

AN ONLINE ADMISSION APPLICATION AND INTERVIEW IN UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

A thesis

Present to

Information and Communication Technology Department
University of Science and Technology of Hanoi



by

Supervisor : Kieu Quoc Viet
December, 2023

Group's members:

Nguyễn Thái Dương _ BI12-117
Nguyễn Trung Kiên _ BI12-224
Nguyễn Khang Ninh _ BI12-341
Hoàng Quốc Minh Quân _ BI12-363
Nguyễn Đức Phúc Tường _ BI12-485
Trần Ngọc Hải _ BI12-150
Tăng Minh Trí _ BI12-445

TABLE OF CONTENTS

Contents

Acknowledgement	6
Abstract	7
I- Introduction:	8
1- Context & Motivation.....	8
2- Report structure:	9
3- Objectives.....	9
3.1- Desired Features:	9
3.2- Expected outcome:.....	10
II- Requirement Analysis:	11
1- Requirement functional and requirement non-functional:	11
2-Use cases:.....	13
2.1- Diagram.....	13
2.2- User characteristics:.....	13
3- Use Cases and Scenario Description:	14
III- Design and Implementations	19
1. Tools and techniques:	19
2- System Architecture:	21
2.1- Client side:.....	22
2.2- Server side:	22
3- Database Design	24
4.1- Login:	26
4.2- Register	27
4.3-Caculate BMI:	28
4.4- Search and manage exercises/meals	29

4.5- Manage Body Tracking:	30
4.6: Manage CRUD System:	31
IV- Result & Discussion	33
1- Result	33
2- Discussion	33
V- Conclusion & Future work:	34
1- Conclusion:.....	34
2- Future work:	34
VI- Appendix	35

LIST OF FIGURES

Figure 1: Use Case Diagram	13
Figure 2. System Architecture Diagram	21
Figure 3. Front-end Component Design	22
Figure 4. Overall Database Design.....	25
Figure 5. Login Sequence Diagram.....	27
Figure 6. Register Sequence Diagram	28
Figure 7. BMI Calculate Sequence Diagram	29
Figure 8. Search And Manage Exercises/Meals Sequence Diagram	30
Figure 9. Manage Body Tracking Sequence Diagram	31
Figure 10. Manage CRUD System Sequence Diagram	32
Figure 11. Login User Interface	35
Figure 12. Register User Interface	36
Figure 13. BMI/BMR Calculation Interface.....	36
Figure 14. TDEE/Water Intake/ Calories Interface.....	37
Figure 15. User Management Interface	37
Figure 16. Create User Interface	38
Figure 17. Roles List Interface.....	38
Figure 18. Roles Permission Interface.....	39
Figure 19. Admin Analytics Interface	39
Figure 20. Profile Settings Interface.....	40
Figure 21. User Analytics Interface	40
Figure 22. Meal Search and Tracking Interface.....	41
Figure 23. Exercise Search Interface.....	41
Figure 24. Exercise Research Interface	42
Figure 25. Exercise Result Interface	42
Figure 26. Exercise Detail Interface	43
Figure 27. Exercise Suggestion Interface	43

LIST OF TABLES

Table 1. Register Basic Flow	14
Table 2. Register Alternative Flow	15
Table 3. Login Basic Flow	15
Table 4. Login Alternative Flow	16
Table 5. BMI Calculation Basic Flow	16
Table 6. Manage Meal Tracking Basic Flow	17
Table 7. Manage Exercise Tracking Basic Flow	17
Table 8. Manage Body Tracking Basic Flow	18

Acknowledgement

Without the invaluable support and assistance extended to us throughout this period, the completion of this project would not have been possible.

Our heartfelt appreciation goes to the University of Science and Technology of Hanoi (USTH), along with all the professors, with a special acknowledgement to Mr. Kieu Quoc Viet, who served as our project supervisor.

Mr. Kieu Quoc Viet has played a crucial role in the successful execution of our project and thesis. His guidance provided us with the conceptual framework for the fitness tracking application and insightful analysis of the challenges we faced. Without his assistance, our project and thesis would lack the depth and detail achieved under his supervision.

While recognizing our imperfections, this project has played a pivotal role in our learning journey, providing valuable insights into the development of a health and fitness website.

Abstract

Nowadays, we can see that there are more and more people having problems with their health due to their lack of experience in this aspect. They can easily become overweight, underweight or easily suffering from many medical conditions. As we observe the current trends, it becomes evident that a significant portion of the population is grappling with health concerns, emphasizing the urgent need for education and awareness campaigns to address the root causes of these issues.

Therefore, we are thinking about a solution that can help people to have more information about their health, find more information and methods to improve their own well-being. That leads us to the idea of making a gym website. We name our site “FitnessRevive” as your target is to help people to get their shape back. Moreover, our website will encourage people to focus more on exercises. We will give them suggestions about diet, based on their BMI.

I- Introduction:

1- Context & Motivation

In recent years, the healthcare landscape has witnessed a paradigm shift with the integration of technology into patient care, and one significant development is the emergence of healthcare apps. These applications serve as invaluable tools in enhancing the overall healthcare experience for both patients and healthcare providers. The context and motivation for using a healthcare app are multifaceted, addressing various aspects of modern healthcare.

Healthcare apps address various challenges in traditional healthcare models by offering convenience and accessibility. They provide easy access to medical information, virtual consultations, and appointment scheduling, reducing the need for physical visits. These apps support personalized health management, allowing users to track and manage their health proactively with features like personalized health plans and medication reminders. Real-time monitoring capabilities enable continuous tracking of vital health metrics, facilitating early detection and intervention for certain medical conditions. Healthcare apps also contribute to health education and awareness by offering educational resources and interactive content. They enhance communication between patients and providers through secure messaging and telemedicine services, fostering a stronger patient-provider relationship. Additionally, these apps promote data-driven decision-making by collecting and analyzing user data, leading to better-informed decisions and personalized treatment plans in the healthcare industry.

In conclusion, the context and motivation for using a healthcare app revolve around improving accessibility, personalizing health management, enabling real-time monitoring, promoting health education, facilitating communication, and embracing a data-driven

approach. As technology continues to advance, healthcare apps play a pivotal role in shaping the future of patient-centered and efficient healthcare delivery.

2- Report structure:

Here is the structure of our report:

- **Section I: Introduction:** we will describe the problem in people's health and our solution to this and the objectives of our project.
- **Section II: Requirement Analysis:** we will delve into scenarios, use cases, object models, and dynamic models, providing a comprehensive functional specification, including navigational paths that depict the sequence of screens.
- **Section III: Design and Implementation:** we will enumerate the tools and techniques employed in the project, elucidate the reasons for their selection, and provide detailed implementations of use cases.
- **Section IV: Results:** we will encompass a comprehensive list of the implemented functionalities within the system.
- **Section V: Future Work & Conclusion:** we will describe our work in the long-term future.

3- Objectives

3.1- Desired Features:

Feature 1: Body-vitals Calculation: helps people check their body condition, BMI , calories needed per day and water intake and set their goal.

Feature 2: Authentication

Sub Feature 2.1: Register: allow users and admins to create a new account

Sub Feature 2.2: Login: allow users and admins to get access to the system

Sub Feature 2.3: Change password: allows users and admins to change their password for security means.

Feature 3: Comprehensive Health Management

Sub Feature 3.1: Create diet chart: allows user to decide their most suitable diet plan

Sub Feature 3.2: Manage meal tracking: keep up with their meal to calculate their consumption.

Sub Feature 3.3: Manage workout plan: help users to keep up with their workout progress

Sub Feature 3.4: Manage exercises tracking

Sub Feature 3.5: Manage user information

Sub Feature 3.6: Manage body tracking: check if the BMI reaches what the user wants

Feature 4: Manage user Full application: allows the admin to manage all of user application

3.2- Expected outcome:

Our system sets the target to help members of our site to keep up the progress in their challenges of improving their state of health. The specific goals include:

- The application should have all the features mentioned in Desired Features
- The web page should be able to run in multiple browsers such as Chrome, Mozilla Firefox, etc.
- Develop the backend engine to manage and control the progress

II- Requirement Analysis:

1- Requirement functional and requirement non-functional:

- Functional:

Our fitness tracking platform is designed to capture essential metrics related to users' health, such as their Body Mass Index (BMI), calories consumption and exercises. Users benefit from easy access to real-time data for each metric, facilitating a visual representation of their fitness journey through interactive graphs. The platform supports use cases such as managing workout plans, enabling users to create and track personalized exercise routines. Additionally, users can efficiently manage meal tracking and exercise monitoring, leveraging a comprehensive library of exercises. The platform extends customization options, allowing users to personalize the interface, choose preferred languages, and manage body tracking for a tailored and user-friendly fitness experience.

- Non-functional:

- Query Optimization:

We optimize our SQL queries to ensure they are efficient and make use of indexes. Analyze and profile our queries to identify any bottlenecks. We also use Sequelize ORM you provided us with performance optimization.

- Connection Pooling:

We use connection pooling to manage database connections efficiently. Instead of opening a new database connection for each user request, a connection pool allows us to reuse existing connections, reducing the overhead of creating and closing connections.

- Optimize Server Configuration:

We adjust MySQL server configurations based on our system requirements. Parameters such as max_connections, innodb_buffer_pool_size, and others can significantly impact performance.

my.ini

2-Use cases:

2.1- Diagram

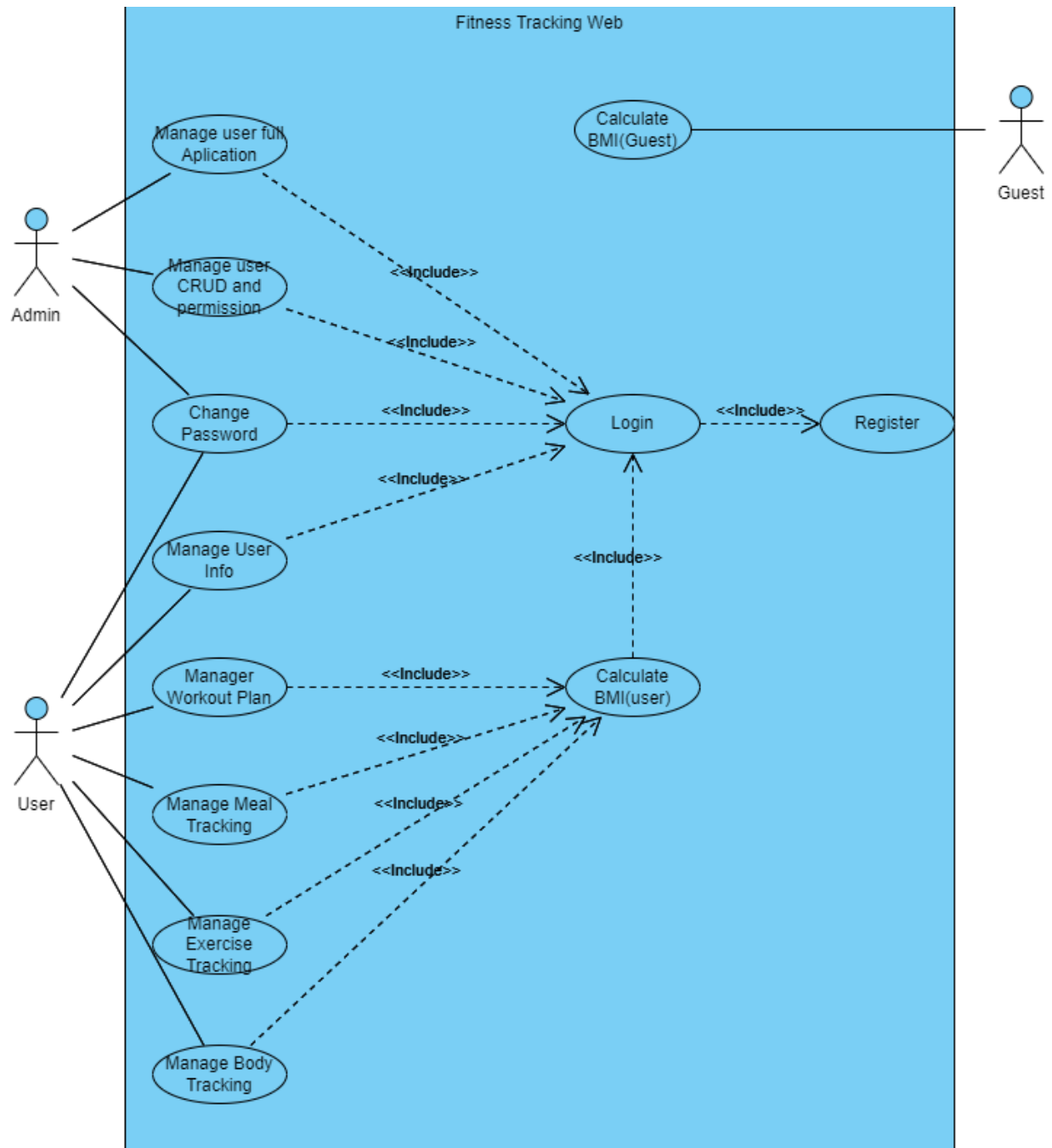


Figure 1: Use Case Diagram

2.2- User characteristics:

- Guest: people who only go to the website to check their BMI.

- User: people who go to the website not only checking their BMI, they also want to find the suitable diet and exercise to improve their health.
- Admin: people who has the total control of the whole website
- Dev (SuperAdmin): give permission and assign roles to admins and users.

3- Use Cases and Scenario Description:

3.1- Register

3.1.1 - Brief description:

This use case describes how a person registers to the FitnessRevive web system

3.1.2- Flow of events

Table 1. Register Basic Flow

Actor Action	System Action	Data
The user wants to register to the system	The system requests the user to fill in their email, password and their basic information	
The user enters the requested information	.The system checks if the entered email is unique from the table “user” in Database. If unique, the system saves the user information and assigns a unique id to the user.	Email and Password

3.1.3- Alternative Flow:

- The email is in use:

Table 2. Register Alternative Flow

Actor Action	System Action	Data
A person enters the username and password	The system validates the username and password. If the username or password is invalid, the system displays an error message and requests the user to re-enters username and password.	Email and password

3.2- Login:

3.2.1 - Brief description:

This use case describes how a person logs into the web system.

3.2.2- Flow of events:

Table 3. Login Basic Flow

Actor Action	System Action	Data
The person wants to enter the system	The system asks the person to fill in email and password	Email and password
The person fills in the requested system	The system validates the email, password and type entered from the user table in the Database. If valid, it will log the person in.	

3.2.3- Alternative flow:

- Invalid login parameters:

Table 4. Login Alternative Flow

Actor Action	System Action	Data
The person enters the username and password	The system validates the username and password. If the username or password is invalid, the system displays an error message and asks the person to retype the username and password again.	Email and password

3.3- BMI Calculation:

3.3.1- Brief Description:

This use case describes how to calculate the Body Mass Index (BMI) of a person.

3.3.2- Flow of events:

Table 5. BMI Calculation Basic Flow

Actor Action	System Action	Data
The user wants to calculate the body-vitals	The system asks the person to fill in their height and weight in order to calculate the body-vital	Height, weight, gender, age, activity level, tdee, water intake, ...
The user fills in the requested information	The system receives the data user request and perform the calculation then return back to display for user	
The user saves the calculated body vitals log to the database	The system saves to the database and display on the screen	Height, weight, gender, age, activity level, tdee, water intake, ...
The user set the goal	The system calculates the energy that user need per day then save to the database then display back on the interface	Calories_goal (energy)

3.4- Manage meal tracking:

3.4.1- Brief description:

This use case describes how to manage the meal tracking of a user.

3.4.2- Flow of events

Table 6. Manage Meal Tracking Basic Flow

Actor Action	System Action	Data
The person searches for food nutritional content.	The system returns the nutritional content that the user needs.	
The person saves the food they want for their meal.	The system saves food users want to add to their meal to the database and displays the calories needed left based on the calories needed per day in the body-vitals calculation.	Food and food nutrition content that users choose and save.

3.5- Manage exercise tracking:

This use case describes how to track the exercise.

3.5.2- Flow of events

Table 7. Manage Exercise Tracking Basic Flow

Actor Action	System Action	Data
The user search for exercise content	The system returns the exercise that is suitable for the user's target	
The user saves the exercise they want	The system saves exercise users want to add to the database and displays the calories needed left based on the calories needed per day in the body-vitals calculation.	The exercise that the user wants to save

3.6- Manage body tracking

3.6.1- Brief description:

This use case describes how to track the data of the body.

3.6.2- Flow of events

Table 8. Manage Body Tracking Basic Flow

Actor Action	System Action	Data
The user wants to see all the results after managing meals and exercise.	The system displays all the exercise picked, food managed and data about body-vitals from the database.	

III- Design and Implementations

1. Tools and techniques:

1.1. React

React is a JavaScript library for building user interfaces, particularly single-page applications where user interactions dynamically update the content. Developed and maintained by Facebook, React follows a component-based architecture, enabling developers to create modular and reusable UI components, facilitating efficient development and maintenance of complex web applications. Known for its declarative and efficient approach, React ensures seamless synchronization between the application's state and the user interface, providing a robust foundation for building interactive and scalable web applications.

1.2. Tailwind CSS

Tailwind CSS is a utility-first CSS framework that streamlines the process of styling web applications by providing a comprehensive set of pre-designed, low-level utility classes. Instead of relying on predefined components, Tailwind encourages developers to compose styles directly within the HTML, offering a highly customizable and flexible approach to styling. Its atomic design philosophy and efficient utility classes enable rapid development while maintaining a clean and maintainable codebase.

1.3- NodeJS

Node.js is an open-source, server-side runtime environment that allows developers to run JavaScript code outside of a web browser. It is built on the V8 JavaScript runtime engine, which is the same engine that powers the Google Chrome browser. It is commonly used for building scalable network applications, real-time applications (such as chat applications and

online gaming), and server-side applications where high concurrency is required. Popular frameworks built on top of Node.js include Express.js for web applications and Socket.IO for real-time applications.

1.4- MySQL

	SQL	NoSQL
Type of database	Relational Database	Non-relational database
Schema	Pre-defined schema	Dynamic schema
Database Categories	Table-based database	Document-based databases, Key-value stores, graph stores, wide column stores
Complex queries	Good for complex queries	Not suitable for complex queries
Hierarchical Data Storage	Not the best fit	More suitable than SQL
Scalability	Vertically Scalable	Horizontally Scalable

In a fitness tracking website, MySQL serves as a robust relational database management system, facilitating the organized storage and retrieval of user data. It plays a crucial role in managing information such as user profiles, workout routines, and nutrition plans. With MySQL's structured approach, the website can efficiently handle complex relationships between different data entities, ensuring seamless navigation and data integrity. This relational database management system enhances the overall performance and reliability of the fitness tracking website, contributing to a secure and well-organized user experience.

1.5- Postman

Postman is a collaboration platform for API development, it simplifies each step of building an API and streamline collaboration. In this project, Postman is used to create collections of

integration tests to ensure the API is working as expected. This helps to save hours of manually testing.

2- System Architecture:

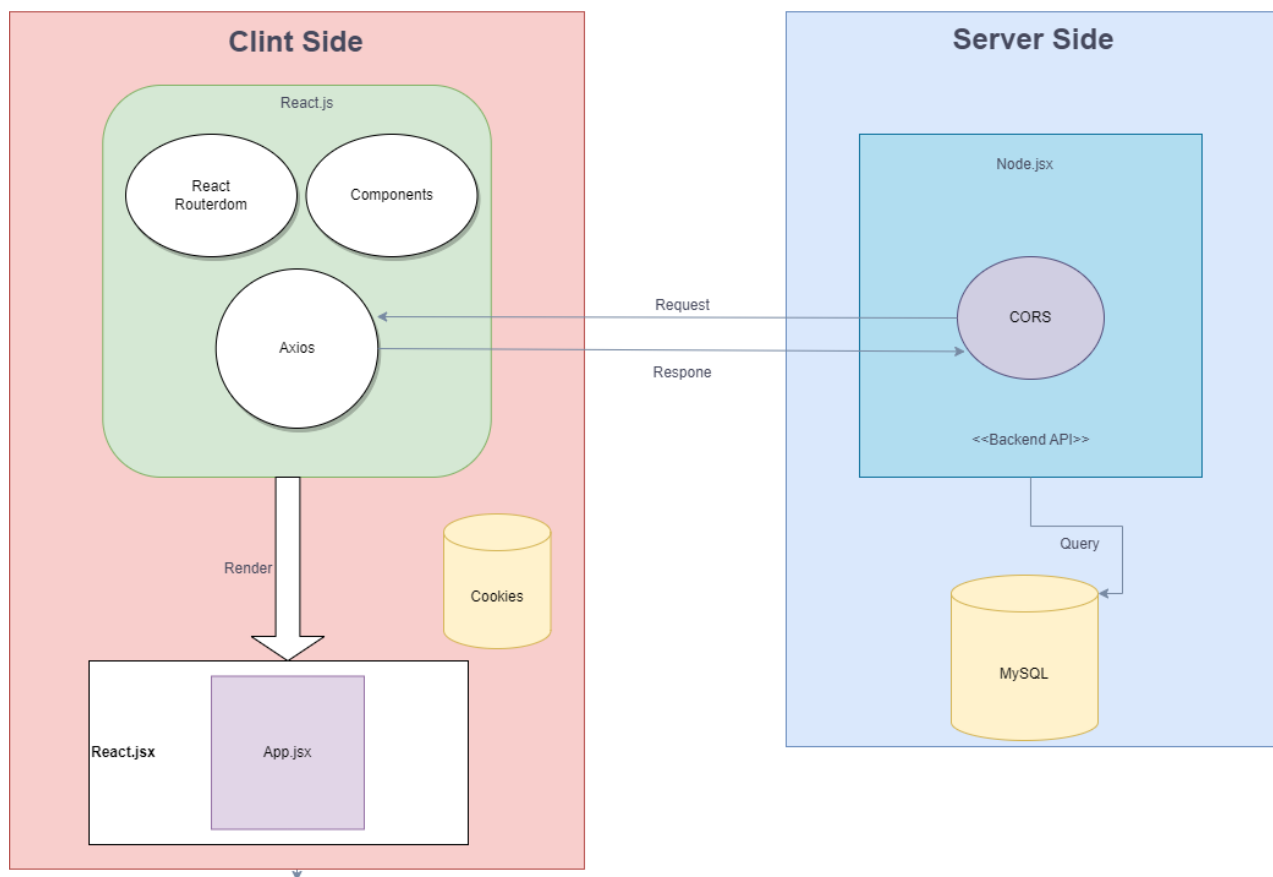


Figure 2. System Architecture Diagram

This Online Admission Application and Interview System is a full side project with a completely developed Client side (Front-end) and Server side (Back-end):

- The Server side receives the request from the Client-side, processes and gets data from the Database then returns a response.
- The Client side interacts with the user, retrieves the user request, sends it to Server side, receives the response from the Server then renders the response data for display.

2.1- Client side:

Client side of the application includes three main parts: React component, React Router and Axios:

- React Router is the official routing library for React.js. It seamlessly integrates with React.js core to simplify the development of Single Page Applications using React.js. This enables us to navigate between different views or pages within our application.
- Axios is imported into components in order to send requests and handle responses from HTTP requests between client and server.
- Components are reusable React instances with a name. Many components are included in the web front-end. These components are arranged as in the below figure. Sharing data between components is divided into two types: pass data from parent component to child component (using prop) and from child component to parent component (by emitting event).

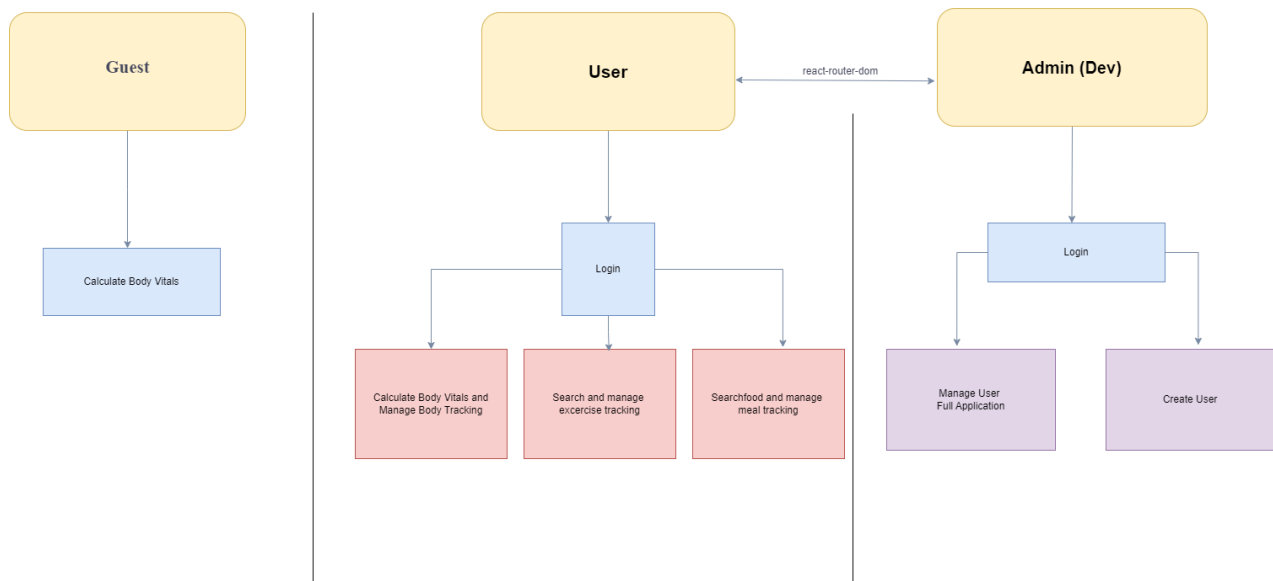


Figure 3. Front-end Component Design

2.2- Server side:

This is a Client side rendering web application, the Back-end takes the responsibility of creating a secure connection between the web browser and system's database. This side include two main parts:

- MySQL: A database management system to manage all the system data.

- NodeJs for API processing (including receiving requests, authenticating requests, communicating with databases, and returning responses in JSON form). Structure of an API contains:

- Define action.
- Request verification with token.
- Query from Database.
- Response:
 - Return a response data in JSON format if successful.
 - Return an error message if there is an error.

3- Database Design

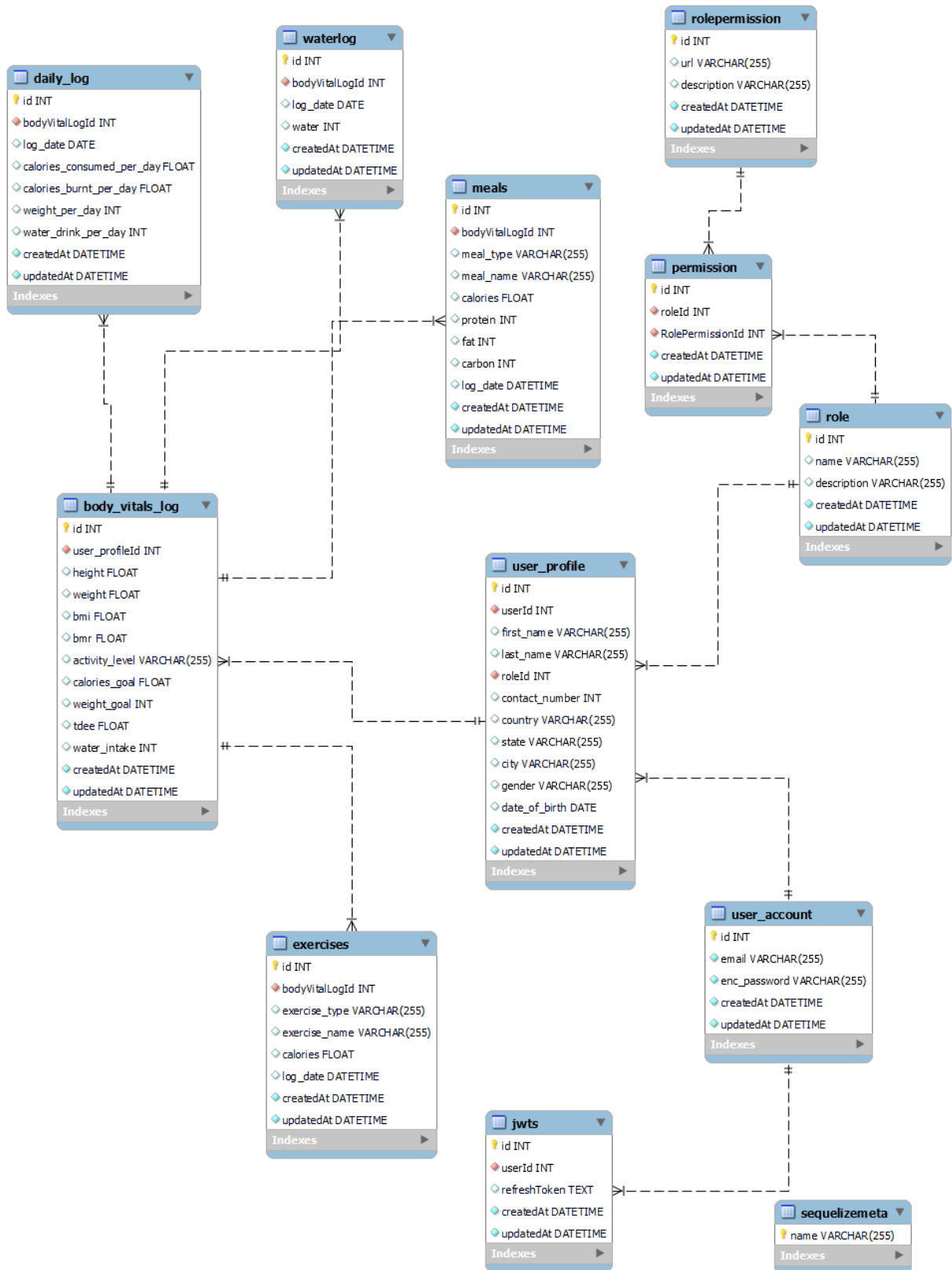


Figure 4. Overall Database Design

- The "user_account" table uniquely identifies users with "id" and stores their email. Encrypted passwords are secured in "enc_password".
- The "user_profile" table uniquely identifies users with "id" and captures personal and geographical details. The "roleId" signifies the user's role.
- The "meals" table holds details about meals, including unique IDs, types, names, and caloric content.
- The "jwt" table manages JWTs, uniquely identified by the "id" field. "userId" links to the associated user, and "refreshToken" stores the refresh token for authentication.
- The `exercises` table stores exercise details, including type, name, calories burned, and timestamps for creation and updates. It serves as a concise repository for managing exercise-related data.
- The `rolepermission` table contains id, URL, description for permissions linked to roles. It plays a key role in defining and organizing role-specific access rights in a system.
- The `role` table stores role information with id, name, description for creation and updates. It serves as a central component for managing and categorizing user roles in a system.
- The `permission` table is vital for access control, assigning permissions to roles. It includes id, roleId, rolePermissionId, and tracking of role-based permissions in a database.
- The `daily_log` table records daily activities and health metrics, including meals, exercise, body vitals, and calorie-related data. It features fields for tracking intake, expenditure, goals, and timestamps for creation and updates. This table provides a concise overview of daily health and wellness information.
- The `body_vitals_log` table records body vitals, including height, weight, BMI, BMR, and activity level for creation and updates. It is associated with user profiles

through user_profile_id, providing a concise way to monitor and analyze individuals' physical health data.

- The "role" table defines system roles, each identified by a unique "id" and characterized by a name and brief description.
- The "Waterlog" table records water consumption data, associated with body vital information. Each entry is uniquely identified by the "id." The "log_date" field captures the date, and "water" stores the amount consumed.

4- Use Cases Implementation

4.1- Login:

We have the case that a person wants to enter the system. User fills in the credentials to login. The system validates the information and creates 2 tokens if success. Then, it sets the cookie with refresh token, save access token to a state and return the homepage. Otherwise, the system will return the error message and display it on the interface.

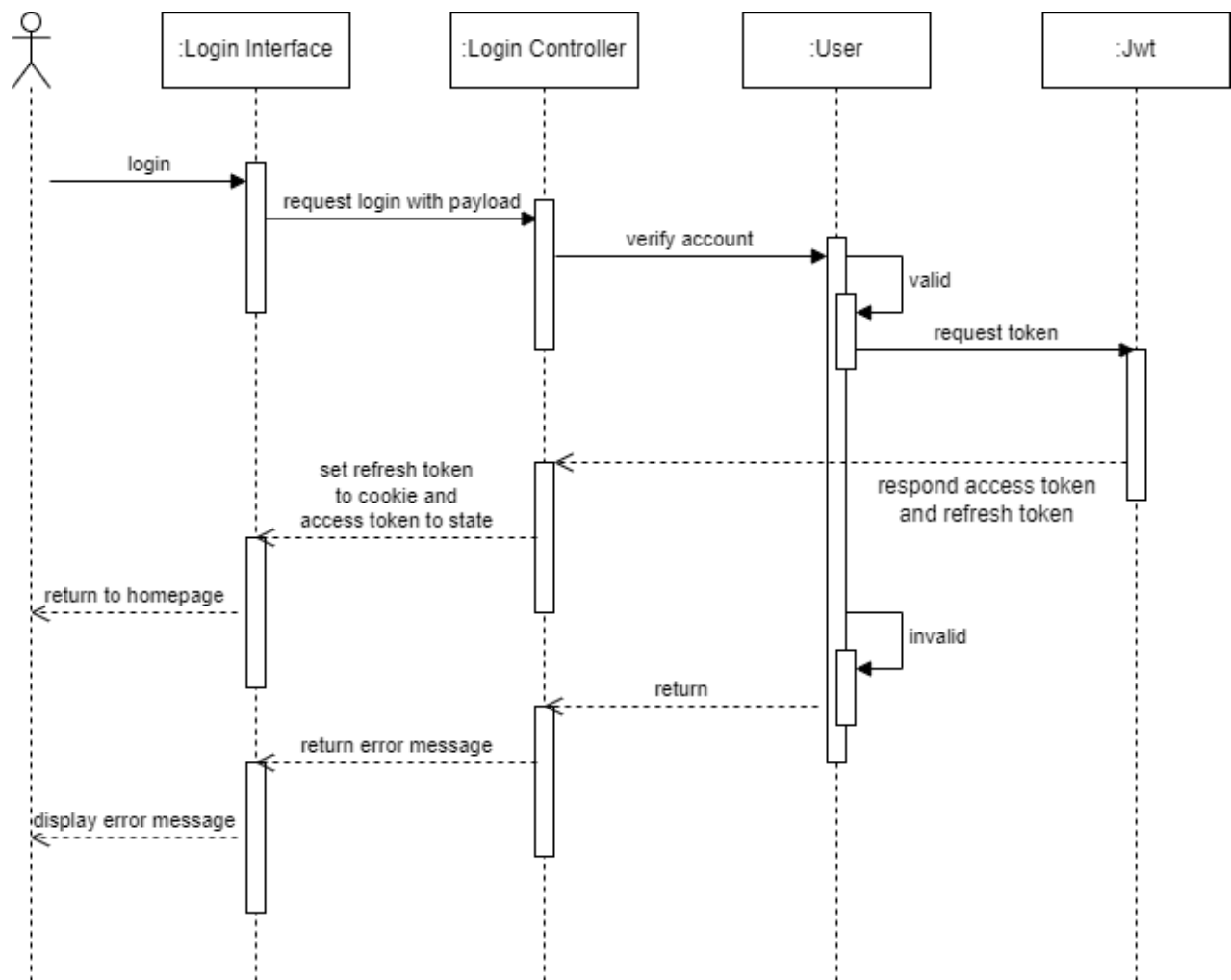


Figure 5. Login Sequence Diagram

4.2- Register

We have the case that a person wants to register into the system. User fills the require information to register. After the user enters the required information, the system checks if the entered email is unique from the table “user_account” in Database. If unique, the system encrypts the password and saves the user information, assigns a unique id to the user and displays a success message. Otherwise, the system displays an error message and asks the user to reenter the information.

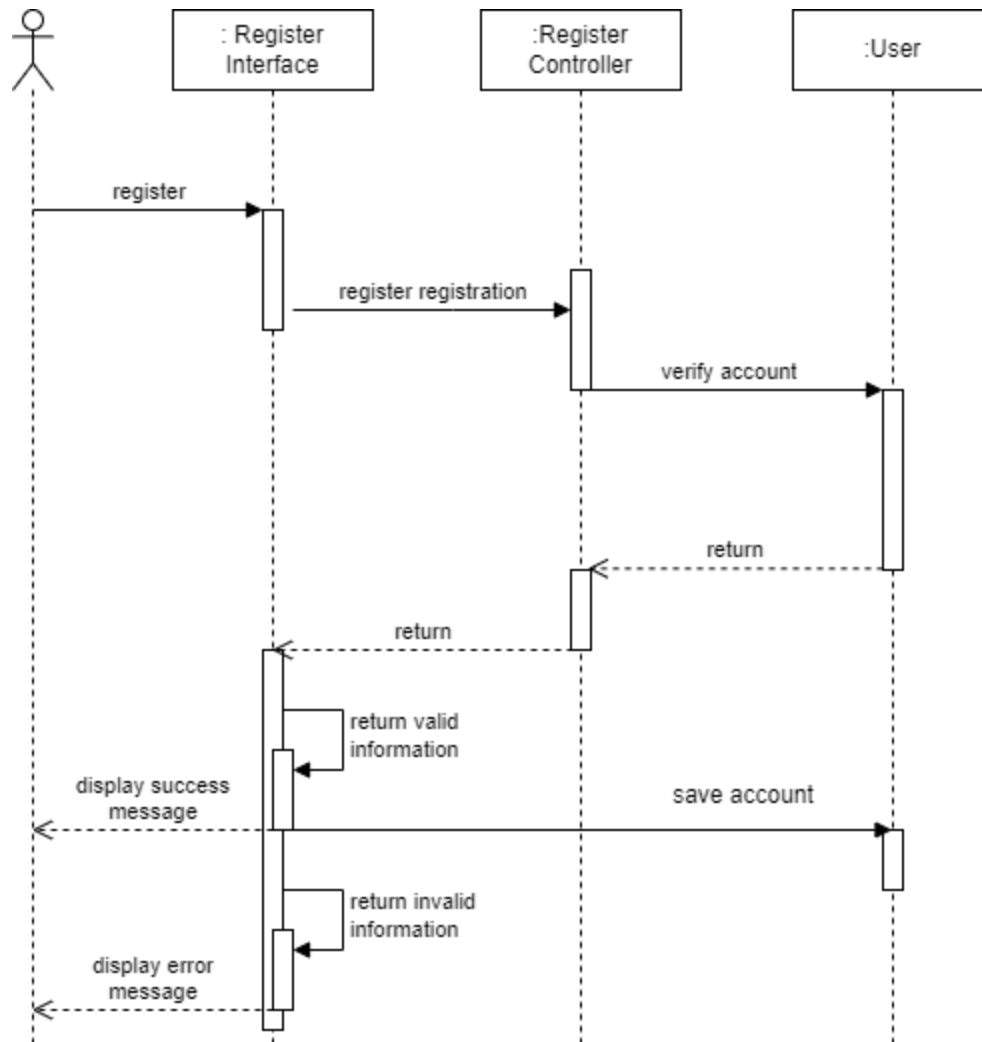


Figure 6. Register Sequence Diagram

4.3-Caculate BMI:

This case describes how users calculate Body vitals such as BMI, BMR and store the result in the Database. After successfully login, user fills in the required credentials for calculating BMI, then the system sends this information to a calculate service. The calculate service performs some calculations and then sends the results back to the interface. The interface displays the results and as long as the user request to save the data, it also saves them to the database.

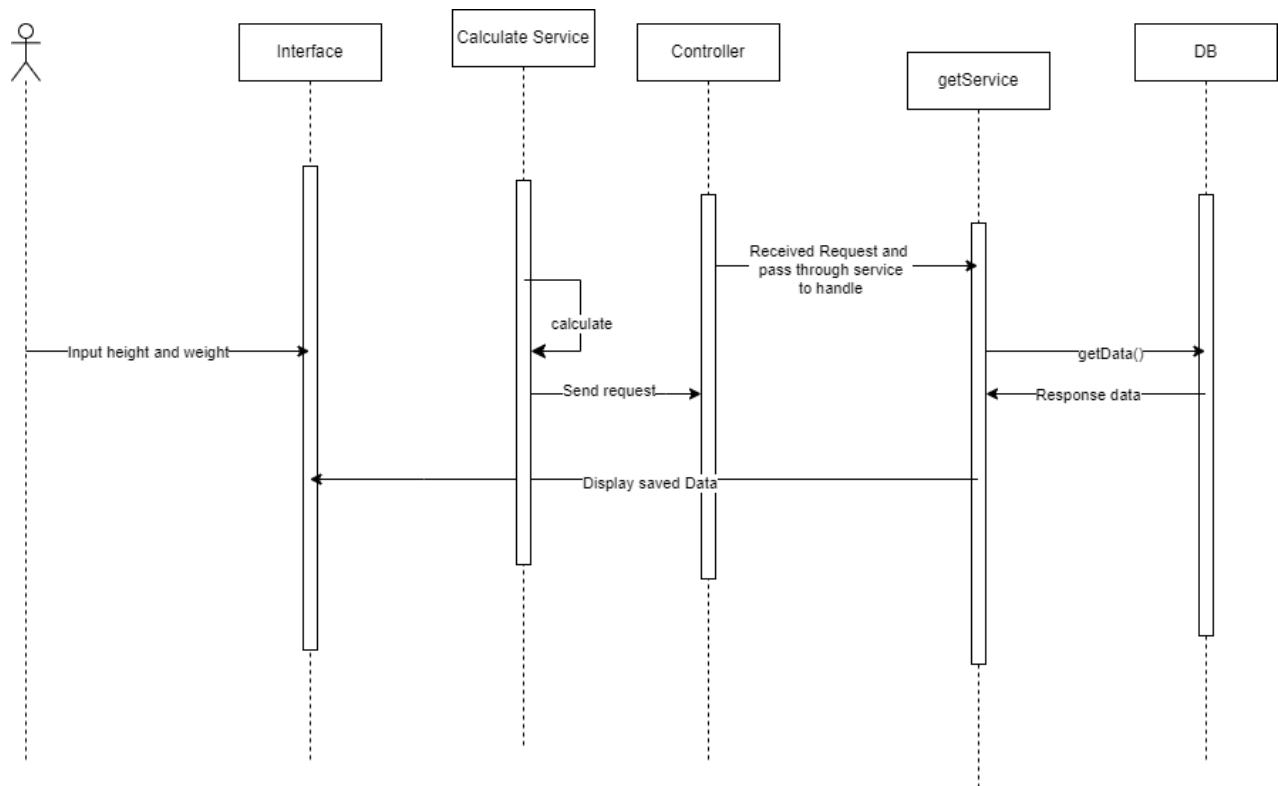


Figure 7. BMI Calculate Sequence Diagram

4.4- Search and manage exercises/meals

This case describes how the user searches for the details of exercises/meals and manages them. Firstly, the user searches for the name of the exercises/meals. This is sent to the SearchAPI, which then sends it to the SearchDB. The SearchDB looks for any exercises or meals that match the user's query. Once the results are found, they are sent back to the SearchAPI, which then passes them to the Controller. The Controller then formats the results and displays them to the user. If the user wants to save any of the results, they can click on the "Save" button. This sends the results to the SaveService, which then saves them to the DB. Once the results are saved, the user can see them by clicking on the "Saved Data" tab. This displays a list of all the saved exercises and meals.

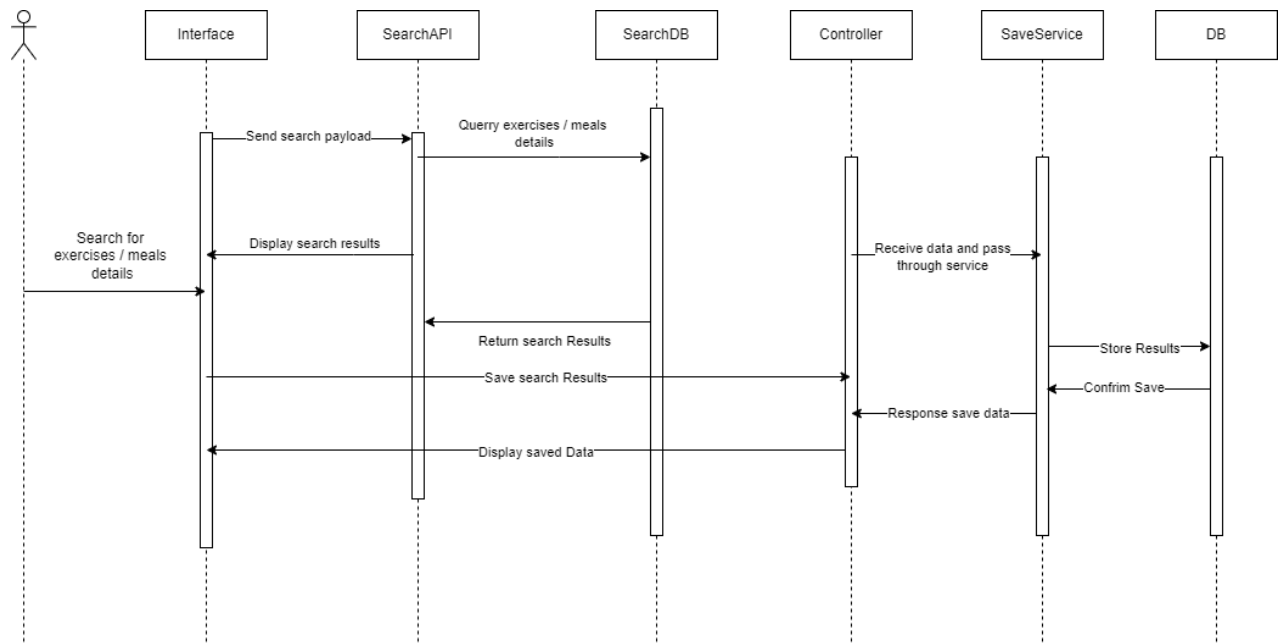


Figure 8. Search And Manage Exercises/Meals Sequence Diagram

4.5- Manage Body Tracking:

This case describes how the user wants to see and manage their body vitals information. The interface sends requests to the server, then the server will perform users' requests (save, edit, update, display) the return data to display for the user on the user interface.

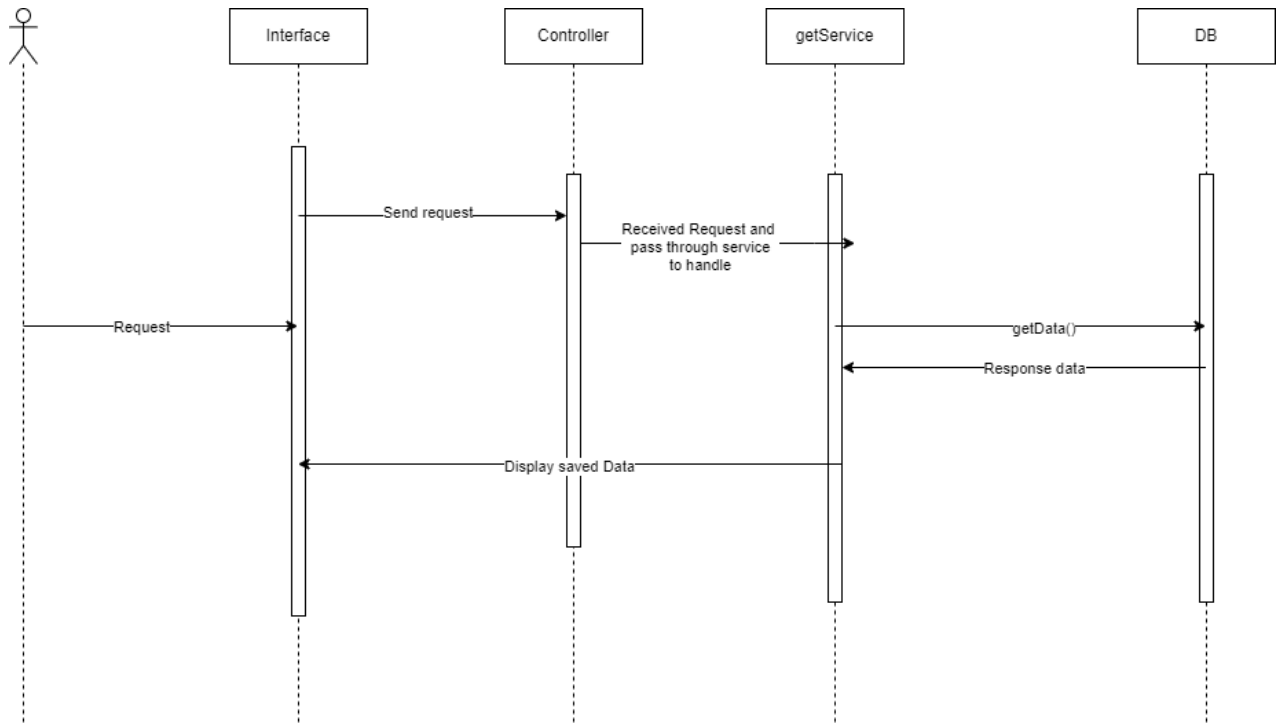


Figure 9. Manage Body Tracking Sequence Diagram

4.6: Manage CRUD System:

This case describes how the admin manages user's CRUD and permission. The first step is for the user to send a request to the interface. The interface then passes the request to the controller. The controller performs the request and sends the results back to the interface. If the request is to create a user, the controller will call the `CreateUserService`. The `CreateUserService` will create a new user in the database and return the user's ID to the controller. If the request is to edit permissions, the controller will call the `EditPermissionsService`. The `EditPermissionsService` will create, update or delete the permissions for the specified role in the database and return the updated permissions to the controller. If the request is to assign permissions to a role, the controller will call the `AssignPermissionsToRoleService`. The `AssignPermissionsToRoleService` will assign the specified permissions to the specified role in the database and return the updated role to the controller. Once the controller has received the results from the service, it will send them back to the interface. The interface will then display the results to the user.

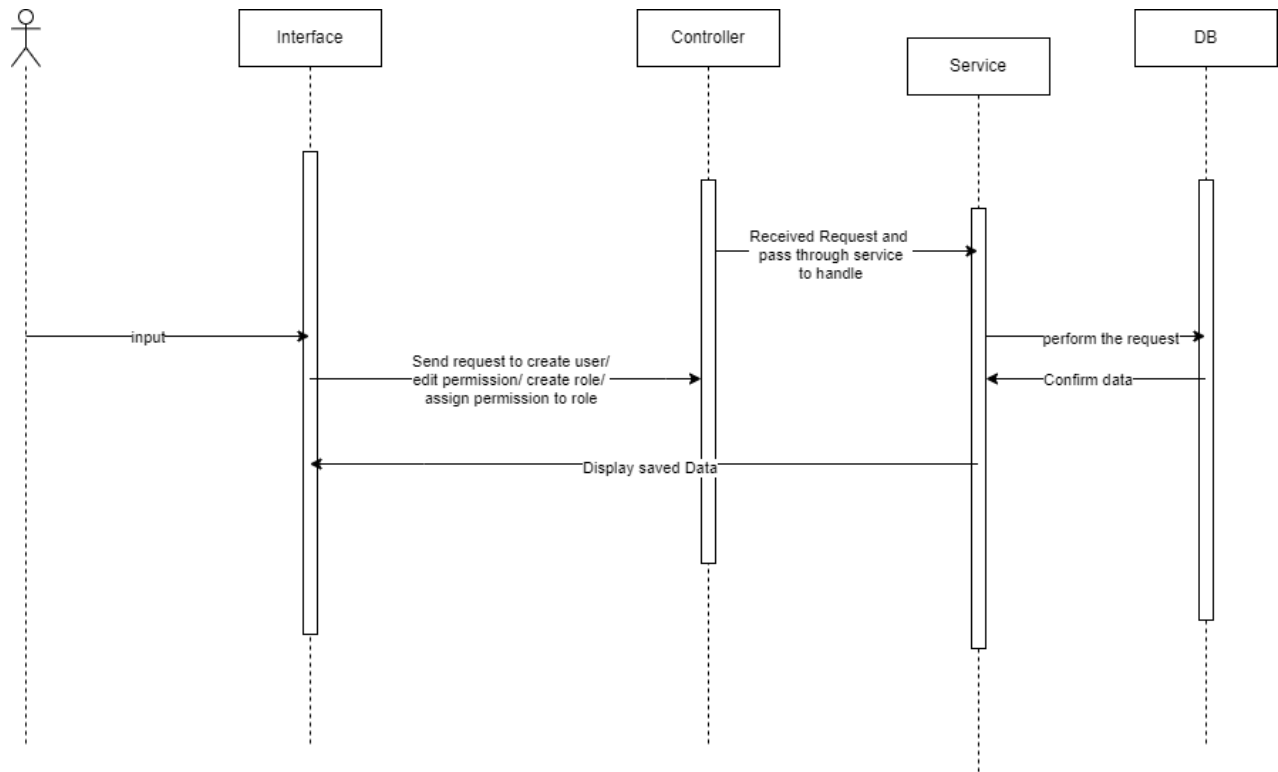


Figure 10. Manage CRUD System Sequence Diagram

IV- Result & Discussion

1- Result

In this system, we have completed the implementation of these functions:

- Register and Login: users can create a new account and get access to the system
- Create diet chart: users can create a suitable diet for their ideal target weight
- Manage the daily logs: help the users keep up with their progress.

2- Discussion

Although we have completed the implementation, the project still has some existing problems:

- A security system is necessary.
- There is a need of a support system so that the admins can help the user in case they have problems with any functions on the application.

V- Conclusion & Future work:

1- Conclusion:

To sum up, the purposes and objectives of the fitness tracking website are achieved. By providing a BMI calculation and suitable plans, we intend that our fitness tracking web application will lend a helping hand to people who want to improve their health.

2- Future work:

To improve the web application in the future, we will have some updates for it:

- Improve the security of the website. We could use double-factor authentication to make it more secure.
- Create an online support system so that users can interact with admins. For example, we should have a hotline contact so that users can contact us if they see any problems appearing.
- Enhance the performance of the application so it can be more user-friendly. We will create a manual to help new user be more familiar with the web application

VI- Appendix

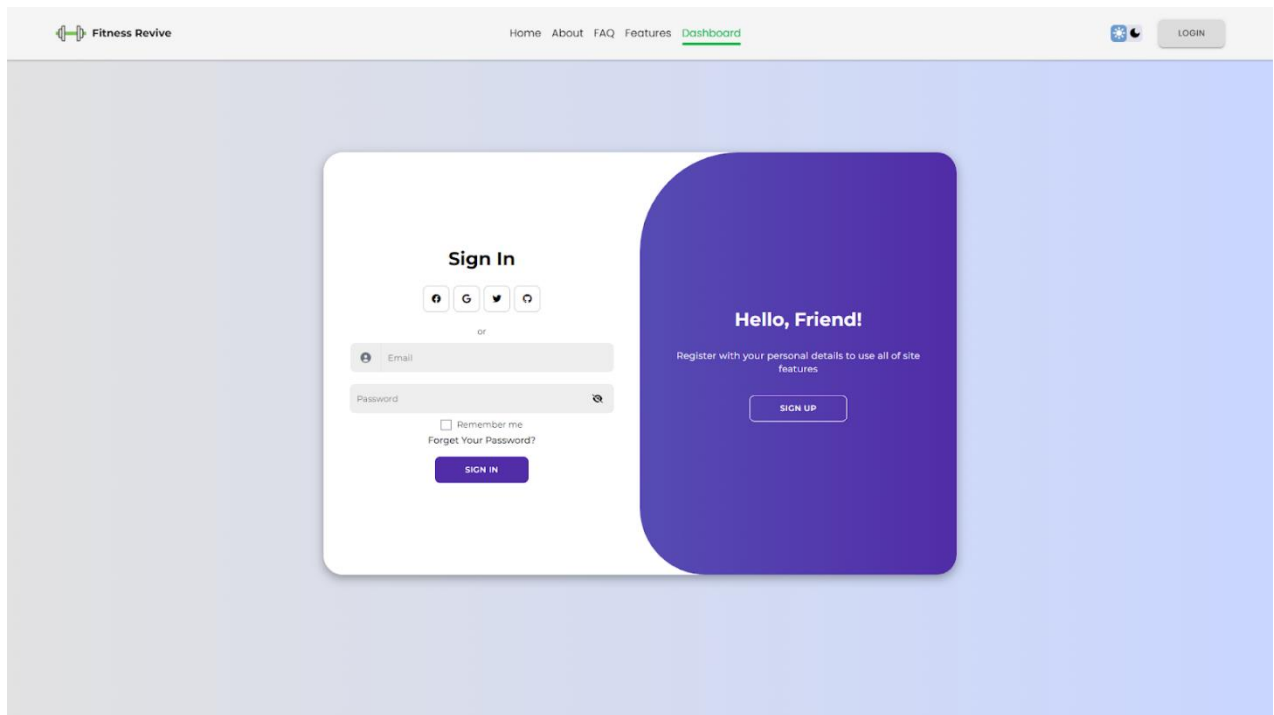


Figure 11. Login User Interface

Welcome Back!
Enter your personal details to use all of site features.
[SIGN IN](#)

Create Account

or use your Email for registration

First Name

Last Name

Email

New Password

Confirm Password

☐ I have read and agreed to the terms

[SIGN UP](#)

Figure 12. Register User Interface

FILL DETAILS AND SAVE YOUR DATA

Your BMI / BMR

Body mass index is a value derived from the mass and height of a person. The BMI is defined as the body mass divided by the square of the body height, and is expressed in units of kg/m^2 , resulting from mass in kilograms and height in metres.

Basal Metabolic Rate represents the amount of energy, measured in calories, that the body requires at rest to maintain normal bodily functions such as breathing, circulation, and cell production. In other words, it is the number of calories your body needs to perform basic physiological functions while at complete rest.

Height / cm

Weight / kg

Age

Male

Sedentary (little or no exercise)

[Calculate](#)

Chart

BMI	WEIGHT STATUS
Below 18.5	Underweight
18.5 - 24.9	Healthy
25.0 - 29.9	Overweight
30.0 - and Above	Obese

Your TDEE

TDEE stands for **Total Daily Energy Expenditure**. It represents the total number of calories that an individual burns in a day, taking into account all activities, including basal metabolic rate (BMR), physical activity, and the thermic effect of food. The TDEE is a crucial factor in determining an individual's caloric needs, especially for those who are interested in managing their weight. To calculate TDEE, you typically start with the BMR, which represents the number of calories your body needs at rest to maintain basic physiological functions. Then, you factor in the calories burned through daily activities and exercise.

Figure 13. BMI/BMR Calculation Interface

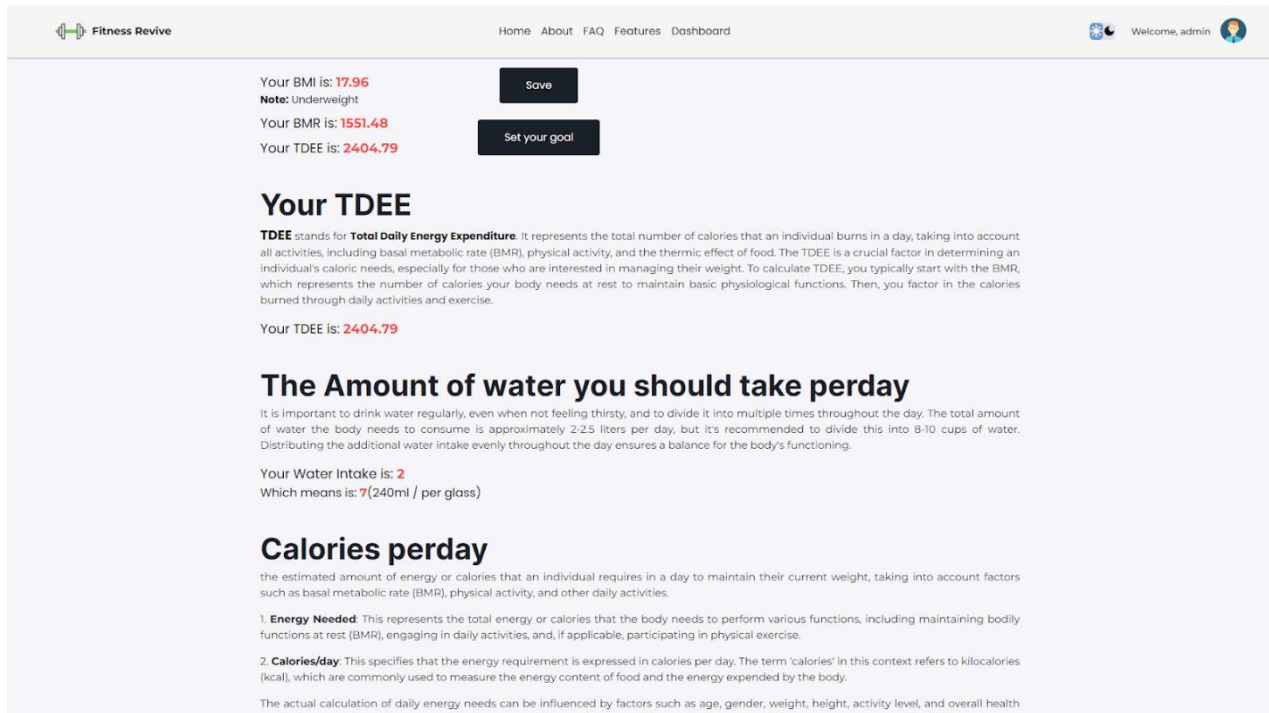


Figure 14. TDEE/Water Intake/ Calories Interface

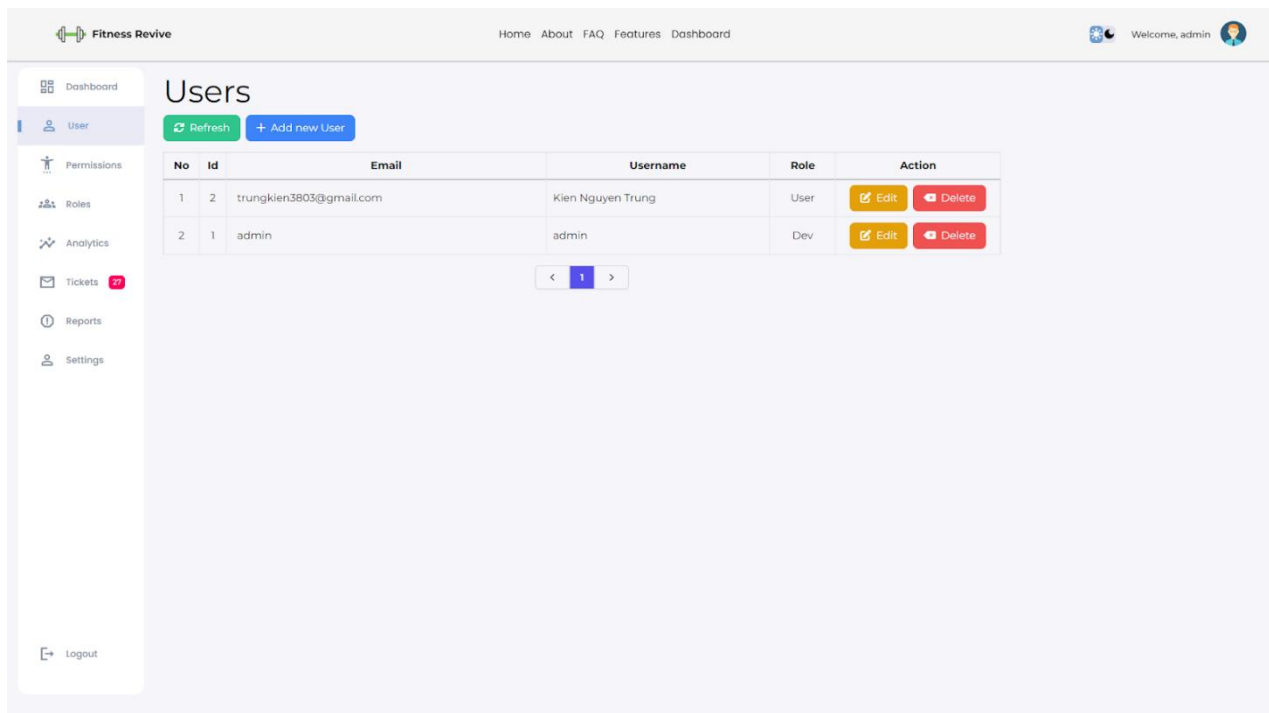


Figure 15. User Management Interface

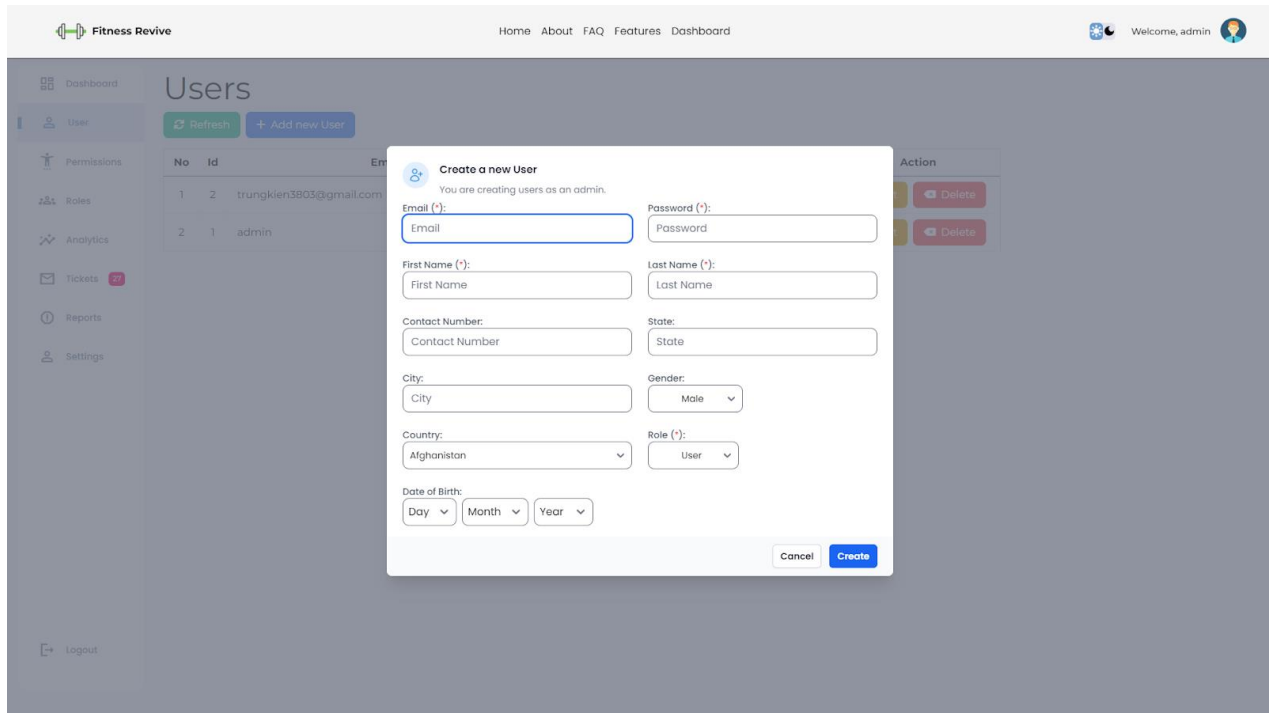


Figure 16. Create User Interface

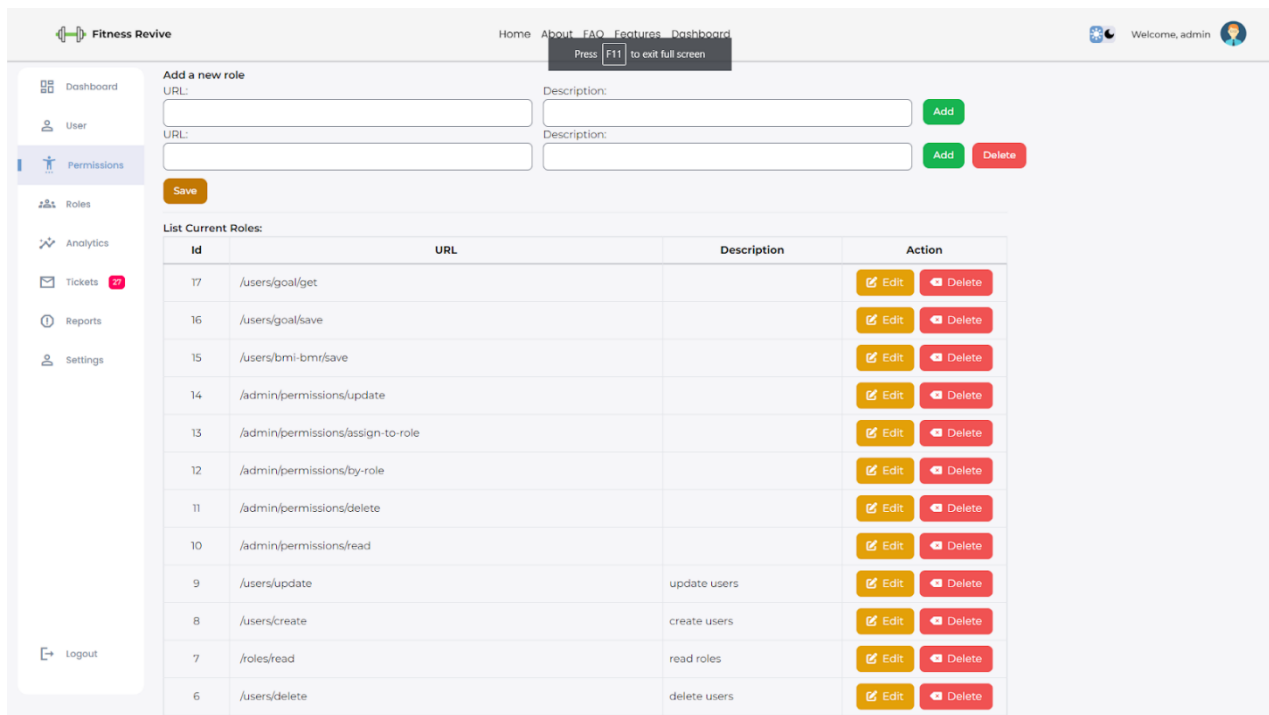


Figure 17. Roles List Interface

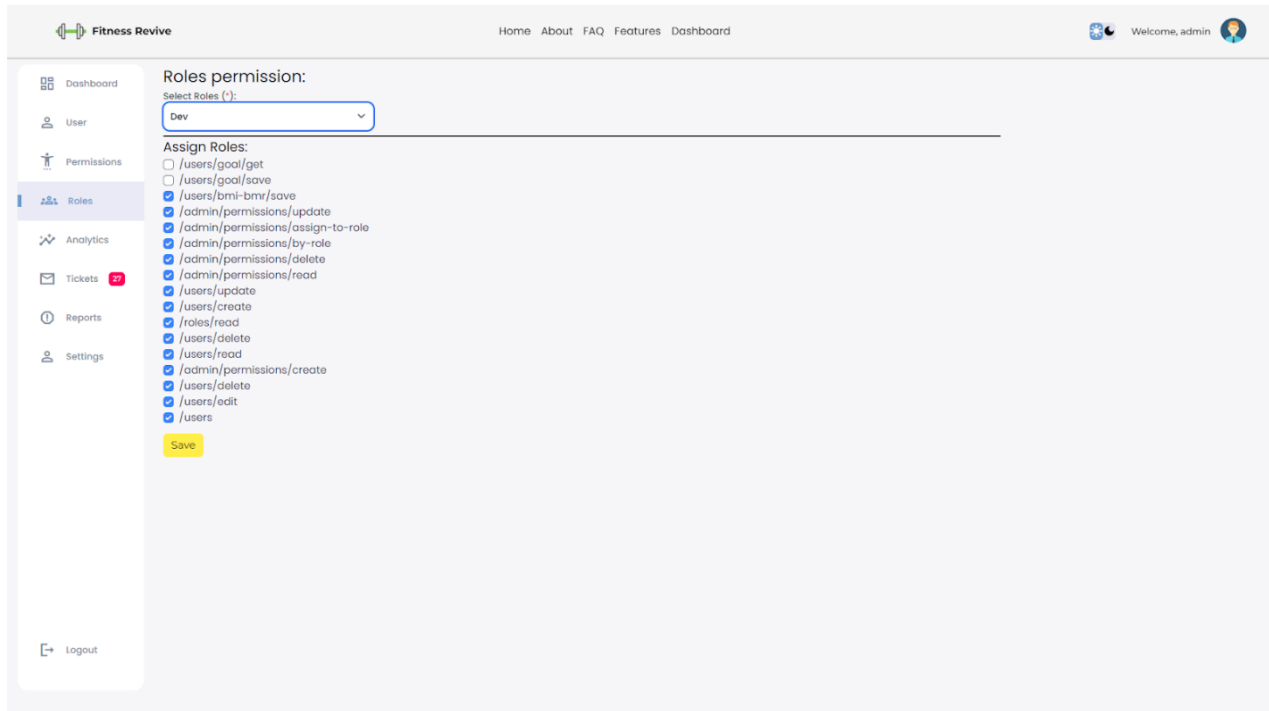


Figure 18. Roles Permission Interface

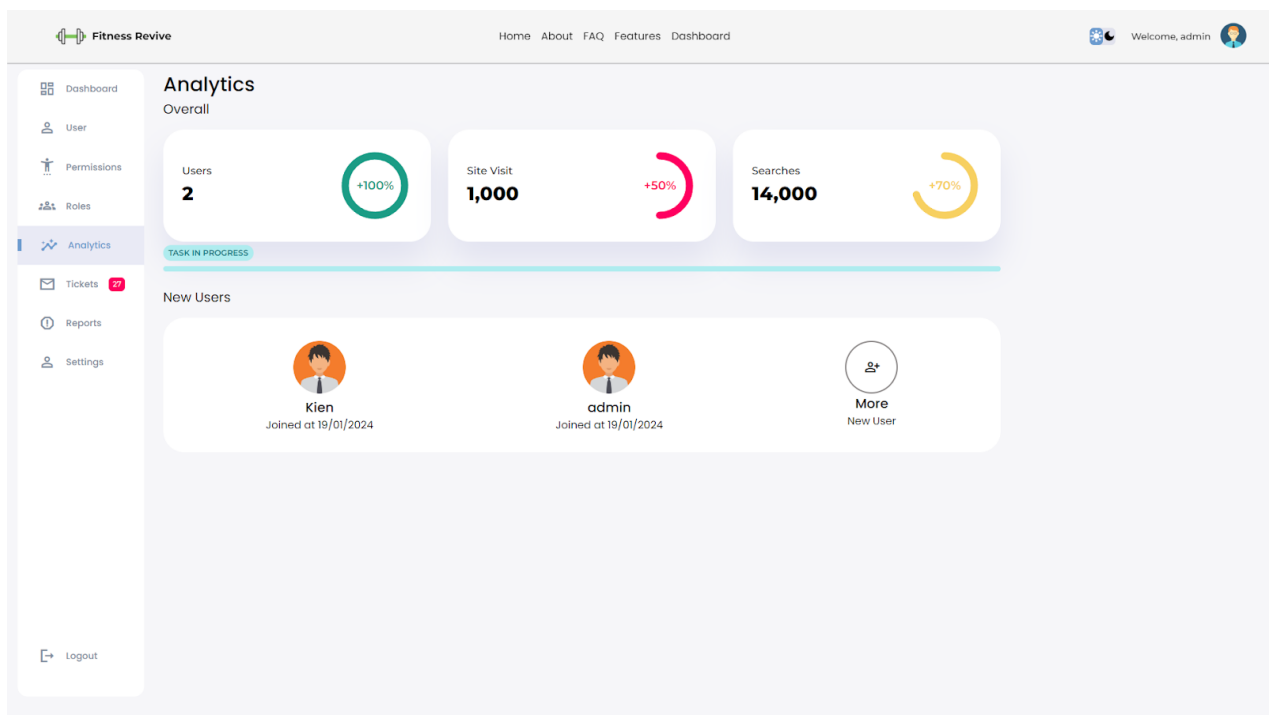


Figure 19. Admin Analytics Interface

[Home](#)
[About](#)
[FAQ](#)
[Features](#)
[Dashboard](#)

Welcome, Nguyen Trung Kien

Dashboard

Your meal

Roles

Roles

Analytics

Tickets 27

Reports

Settings

Logout

Profile

This information will be displayed publicly so be careful what you share.

Username

workation.com/ janesmith

About

Write a few sentences about yourself.

Photo

Change

Cover photo

Upload a file or drag and drop
PNG, JPG, GIF up to 10MB

Personal Information

Use a permanent address where you can receive mail.

First name

Last name

Figure 20. Profile Settings Interface



Figure 21. User Analytics Interface

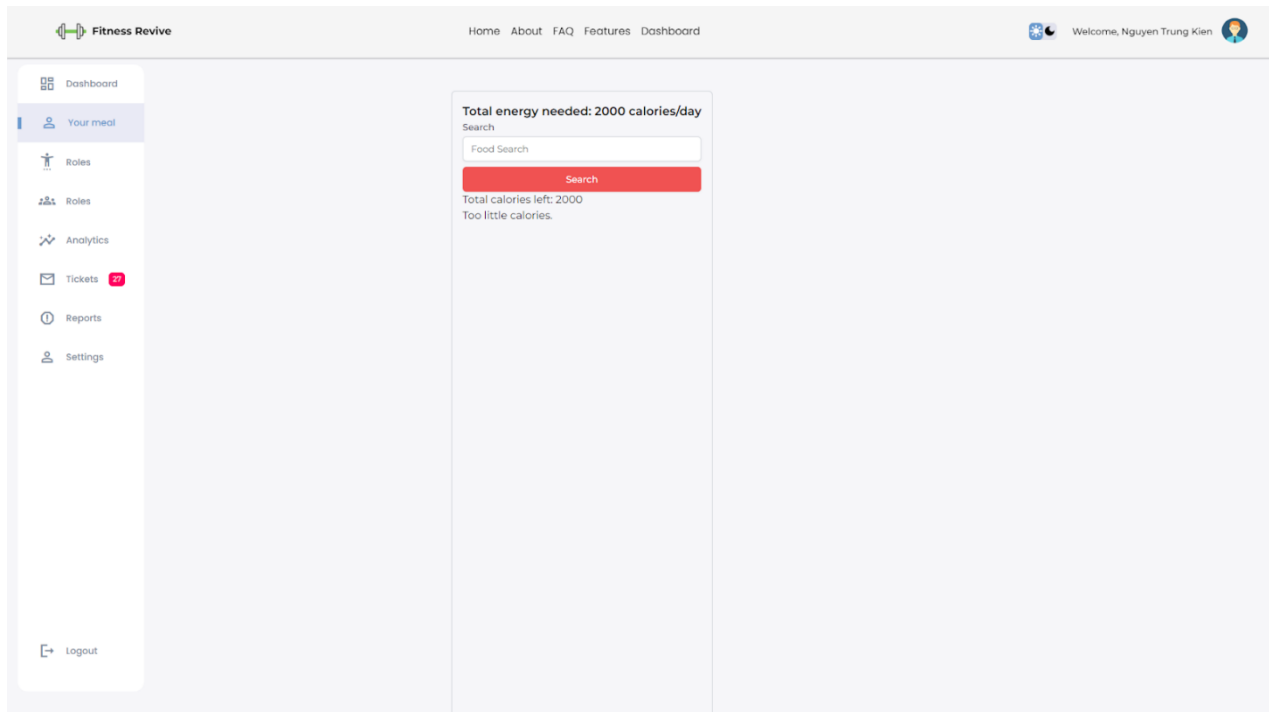


Figure 22. Meal Search and Tracking Interface

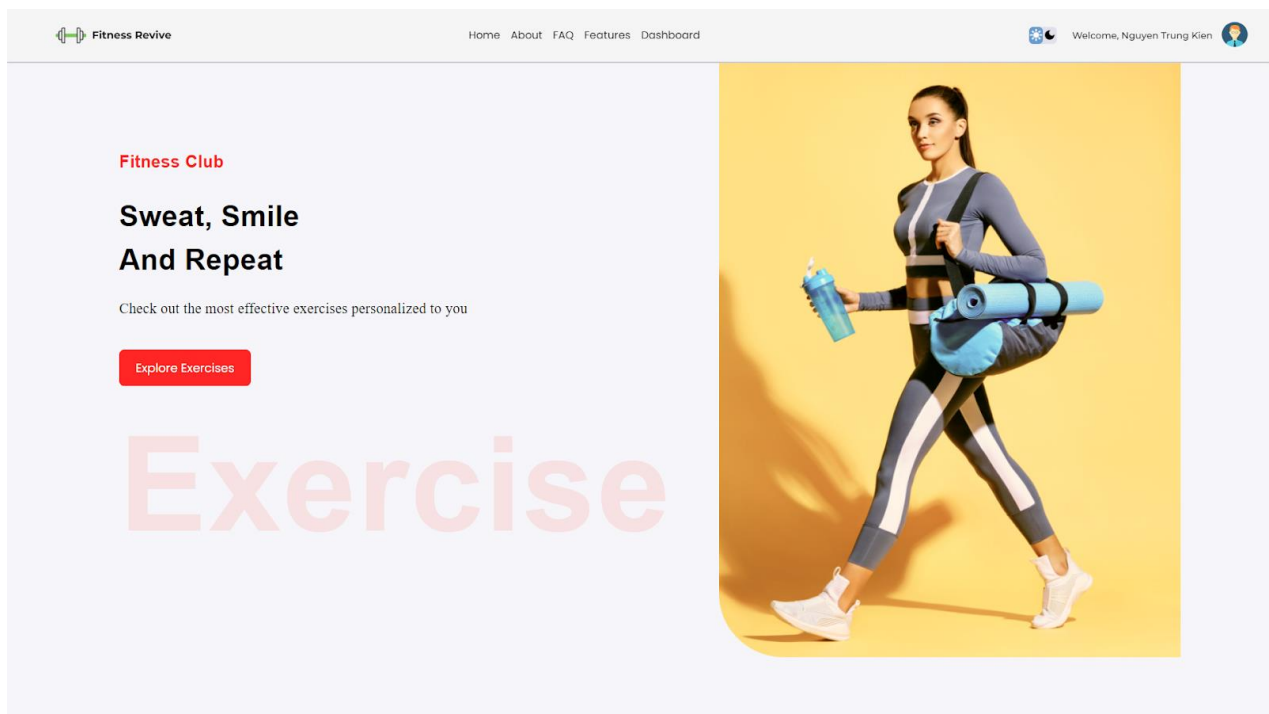


Figure 23. Exercise Search Interface

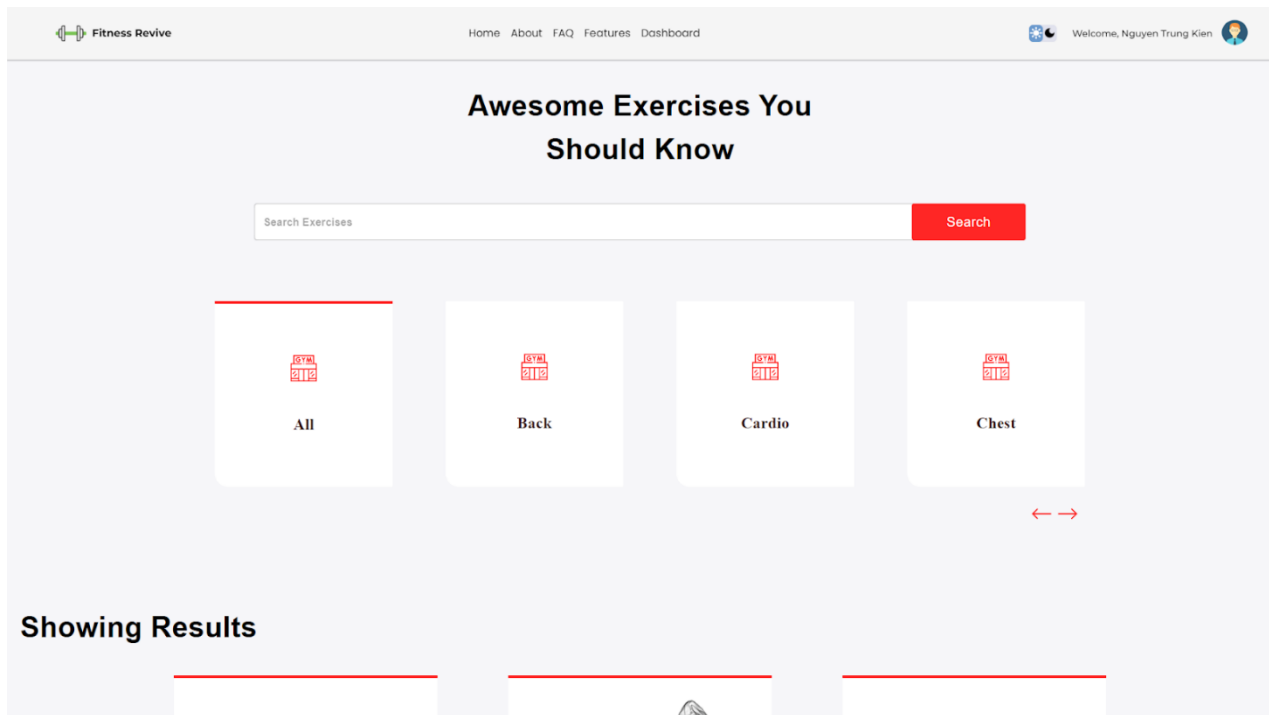


Figure 24. Exercise Research Interface

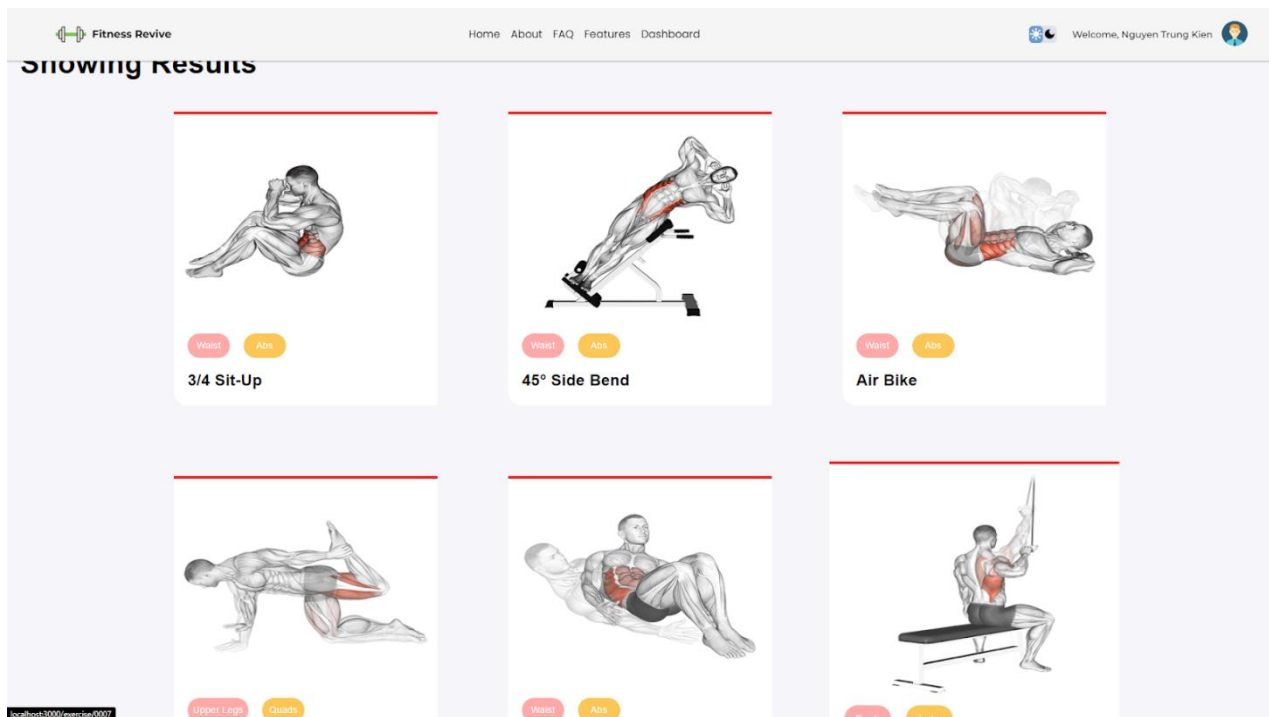


Figure 25. Exercise Result Interface

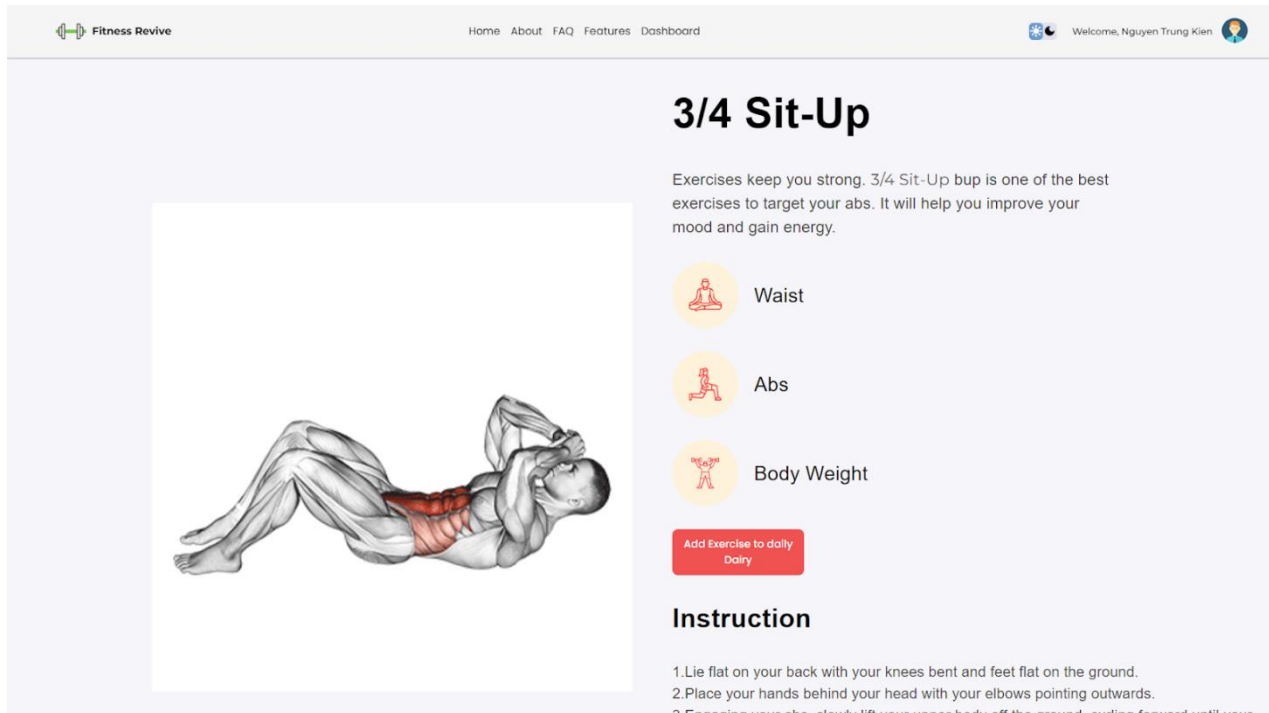


Figure 26. Exercise Detail Interface

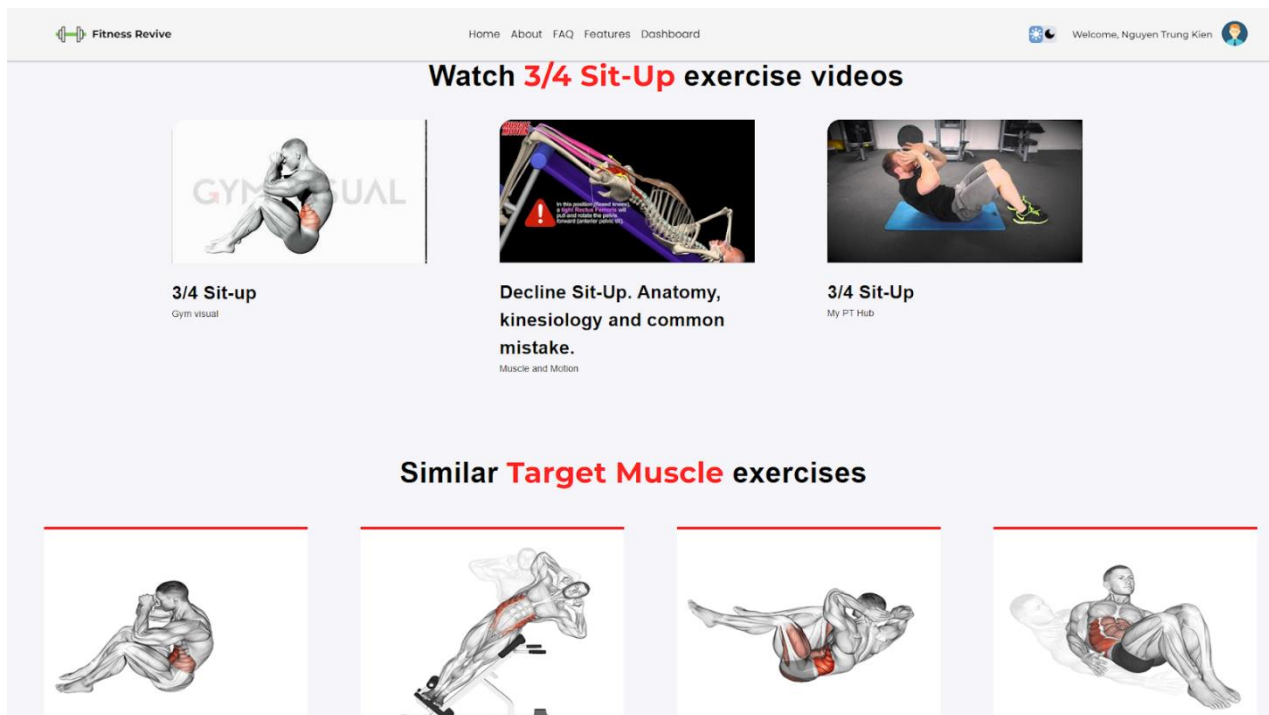


Figure 27. Exercise Suggestion Interface

