SW Engineering CSC648/848 Summer 24
Summer 2024

FitNutri Hub
Team 03

Student	Full Name	SFSU Email	Role
# 1	Michelle Nguyen	mnguyen62@sfsu.edu	Team-Lead
#2	Mitchell Caine	mcaine@sfsu.edu	Frontend-Lead
#3	Shreejana Bartaula	sbartaula@mail.sfsu.edu	Docs-editors and Frontend
#4	Ali Almusawi	aalmusawi@mail.sfsu.edu	
#5	Eduardo Enrique Muñoz Alvarez	emunozalvarez@sfsu.edu	Database-admin
#6	Nilofar Ali	nmohammadali@mail.sfsu.edu	Frontend-Lead
#7	Uzair Hamed Mohammed	umohammed@sfsu.edu	Backend
#8	John Collins	jcollins9@sfsu.edu	Backend
#9	Ali Hadwan	hadwanali41@gmail.com	Frontend

MILESTONE 4
Date: 07/25/2024

History Table

Milestone	**Version**	**Date Submitted**
Milestone 1	V1	6/19/24
Milestone 1	V2	7/8/2024
Milestone 2	V1	7/8/2024
Milestone 3	V1	7/24/2024
Milestone 3	V2	7/29/2024
Milestone 4	V1	7/29/2024

Table of Contents

1. Product Summary

FitNutri is a platform where fitness meets nutrition. What does that mean and what do we provide? Our platform is dedicated to enhancing your health and wellness journey. Unlike conventional fitness apps, FitNutri integrates personalized workout plans and dietary guidance tailored to your individual goals and preferences. With our approach here at FitNutri, it will ensure that you will receive the support your need to achieve your fitness and nutrition objectives. We focus on user-centered design, providing an intuitive and engaging experience for every user. Whether you're a fitness enthusiast or a beginner, FitNutri accommodates all levels, allowing you to create a profile, track your progress, and stay motivated. Users can securely sign up, log in, and manage their accounts, ensuring a smooth and secure experience.

You may ask: How might we stand out from other applications? With FitNutri, we have personalized plans that generate customized nutrition and workout plans based on your health information and goals. Second, activity tracking monitors calories burned and steps taken throughout the day. Third, we have a library where we have access to a wide range of recipes, exercises, and educational resources. Fourth, we have a place for community engagement. You will be able to join live streaming and on-demand classes led by professional trainers. This way, you will feel a sense of motivation! Lastly, we have interactive features where users can share, edit, and print their fitness and nutrition progress, and engage with other users through posts and feedback. Unlike other "similar" fitness health applications, FitNutri combines real-time interaction with professional trainers and advanced algorithms to provide a personalized and motivating experience with these living streaming and on-demand classes. We help bring an interactive and supportive environment, making fitness accessible and enjoyable for everyone.

FitNutri is where our fitness goes with health and nutrition. Explore FitNurti where you will have everything you need to gain that healthy lifestyle. Take the first step towards achieving your health and fitness goals!

Functional Requirements - Priority 1

1. **User Account Management**
   1.1 A user shall securely sign up.
   1.2 A user shall securely sign in.
   1.3 A user should be able to create their profile (age, weight, height, fitness goals).
   1.4 A user should be able to do password recovery/reset.
   1.5 A user should be able to update their account.
   1.6 A user should be able to delete their account.
   1.7 Each user shall have exactly one account.
   1.8 A user shall log into their account using secure authentication.
   1.9 A user shall fill out one health information form.
   1.10 A user shall be able to choose many types of nutrition
   1.11 A user shall be able to choose many types of workouts
   1.12 Users shall write many dietary restrictions
   1.13 A user shall generate one personal nutrition plan
   1.14 A user shall generate one personal workout plan
   1.15 A user shall view many recipes
   1.16 A user shall view many exercises
   1.17 Users shall search for recipes and workouts using keywords.
   1.18 Users shall be able to share, edit and print fitness and nutrition progress many times.
2. **Account**
   2.1An account shall belong to one user at most.
   2.1An account shall be associated with one email at most.
3. **Tracking**
   3.1. A user shall be able to track their calories burned throughout their day.
   3.2. A user shall be able to track their total steps throughout their day.
4. **Health information**
   4.1A health information shall be filled by one user
   4.2 A health information shall contain many Types of Nutrition
   4.3 A health information shall contain many Types of workout
   4.3A health information shall contain one article based on a personal nutrition plan.
   4.3 health information shall contain one article based on a personal fitness
   4.4 Health information shall include user goals related to weight, fitness level, and dietary preferences
5. **Types of Nutrition**
   5.1 A type of Nutrition shall be chosen by many users
   5.2 A type of Nutrition shall contain one article
6. **Types of Workout**
   6.1 A type of workout shall be chosen by many users
   6.2 A type of workout shall contain one article
7. **Exercises**
   7.1 An exercise shall include instructional videos and images
   7.2 An exercise shall provide estimated calories burned
   7.3 An exercise shall be viewed by many users
8. **Recipe**

8.1 A recipe shall consist of many dishes
8.2 A recipe shall be categorized by diet type (ex., keto, vegan)
8.3 A recipe shall be viewed by many users
8.4 A recipe shall include instructional videos and images

9. Personal nutrition plan

9.1 A personal nutrition plan shall be generated by many users
9.2 A personal nutrition plan shall be shared by many users

10. Personal fitness plan

10.1 A personal workout plan shall be generated by many users
10.2 A personal workout plan shall be shared by many users

11. Dietary Restrictions

11.1 A Dietary Restrictions shall be written by many users

12. Assess

12.1 An assess shall be generated by many users

13. Form

13.1 A form shall be submitted by one users
13.2 A form shall be categorized by topic (ex., nutrition, workouts, achievements)
13.3 A form shall allow users to request support or submit feedback directly to the service team

14. Post

14.1 shall be liked by many users
14.2 Posts can be categorized by topic (Ex, nutrition, workouts, achievements)

15. Article

15.1 An article shall contain one type of nutrition
15.2 An article shall contain one type of workout

16. Nutrition progress

16.1 Nutrition progress shall be editable by one user
16.2 Nutrition progress shall display food intake(ex., breakfast, lunch ,dinner), micros and vitamins to one user

17. Fitness progress

Fitness progress shall be editable by one user
Fitness progress shall display activity(ex. number of workout, minutes and days of strike), weight, lb and achievements to one user

2. Usability Test Plan

Usability Test Procedure

1. A user should be able to fill in their health information
2. A user should be able to generate a nutrition plan
3. A user should be able to generate a workout plan
4. A user should be able to view a recipe
5. A user should be able to view a workout

Test for FitNutri

Filling in Health Information

- Objective:

  The objective of this test is to check how usable filling health information is on our website. The test will involve finding how easy it is for users to fill out the health information section with prompts for what to fill out and if the overall experience of inputting their data is easy to accomplish.
  The importance of inputting user data is because our website aims to create accurate suggestions for both exercise and recipes. Without being able to properly input this information, the user would not have a custom plan tailored to what they provided.
  In testing this metric, our team is able to properly assess the usability of this function in order to create their custom workout and recipe plans so that they are happy with the regimen given to them and so it will be more effective in helping them do what the website was designed to do.

- Description:

  System Setup - The setup assumes that the user is on the web page and been directed to the health information page.

Starting Point - The user will be on the health information page and directed to the top most textbox of the page.
Intended Users - People who wish to manage their health and get custom plans to manage it.
URL - http://13.57.220.69/HealthInfo
Measurement - Amount of time to fill out health information, if the user is able to enter information incorrectly, and user happiness while filling out the form.

Generating a Nutrition Plan

- Objective:

The purpose of this test is to find out how accurate the plan generation is. This will show the process of how accurate a plan will be after entering in health information.
Creating a personal plan is one of the cornerstones of the FitNutri application. By tailoring recommendations to our users, we are able to provide a means of reaching the health goals users are looking for. The test will find any problems with generating a plan which could be caused by difficult navigation or unclear instructions.
In measuring the time it takes to generate the plan, the users' understanding of what is happening, and the satisfaction of the plan given, we are able to address the needs of our users and help them achieve the goals they set out to do.

- Description:

System Setup - Health information has already been input by the user.
Starting Point - The user moves to and presses the generate plan button.
Intended Users - Users who have input their health information and who want a custom Personal nutrition plan.
URL - http://13.57.220.69/HealthInfo
Measurement - Time from button press until the plan is generated and happiness with the generated plan

Generate a Workout Plan

- Objective:

  We are doing this test in order to check the usability of the workout plan generation. This test will attempt to figure how easily a user will create a workout plan after inputting their health data. We wish to understand how intuitive the interface is, the ease of use, and user experience.
  Being able to have custom workouts generated for our users will help them reach the specific goals they want to achieve. By testing against this case, we will identify any problems with generating a workout plan which might be caused by unintuitive steps or slow processing times.
  The overall goal is to make sure users can properly generate a workout plan which is made based on their preferences. The test will watch for any errors when generating a plan, the time it takes to generate, and how happy the user is with the generated plan.

- Description:

  System Setup - Health information has already been input by the user.
  Starting Point - The user moves to and presses the generate plan button.
  Intended Users - Users who have input their health information and who want a custom nutrition plan.
  URL - http://13.57.220.69/PersonalWorkouts
  Measurement - Time from button press until the plan is generated and happiness with the generated plan.

View a Recipe

- Objective:

  For this test, we are testing the experience of viewing the recipe page. The focus will be on how easy it is for users to access recipes and understand how to use them. The clarity of the instructions as well as finding recipes will be key.

Viewing recipes is an imperative aspect of our website because it is part of the custom nutrition plan generated during sign up. We aim to find problems users might run into when navigating to and interacting with the page.
Our test will determine if users are able to find recipes and view them. The recipes should be easy to follow and meet their dietary needs. The test will measure how long it takes to find a recipe, the ease of following the recipe, and how the user feels about the navigation and viewing experience.

- Description:

System Setup - The website has a list of recipes ready.
Starting Point - The user will start on the recipes section and select a recipe to view.
Intended Users - Users who wish to pair their workouts with nutrition which helps with their goals.
URL - http://13.57.220.69/PersonalRecipes
Measurement - Amount of time to find and view a recipe, ease of understanding recipe instructions, and user happiness with the experience.

View a Workout

- Objective:

This test will be understanding how usable the workout page is. We will be testing how easy it will be for users to access the workout routines and understand how to do them. The ease of navigation and instructions will be the focus of this test.
Being able to find and use the workout routine is one of the most important aspects of our website. We plan to identify any problems which might cause problems preventing them from access to the workouts they need. The issues include difficulty getting to the workout page and understanding the workout routine.
The goal is to make sure users are able to view and understand their routines so that they may follow along. We will be measuring the time it takes to navigate to the workouts, the effectiveness of reading the workout, and how happy the user is from viewing the workout.

- Description:

  System Setup - The website has a list of workouts ready.

  Starting Point - The user will start on the workouts section and select a workout to view.

  Intended Users - Users who wish to pair their nutrition with workouts which helps with their goal.

  URL - http://13.57.220.69/PersonalWorkouts

  Measurement - Amount of time to find and view a workout, ease of understanding workout instructions, and user happiness with the experience.

Usability Test Table (Effectiveness & Efficiency)

Test/Use Case	% Completed	Errors	Time	Comments	Satisfaction (Likert Questionnaire)
1	100%	None	00:54:19	It's unclear if height requested is in cm or ft	1. The process of creating an account was straightforward and easy to follow. 2. I found the health profile questions to be relevant and comprehensive. 3. I felt confident that my personal health informatio

					n was secure during the account creation process.
2	100%	None			1. The nutrition plan generated by the app meets my dietary needs and preferences. 2. I found the nutrition plan easy to understand and follow. 3. The app provided sufficient variety in the nutrition plans offered.
3	100%	None			1. The workout plan generated by the app aligns with my fitness goals. 2. I found the workout plan to be appropriately

					challenging for my fitness level. 3. The app provided clear instructions and demonstrations for the exercises given in the workout plan.

3. QA Test Plan :

1. Objective: Information related to the user such as their password, daily activity data (steps, average heart rate, etc..), recorded workout data shall be encrypted and appear unreadable without the decryption key.

HW/SW setup: AWS EC2 instance running Ubuntu and Apache Web Server. MySQL database paired with the MySQL workbench application. http://13.57.220.69/

No.	Test	Description	Test Input	Expected Output	Test Results
1	Fitness Data Encryption	Check database for successful encryption of use fitness data	Logged fitness data of user	Data appears encrypted in database	Pass
2	Workout Data Encryption	Check database for successful encryption of use workout data	Recorded workouts of user	Data appears encrypted in database	Pass
3	Activity Data Encryption	Check database for successful encryption of use activity data	Daily activity data of user	Data appears encrypted in database	Pass

2. Objective: The application shall load reasonably quickly regardless of what device form factor (phone, tablet, laptop, etc.) is being used to access the application.

HW/SW setup: Android device running Android 14 and using the Samsung Internet browser. Windows 11 device using Firefox 128.0.3. http://13.57.220.69/

No.	Test	Description	Test Input	Expected Output	Test Results
1	Access dashboard on Android	Measure the amount of time it takes for a user's populated dashboard to load on an Android device	Various bits of data from a user's day, such as their steps, average heart rate, and calories burned	<2 seconds	Pass

2	Access dashboard on Windows	Measure the amount of time it takes for a user's populated dashboard to load on a Windows device	Various bits of data from a user's day, such as their steps, average heart rate, and calories burned	<2 seconds	Pass
3	Access workout summary on Android	Measure the amount of time it takes for the summary of a single recorded workout to load on an Android device	Data collected during a user's workout, such as their step count and heart rate, combined with calculated data such as calories burned, displayed in a neat summary	<4 seconds	Pass

3. Objective: The application should have a 99.9% uptime and always be available to the user, except during times of scheduled maintenance.

HW/SW setup: AWS EC2 instance running Ubuntu and Apache Web Server. Server has 16 GB RAM at disposal. http://13.57.220.69/

No.	Test	Description	Test Input	Expected Output	Test Results
1	Android access test during midday	Users, especially office workers, tend to take a	Activity data collected from users	Application loads successfully	Pass

		lunch break and go for a walk. The result is a large group of users recording activities. An Android device shall attempt to access the application during these hours			
2	Android access test during early morning hours	Early morning hours see low levels of users recording workouts. An Android device shall attempt to access the application during these hours	Activity data collected from users	Application loads successfully	Pass
3	Linux access test during evening hours	Evening hours see moderate to high levels of user activity. A Linux device shall attempt to access the application during these hours	Activity data collected from users	Application loads successfully	Pass

4. Objective: User Interface

5.7 The user interface shall be visually appealing and customizable to enhance user engagement and satisfaction.

HW/SW setup: AWS EC2 instance running Ubuntu and Apache Web Server. Server has 16 GB RAM at disposal. http://13.57.220.69/

No.	Test	Description	Test Input	Expected Output	Test Results
1	Appealing test visual	Look at the aesthetic quality of the user interface.	Visual inspection	Interface should be visually	Pass (Chrome)
2	Customization Options	Test the availability and functionality of customization options.	Modify UI settings such as theme layouts.	Users can/should customize the interface easily.	Pass (Firefox)
3	Consistency Test	Navigate through a different page	Navigate through pages	Pages should have a consistent design and layout.	Pass (Safari)

5. Objective: User Interface

The application shall have an FAQ for customers to answer their own questions.

HW/SW setup: AWS EC2 instance running Ubuntu and Apache Web Server. Server has 16 GB RAM at disposal. http://13.57.220.69/

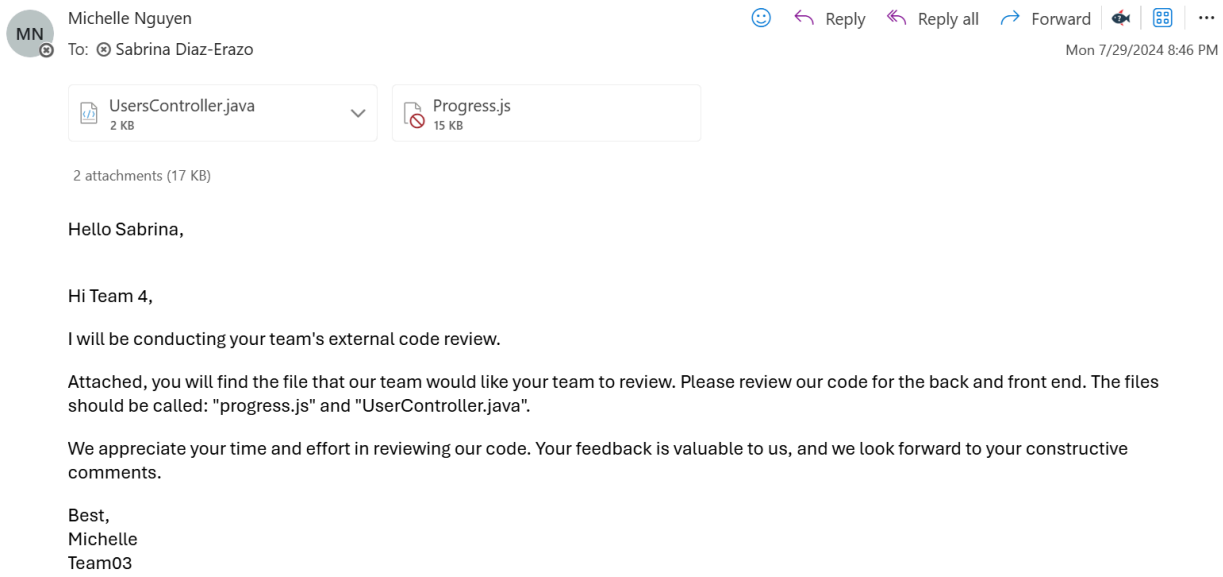
No.	Test	Description	Test Input	Expected Output	Test Results
1	FAQ Availability test	FAQ should be able to be accessible from the main menu.	Navigate to the FAQ section from the main menu.	FAQ section should be easily accessible from the main menu.	PASS

2	FAQ Content test	This section should contain a review for the content of the FAQ section.	Review the content of the FAQ section.	It should be accurate, helpful and comprehensive answers.	PASS
3	FAQ Search	Use search functionality within FAQ	User search functionality within FAQ.	Search should return relevant results based on the query.	PASS

4. Code Review

They need a better layout/design of their application. Compared to our team03, we chose the MVC design pattern for the backend, and thanks to it, our files are very organized and easy to find. We are not sure how javascript projects work, but it looks like they dumped everything in one file.

Email Exhange:



Michelle Nguyen
To: Sabrina Diaz-Erazo

Reply Reply all Forward

Mon 7/29/2024 8:46 PM

UsersController.java 2 KB

Progress.js 15 KB

2 attachments (17 KB)

Hello Sabrina,

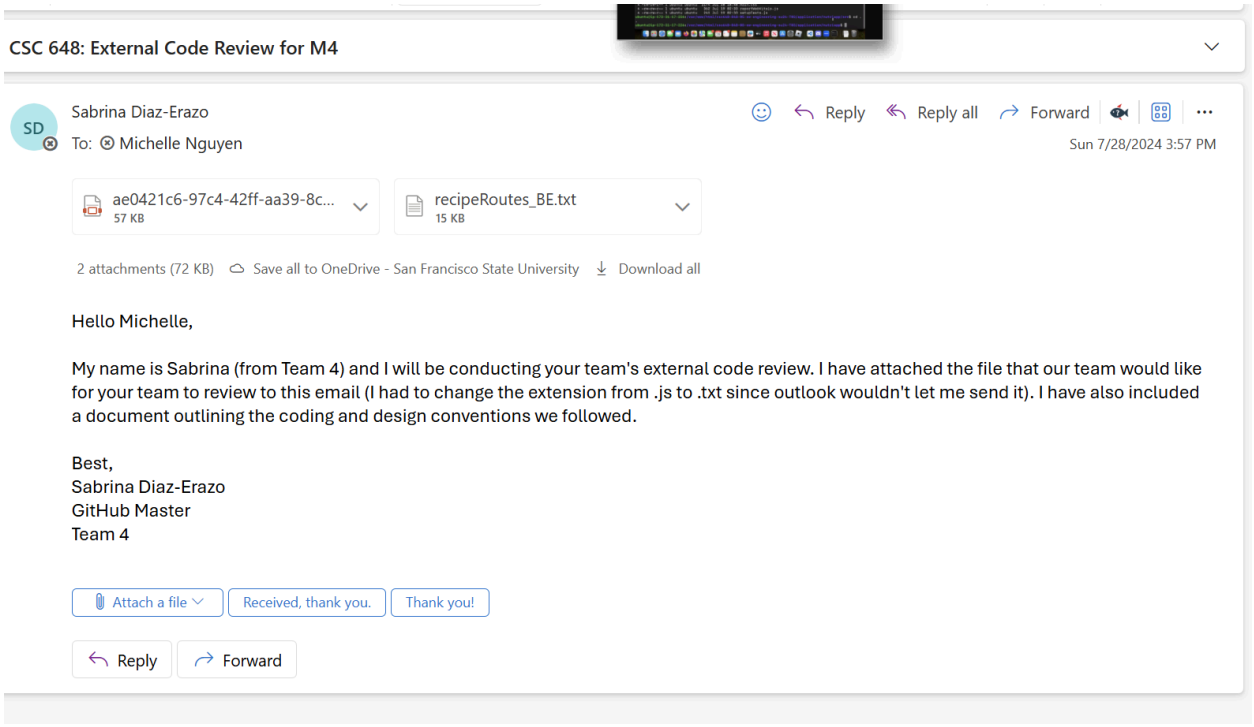
Hi Team 4,

I will be conducting your team's external code review.

Attached, you will find the file that our team would like your team to review. Please review our code for the back and front end. The files should be called: "progress.js" and "UserController.java".

We appreciate your time and effort in reviewing our code. Your feedback is valuable to us, and we look forward to your constructive comments.

Best,
Michelle
Team03



CSC 648: External Code Review for M4

SD Sabrina Diaz-Erazo
To: Michelle Nguyen

Reply Reply all Forward
Sun 7/28/2024 3:57 PM

ae0421c6-97c4-42ff-aa39-8c... 57 KB
recipeRoutes_BE.txt 15 KB

2 attachments (72 KB) Save all to OneDrive - San Francisco State University Download all

Hello Michelle,

My name is Sabrina (from Team 4) and I will be conducting your team's external code review. I have attached the file that our team would like for your team to review to this email (I had to change the extension from .js to .txt since outlook wouldn't let me send it). I have also included a document outlining the coding and design conventions we followed.

Best,
Sabrina Diaz-Erazo
GitHub Master
Team 4

Attach a file | Received, thank you. | Thank you!

Reply Forward

Feedback

MN Michelle Nguyen
To: Sabrina Diaz-Erazo
Cc: Uzair Hamed Mohammed

Reply Reply all Forward
Mon 7/29/2024 11:20 PM

Hello Sabrina,

Our team (Uzair and I) reviewed your code that you gave us and here are some feedback we saw in your code. You all need a better layout/design of your application. For example we chose the MVC design pattern for the backend, and thanks to it, our files are very organized and easy to find. We are not sure how javascript projects work, but it looks like you all dumped everything in one file?

Additionally...

1. 1. Avoid hardcoding credentials in the code.
2. 2. Use environment variables or a secure vault to manage sensitive information
3. 3. The error message in the res.status(400).json response should be more user-friendly and avoid exposing internal error codes directly to the user. Consider logging the detailed error internally and providing a generic message to the user
4. 4. The comments are helpful, but ensure they are up-to-date and accurate. The comment about IS_LOGGED_IN is unclear without additional context.

I hope we were able to help you review your code. Let us know (Uzair and I) if you have any concerns or questions.

Best,

Michelle & Uzair

Team03

Reply Reply all Forward

```
const connection = mysql.createPool({
    host:       'csc648database.cfgu0ky6ydzi.us-east-2.rds.amazonaws.com',
```

```
    user:       'backend_lead',
    password:   'password',
    database:   'ScholarEats'
});
```

Avoid hardcoding credentials in the code. Use environment variables or a secure vault to manage sensitive information

```
if (userUniversityInfo == 0 ) { // No user info
                return res.status(400).json({ error: 'There was an error
retreving user data. Please try again later. Error code: RR_BE:47'}) //
Error code refrences which line tripped
            }

            const userUniversity = userUniversityInfo[0].university; //
User's university

            // Find the user's university ID from the univeristy name
            const universityIdQuery = `SELECT university_id FROM
university WHERE name = ?`;
            const [universityIdInfo] = await
connection.execute(universityIdQuery, [userUniversity]);

            if (universityIdInfo === 0) {
```

The error message in the res.status(400).json response should be more user-friendly and avoid exposing internal error codes directly to the user. Consider logging the detailed error internally and providing a generic message to the user.

```
/*
* Reserve a recipe and send a notification to the admin of the university
* MERGED FROM reserveRecipeButton_BE.js, since this file is already
* hooked up to the individual recipes page.
*/
```

```
router.post('/:id', IS_LOGGED_IN, async (req, res) => {
    try {
        const recipeId = req.params.id; // Recipe ID
        debugMsg(recipeId);
```

The comments are helpful, but ensure they are up-to-date and accurate. The comment about IS_LOGGED_IN is unclear without additional context.

5. Self Check on Best Practices for Security

5.1: Protected Assets:

5.1.1: User password

5.1.2: User vitals data

5.1.3: User daily activity data

5.1.4: User workout data

5.1.5: User personalized recipes

5.1.6: User nutrition plan

5.1.7: User height, weight, gender, and medical conditions

5.2: Password Encryption in Database: When a new user is created or an existing user's password is updated, the password is encoded using a `PasswordEncoder` before being saved to the database. Additionally, during the login process, the provided password is compared with the stored encoded password using the `PasswordEncoder` object's `matches` method. As the name suggests, this method checks if the provided password, when encoded, matches the stored encoded password.

```java
@Configuration
@EnableWebSecurity
public class SecurityConfig {

    Uzair Mohammed
    @Bean
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
```

```java
Uzair Mohammed *
@PostMapping("/createUser")
public ResponseEntity<User> addUser(@Valid @RequestBody User user){
    user.setPassword(passwordEncoder.encode(user.getPassword()));
    return ResponseEntity.ok(userService.createUser(user));
}

new *
@PutMapping("/updateUser/{id}")
public ResponseEntity<User> updateUser(@PathVariable Long id, @Valid @RequestBody User userData){
    userData.setPassword(passwordEncoder.encode(userData.getPassword()));
    return ResponseEntity.ok(userService.updateUser(id, userData));
}
```

```java
Uzair Mohammed
@PostMapping("/login")
public ResponseEntity<?> loginUser(@RequestBody LoginRequest loginRequest){
    User user = userService.findByEmail(loginRequest.getEmail());
    if(user != null && passwordEncoder.matches(loginRequest.getPassword(), user.getPassword())){
        return ResponseEntity.ok().body("User authenticated successfully");
    }else{
        return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Invalid email or password");
    }
}
```

5.3: Input Data Validation: User input is validated using Hibernate Validator, which is integrated with Spring Boot. Hibernate makes available annotations such as `@NotNull`, `@Size`, and `@Email` as defined by the JSR-303/JSR-380 APIs. These annotations ensure that the input data

meets the specified criteria before it is processed or stored. Provided below is an example of validation in the User model and controllers.

```
@NotNull
@Size(min = 1, max = 50)
private String first_name;

3 usages
@Size(min = 1, max = 50)
private String last_name;

3 usages
@NotNull
@Email
private String email;
```

```
Uzair Mohammed *
@PostMapping("/createUser")
public ResponseEntity<User> addUser(@Valid @RequestBody User user){
    user.setPassword(passwordEncoder.encode(user.getPassword()));
    return ResponseEntity.ok(userService.createUser(user));
}
```

6. Self Check: Adherence to Original Non-Functional Specs

1. **User Information**

1.1. The application shall have data encrypted and saved to the database. **DONE**

1.2 Data in the application must be encrypted and stored in the database. **DONE**

1.2 The login process of the Application System requires Two-factor authentication.

ISSUE: We have not reached this part of the prototype to implement this feature.

1.3 The app must guarantee the privacy of user data by following laws like GDPR and CCPA.

ISSUE: We cannot find any engines to be able to implement to the system to stick to the GDPR and CCPA regulations, well there are engines but it is a bit difficult to install them and figured it would be best to skip this portion of the document.

1.4 The app will allow customers to see, change, and remove their personal information while following privacy laws.

ISSUE: We have not been able to reach this part of the prototype to be able to do this.

1.5 The app will allow customers to keep their information private. **DONE**

2. **Application Database**

2.1. The application System should perform regular data backups. **ON TRACK**

2.2. The application shall log customers' activity. **DONE**

2.3. The database should support high availability and automatic failover to ensure continuous operation. **DONE**

2.4. The application shall provide data recovery procedures in case of data loss or corruption. **DONE**

2.5. The database shall be optimized for both read and write operations to ensure quick access and storage of user data. **DONE**

2.6. The database shall be designed to handle a growing volume of data as the user base increases. **DONE**

3. **Performance**

3.1. Application response time should load the App within 3 seconds. **DONE**

3.2. Application Systems should maintain detailed logs of application usage, errors, and security events to support troubleshooting and improve system reliability. **DONE**

3.3. The system should have a 99.9% uptime, excluding scheduled maintenance. **DONE**

3.4. The system should implement caching strategies to reduce load on the database and improve response times. **DONE**

3.5. The application should support concurrent usage by a large number of customers without significant performance degradation. **DONE**
3.6. The application shall be stress-tested to ensure it can handle peak loads and high traffic periods. **DONE**
3.7. The application shall be designed to handle an expected load of up to 1 million users concurrently. **DONE**

4. **Notifications**
   4.1. The application should implement robust error handling and user-friendly error messages to ensure the user experience is minimally impacted during failures. **DONE**
   4.2. The application shall send timely notifications to customers for important events, such as reminders for workouts, meal plans, or health tips. **ON TRACK**
   4.3. The notification system should be configurable to allow customers to choose their preferred notification methods (e.g., push notifications, emails, SMS). **ON TRACK**
   4.4. The application shall support localization of notifications to cater to customers in different regions and languages. ISSUE: As mentioned earlier in an earlier issue, we have not been able to reach this part of the prototype to initialize something like this.

5. **User Interface**
   5.1. The application shall ensure responsiveness and an intuitive user interface that works seamlessly across various devices and screen sizes. **DONE**
   5.2. The application should be designed to scale horizontally to accommodate increasing user loads without significant changes to the underlying architecture. **DONE**
   5.3. The application shall have an FAQ for customers to answer their own questions. **ON TRACK**
   5.4. The application shall include a helpdesk or support ticketing system to handle user inquiries and issues effectively.
   ISSUE: As a team we realized this was unobtainable and decided to leave this for last and see if we are able to implement this as there were more important issues to deal with.

   5.5. The application shall be built using a modular architecture to facilitate easier updates and maintenance. **DONE**
   5.6. The application shall adhere to accessibility standards to ensure it is usable by people with disabilities.

ISSUE: As a group we also realized that implementing features for the disabled would take up a lot of our time and decided to leave this for last as well and if we were able to get to it we would implement it.

5.7. The user interface shall be visually appealing and customizable to enhance user engagement and satisfaction. **DONE**

6. **6. Security**

6.1. The application shall conduct regular security audits and vulnerability assessments to identify and mitigate potential security risks. **DONE**

6.4. The application shall enforce strong password policies, including complexity requirements and regular password updates. **DONE**

7. **Compatibility**

7.1. The application shall be compatible with the latest versions of major operating systems, including iOS and Android.

ISSUE: Too much of our time is being spent on creating the prototype, we are not going to be able to create a application for the website

7.2. The application shall be tested across a variety of devices and browsers to ensure compatibility and optimal performance. **DONE**

7.3 The application shall support the latest versions of major web browsers, including Chrome, Firefox, Safari, and Edge. **DONE**

7.4 The application shall support major desktop operating systems, including Windows, macOS, and Linux, ensuring functionality across different environments. **DONE**

7.5 The application shall be designed to leverage OS-specific features and optimizations, such as widgets on iOS and Android, to enhance the user experience.

ISSUE: Too much of our time is being spent on creating the prototype, we are not going to be able to create a application for the website

7.6 The application shall provide seamless updates through app stores (Apple App Store, Google Play Store) and support for over-the-air updates to ensure users always have the latest version.

ISSUE: Too much of our time is being spent on creating the prototype, we are not going to be able to create a application for the website

8. **Maintainability**

8.1. The application shall be developed using clean code principles and best practices to facilitate easy maintenance. **DONE**

8.2. The application shall include comprehensive documentation for developers, including API documentation, user guides, and troubleshooting guides. **DONE**

8.3. The application shall use version control for source code management to track changes and facilitate collaboration among developers. **ON TRACK**

9. Localization

9.1. The application shall support multiple languages and regional settings, allowing customers to select their preferred language and units of measurement.

ISSUE: We have been focusing our attention mostly on getting the prototype up and running and focusing on the main issues of the prototype, we hope to implement this by the next milestone.

9.2. The application shall provide localized content, including region-specific health tips, dietary recommendations, and exercise guidelines. **DONE**

9.3. The application shall adjust time zones automatically based on the user's location to ensure the accuracy of logs and reminders. **DONE**

10. Regulatory Compliance

10.1. The application shall comply with all relevant local, national, and international laws and regulations concerning health, fitness, and nutrition data. **DONE**

10.2. The application shall provide clear and accessible terms and conditions, including a privacy policy, which customers must accept before using the app. **ON TRACK**

11. Storage

11.1 The application shall provide scalable storage solutions to accommodate increasing amounts of user data. **DONE**

11.2 The application shall use data compression techniques to optimize storage usage and improve performance. **DONE**

11.3 The application shall ensure data integrity and prevent data loss through robust storage management practices. **DONE**

12. Fault tolerance

12.1 The application shall include mechanisms to detect, report, and recover from software and hardware failures with minimal impact on the user experience.

ISSUE: We were not able to implement an engine to automate the report process for software and hardware failures.

12.2 The application architecture shall support redundancy to ensure service continuity in case of component failures. **DONE**

12.3 The application shall include automated failover processes to switch to backup systems in case of primary system failure.

ISSUE: Creating and finding a way to do this while not using too much storage is a bit difficult, time consuming and also costly. Unsure if we will be able to proceed with this.

10. Detailed List of Contributions

NO.	Member	Contribution	Rating

1	Michelle Nguyen (Team Lead)	● Wrote the product summary for FitNutri. ● Help create and push the nutrition page for the prototype. ● Shared and created the M4 Doc. ● Organized, edited, and assigned tasks for the milestone. ● Collaborated with Ali H to edit the document and go over the code. ● Code review with team 4. ● Worked with Uzair to fix the QA testing. ● Asked family and friends to test the application.	
2	Mitchell Caine	● Lead discussion for what needs to be done for the prototype. ● Set up dates and meetings for the prototype. ● Helped troubleshoot back and front end. ● Worked with the frontend team to connect the backend to the frontend.	
3	Shreejana Bartaula	● In meetings trying to fix the front-end frames. ● Update about page and added css. ● 2 full vegan recipe diet to implement for the front end. ● Updated workout frames content to Ali H. ● Found images for keto diet and workouts. ● QA test plan with Nilo.	
4	Eduardo Enrique Muñoz Alvarez	● Troubleshoot the back and front end of the prototype. ● Implement controllers ● Worked with the frontend team to connect the backend to the frontend. ● Helped get the prototype going. ● Self Check: Adherence to Original Non-Functional Specs	
5	Nilofar Ali	● Wrote the QA test plan with Shree. ● Worked on frontend. ● Added and pushed code to the prototype. ● Collaborated with Shree on doc and prototype work.	
6	Uzair Hamed Mohammed	● Troubleshoot the back and front end of the prototype. ● Implement controllers ● Worked with the frontend team to connect the backend to the frontend. ● Worked with team lead with QA testing. ● Asked family and friends to test the application. ● Worked with code review.	

7	John Collins	● Implement controllers ● Worked with the frontend team to connect the backend to the frontend. ● Helped troubleshoot back and front end.	
8	Ali Hadwan	● Troubleshoot the back and front end of the prototype. ● Helped with the front end. ● Lead the front-end discussions. ● Worked on the front end. ● Worked with the team lead to ensure the doc is ready to submit. ● Code review making sure our code is efficient. ● Helped and collaborated with the team lead to make sure we are in track.	
9	Ali A		