Sri Lanka Institute of Information Technology

Programming Applications and Frameworks

(IT3030)

Continuous Assignment

BATCH ID: Year 03 Semester 02

GROUP ID:

IT_JUN_WE_47

Members Details

Name	Registration Number	Email Address	Student Group
Gangabadage U.D	IT21157400	IT21157400@my.sliit.lk	Y3.S2.WE.IT.02.01
Kumara H.M.I.M	IT21183904	IT21183904@my.sliit.lk	Y3.S2.WE.IT.02.01
Dissanayake D.M.W.B.T	IT21164194	IT21164194@my.sliit.lk	Y3.S2.WE.IT.02.01

Contents

MEMBER DETAILS AND WORKLOAD DISTRIBUTION

Name	Registration Number	Contribution
Dissanayake D.M.W.B. T	IT21164194	• Workout plan Management • Workout Status Management
Gangabadage U. D	IT21157400	• Meal Plan Management • User Register and login • User Profile
Kumara H.M.I.M	IT21183904	• Post Creation • Post comment. • Post sharing. • Post like

PROJECT INTRODUCTION

Our project focuses on developing a social networking platform tailored specifically for fitness enthusiasts, providing a space for them to connect, share experiences, and support each other on their fitness journeys. Users will have the ability to follow other members, like and comment on posts, and receive notifications, while also being able to create their own posts featuring workout routines, progress updates, and dietary insights. Our platform will be accessible as a web application, ensuring security with OAuth-based authentication to protect user information.

We have chosen a technology stack that includes Spring Boot for the backend, react for the frontend, MongoDB as the database, and Spring Security for authentication and authorization, guaranteeing a robust and secure platform for our users.

Here are the key functionalities of our fitness-focused social media web application,

Post Creation:
Users can create posts with uploaded media files, including photos or videos, and add descriptions. We ensure data validation for correct formatting and adherence to length restrictions.

Media Upload:
Users can upload up to three photos or videos showcasing fitness activities. Videos are limited to 30 seconds, and we enable direct acceptance of media files from user devices.

Workout plan Management:
For workout plans, users can generate, modify, and remove plans using templates for routines, exercises, sets, and repetitions. Customization options align plans with changing fitness objectives.

Workout Status Update:
Users can share ongoing workout progress updates, using templates for workout metrics and concise descriptions of achievements.

Meal Plan Management:
Meal planning features allow users to create, modify, and remove plans, inputting recipes, nutritional details, and serving sizes. Plans can be organized based on dietary preferences like vegan or keto options.

Data validation and Error Handling:
We validate input data for consistency and integrity, with error-handling mechanisms for informative user messages.

User Authentication and Authorization:
Authentication verifies user identities securely, while authorization rules control endpoint access based on user roles and permissions.

FUNCTIONAL REQUIREMENTS

- **For REST API:**

1. User registration and login using OAuth-based authentication for secure access.
2. Create, update, and delete posts that include workout photos/videos and descriptions.
3. Follow and unfollow other fitness enthusiasts to stay connected and build a supportive community.
4. Like and dislike posts to express appreciation or preference.
5. Comment on posts and could edit or delete your own comments.
6. Receive notifications for comments and likes on your posts to stay engaged with your audience.
7. Receive personalized recommendations for fitness enthusiasts to follow based on shared interests or activities.
8. View your own and other users' posts and profiles to explore different fitness journeys and insights.

- **For Client Web Application:**

1. User-friendly interface with OAuth-based authentication for seamless registration and login processes.
2. Create, update, and delete posts showcasing workout routines, progress updates, and dietary insights, with the option to include images and descriptions.
3. Follow or unfollow other users to build connections and support each other in fitness goals.
4. Like or dislike posts to express interest or preference.
5. Comment on posts and have the flexibility to edit or delete your own comments as needed.
6. Receive real-time notifications for comments and likes on your posts to stay engaged and responsive.
7. Get personalized recommendations for fitness enthusiasts to follow, enhancing community interaction and discovery.
8. Explore your own and other users' posts and profiles to gain insights, inspiration, and motivation from diverse fitness journeys.

NON-FUNCTIONAL REQUIREMENTS

- **For REST API:**

1. Ensure the API can scale horizontally by adding more resources or instances and implement load balancing mechanisms for evenly distributing incoming requests across multiple server instances to accommodate increased loads.
2. Security measures for the API include using JWT or OAuth for user authentication, employing HTTPS encryption to protect data during transmission, and implementing input validation and output encoding to prevent common cyberattacks like SQL injection or Cross-Site Scripting (XSS).
3. The API should respond to requests quickly, usually within milliseconds, and manage multiple requests efficiently to ensure consistent response times even during high loads and different traffic patterns.
4. The API is always available by reducing downtime, quickly recovering from failures, and handling unexpected errors without causing system disruptions.

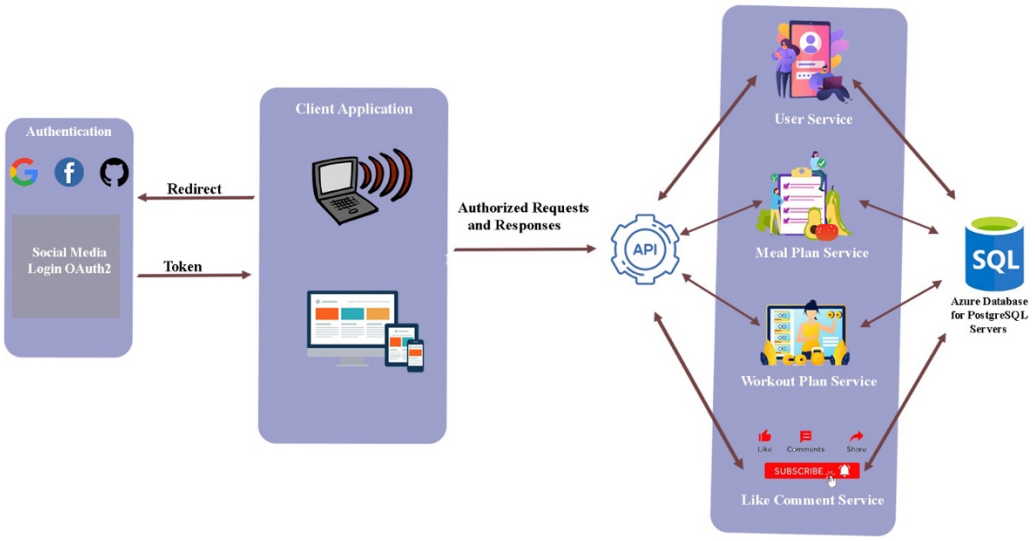
5. The API needs to work well with different platforms, programming languages, and frameworks to make it easy for users to access and use, and regular testing should be done to make sure it works smoothly with different client applications and systems.

6. The API follows industry standards and protocols to work smoothly with other systems and services and allow easy integration with third-party APIs and services using clear interfaces and protocols.

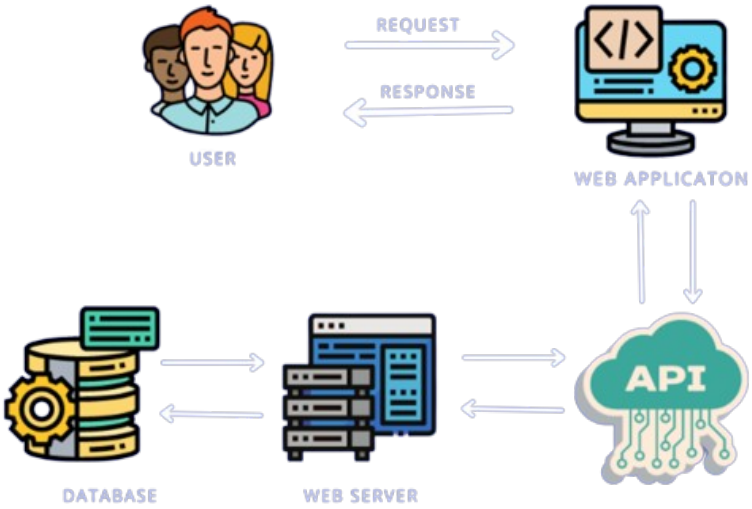
- **For Client Web Application:**

1. The application should load quickly and respond promptly to user interactions for a smooth user experience. Implement smooth animations and transitions for enhanced engagement.
2. Design the application to handle a growing user base and increased traffic without sacrificing performance by scaling horizontally or vertically.
3. Ensure the application is stable and reliable with minimal downtime or disruptions. Include error handling mechanisms to manage unexpected errors gracefully.
4. Secure data transmission using HTTPS encryption and implement measures like input validation and output encoding to mitigate security threats.
5. Design an intuitive user interface with clear labeling and consistent design patterns. Incorporate accessibility features and user-friendly error messages.
6. Conduct compatibility testing to ensure the application works seamlessly across various devices, browsers, and operating systems.
7. Maintain a well-organized and documented codebase for easy modification. Use version control, automated testing, and continuous integration for efficient development.
8. Ensure interoperability with other systems and services through well-defined APIs or protocols, supporting integration with third-party resources.

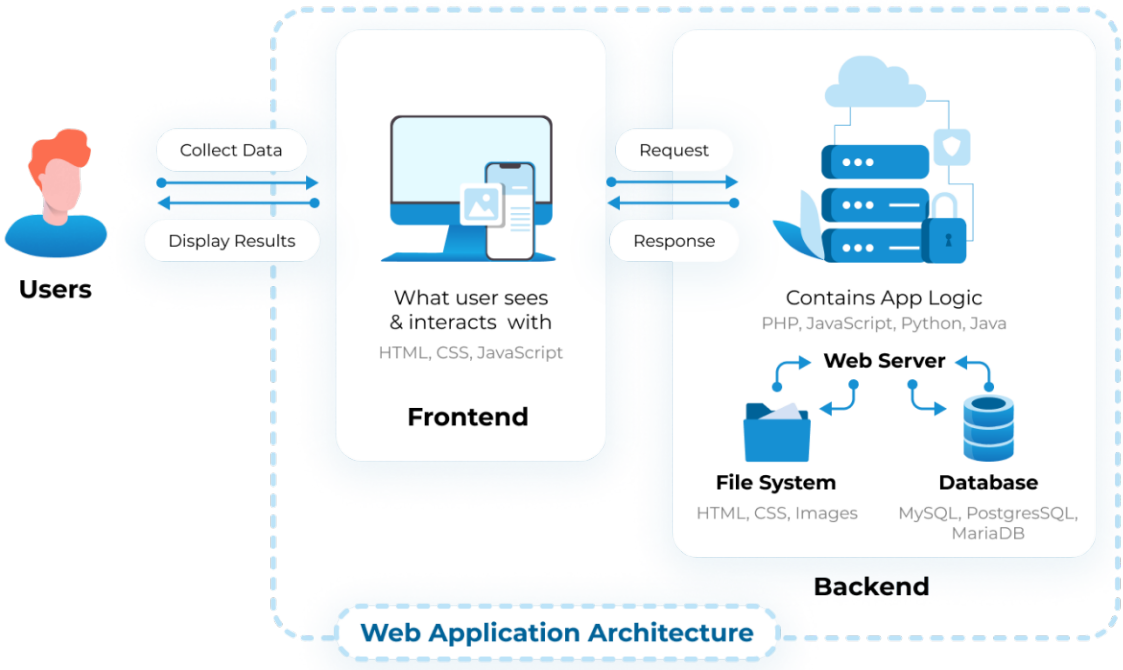
An Overall Architecture Diagram



A Detailed Architecture Diagram for the REST API



A Detailed Architecture Diagram(s) for the Client Web Application



Design Details for Front-End and Backend with Reasons for the Design Decisions

• Front-End Design Details

1. **Framework:**
   We'll opt for React as our front-end framework due to its component-based structure, aiding in code modularity, reusability, and efficient testing. Its extensive component library simplifies UI development.

2. **State Management:**
   Redux will serve as our state management library to centralize application state. It facilitates data sharing among components and manages global state changes effectively.

3. **Styling:**
   We'll leverage CSS modules for styling to ensure scoped styles per component, reducing conflicts and enhancing CSS modularity.

4. **APIs:**
   The front end will consume REST APIs from the Spring Boot backend, using Axios or Fetch for HTTP requests.

• Backend Design Details

1. **Framework:**
   Spring Boot is chosen for the backend due to its streamlined development of RESTful web services and built-in features for OAuth2 authentication and database integration.

2. **Database:**
   A relational database like PostgreSQL will store user and application data, providing efficient data organization and relationship management.

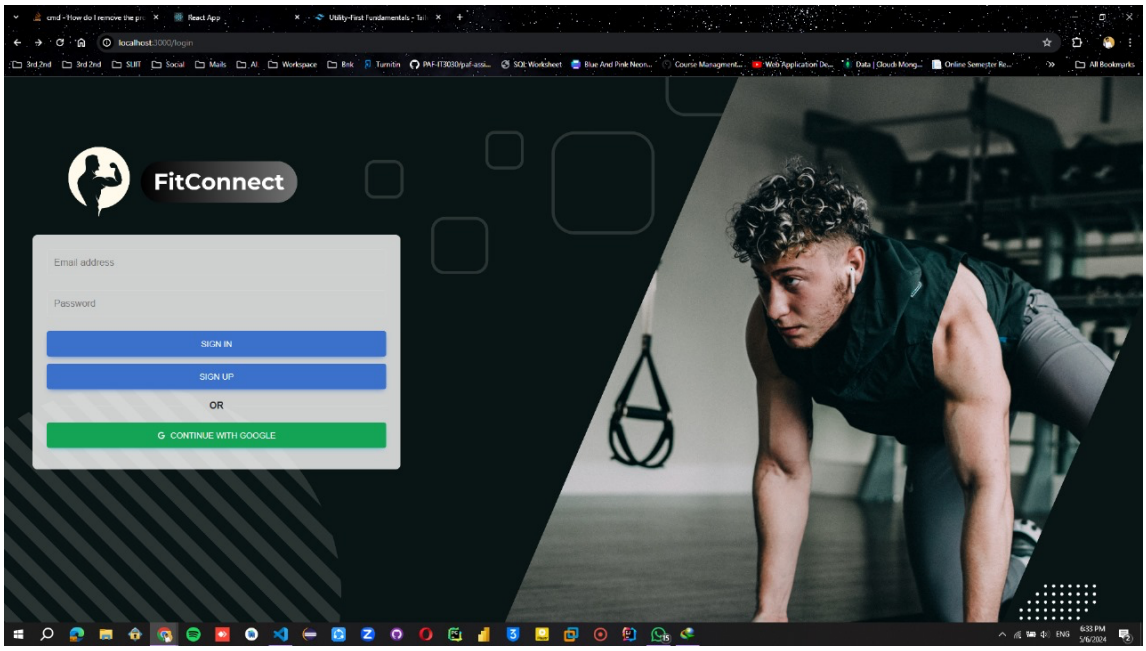
3. **Authentication:**
   OAuth2-based authentication is implemented for its secure user authentication and resource access control, well-supported by Spring Boot.

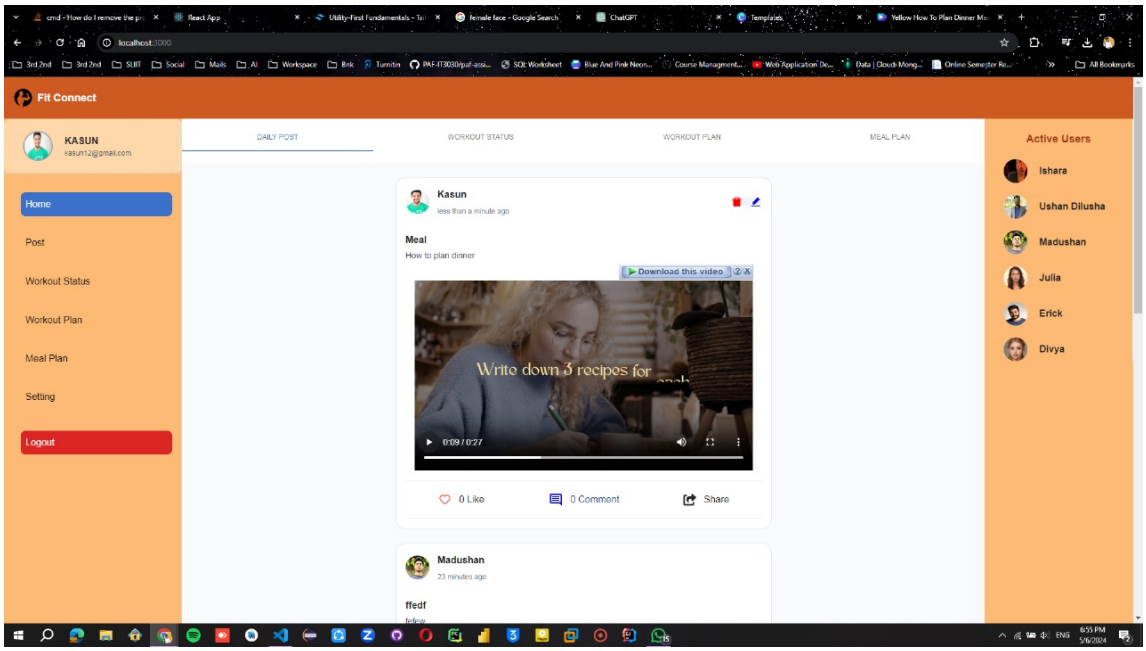
4. **REST API Design:**
   We'll adhere to RESTful API principles, ensuring unique URIs for each resource, using standard HTTP methods for resource manipulation, and employing meaningful status codes for API request outcomes.

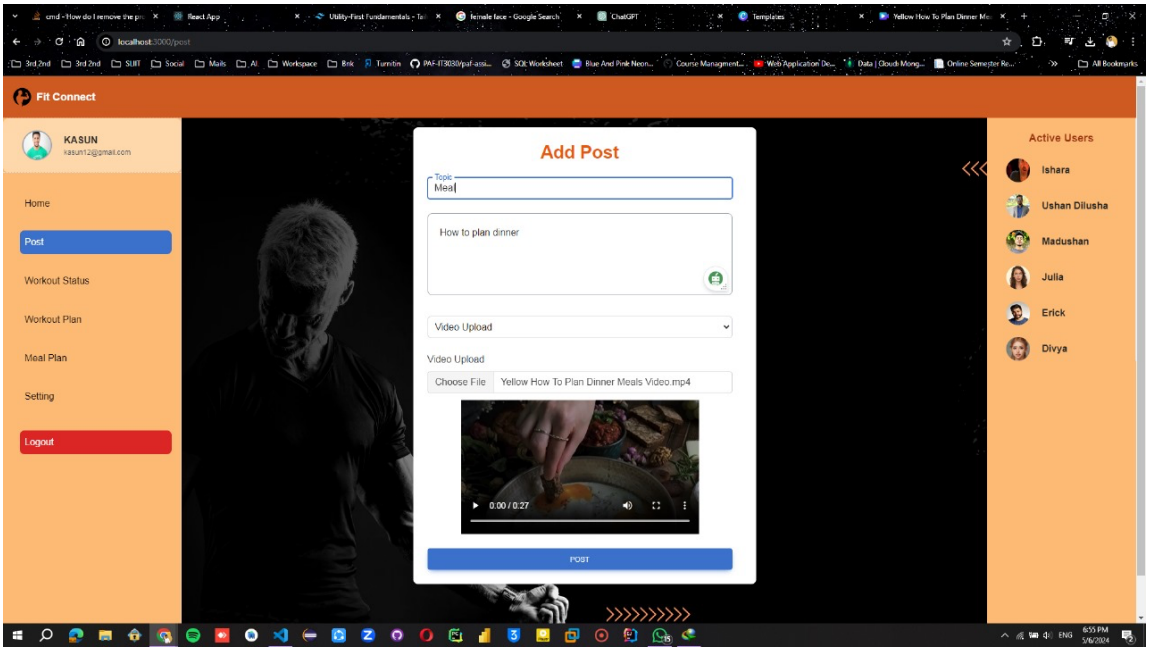
GitHub Classroom – Repository Link

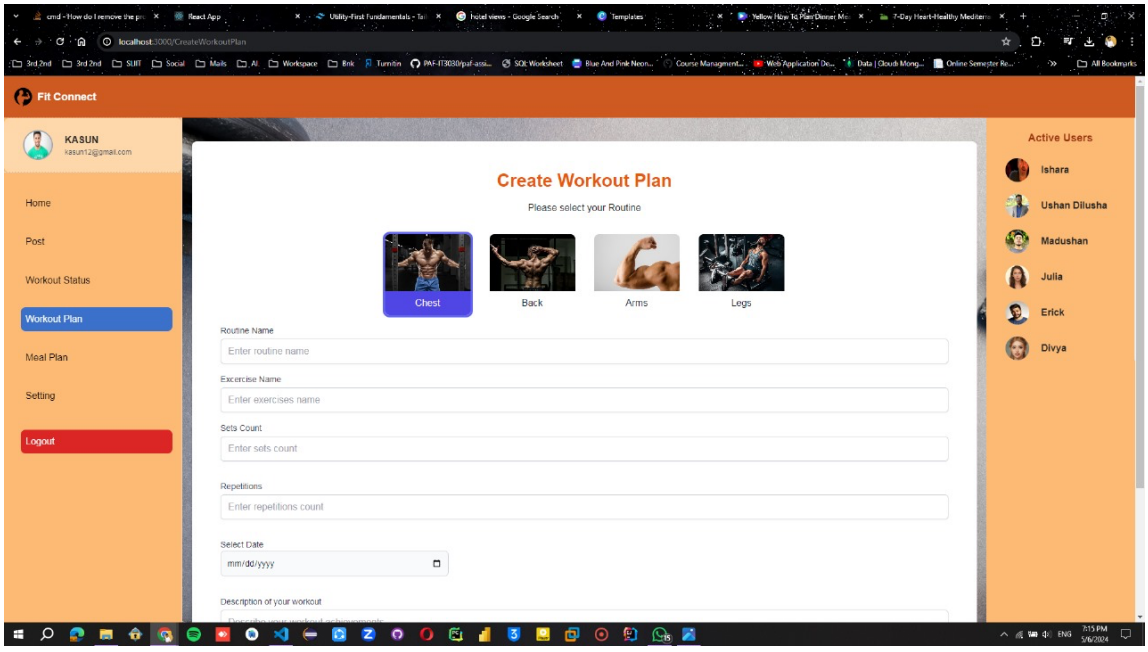
URL: PAF-IT3030/Team_IT_JUN_WE_47_2024 (github.com)

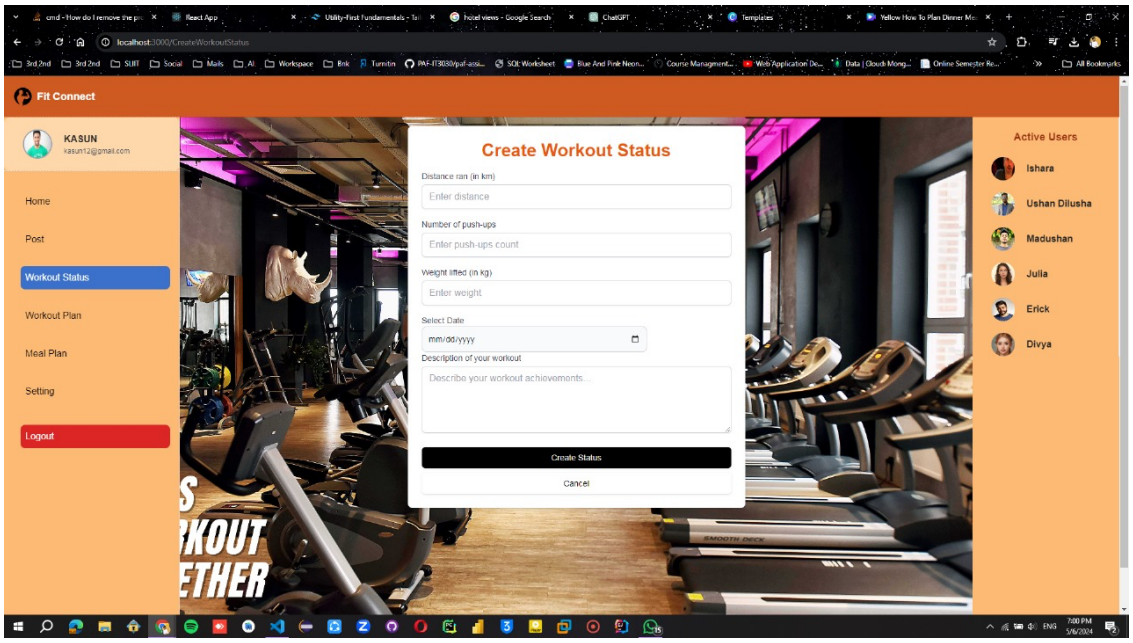
Interfaces

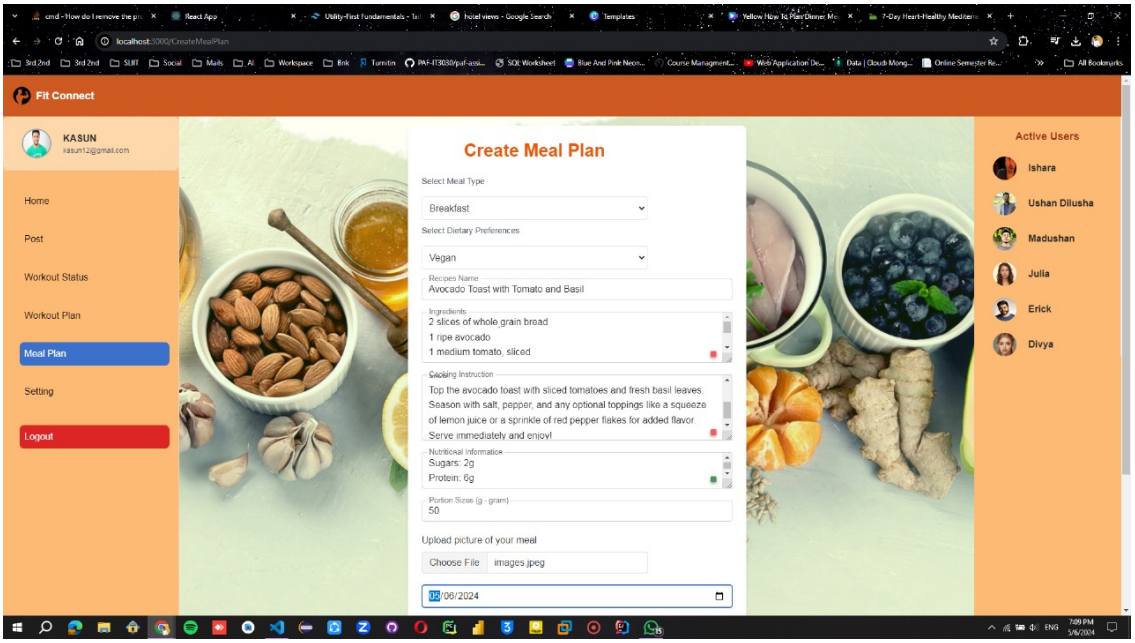





Fit Connect
KASUN
Home
Post
Workout Status
Workout Plan
Meal Plan
Setting
Logout
Add Post
Topic
Meal
How to plan dinner
Video Upload
Video Upload
Choose File
Yellow How To Plan Dinner Meals Video.mp4
POST
Active Users
Ishara
Ushan Dilusha
Madushan
Julia
Erick
Divya


Fit Connect
KASUN
Home
Post
Workout Status
Workout Plan
Meal Plan
Setting
Logout
Create Workout Plan
Please select your Routine
Chest
Back
Arms
Legs
Routine Name
Enter routine name
Excercise Name
Enter exercises name
Sets Count
Enter sets count
Repetitions
Enter repetitions count
Select Date
mm/dd/yyyy
Description of your workout
Active Users
Ishara
Ushan Dilusha
Madushan
Julia
Erick
Divya


Fit Connect
KASUN
Home
Post
Workout Status
Workout Plan
Meal Plan
Setting
Logout
Create Workout Status
Distance ran (in km)
Enter distance
Number of push-ups
Enter push-ups count
Weight lifted (in kg)
Enter weight
Select Date
mm/dd/yyyy
Description of your workout
Describe your workout achievements...
Create Status
Cancel
Active Users
Ishara
Ushan Dilusha
Madushan
Julia
Erick
Divya


Fit Connect
KASUN
Home
Post
Workout Status
Workout Plan
Meal Plan
Setting
Logout
Create Meal Plan
Select Meal Type
Breakfast
Select Dietary Preferences
Vegan
Recipes Name
Avocado Toast with Tomato and Basil
Ingredients
2 slices of whole grain bread
1 ripe avocado
1 medium tomato, sliced
Cooking Instruction
Top the avocado toast with sliced tomatoes and fresh basil leaves.
Season with salt, pepper, and any optional toppings like a squeeze
of lemon juice or a sprinkle of red pepper flakes for added flavor.
Serve immediately and enjoy!
Nutritional Information
Sugars: 2g
Protein: 6g
Portion Sizes (g - gram)
50
Upload picture of your meal
Choose File
images.jpeg
Active Users
Ishara
Ushan Dilusha
Madushan
Julia
Erick
Divya

Git Commit Graph