

COMP 3900 Project Proposal

Team YYDS-2021

Weiqiang Zhuang z5183982 (SCRUM MASTER)

Xiao Hu z5223731 (DEVELOPER)

KE WU z5158146 (DEVELOPER)

Boyu Cai z5183721 (DEVELOPER)

Kylee Bowers z5304138 (DEVELOPER)

Table of Content:

Team YYDS-2021	0
Table of Content:	1
1. Overview	3
1.1 Background	3
1.2.1 Presentation Layer	6
1.2.2 Application Layer	6
1.2.3 Data Layer	6
2. Data Model	6
2.1 ER Diagram	7
2.2 Database	7
3. Functionalities	7
3.1 User Authentication	7
3.1.1 Login	8
3.1.2 Register	8
3.1.2 Email Verification	9
3.2 Homepage	10
3.2.1 Recipe Drop-Down Menu	10
3.2.2 My Profile Drop-Down Menu	11
3.2.3 Access Permission	12
3.3 Search	13
3.4 News Feed	13
3.4.1 Recipe Page	14
3.4.2 Share Opinion	15
3.4.3 Upload Recipe	17

4. Third-Party Functionalities	21
4.1 Front-End	21
4.2 Back-End	22
4.2.1 Springboot Framework	22
4.2.2 MyBatis	22
4.2.3 Log4j2	22
4.2.4 lettuce	22
4.2.5 Java-Jwt	22
4.2.6 Aliyun-OSS	22
4.2.7 Aliyun-ECS3	22
4.2.8 Aliyun-Mysql	22
5. Implementation Challenges	22
5.1 Front-End	22
5.2 Back-End	22
5.2.1 JWT	22
5.2.2 Custom HTTP status code	22
5.2.3 Shiro conflict	22
5.2.4 Nginx to resolve cors	22
6. User Document/Manual	22
6.1 Front-End Setup	22
6.2 Back-End Setup	22
6.3 Deployment Version	22

1. Overview

1.1 Background

Nowadays, more and more people are using the internet to help them decide what to cook. Blogs, aggregators of chefs' recipes and online compendiums of family cooking are snatching at the apron-tails of magazines and cookbooks. Online recipes range from home cooks who enjoy sharing their favorite creations, to the famous top chef who uses the virtual space to share their ultimate recipes and top tips. However, they all share one common thing: they were born out of a passion for food. Without a doubt, the world is a better place now they are in it.

Even though there exist many great recipe websites on the internet, there are two main issues we observed involving the existing recipes websites that we aim to resolve. The first issue is that there is no means of interaction between the content creators and the recipe seekers on most recipe viewing platforms. The second issue is that there needs to be a supervised management system to prevent inappropriate activities. Our solution to resolve the first issue is to develop an interactive recipe website. Recipe seekers can convey their opinion for any recipe in the comment section. Content creators can then modify their recipes based on the suggestions they have gathered from their viewers, or get the inspiration to create a new recipe. The solution for the second issue is to construct a supervised management system (SMS). Content quality is one of the most important aspects of our website. Therefore, inappropriate comments or irrelative advertisements will be removed.

For this project, the existing websites we are referencing are BBC Good Food and Allrecipes. Although these websites offer fairly comprehensive functionalities such as a search bar to search recipes, an explorer to view featured recipes, a comment section for users to convey their opinions, and et cetera, there are some drawbacks in these websites. The two main drawbacks are inappropriate comments and irrelevant advertisements.

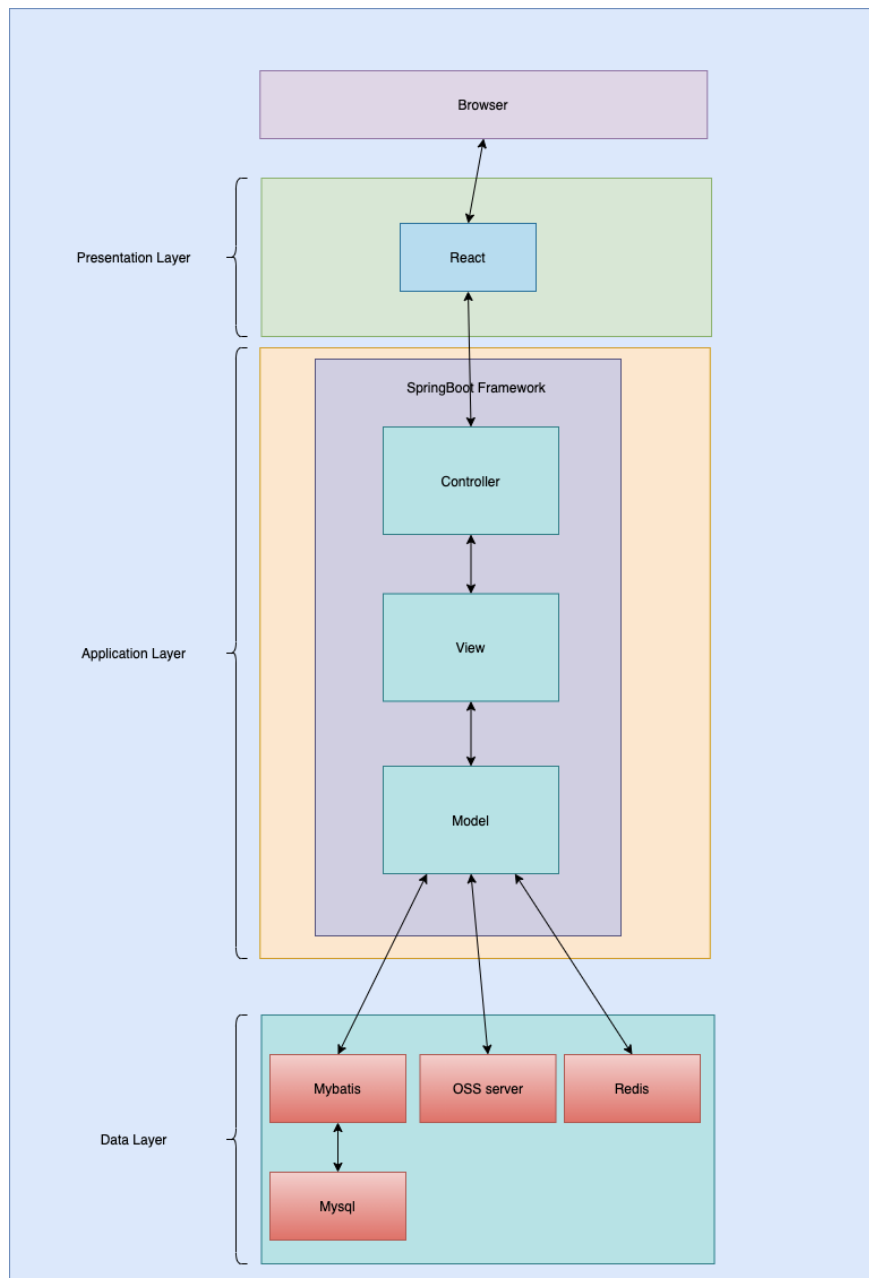
Even though users can share their opinions on their sites, recipe authors are not able to respond to their comments which can be inconvenient for users to ask questions. Moreover, without supervision, unfriendly comments can be posted on the website. Another problem is that there exist irrelevant or inappropriate advertisements which most users find distracting and irritating.

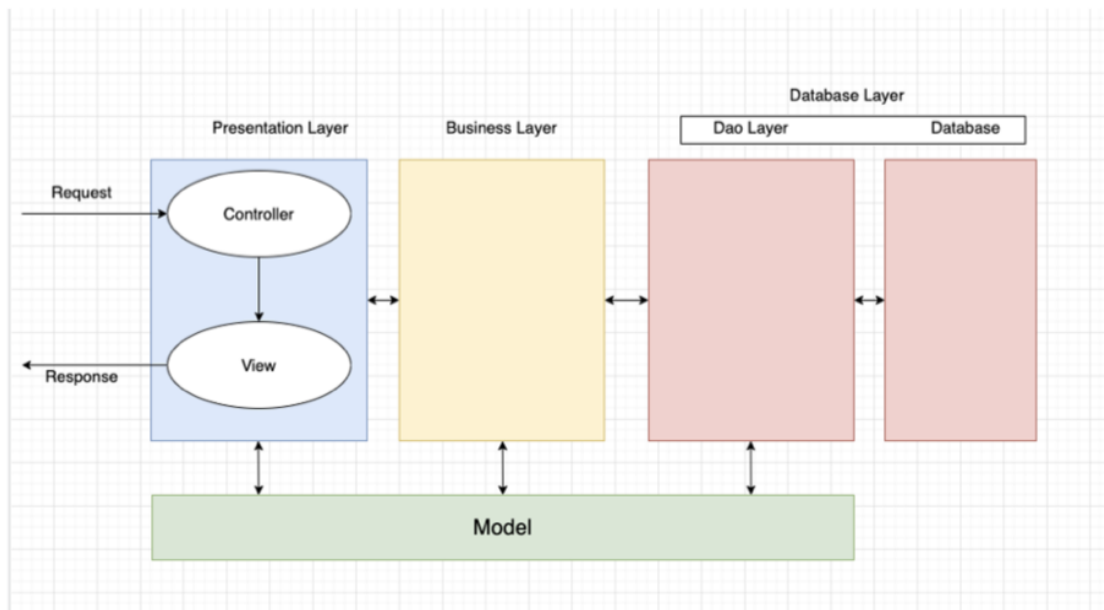
The main drawbacks above are common issues across most existing food recipe websites. Our team will build a website in order to resolve these problems. Our website will not only provide the most features of existing recipe websites, but also modify these functionalities in order to achieve better results.

The YYDS Recipe looks to cater for people who like to cook, are on the lookout for interesting recipes, and want to share recipes with others. YYDS Recipe can help explorers find the recipes they are interested in by providing them with recipe search and allowing them to share their opinions on recipes through comments and likes. Furthermore, YYDS Recipe provides opportunities for contributors range from stay-home mothers to famous top chefs to share their recipes. Additionally, YYDS Recipe allows the contributors to maintain a profile which includes a list of recipes that are published for explorers on the platform to scroll through for inspirations and collect to use them later. Furthermore, explorers can subscribe to contributors and their news feed will be personalised based on the contributors that they subscribed to. The order of the news feed is based on the date of these recipes are created or updated and the likes that the explorer has for the contributor of each of the recipes. The news feed will show the explorer a list of the summary for the recipes, and the explorers can view the full details of the recipes if they wish.

This report focuses on demonstrating the architecture of the system, the design of functionalities, implementation challenges, as well as the user documentation manual.

1.2 Software Architecture





1.2.1 Presentation Layer

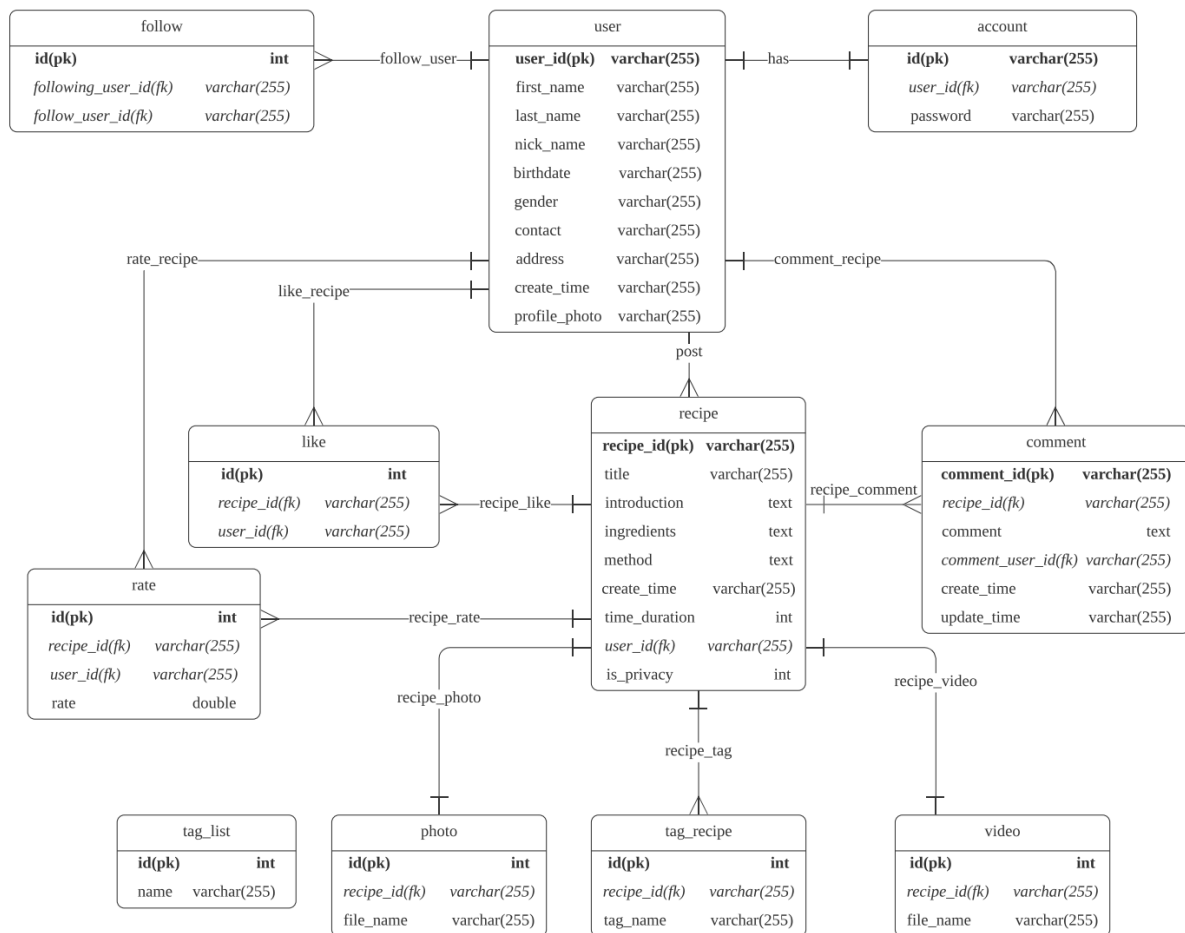
1.2.2 Application Layer

For application layer, the recipe website uses SpringBoot Framework and it is a classic and well-known three-layer MVC framework.

- **Model Layer**
Responsible for data persistence, control data model and connect to the database to exchange data.
- **View Layer**
For this project, frontend instead of view layer, we only need control model layer and present data to the frontend
- **Controller Layer**
Accept and forward HTTP request and return result to the View Layer.

1.2.3 Data Layer

ER Diagram



Database

Database user is use for saving the user data

Database account is the user who own an unique account

Database follow is use for saving the user who follow the other users

Database recipe is use for saving the user who post the recipe

Database comment is use for saving the user who comment the recipe

Database like is use for saving the user who like the recipe

Database rate is use for saving the rate which is users rate the recipe

Database tag_recipe is use for classification the recipe

Database video is use for saving the recipe video which is user uploaded

Database photo is use for saving the recipe photo which is user uploaded

3. Functionalities

3.1 User Authentication

For the Profile page and the functionalities that allows users to upload their recipes, users would need to log in in order to get the permission to view these pages and the functionalities.

For most request it should hold the token in the request header, therefore, each request the system will intercept and decode the token to make sure if is expired or the userId is not belong the system.

3.1.1 Login

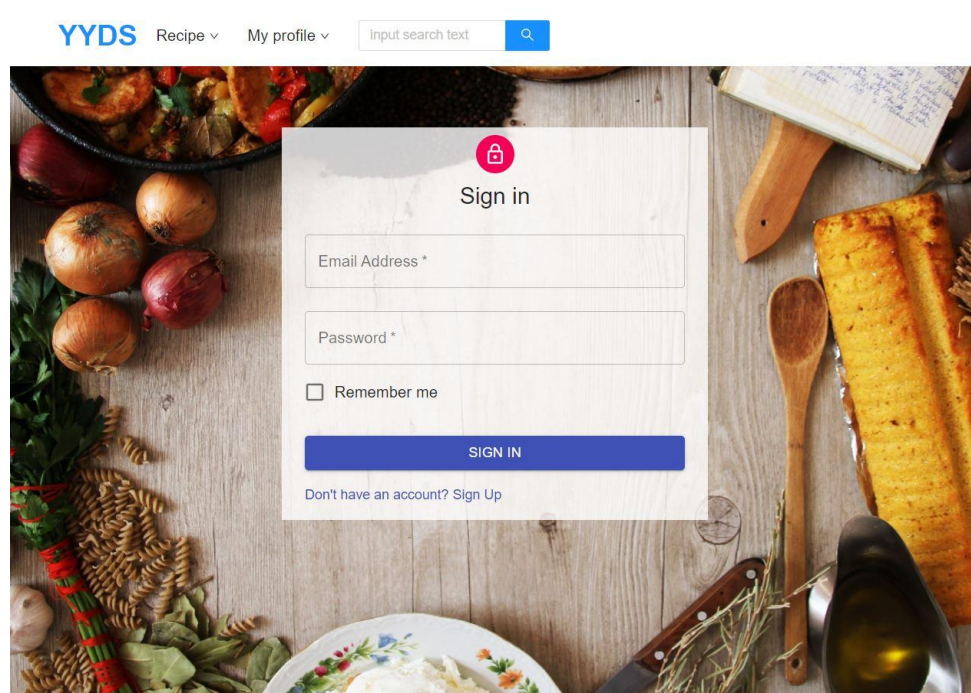


Figure 3.1.1 Sign-up Page

As the image shown above, users can sign in using the same email and password they used to sign up. Users can choose to tick the “Remember me” to save their login information so that they don’t have to enter the same information again. If the login information that the user entered is invalid, a pop up error message will show up to let the user know (*Figure 3.1.1 Login Error*).

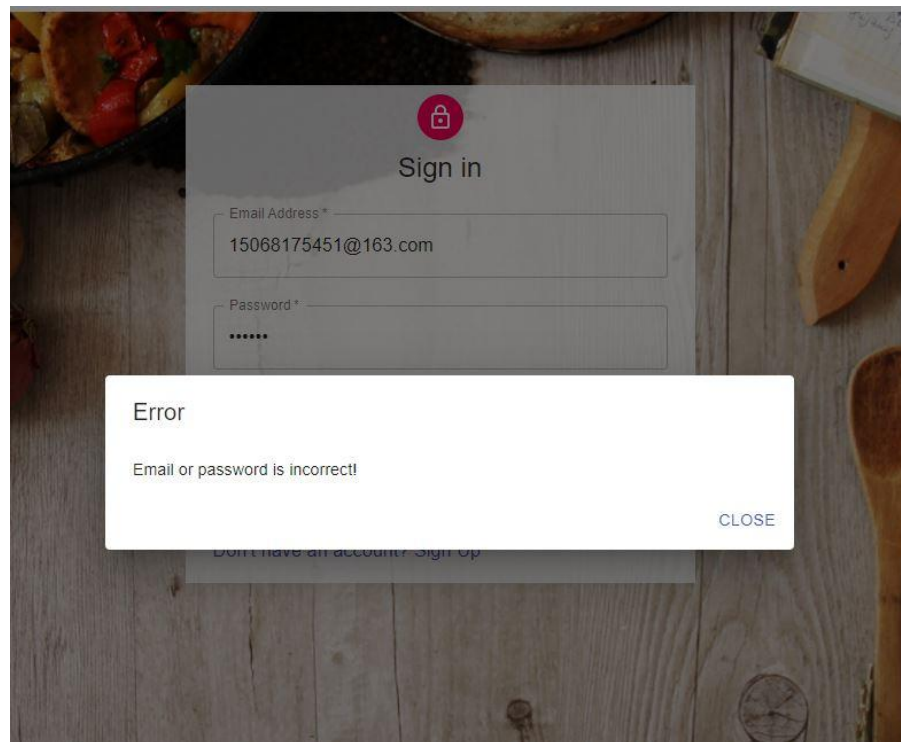


Figure 3.1.1. Login Error

3.1.2 Register

If the user is not a member of our website, they can sign up by filling in their personal information into our registration form for us to save their data into our database.

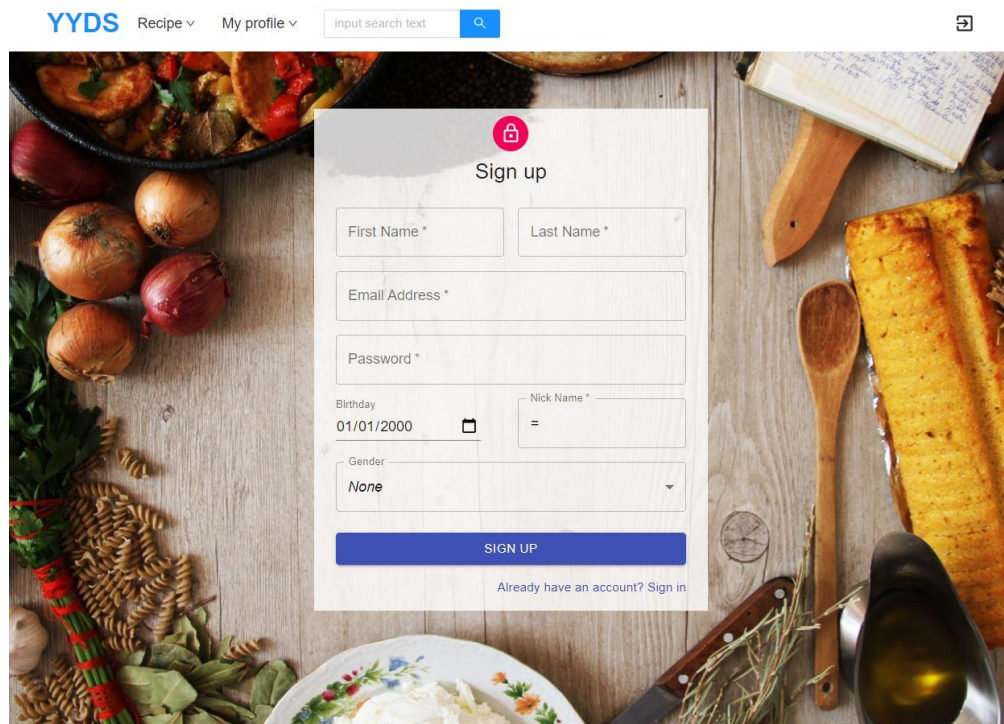


Figure 3.1.2 Register Page

The register page is shown when the user clicks “Don’t have an account? Sign up” link under the “SIGN IN” button in Figure 3.1.1 Register Page. When the users are signing up, they will be required to provide their first and last name, email address, and birthday. If the entry for each item is invalid, our website will pop up an error message like the image shown below (Figure 3.1.2. Email Error Message).

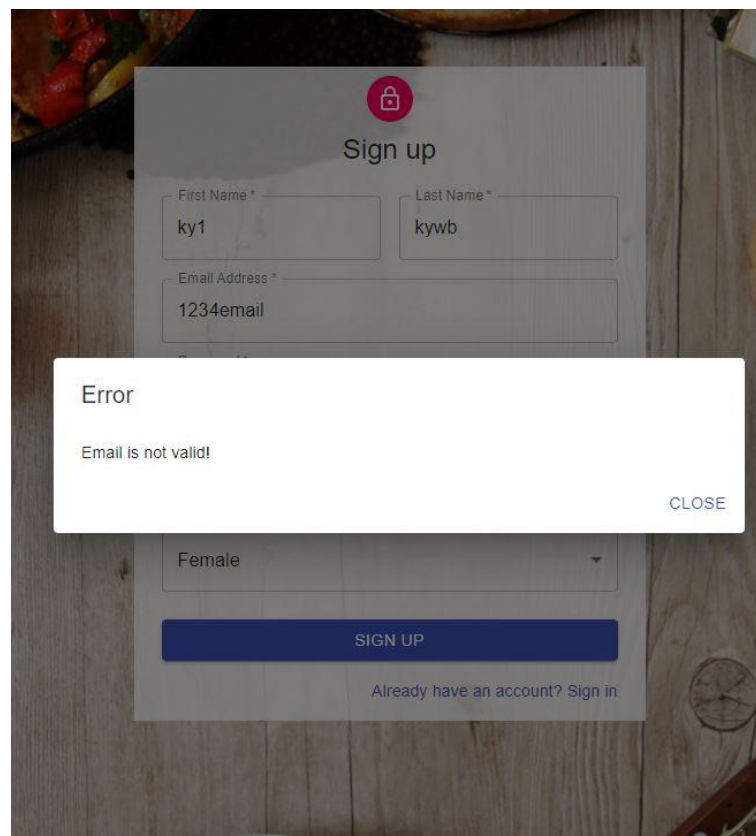


Figure 3.1.2. Email Error Message

3.1.3 Email Verification

In order to register an account successfully, users will also need to verify their emails. After the user fills out their personal information and clicks the “SIGN UP” button, a new page will show up and inform the user that YYDS Recipe has sent an email to the user to verify their email (Figure 3.1.3 Verify Email Message). In addition to that, the user has to verify their email within a certain limited time otherwise the session will expire and the user will have to register again. In the email that YYDS Recipe sent to the user (Figure 3.1.3 Verify Email Sent), a link is provided to the user to verify their email. Lastly, when the user clicks the link, a new web page will tell the user that he/she has successfully registered an account.

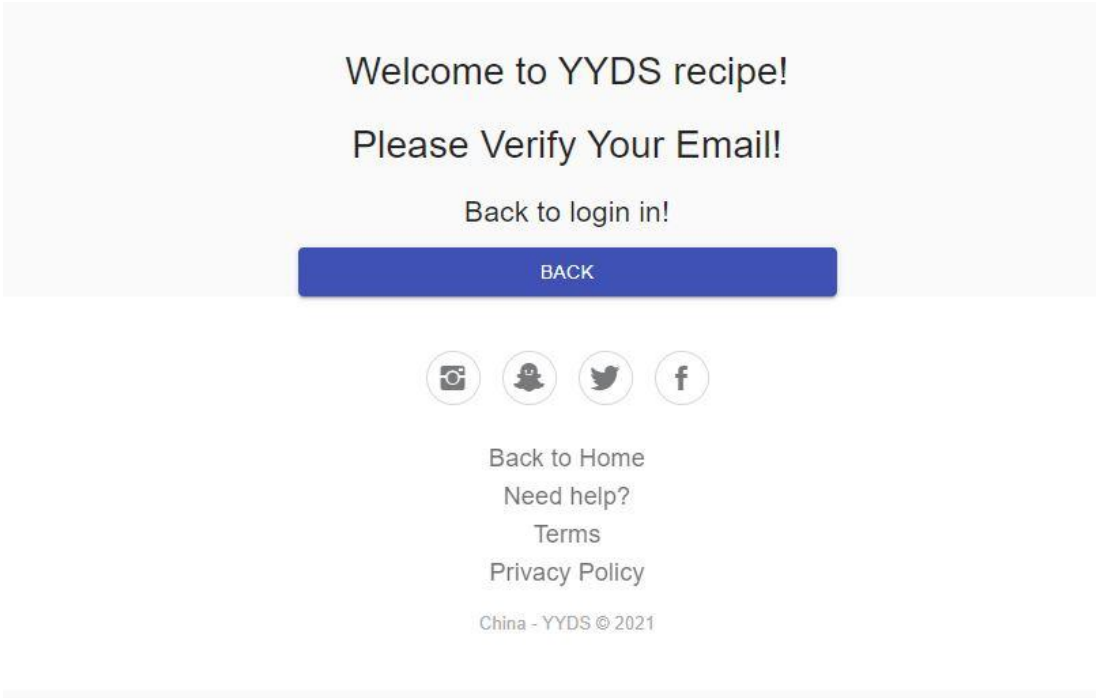


Figure 3.1.3 Verify Email Message

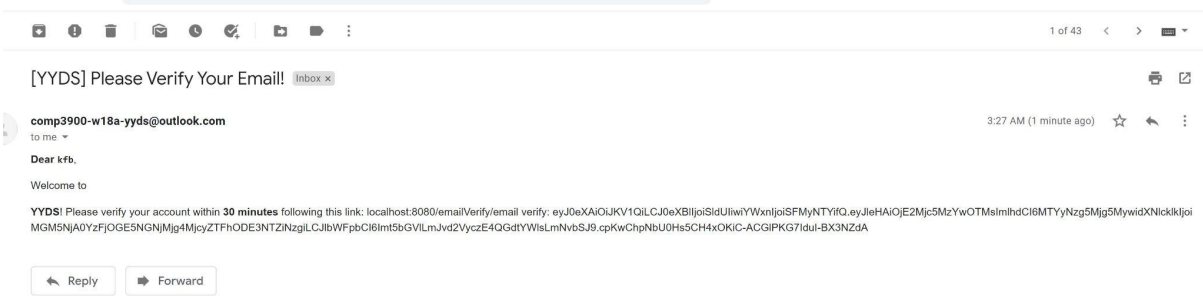


Figure 3.1.3 Verify Email Sent

[image of the webpage that the link directed the user to]

3.2 Homepage

When users enter our website, a home page is presented to users where they can browse through and search for recipes.

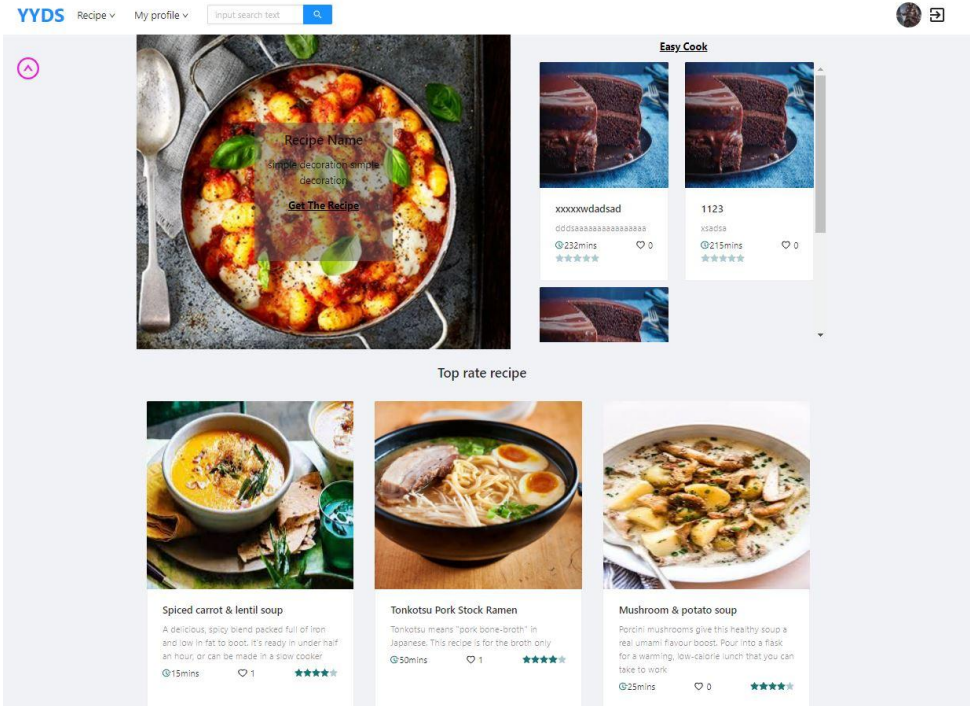


Figure 3.2 Home Page

As shown in the image above, our home page has a “Recipe” drop-down menu, “My Profile” drop-down menu, a search bar, a news feed, and an upload-recipe button.

3.2.1 Recipe Drop-Down Menu

When the user hovers over “Recipe” in the top bar, a drop down menu is displayed to the user. As shown in Figure 3.2.1, the drop-down menu shows the user a list of categories that can be chosen from.

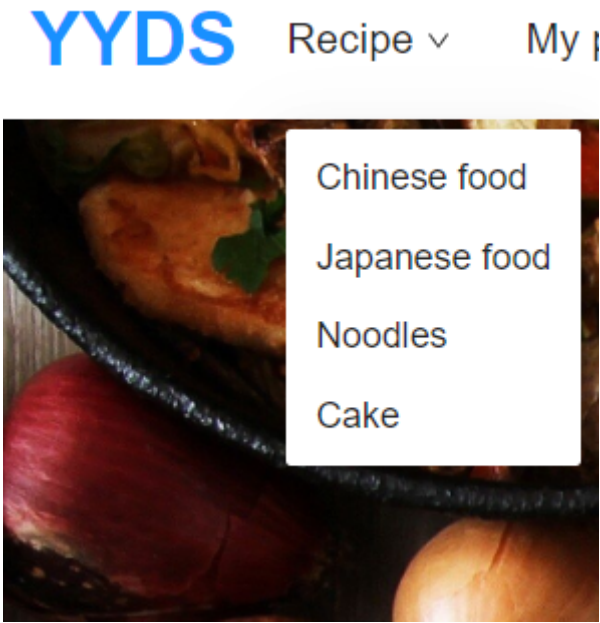


Figure 3.2.1 Recipe Drop-Down Menu

When the user clicks on one of the categories, users will be directed to a new page that shows a collection of recipes in the news feed section. The image below shows an example of what users will see when they choose Japanese food from the list of categories.

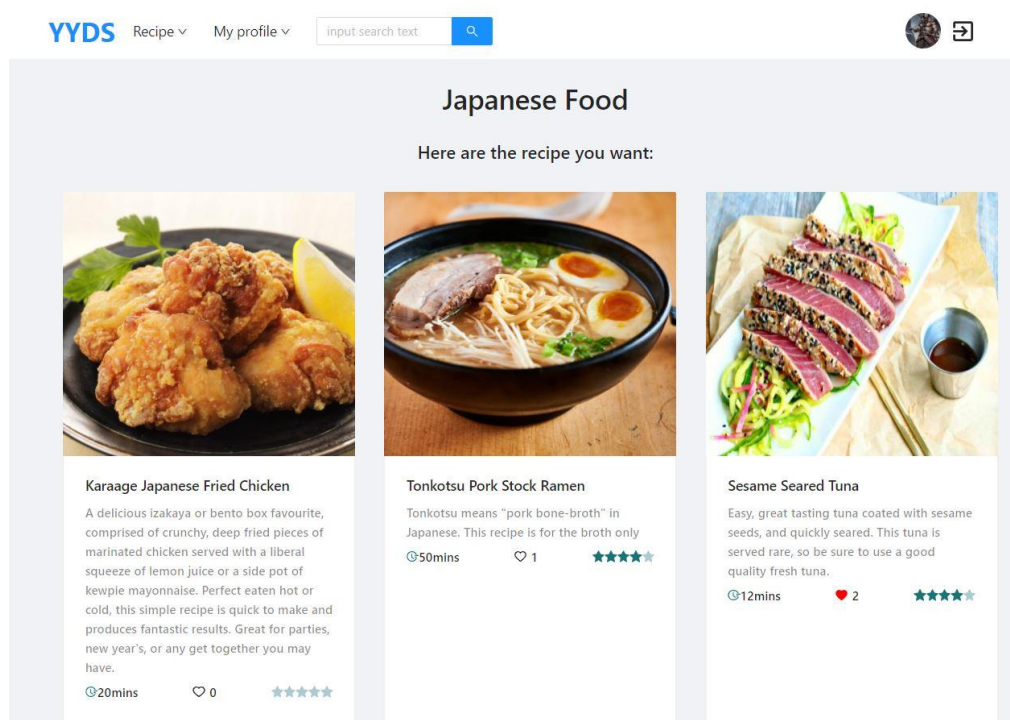


Figure 3.2.1 Japanese Food Collection

3.2.2 My Profile Drop-Down Menu

When the user hovers over the "My Profile" in the top bar, a drop down menu will be displayed to the user to manage their account related information. The image shown below is a list of options that the user will see.

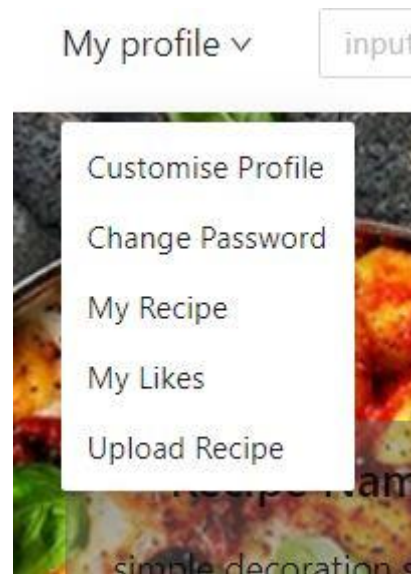


Figure 3.2.2 Profile Drop-Down Menu

Customize Profile: the user will be able to change their profile picture, first and last names, nick name, birthday, and gender.

Change Password: The user will be able to change their password

My Recipe: the user will see a list of recipes that he/she has created

My Likes: the user will be able to see a list of recipes that he/she liked previously

Upload Recipe: the user can upload a recipe to our website.

3.2.3 Access Permission

Although all users are allowed to browse through our website, there are some of the features that are only restricted to members only. Non-member users will not be able to access any pages under the “My Profile” drop-down menu. Furthermore, non-member users will not be able to see the details of any recipes when they click on the recipes. An error message will be displayed when a non-member user tries to do any of the restricted actions. The image below is an image of the homepage with the error messages (*Figure 3.2.3 Access Error*).

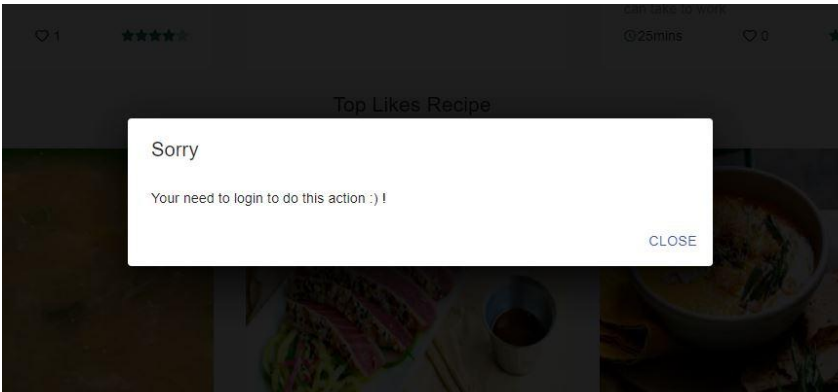


Figure 3.2.3. Access Error

3.3 Search

Users can search for a recipe by typing keywords in the search bar, then the website will load a list of recipes that are relevant to the inputted keywords. The image below shows an example of the result from the search bar when we typed in “Soup”.

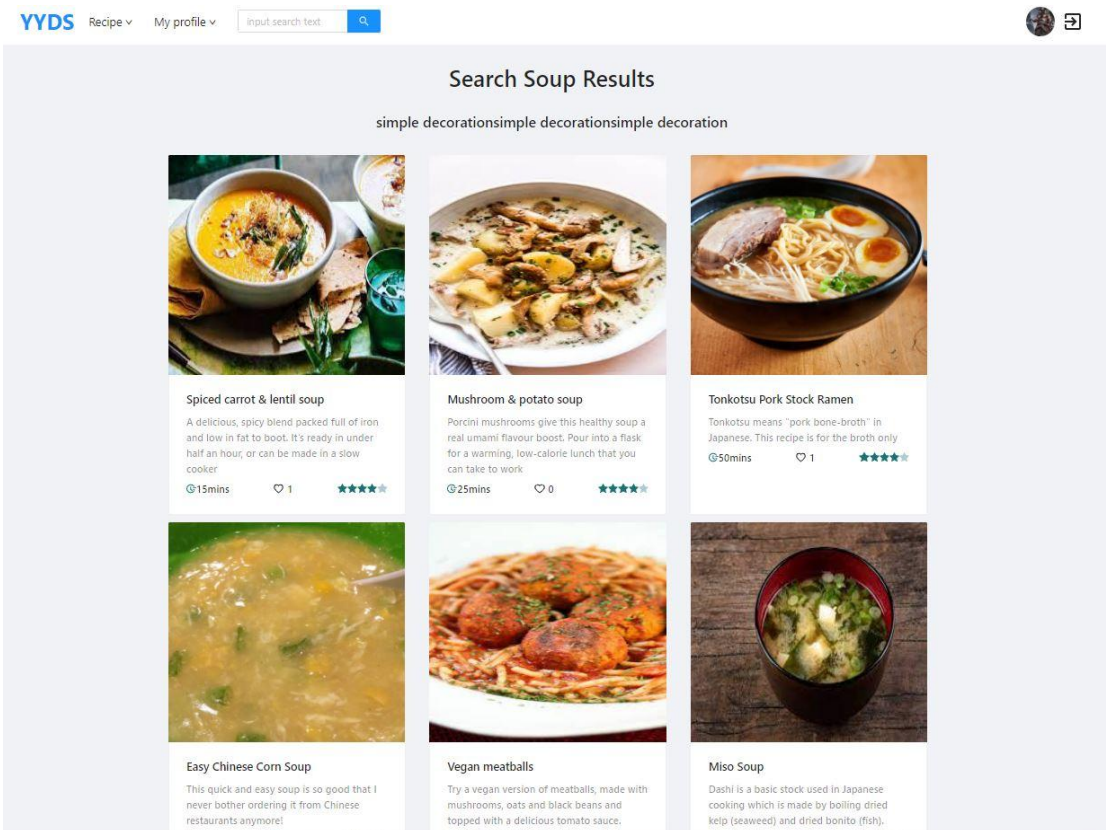


Figure 3.2.3 Result from Search Bar

3.4 News Feed

The news feed is separated into four different parts: featured recipe, “Easy Cook” recipes, “Top Rated Recipes”, and “Top Likes Recipe”.

Featured Recipe: a recipe is being featured by YYDS Recipe

“Easy Cook” recipes: recipes displayed in the “Easy Cook” section are the recipes that were tagged with “Easy Cook” when the contributor created them

“Top Rated Recipe”: this shows a list of top rated recipes that other users rated with the five-star system.

“Top Likes Recipe”: this shows a list of top recipes with the most likes

The image below shows the “Top Rated Recipe” section and the “Top Likes Recipe” section

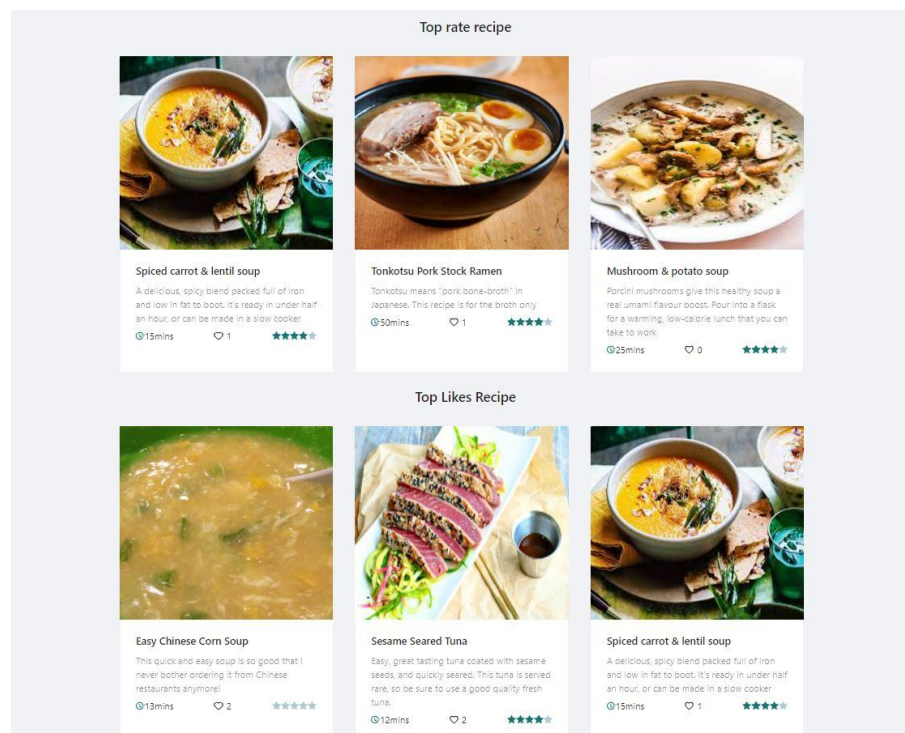



Figure 3.4 Top Rated and Top Liked Recipes

3.4.1 Recipe Page

Users can view the details of the recipes by clicking on the recipe card. The image below is an image of what the webpage looks like when the user clicks on a recipe card.



Spiced carrot & lentil soup

by [KeKe](#)

Ratings: 4 Rate: ★★★★★ Your score

Time use: 15

Ingredients:

2 tsp cumin seeds pinch chilli flakes 2 tbsp olive oil 600g carrots, washed and coarsely grated (no need to peel) 140g split red lentils 1l hot vegetable stock (from a cube is fine) 125ml milk (to make it dairy-free, see 'try' below) plain yogurt and naan bread, to serve

Method:


STEP 1 Heat a large saucepan and dry-fry 2 tsp cumin seeds and a pinch of chilli flakes for 1 min, or until they start to jump around the pan and release their aromas. STEP 2 Scoop out about half with a spoon and set aside. Add 2 tbsp olive oil, 600g coarsely grated carrots, 140g split red lentils, 1l hot vegetable stock and 125ml milk to the pan and bring to the boil. STEP 3 Simmer for 15 mins until the lentils have swollen and softened. STEP 4 Whizz the soup with a stick blender or in a food processor until smooth (or leave it chunky if you prefer). STEP 5 Season to taste and finish with a dollop of plain yogurt and a sprinkling of the reserved toasted spices. Serve with warmed naan breads.


Recipe Tags:

Desserts Breakfast Soup Snacks Other

A delicious, spicy blend packed full of iron and low in fat to boot. It's ready in under half an hour, or can be made in a slow cooker

1 replies

 KeKe in 9 hours
I like it!



Add Comment

Figure 3.4.1 Details of Recipe

The webpage for the recipe includes an image of the food, a summary from the contributor, a title, the name of the recipe contributor, a rating for the recipe using the five-star system, the cooking time, the ingredient list, cooking instructions, recipe tags, and a comment section.

3.4.2 Share Opinion

Users can convey their opinion on a recipe by comments and likes. As shown in Figure 3.4.1 Details of Recipe, there is a comment section at the bottom. Images shown below show an example of when a user shares a comment for a recipe. User's comment will be added to the comment section when the user clicks on the "Add Comment" button.

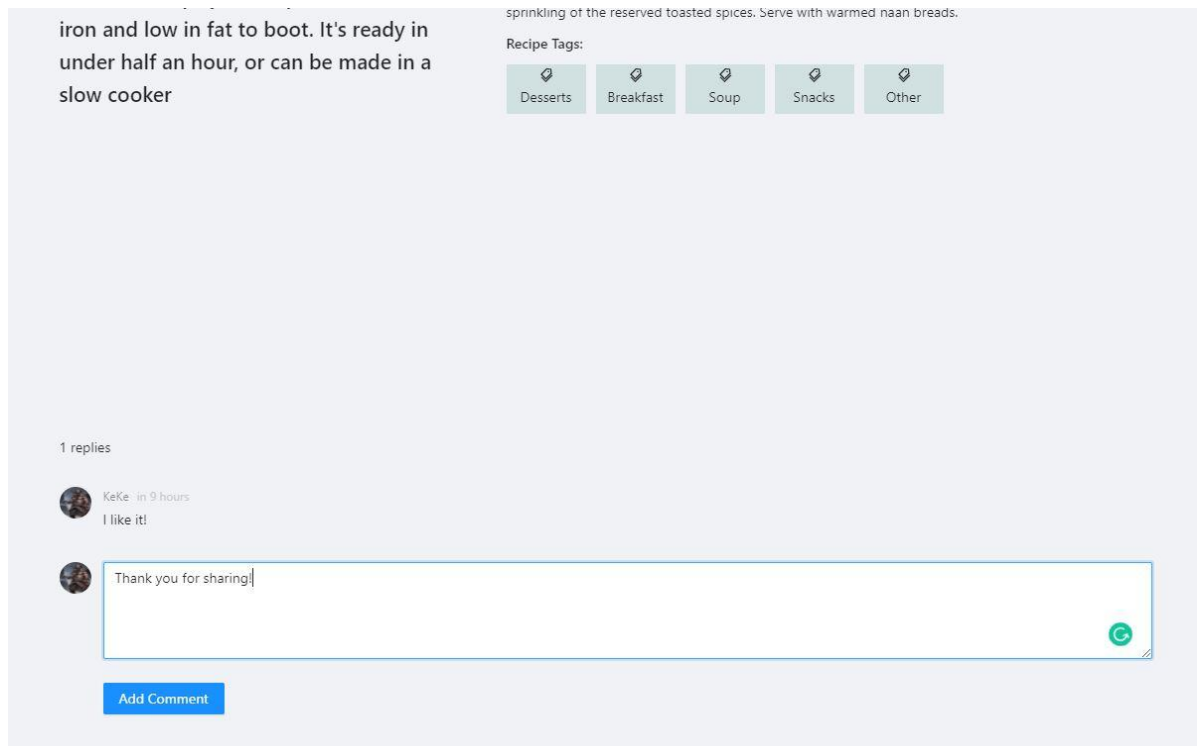


Figure 3.4.1 Input in the Comment Section

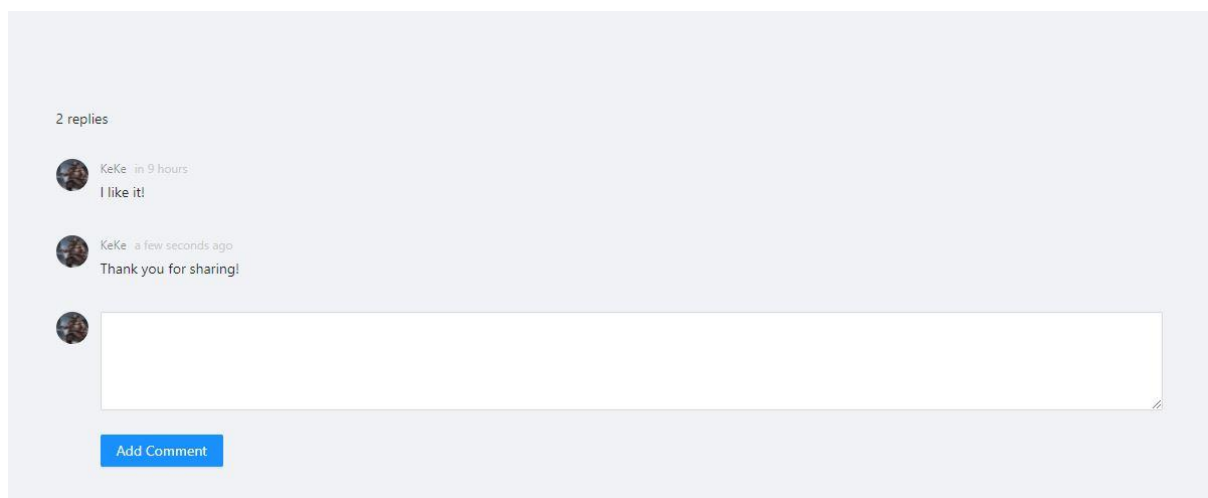


Figure 3.4.1 Comment Appears in Comment Section

Users can also share their opinion by liking recipes. Users can like a recipe by clicking the heart button on the recipe, then the heart will turn red. Users can also unlike the recipe by clicking on the heart again, and the heart will turn into an empty heart (Figure 3.4.1 Recipe Like).

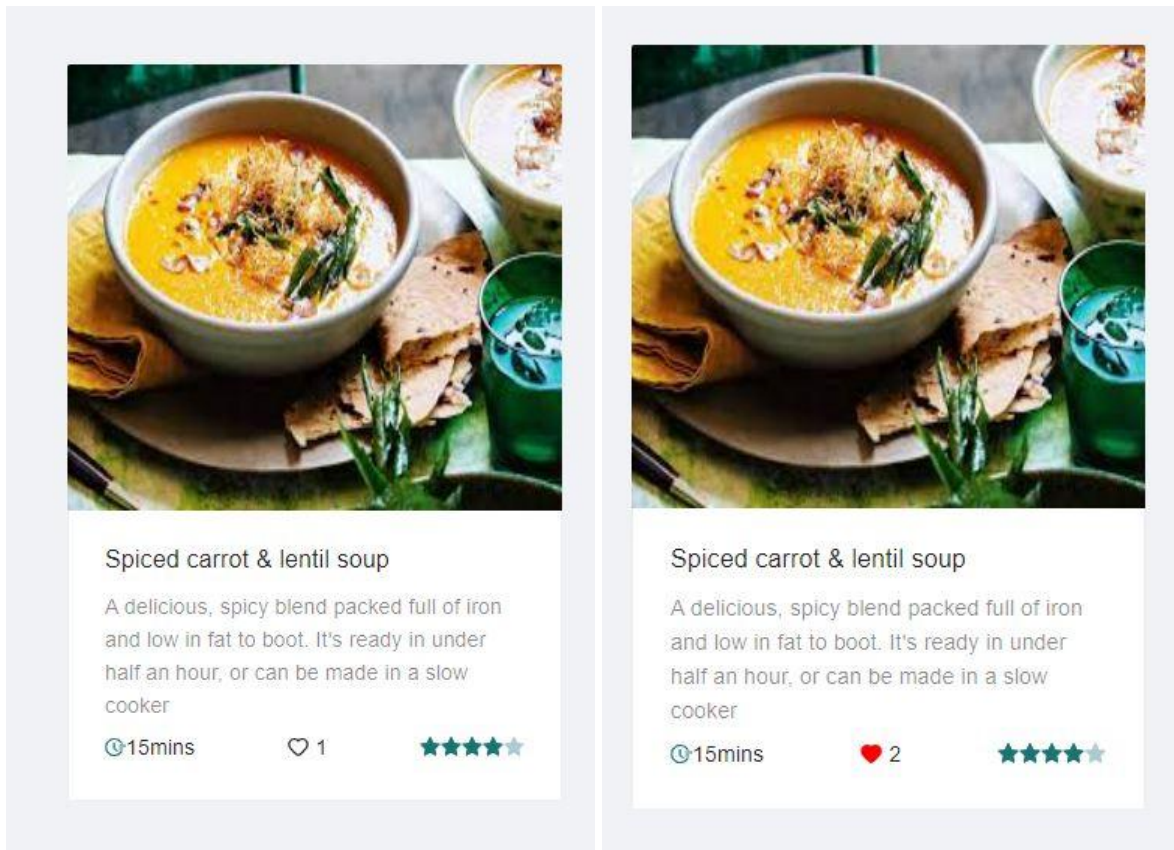


Figure 3.4.1 Recipe Like

3.4.3 Upload Recipe

Members of YYDS Recipe can share their recipes by clicking “Upload Recipe” under the “My profile” drop-down menu. Then a window will pop up for the user to fill in the information about the recipe. The input fields that the user will need to fill are an image of the food, a short title, tags for the recipe, cooking time, an introduction, ingredients to make the food, and the instruction to how to make the food (*Figure 3.4.3 Upload Recipe Form*).

The image shows a web application interface for uploading a recipe. A modal window titled "Upload Recipe" is centered on the screen, overlaying a background of various food images. The modal contains the following fields and controls:

- * upload Photo/video:** A button labeled "Choose Files" and the text "No file chosen".
- * recipe title:** A text input field.
- * Select a tag:** A dropdown menu currently showing "Tags Mode".
- * Time use(mins):** A text input field.
- * Introduction:** A larger text area with a small icon for expanding the text.
- * Ingredients:** A text area with a small icon for expanding the text.
- * Method:** A text area with a small icon for expanding the text.
- submit:** A blue button at the bottom of the modal.

The background website shows recipe cards with images and text, including "ed carrot & le", "ious, spicy blend pe", "ron and low in fat to", "dy in under half an", "made in a slow coo", "EDIT", "age Japanese", "XXXXXXXXXXXXXXXXXX", and "XXXX".

Figure 3.4.3. Upload Recipe Form

The following conditions must be met in order to upload a recipe successfully, otherwise, error messages will appear (*Figure 3.4.3. Error Messages*)

- The title of the recipe cannot be too long (limited to 50 characters)
- An image must be included
- At least one tag is added to recipe
- The time used must be a number
- All fields must not be empty

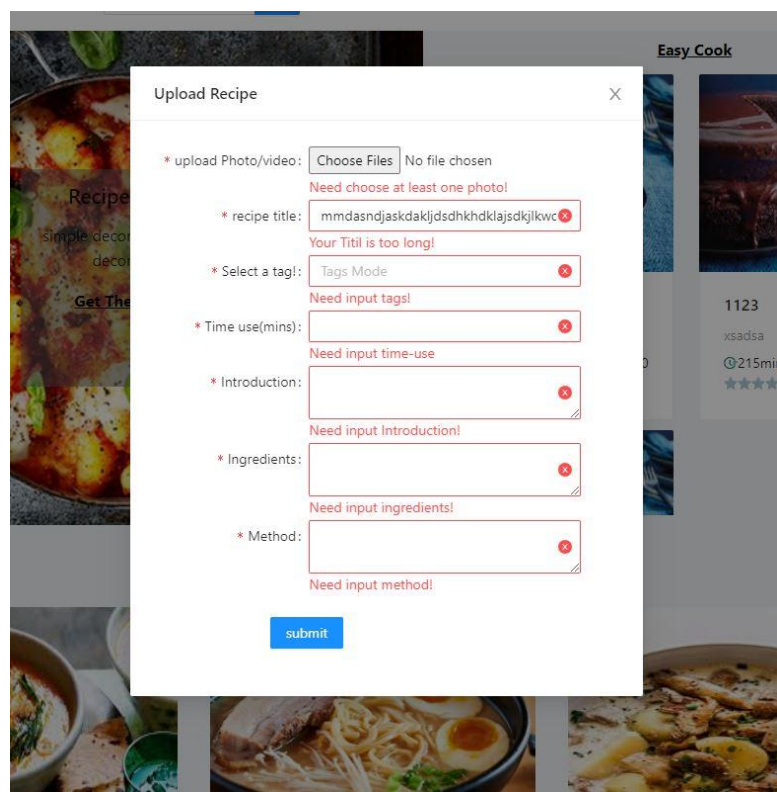


Figure 3.4.3 Error Messages

Users can upload an image for a recipe that is stored in their local machine (*Figure 3.4.3. Image from Local*). After the user filled out all the information for the recipe like in Figure 3.4.3. Recipe Information, users can click the “Submit” button to upload the recipe and a message will pop up when the recipe is uploaded successfully (*Figure 3.4.3. Upload Recipe Successfully*). The recipe then will appear on our website (*Figure 3.4.3. New Recipe on Website*)

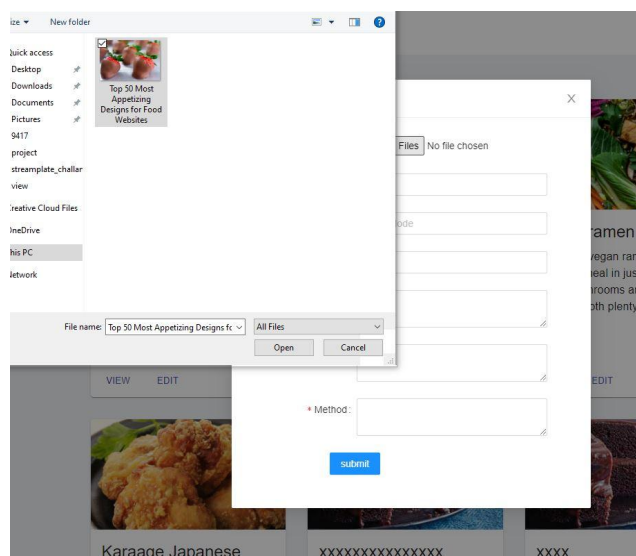


Figure 3.4.3. Image from Local

Upload Recipe

* upload Photo/video:

Choose Files

Top 50 Mos...ebsites.jpg

* recipe title:

Chocolate Covered Strawberries

* Select a tag!:

Appetizers x

Desserts x

Snacks x

* Time use(mins):

30

* Introduction:

Chocolate covered strawberries are one of everyone's favorite treats

* Ingredients:

Fresh strawberries

Coconut oil

* Method:

Clean strawberries: Scroll down to see how best to do this.

submit

Figure 3.4.3. Recipe Information

* Select a tag!:

Tags Mode

Congratulations

Your recipe has been Submit Successfully

CLOSE

Figure 3.4.3. Upload Recipe Successfully

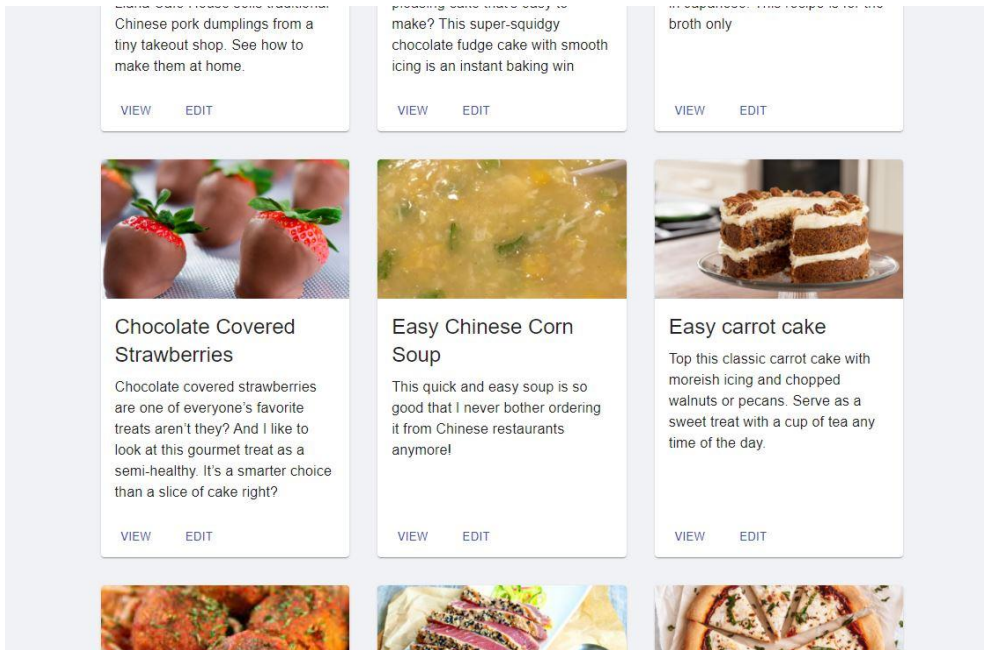






Figure 3.4.3. New Recipe on Website

Users can get help or give advices to Web by sent email showing blew (Figure 3.5.1)

For All Web users:

If you have any confusion, please send your question or advices to email 425293647@qq.com

This is only a test version, welcome to the above email us any suggestions, for this site we have all the right to explain



Back to Home

Need help?

China - YYDS © 2021

Figure 3.5.1. New Recipe on Website

4. Third-Party Functionalities

4.1 Front-End(React)

4.1.1 Virtual DOM

In traditional web applications, operating the DOM is generally a direct update operation, but we know that DOM updates are usually more expensive. In order to minimize the manipulation of the DOM, React provides a different and powerful way to update the DOM instead of direct DOM manipulation which is Virtual DOM, a lightweight virtual DOM, is an object abstracted by React, describing what the dom should look like and how it should be presented. The real DOM is updated through this Virtual DOM, and the update of the real DOM is managed by this Virtual DOM.

4.1.2 Components

The nodes on the DOM tree are called elements, but different in react, they are called components on the Virtual DOM. The node of Virtual DOM is a complete abstract component, composed of components.

4.1.3 State and Render

The state attribute contains some data needed to define the component. When the data changes, Render will be called to reproduce the rendering. Here, the data can only be updated through the provided setState method.

4.1.4 Ant Design and Material

React UI component library based on Ant Design and Material Design system which save us lots of time.

4.2 Back-End

4.2.1 Springboot Framework

A well-known Java based Enterprise Framework, which is a classic MVC architecture, our group uses this framework to build the backend. Then it's built-in function also used in the project, we use spring boot security to encrypt the password and match password, spring boot mail server to send verified email to user registry, and also spring boot interceptor to intercept all requests then doing authentication and authorization.

4.2.2 MyBatis

MyBatis is a persistence framework which can transfer models to databases and solve complex sql statements in tricky ways in XML files.

4.2.3 Log4j2

A Logger Library which helps to debug the exception and record system log, it helps a lot when the frontend connects with the backend.

4.2.4 lettuce

A Java redis client which is used to connect and transfer data with redis.

4.2.5 Java-Jwt

Java web token used in Java and jwt is important in our project. Almost every request needs to hold it in the request header and our system will decode this token to do authorization and authentication.

4.2.6 Aliyun-OSS

The OSS server can help to store the recipe photos and videos in this project and it is important for others to access other user's photos or videos.

4.2.7 Aliyun-ECS3

I deployed redis server in ecs3 server and I deployed our project in this ecs by using nginx. Deploy version can directly access on : www.comp3900-yyds.com

4.2.8 Cloud Mysql

Cloud database Service is useful for working from home, we can easily share the data and improve development efficiency.

We have prepared 2 kinds of Cloud, one is used by people who stay in China, another is used for those who stay in Sydney because of the database connection speed and delay.

5. Implementation Challenges

3.Implementation challenges :

descriptions of any tricks, non-trivial algorithms, special architecture/design, etc.

5.1 Front-End

5.1.1 Structure

In the first coding weeks, the front-end code has poor modularity, as project process goes, we extracted Navigation and Footer, and we only have a few pages, hence components render most of the pages which is way faster.

5.1.2 Style

Actually those two component library are design for backstage management,

5.2 Back-End

5.2.1 Shiro Conflict

At the beginning of the project, we prepared using shiro to do authentication and authorization, however, it conflicts with jwt token, I configured it for a long while and finally gave this framework and using jwt and store redis to do authentication and authorization.

5.2.2 Custom HTTP status code

We have our own built-in status code and it will return the http status code, our front end can figure out and give the user a better experience.

5.2.3 login Interceptor

For Authorization, I write a login interceptor filter to intercept all requests and get the request header to decode the token and for some api it does not need authorization, i will let it pass.

```
@Configuration
public class InterceptorConfig implements WebMvcConfigurer {
    @Override
    public void addInterceptors(InterceptorRegistry registry) {
        registry.addInterceptor(loginInterceptor())
            .addPathPatterns("/**")
            .excludePathPatterns(List.of(
                "/home",
                "/login",
                "/visitor/recipe_list",
                "/register",
                "/emailVerify/**",
                "swagger/**",
                "/swagger-ui.html",
                "swagger-ui.html",
                "/webjars/**",
                "/swagger-ui.html/*",
                "/swagger-resources",
                "/swagger-resources/**",
                "/test",
                "/tags/tag_list"
            ));
    }
}

@Bean
public LoginInterceptor loginInterceptor() { return new LoginInterceptor(); }
```

```

@Override
public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {
    String token = request.getHeader("token");
    if (!(handler instanceof HandlerMethod)) {
        return true;
    }
    if (StringUtils.isEmpty(token)) {
        throw new AuthorizationException();
    }

    String userId = null;
    try {
        userId = JwtUtil.decodeToken(token).getClaim("userId").asString();
    } catch (Exception e) {
        throw new AuthorizationException();
    }
    User user = userMapper.getUserById(userId);
    if (user == null) {
        throw new AuthorizationException();
    }
    if (!redisTemplate.hasKey(token)) {
        throw new AuthorizationException();
    }
    return true;
}

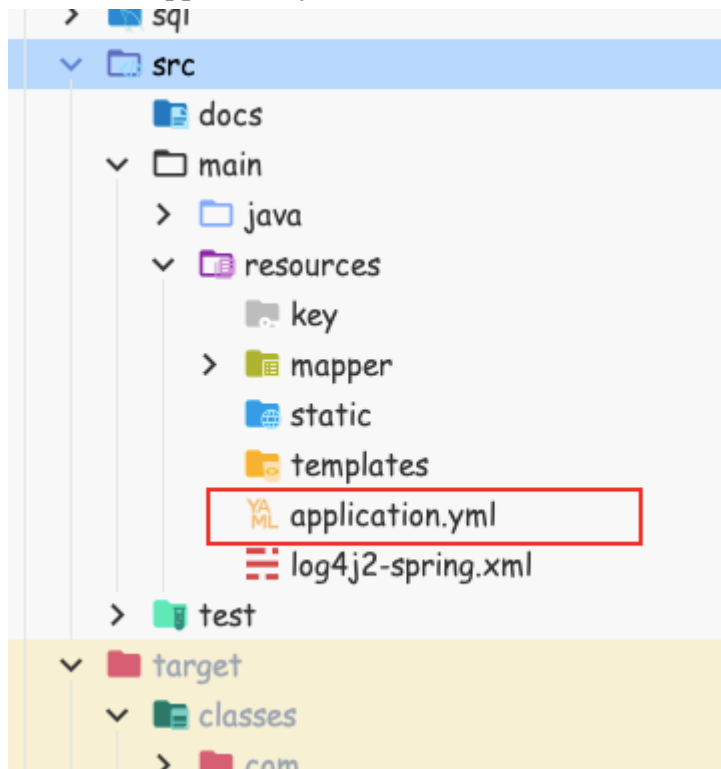
```

6. User Document/Manual

6.1 Back-End Setup

cd src/main/resources

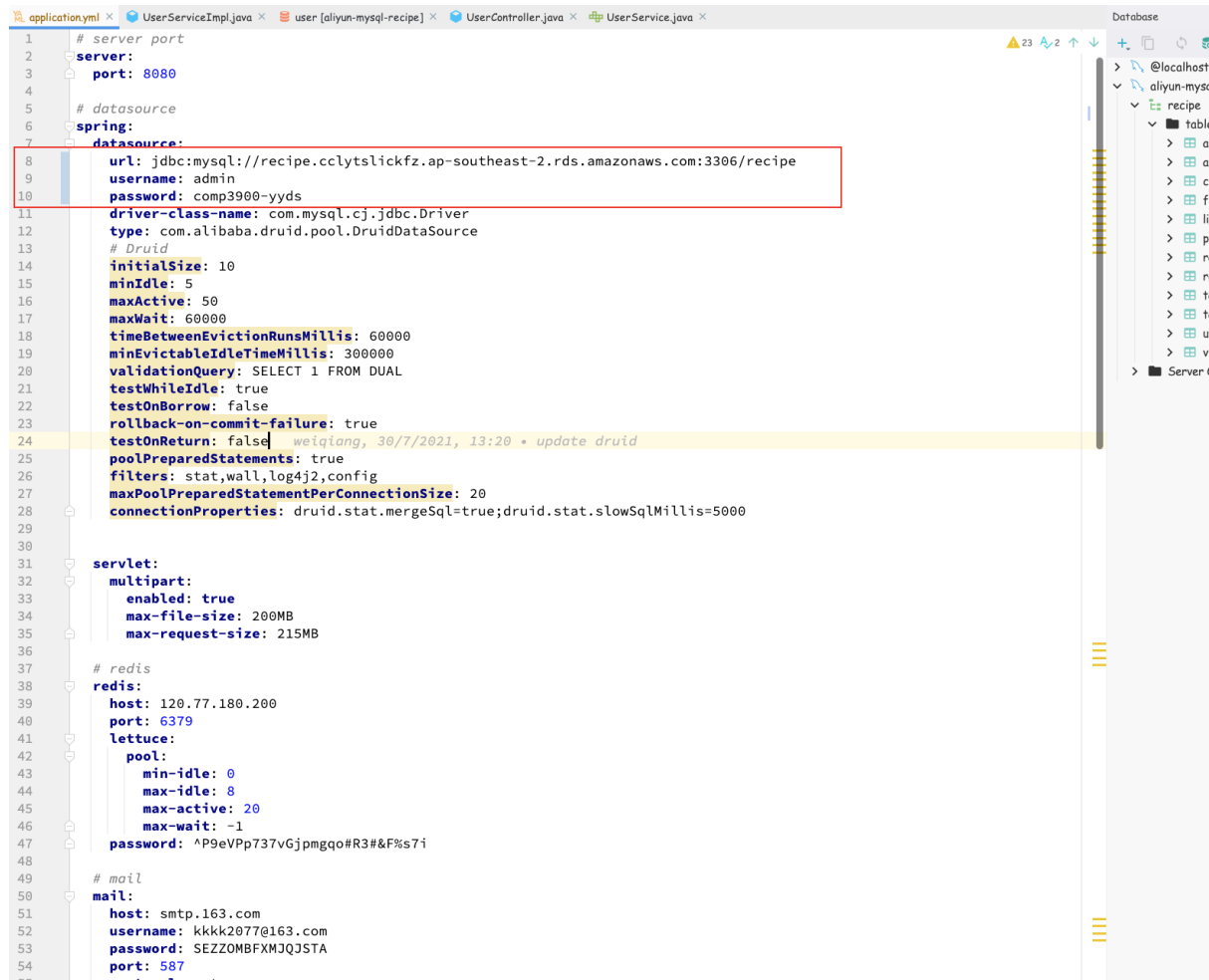
There is an application.yml.



1. If you are in Australia try this

url: jdbc:mysql://recipe.clytslickfz.ap-southeast-2.rds.amazonaws.com:3306/recipe
username: admin
password: comp3900-yyds

Paste in into



```

1  # server port
2  server:
3    port: 8080
4
5  # datasource
6  spring:
7    datasource:
8      url: jdbc:mysql://recipe.clytslickfz.ap-southeast-2.rds.amazonaws.com:3306/recipe
9      username: admin
10     password: comp3900-yyds
11     driver-class-name: com.mysql.cj.jdbc.Driver
12     type: com.alibaba.druid.pool.DruidDataSource
13     # Druid
14     initialSize: 10
15     minIdle: 5
16     maxActive: 50
17     maxWait: 60000
18     timeBetweenEvictionRunsMillis: 60000
19     minEvictableIdleTimeMillis: 300000
20     validationQuery: SELECT 1 FROM DUAL
21     testWhileIdle: true
22     testOnBorrow: false
23     rollback-on-commit-failure: true
24     testOnReturn: false
25     poolPreparedStatements: true
26     filters: stat,wall,log4j2,config
27     maxPoolPreparedStatementPerConnectionSize: 20
28     connectionProperties: druid.stat.mergeSql=true;druid.stat.slowSqlMillis=5000
29
30
31  servlet:
32    multipart:
33      enabled: true
34      max-file-size: 200MB
35      max-request-size: 215MB
36
37  # redis
38  redis:
39    host: 120.77.180.200
40    port: 6379
41    lettuce:
42      pool:
43        min-idle: 0
44        max-idle: 8
45        max-active: 20
46        max-wait: -1
47    password: ^P9eVPp737vGjpmgqo#R3#&F%$7i
48
49  # mail
50  mail:
51    host: smtp.163.com
52    username: kkkk2077@163.com
53    password: SEZZOMBFXMJQJSTA
54    port: 587
55
  
```

2. If you are in China try this

url: jdbc:mysql://rm-wz98eu202hx5fv84gzo.mysql.rds.aliyuncs.com:3306/recipe
username: root
password: qXd8b#5Y%!e2P4PcMR9zPj5XPA**

```

2  server:
3  port: 8080
4
5  # datasource | weiqiang, 17/6/2021, 21:07 * style: change from properties to yml
6  spring:
7  datasource:
8  url: jdbc:mysql://rm-wz98eu202hx5fv84gzo.mysql.rds.aliyuncs.com:3306/recipe
9  username: root
10 password: qXd8b#5Y%le2P4PcMR9zPj5XPA**
11 driver-class-name: com.mysql.cj.jdbc.Driver
12 type: com.alibaba.druid.pool.DruidDataSource
13 # Druid
14 initialSize: 10
15 minIdle: 5
16 maxActive: 50
17 maxWait: 60000
18 timeBetweenEvictionRunsMillis: 60000
19 minEvictableIdleTimeMillis: 300000
20 validationQuery: SELECT 1 FROM DUAL
21 testWhileIdle: true
22 testOnBorrow: false

```

Copy and replace it

Go to the root directory

run mvn install

```

# lubuntu @ lubuntu2004 in ~/Desktop/setup/recipe on git:dev/backend * [6:48:33]
$ mvn install
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/usr/share/maven/lib/guice.jar) to method java.lang.ClassLoader.defineClass(java.lang.
String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Scanning for projects...
[INFO]
[INFO] -----[ jar ]-----
[INFO] Building recipe 0.0.1-SNAPSHOT
[INFO]
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-mail/2.5.0/spring-boot-starter-mail-2.5.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-mail/2.5.0/spring-boot-starter-mail-2.5.0.pom (2.5 kB at 1.0 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/spring-context-support/5.3.7/spring-context-support-5.3.7.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/spring-context-support/5.3.7/spring-context-support-5.3.7.pom (2.4 kB at 6.7 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/sun/mail/jakarta.mail/1.6.7/jakarta.mail-1.6.7.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/sun/mail/jakarta.mail/1.6.7/jakarta.mail-1.6.7.pom (5.7 kB at 16 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/sun/mail/all/1.6.7/all-1.6.7.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/sun/mail/all/1.6.7/all-1.6.7.pom (24 kB at 59 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/activation/activation-1.2.2/activation-1.2.2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/activation/activation-1.2.2/activation-1.2.2.pom (4.6 kB at 13 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-data-redis/2.5.0/spring-boot-starter-data-redis-2.5.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-starter-data-redis/2.5.0/spring-boot-starter-data-redis-2.5.0.pom (2.5 kB at 7.1 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/data/spring-data-redis/2.5.1/spring-data-redis-2.5.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/data/spring-data-redis/2.5.1/spring-data-redis-2.5.1.pom

```

After that

```

Downloaded from central: https://repo.maven.apache.org/maven2/org/tukaani/xz/1.8/xz-1.8.jar (109 kB at 41 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.3.0/plexus-utils-3.3.0.jar (263 kB at 47 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-compress/1.19/commons-compress-1.19.jar (615 kB at 81 kB/s)
[INFO] Building jar: /home/lubuntu/Desktop/setup/recipe/target/recipe-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.5.0:repackage (repackage) @ recipe ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ recipe ---
Downloaded from central: https://repo.maven.apache.org/maven2/junit/junit/3.8.1/junit-3.8.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/junit/junit/3.8.1/junit-3.8.1.pom (998 B at 1.0 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.6/commons-codec-1.6.pom
Downloaded from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.6/commons-codec-1.6.pom (11 kB at 23 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-parent/22/commons-parent-22.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-parent/22/commons-parent-22.pom (42 kB at 32 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/0.4/maven-shared-utils-0.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/0.4/maven-shared-utils-0.4.pom (4.0 kB at 8.4 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/junit/junit/3.8.1/junit-3.8.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.15/plexus-utils-3.0.15.jar
Downloaded from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.6/commons-codec-1.6.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/0.4/maven-shared-utils-0.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/code/findbugs/jsr305/2.0.1/jsr305-2.0.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/code/findbugs/jsr305/2.0.1/jsr305-2.0.1.jar (32 kB at 55 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.15/plexus-utils-3.0.15.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.15/plexus-utils-3.0.15.jar (239 kB at 52 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.6/commons-codec-1.6.jar (233 kB at 157 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/junit/junit/3.8.1/junit-3.8.1.jar (121 kB at 79 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/0.4/maven-shared-utils-0.4.jar (155 kB at 101 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.15/plexus-utils-3.0.15.jar (239 kB at 94 kB/s)
[INFO] Installing /home/lubuntu/Desktop/setup/recipe/target/recipe-0.0.1-SNAPSHOT.jar to /home/lubuntu/.m2/repository/com/yyds/recipe/0.0.1-SNAPSHOT/recipe-0.0.1-SNAPSHOT.jar
[INFO] Installing /home/lubuntu/Desktop/setup/recipe/pom.xml to /home/lubuntu/.m2/repository/com/yyds/recipe/0.0.1-SNAPSHOT/recipe-0.0.1-SNAPSHOT.pom
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 04:19 min
[INFO] Finished at: 2021-08-02T06:53:04-04:00
[INFO]

```

cd target/

```
# lubuntu @ lubuntu2004 in ~/Desktop/setup/recipe on git:dev/backend x [6:55:30]
$ cd target

# lubuntu @ lubuntu2004 in ~/Desktop/setup/recipe/target on git:dev/backend x [6:55:33]
$
```

```
# lubuntu @ lubuntu2004 in ~/Desktop/setup/recipe/target on git:dev/backend x [6:55:33]
$ ls
classes generated-sources generated-test-sources naven-archiver naven-status recipe-0.0.1-SNAPSHOT.jar recipe-0.0.1-SNAPSHOT.jar.original surefire-reports test-classes
```

java -jar recipe-0.0.1-SNAPSHOT.jar

```
# lubuntu @ lubuntu2004 in ~/Desktop/setup/recipe/target on git:dev/backend x [6:55:36]
$ java -jar recipe-0.0.1-SNAPSHOT.jar

  ____ _
 / ___| | | |
| |___| | | |
 \___|_| |_|_|
:: Spring Boot ::
(v2.5.0)

[06:56:41:324] [DEBUG] - org.jboss.logging.LoggerProviders.logProvider(LoggerProviders.java:152) - Logging Provider: org.jboss.logging.Log4j2LoggerProvider
[06:56:41:384] [INFO] - org.hibernate.validator.internal.util.Version.<clinit>(Version.java:21) - HV000001: Hibernate Validator 6.1.5.Final
[06:56:41:420] [DEBUG] - org.hibernate.validator.internal.xml.config.ValidationXmlParser.getValidationXmlInputStream(ValidationXmlParser.java:120) - Trying to load META-INF/validation.xml for XML based Validator configuration.
[06:56:41:425] [DEBUG] - org.hibernate.validator.internal.xml.config.ResourceLoaderHelper.getResettableInputStreamForPath(ResourceLoaderHelper.java:60) - Trying to load META-INF/validation.xml via TCCL
[06:56:41:427] [DEBUG] - org.hibernate.validator.internal.xml.config.ResourceLoaderHelper.getResettableInputStreamForPath(ResourceLoaderHelper.java:66) - Trying to load META-INF/validation.xml via Hibernate Validator's class loader
[06:56:41:428] [DEBUG] - org.hibernate.validator.internal.xml.config.ValidationXmlParser.getValidationXmlInputStream(ValidationXmlParser.java:127) - No META-INF/validation.xml found. Using annotation based configuration only.
[06:56:41:453] [DEBUG] - org.hibernate.validator.internal.engine.resolver.TraversableResolvers.getDefault(TraversableResolvers.java:60) - Cannot find javax.persistence.Persistence on classpath. Assuming non JPA 2 environment. All properties will per default be traversable.
[06:56:41:483] [INFO] - org.springframework.boot.StartupInfoLogger.logStarting(StartupInfoLogger.java:55) - Starting RecipeApplication v0.0.1-SNAPSHOT using Java 11.0.11 on lubuntu2004 with PID 106359 (/home/lubuntu/Desktop/setup/recipe/target/recipe-0.0.1-SNAPSHOT.jar started by lubuntu in /home/lubuntu/Desktop/setup/recipe/target)
[06:56:41:514] [DEBUG] - org.springframework.boot.StartupInfoLogger.logStarting(StartupInfoLogger.java:56) - Running with Spring Boot v2.5.0, Spring v5.3.7
[06:56:41:516] [INFO] - org.springframework.boot.SpringApplication.logStartupProfileInfo(SpringApplication.java:663) - No active profile set, falling back to default profiles: default
[06:56:41:623] [DEBUG] - org.hibernate.validator.messageinterpolation.ResourceBundleMessageInterpolator.buildExpressionFactory(ResourceBundleMessageInterpolator.java:172) - Loaded expression factory via original TCCL
[06:56:42:022] [DEBUG] - org.hibernate.validator.internal.engine.ValidatorFactoryConfigurationHelper.determinePropertyNameProvider(ValidatorFactoryConfigurationHelper.java:300) - HV000252: Using org.hibernate.validator.internal.engine.DefaultPropertyNameProvider as property node name provider.
[06:56:42:054] [DEBUG] - org.hibernate.validator.internal.engine.ValidatorFactoryConfigurationHelper.logValidatorFactoryScopedConfiguration(ValidatorFactoryConfigurationHelper.java:375) - HV000234: Using org.hibernate.validator.messageinterpolation.ResourceBundleMessageInterpolator as ValidatorFactory-scoped message interpolator.
[06:56:42:055] [DEBUG] - org.hibernate.validator.internal.engine.ValidatorFactoryConfigurationHelper.logValidatorFactoryScopedConfiguration(ValidatorFactoryConfigurationHelper.java:376) - HV000234: Using org.hibernate.validator.internal.engine.resolver.TraverseAllTraversableResolver as ValidatorFactory-scoped traversable resolver.
[06:56:42:056] [DEBUG] - org.hibernate.validator.internal.engine.ValidatorFactoryConfigurationHelper.logValidatorFactoryScopedConfiguration(ValidatorFactoryConfigurationHelper.java:377) - HV000234: Using org.hibernate.validator.internal.util.ExecutableParameterNameProvider as ValidatorFactory-scoped parameter name provider.
[06:56:42:058] [DEBUG] - org.hibernate.validator.internal.engine.ValidatorFactoryConfigurationHelper.logValidatorFactoryScopedConfiguration(ValidatorFactoryConfigurationHelper.java:378) - HV000234: Using org.hibernate.validator.internal.engine.DefaultClockProvider as ValidatorFactory-scoped clock provider.
[06:56:42:059] [DEBUG] - org.hibernate.validator.internal.engine.ValidatorFactoryConfigurationHelper.logValidatorFactoryScopedConfiguration(ValidatorFactoryConfigurationHelper.java:379) - HV000234: Using org.hibernate.validator.internal.engine.scripting.DefaultScriptEvaluatorFactory as ValidatorFactory-scoped script evaluator factory.
[06:56:42:909] [WARN] - org.springframework.core.log.CompositeLog.warn(CompositeLog.java:127) - For Jackson Kotlin classes support please add "com.fasterxml.jackson.module:jackson-module-kotlin" to the classpath
```

6.2 Front-End Setup

cd frontend/

```
# lubuntu @ lubuntu2004 in ~/Desktop/setup/recipe on git:dev/backend x [6:45:58]
$ cd frontend
```

yarn install or npm install

```

# lubuntu @ lubuntu2004 in ~/Desktop/setup/recipe/frontend on git:dev/backend x [6:46:06] C:127
$ npm install
> core-js@2.6.12 postinstall /home/lubuntu/Desktop/setup/recipe/frontend/node_modules/core-js
> node -e "try(require('./postinstall'))catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

> core-js-pure@3.14.0 postinstall /home/lubuntu/Desktop/setup/recipe/frontend/node_modules/core-js-pure
> node -e "try(require('./postinstall'))catch(e){}"

> uglifyjs-webpack-plugin@0.4.6 postinstall /home/lubuntu/Desktop/setup/recipe/frontend/node_modules/uglifyjs-webpack-plugin
> node lib/post_install.js

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules/webpack-dev-server/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules/watchpack-chokidar2/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules/react-scripts/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})

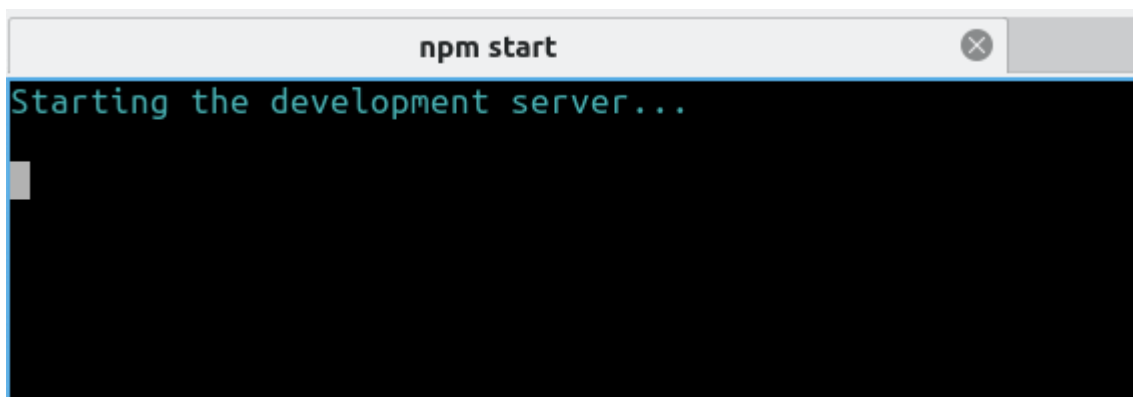
added 1658 packages from 903 contributors and audited 1666 packages in 126.422s

57 packages are looking for funding
  run `npm fund` for details

found 79 vulnerabilities (33 low, 36 moderate, 10 high)
  run `npm audit fix` to fix them, or `npm audit` for details

```

yarn start or npm start



frontend Set Done

Then it will skip to the project homepage.

6.3 Deployment in ECS

We have deployed in Aliyun ECS3

www.comp3900-yyds.com