novel functionality. In usual, the search functionality is activated by the text such as name, ingredient and title. Nevertheless, the novel search way of image search for a recipe in which this caters to the demand of users in the era of AI. In this case, we considered converting the image into base64 to the path directed to the backend but the problem occurs as base64 always exceeds the limitation of parameter of path. Also, the file of the image is not accepted by path because this is an object.

After the exploration and research, this problem has been resolved by local storage manipulation in which the image is passed to local storage and then the backend will go to local storage and extract the image to operate the following process.

Furthermore, after we learn the knowledge of local storage, we also solve another challenge of the functionality of remaining login. The local storage is employed to store the data of personal identity verification to keep the status of remaining login.

III. Optimize the interface

It is common that a good design of interface can enhance the user's willingness to choose and use the software in the long term. Therefore, the reasonable and clear human-computer interaction can be considered as a challenge.

Firstly, we optimize the visualization of uploading the image significantly since the look of default upload functionality in HTML is pale and not consistent with the interface design. Indeed, it cannot be changed. After the exploration, we found that the DOM element solves this problem in which it can hide the base box of input and we can write a new button to monitor the movement of clicking. Once the user clicks, it will mock the CSS and hide the box. Moreover, the degree of convenience for users will increase to achieve the effectiveness.

Secondly, we employ the Map function of useState in all the pages of contributing the new recipe and then the filled text in the box will not be saved and displayed after we add a new line of ingredients. Then we make a lot of effort and find we need to make the department of adding the ingredient isolated, that is, useState only works in this part.

4.2 Backend

4.2.1 Recommendation Challenge

I. Model selection

There are many functionalities that we need to employ the recommendation system such as recommendation on the recipe page and search the recipe by image (novel). After the research, we made a first try with the "word2vec" model of NLP. Generally, Word2vec is to convert the name and ingredient into the vector based on its own lexicon and then perform similarity search. However, this model has some obvious drawbacks in the aspects of effect, the difficulty of training and the content of the library. This model doesn't have any pre-trained model so we need to spend a great deal of time and way to train its best effect but this project of Agile methodology is time-limited.

To solve this challenge with the optimal solution, we make a huge effort to research other relevant NLP models. Finally, Giao et al.(2021) has delivered the "SimCSE" model which solved this problem effectively in which this model has a pre-trained model so we can perform fine-tuning. Also, the theory of SimCSE is to convert names into vectors to perform similarity search and put these into "Milvus". Also, Milvus has multiple utilities in which it also performs text search. At last, we achieve better performance with less time-consuming and space-consuming.

II. Data type embedding

High-fidelity demonstration and usability of the software is what we want since the origin user needs some existing recipes to operate, so we gather 30,000 recipes from "Scrapy" of Python in complete detail. In this case, we extract the image in the form of a URL since the dataset is

enormous. In the first thought of the “SimCSE” model, we consider the convenient way to search image by image.

However, the error occurred in this way in which the highest similar recipe always is the new recipe created because only the new recipe stores the image in the form of png or jpg. Polymorphism can resolve this challenge in terms of the “clip” model which is a multi-template model in SimCSE (Giao et al, 2021). In this case, “clip” model can embed the text (name and ingredient) and image to perform a similarity calculation.

4.2.2 Database Challenge

I. Image transmission

The common way to transmit the image is to convert it to base64 and pass to the backend attached in the JSON file so that we employ this. However, the browser request and dataset storage for the size of image is limited. In this case, we filter the unnecessary part of transmission which will not be affected by the backend and set the store size of image from default 16KB to 16 MB.

II. Consistency

Boundary problem is another challenge in joining between frontend and backend especially since we have various types of function and interface. We resolve this through checking the consistency of type and null of data passed from JSON in which all the teammates perform unit testing and other real users perform beta testing. Not only this, the consistency from sql to milvus is necessary, some error occurred before since inconsistency between this. In this case, we resolve the challenge through the consistent index between two containers and synchronous existence.

5. User manual

5.1 Load the Frontend

1. Clone or download the frontend folder.
2. Make sure you have these dependencies: npm, node, yarn.
3. Next, enter the dictionary of frontend, and use the command lines:

   $ yarn install

   $ yarn start

4. After 1~2 minute, the fronted code start, and you can see the frontend launch success message:

```
$ yarn start
yarn run v1.22.19
$ react-scripts start
(node:1792) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarnin
g: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewar
es' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:1792) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarni
ng: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlew
ares' option.
Starting the development server...

Compiled successfully!

You can now view frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.2.7:3000

Note that the development build is not optimized.
To create a production build, use yarn build.

webpack compiled successfully
```

5. The default browser will automatically open the frontend page.

5.2 Load the Backend

1. Clone or download the backend folder.
2. Install MySQL from docker:

   docker pull mysql

   docker run --name mysql -p 11013:3306 -e MYSQL_ROOT_PASSWORD=vMVUwaFri5KRqlUU -d mysql

3. Install Milvus from docker:

wget https://github.com/milvus-io/milvus/releases/download/v2.1.4/milvus-standalone-docker-compose.yml -O docker-compose.yml
sudo docker-compose up -d

4. Download dataset:
   git lfs install
   git clone https://huggingface.co/datasets/Shengtao/recipe

5. Download 2 files 'simcse_recipe_ingredients' and 'simcse_recipe_title' from hugging face automatically when you run the backend code for the first time:

```
Downloading:   6%|▌         | 28.5M/499M [00:05<01:06, 7.06MB/s]
```

6. Dependencies:
   - Load the requirements.txt in the backend folder to get most of the dependencies.
   - Use pip to install the following 2 dependencies:
     $ pip install ftfy regex tqdm
     $ pip install git+https://github.com/openai/CLIP.git

7. Activate the program:
   python upload2mysql.py
   python upload2milvus.py
   python server.py

After 1~2 minute, you will get this message:

```
FLASK_APP = server.py
FLASK_ENV = development
FLASK_DEBUG = 0
In folder C:/Users/Administrator/Desktop/backend
D:\Anaconda3\python.exe -m flask run
 * Serving Flask app 'server.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

It means the backend has run successfully.

5. Modify the config.js file in the frontend folder and replace the IP and port with the IP and port painted up in the red box in the previous step 4. The front end and back end are now successfully connected. You can enjoy exploring our 'Taste studio' now!

Reference

Gao, Tianyu & Yao, Xingcheng & Chen, Danqi. (2021). SimCSE: Simple Contrastive Learning of Sentence Embeddings. 6894-6910. 10.18653/v1/2021.emnlp-main.552.

Overview, 2022. *Material UI*. [online] Material UI. Available at: https://mui.com/zh/material-ui/getting-started/overview/ [Accessed 17 November2022].

Introduction Milvus vector database documentation. Available at: https://milvus.io/docs/v2.1.x/overview.md[Accessed 17 November2022].

Appendix

User stories

User type

- Explorer (EX) - Users who want to find the recipe and are recommended some potential interesting recipes.
- Contributor (CO) - Users who want to contribute and share their recipe to others.
- Admin (AD) - Users have elevated privileges and are able to modify the database. (*this part is conducted by the backend SQL implementation so we won't display in the report*)
- Registered User (RU) - Users who own persistent data and can be authenticated, and users are compulsory to register.

Code	User Stories	Acceptance Criteria
EX1	As a recipe explorer, I want to search the recipe through the name and tag so that I can find the recipe in a relevant topic quickly and easily.	● Users can type the name or tag into the search bar. ○ At most enter 50 characters. ● Keyword can match the tag: ○ Ingredient ○ Cooking Method ○ Meal type ○ Description
EX2	As a recipe explorer, I want to see all the relevant details of the recipe so that I can decide if it is worth adopting.	● Users can view the recipe's detail: ○ Title ○ Description/Tips ○ Ingredient ○ Cooking Method ○ Meal type

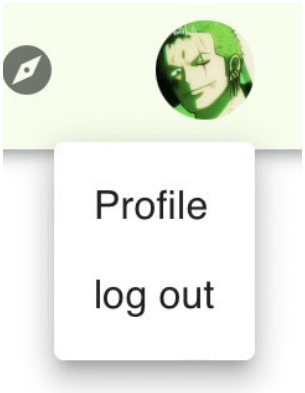
		○ Detailed cooking step ○ review/comment ○ The number of liking ○ Cooking time
EX3	As a recipe explorer, I want to filter the recipe based on its category so that I can find the recipe matching my interest quickly.	● User can choose these tag to filter: ○ Cooking Method ○ Ingredient ○ Meal type
EX4	As a recipe explorer, I want to be recommended recipes relevant to those I have navigated so that I can cook new dishes similar to my hobby.	● Users will be recommended by their visited recipes through these features: ○ Cooking Method ○ Ingredient ○ Meal type
EX5	As a recipe explorer, I want to bring my opinion to the recipe so that good recipes can be found by other users and bad recipes can be noticed.	● Users can like the recipe. ● Users can withdraw their liking to the recipe. ● Users can comment on the recipe freely and other users can see it. ● Users can delete the comment they posted.
EX6	As a recipe explorer, I want to view the details of the recipe through its thumbnail so that I can find some recipe that looks attractive.	● Users can see the thumbnail of recommended recipes on the mainpage. ● Users can click the thumbnail to view the details of them.
EX7	As a recipe explorer, I want to subscribe to some other users who have posted attractive recipes so that I can follow their new recipe.	● Users can subscribe and unsubscribe the contributor. ● Users can see the new post from their subscribed contributor on the subscription page.
EX8	As a recipe explorer, I want to see the recent or best post on the subscription so that I can see and choose the hot dishes and the new dishes easily.	● Users can choose the different sorted ways on the page of subscription: ○ newest ○ most liked
EX9 (novel)	As a recipe explorer, I want to see the calories and whether I can accept the calories in this recipe so that I can have a healthy diet.	● Users can see the approximate calories on the recipe page: ○ Each ingredient ○ Total dish ● Users can add the intaken recipe into today's calories (cont.) ● Users can see the warning if they will

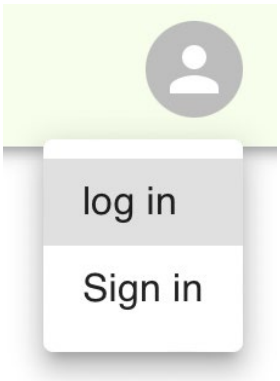
		exceed the threshold of calories when they adopt this recipe.
EX10 (novel)	As a recipe explorer, I want to collect and save the recipe into a place so that I can find some usable recipes easily.	● Users can save their favorite recipes into their profile ● Users can remove unwanted recipes in their favorite list
EX11 (novel)	As a recipe explorer, I want to see a wanted recipe that I am not allergic to and disliked so that it reduces the workload to exclude the unwanted recipe.	● Users can choose allergic and disliked food on the sign up page. ○ At most 100 allergic foods ○ At most 100 dislike foods ● Users can adjust allergic and disliked food on their settings page. ● Users won't be recommended the recipes labeled with these unwanted tags.
EX12 (novel)	As a recipe explorer, I want to know the recipe of dishes in restaurants so that I can know its ingredients and cooking step and store this recipe to cook at home.	● Users can upload the photo of dishes to the system and search the recipe. ● The size of photo cannot exceed 3MB ● Users cannot upload other things: ○ Ingredient ○ Not cookable things ● Users cannot search for some recipe not in the repository.
CO1	As a recipe contributor, I want to upload the recipe to the system so that others can see and learn the recipe to cook more dishes they might not know.	● Users can contribute their recipe to the system including the details: ○ Title ■ At most 50 characters ○ Photo of recipe cover ■ At most 3MB ○ Description and tips of recipe ■ At most 800 characters ○ Tag ■ At most 20 tags ○ Ingredients ■ At most 20 ingredients ○ Recipe steps ■ At most 20 steps ■ Each photo at most 3MB ■ Each description of step at most 800 characters ○ Calories (elective) ■ 4 digits only ● Users cannot upload more than 500 recipes.

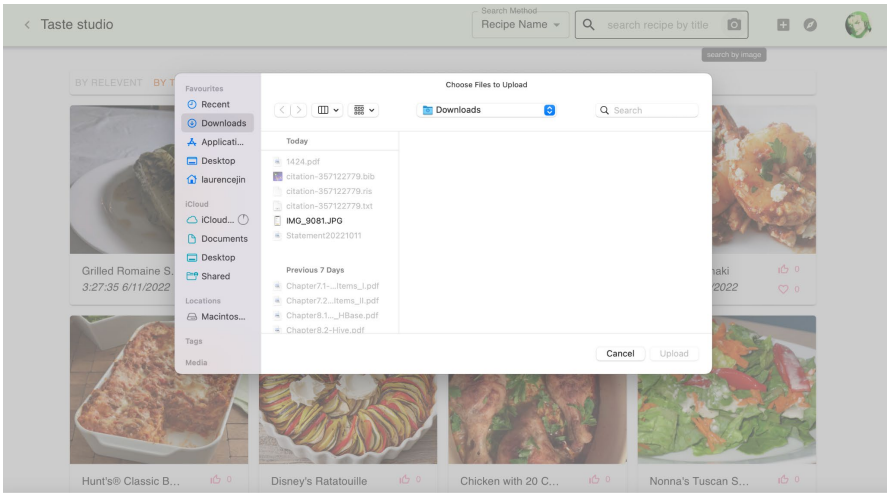
CO2	As a recipe contributor, I want to modify the recipe to the system so that I can modify some accident mistakes easily.	● Users can conduct the recipe through: ○ Edit each visible detail ○ Delete ● Users cannot modify the like and comment on their contributed recipe.
CO3	As a recipe contributor, I want to view the published recipes in the same area so that I can conduct and check them easily.	● Users can view their published recipes on the profile.
RU1	As a registered user, I want to be able to have a customized account so that I can get recommended and contribute recipes.	● Users can create an account through the valid: ○ Email ○ Password ● Users are asked to retry if: ○ The email has already taken ○ The email is invalid ○ Password is less than 6 characters ● Users can log in automatically once the account has already been created.
RU2	As a registered user, I want to access my account from various devices so that I can sync my customization and personal information easily.	● Users can log in using: ○ Username ○ Email address has been registered ● Users can log in to various devices simultaneously. ● If a user fails to enter the valid password after 6 attempts, they will be prompted to reset the password.
RU3	As a registered user, I want to log out my account in the system so that my account won't be modified and stolen maliciously.	● Users can log out from the current platform ● The other platform's account remains constant if the current platform's one has logged out.
RU4	As a registered user, I want to be able to reset the password so that I won't lose this account if I forget the password or be blocked by the system.	● Users can reset the password by entering the valid email address. ○ If it is valid, the system will send verification code to this email address. ● Users cannot be able to know if the email they entered is valid.
RU5	As a registered user, I want to remain logged in the same device so that it reduces the meaningless workload to login every time.	● Once users have logged in, they will remain this unless they sign out manually.

RU6	As a registered user, I want to modify the profile so that I can change my information and preference freely.	• Users can view their profile and click the setting page to modify their own information.

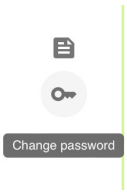
More Software human-interaction in 2. Functionality developer



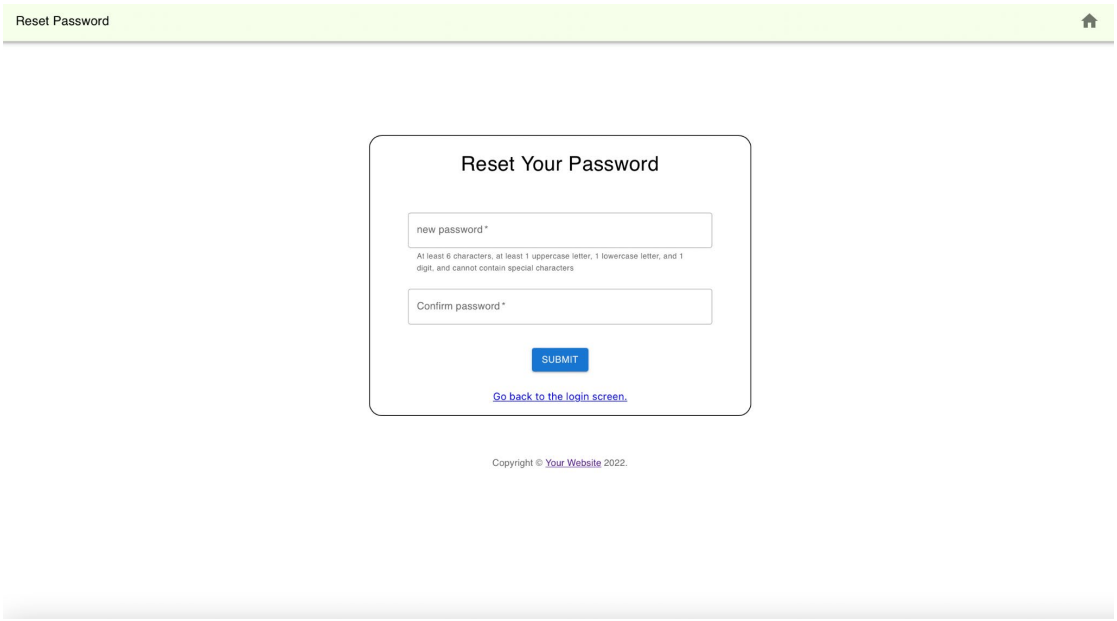




Upload the image to search



hint when the mouse stays on the icon.



Reset the password

```
from sqlalchemy import Column, String, Integer, BLOB,Text,DateTime
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.sql.sqltypes import BigInteger

from conf.dao_config import comment_table_name
from dao.mysql_server import MysqlServer
from sqlalchemy.sql import func


Base = declarative_base()

class Comment(Base):
    """data of comment
    """
    __tablename__ = comment_table_name |
    commentid = Column(Integer(), primary_key=True, autoincrement=True)
    content = Column(String(500))
    recipeid = Column(String(50))
    curtime = Column(DateTime(timezone=True), server_default=func.now())
    userid = Column(Integer(), primary_key=True)
    def __init__(self):
        # Mapping relationship with database
        engine = MysqlServer().get_comment_engine()
        Base.metadata.create_all(engine)
    def new(self,userid,content,recipeid):
        self.userid = userid
        self.content = content
        self.recipeid =  recipeid
```

Figure 3.2.1.1