# Sri Lanka Institute of Information Technology

## Programming Applications and Frameworks (IT3030)

Assignment – 2024, Semester 2

**Final Document**

GROUP ID: **JUN_WE_158**


| Name with Initials | Registration Number |
|---|---|
| IT21228094 | Mendis A.R. P |
| IT21225024 | Bhagya P. S |
| IT21215292 | Madhusanka J.A. A |
| IT21231100 | Sandaruwan W.M.I.M |

# Content

# 1. Project Description

Our group's task is to develop a social media network with a focus on fitness. This website aims to provide a dedicated space for users to exchange workout routines, fitness adventures, and healthy living guidance. Through an intuitive interface, users can upload photographs and videos that showcase their actions, exchange training programs, and even share their meal plans, replete with recipes and dietary instructions. Our goal is to make it possible for users to track their progress, inspire others, and build relationships with a group of like-minded fitness enthusiasts—all while prioritizing simplicity of use and customization.

These are the functions that related to our web application.

- Functionality for Uploading Media and Adding Descriptions

With this feature, users can share up to three images or videos (with a maximum duration of 30 seconds) that highlight their exercises, fitness routines, nutritious meals, and advancements. It should be possible for users to choose and upload media files straight from their devices, along with the ability to add captions to their media posts. For users to share their fitness journeys and accomplishments with the community, this tool is essential.

- Functionality for Sharing Plans and Workout Status:

This feature allows users to use pre-made templates to generate and post updates on their current workout status, including distance run, number of pushups accomplished, weight lifted, etc. It should also be possible for users to share their training schedules, which should include sets, repetitions, routines, and exercises. These plans can be updated at any moment. Sharing exercise programme with others and keeping track of progress both depend on this function.

- Functionality for Sharing and Managing Meal Plans:

This feature enables users to exchange meal plans with resources for meal planning and documentation, including recipes, nutritional data, and portion sizes. It must to be possible for users to group their meal plans according to specific dietary requirements, such vegetarian, vegan, keto, etc. In order to share recipes and healthy eating practices with the community, users must utilize this function.

- Functionality for Community Engagement and Interaction:

By enabling social sharing, this option lets users join a variety of fitness challenges and activities with friends and family. It promotes a sense of community and inspiration by encouraging users to share their results and accomplishments. The development of a community of encouragement among fitness lovers depends on this feature.

# 2.The functional requirements for the REST API and the client web application.

## 1.1. REST API

- Endpoints for Resources:
  Specify the URLs, HTTP methods, and request/response formats for every resource.
  As an example, /users (POST, GET), /users/{id} (PUT, DELETE, GET).

- Rate restriction
  In order to deter misuse and guarantee equitable use, the API ought to incorporate a mechanism that restricts the quantity of requests that can be submitted in a given amount of time.

- Post Management
  Posts ought to be created, retrieved, updated, and deleted by users. and Different forms of posts (pictures, videos, workout status updates, meal plans) should be supported.

- Error Handling
  The API For different situations (such as failed authentication, server problems, and client issues), define error codes and messages. and
  Indicate the error messages that clients should get (e.g., HTTP status codes, error objects).
  Example: Include informative error messages in the response body and use common HTTP status codes (401 Unauthorized, 404 Not Found, etc.).

- Documentation
  Give specifications for the API documentation, such as usage recommendations, request/response samples, and endpoint descriptions.
  Example: Use open API or Swagger to create interactive API documentation.

## 1.2. Client Web Application

- User Authentication
  Let users to Establish a system that allows users to sign up, log in, and log out. Provide support for generating passwords for secure storage. and Give users the option to reset forgotten passwords.

- User Profile management
  Let users to access and modify their personal profiles. Provide options for updating personal data, including a profile picture and email address.

- Dashboard
  Allow Create a dashboard that shows a summary of the user's activity or personalized information. And add modules or widgets to provide easy access to important information or functions.

- Searching and filtering
  Make search features available so users can find particular information or data.
  Enable the filtering options to customize search results according to different standards.

- Notifications
  Install a notification system to inform users of changes or significant events. Give users the option to view and modify their notification preferences.

- Data Visualization
  Use interactive tables, graphs, and charts to effectively convey data. as well Permit users to alter visualizations according to their own preferences.

- Error handling and Feedback
  When mistakes occur, give users relevant error messages and comments. To assist users in fixing errors, include validation messages and guidelines.

# 3.The non-functional requirements for the REST API and the client web application.

## 1.3. REST API

- Usability
  The API must conform to RESTful principles and have a standardized, user-friendly interface. Developers integrating with the API should be able to quickly understand standard error messages and status codes.

- Performance
  Even with a high volume of traffic, the API should reply to queries within a given amount of time. and it is important to optimize response times for frequent tasks like fetching postings.

- Reliability
  Clients should be able to receive informative error messages from the API and it should have procedures in place to handle mistakes graciously. With the proper backup and recovery procedures in place, it should be resilient to failures.

- Security
  Industry-standard techniques should be used to protect user authentication and data transfer. And Sensitive gateway access, such as updating user profiles, needs to be secured with the right authorization procedures.

- Scalability
  An increasing number of users and postings should not cause the API's performance to noticeably decline. In order to add extra servers or instances as needed, it should be designed to develop dynamically.

## 1.4. Client Web Application

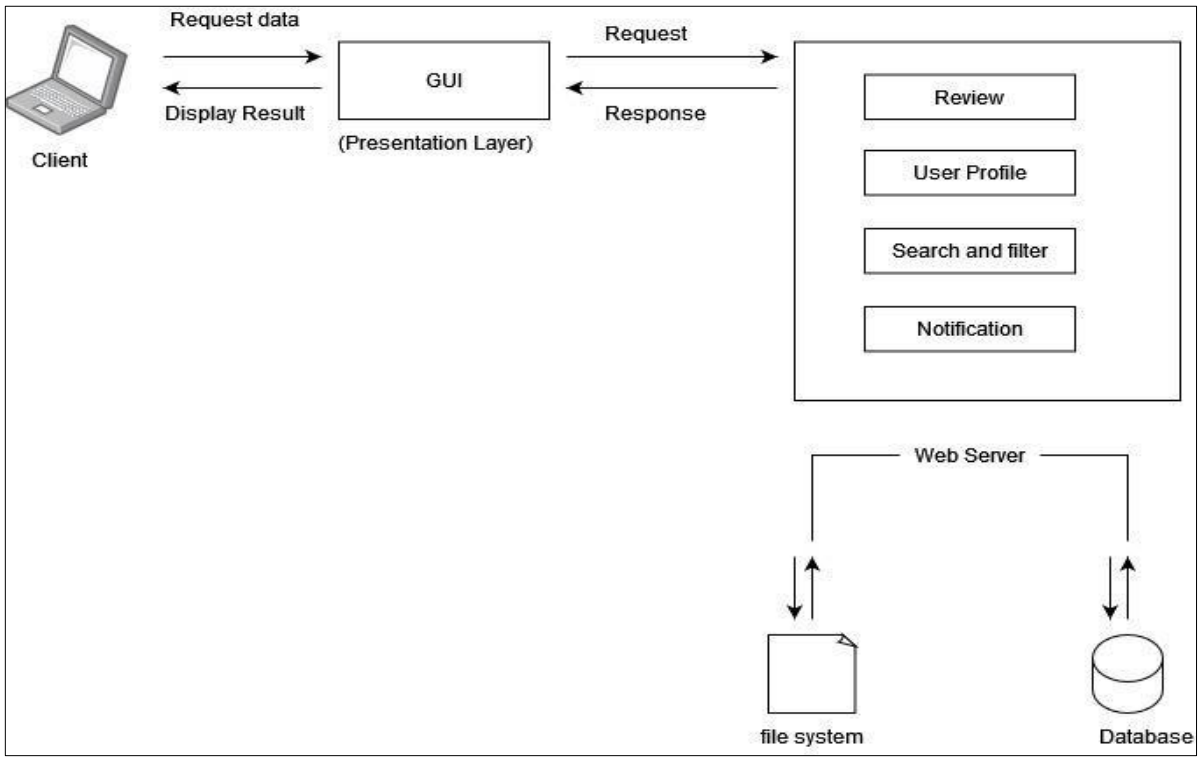
- Performance
  **Reaction time:** The web application must load rapidly and react to user input in a timely manner.
  **Efficiency:** The system should be able to manage a specific quantity of users or requests at once without experiencing any performance impairment.
  **Scalability:** In order to handle increased load or usage, the application must be able to scale both horizontally and vertically.

- Security
  **Data encryption:** To avoid unwanted access or interception, sensitive data sent between the client and server should be encrypted.
  **Authorization and authentication:** To guarantee that only authorized users may access particular features or data, the application should impose access control and securely authenticate users.
  **Compliance:** The system is bound by all applicable security guidelines and laws, including PCI DSS, GDPR, and HIPAA.

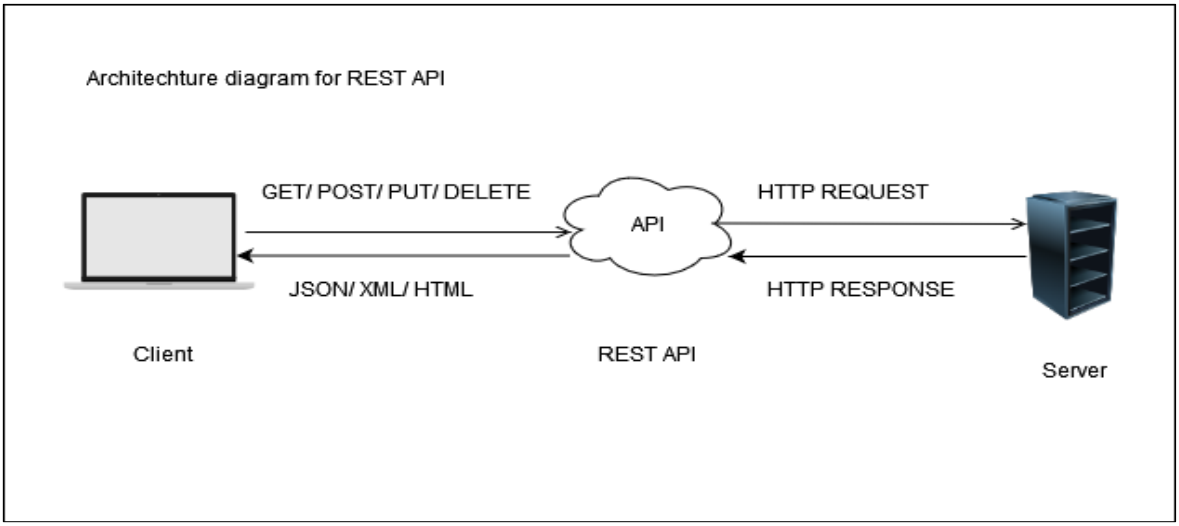
- Usability
  **User interface design:** The programme must to have an easy-to-use interface that is simple to navigate and understand.
  **Accessibility:** The application must adhere to WCAG principles and standards for accessibility in order to be used by people with impairments.
  **Compatibility:** There should be no significant problems with the application's ability to function consistently across a variety of web browsers, hardware, and operating systems.

- Maintainability
  **Code maintainability:** To enable continuous maintenance and upgrades, the application code needs to be flexible, well-documented, and well-structured.
  **Version control:** Git and other version control systems, along with well-defined branching and merging procedures, should be used to manage the source code.
  **Logging and monitoring:** To facilitate troubleshooting and monitoring, the application should log pertinent events and metrics. It should also have tools in place to analyze and address problems.

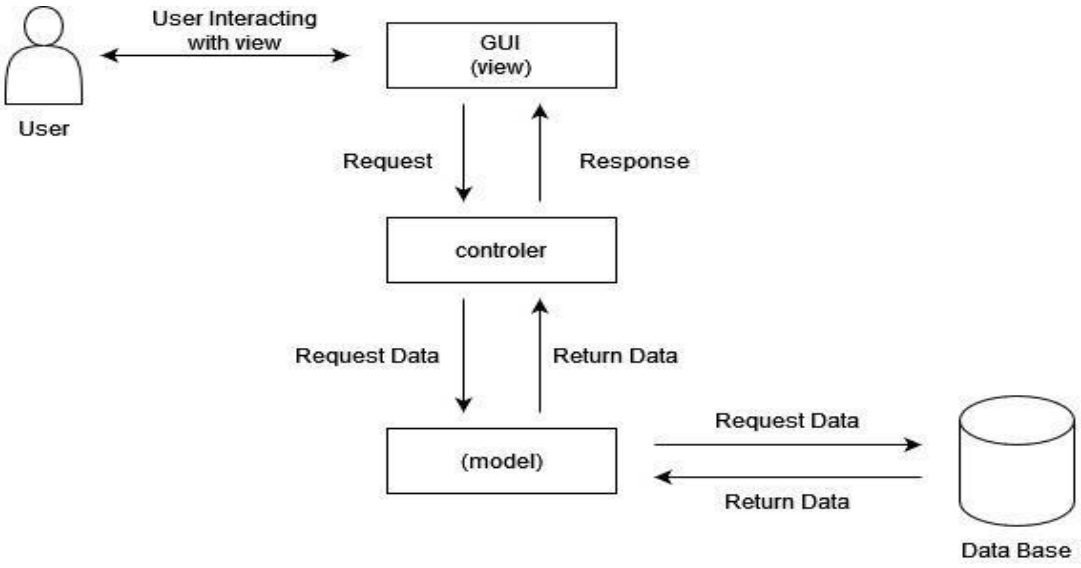
## 4. Architecture Diagram for the System



## 5. Architecture Diagram for REST API




## 6. Architecture Diagram for Client Web Application



# 7.system functions

**The capability to Add Descriptions and Upload Media (IT21231100- Sandaruwan W.M.I.M)**

**Backend**

1. Fetching Posts: Extract all entries from the "posts" collection in the database.
2. Converting a post into an organized database document object is known as post document creation.
3. Post Retrieval: Permit searching for particular posts in the database.
4. Post Deletion: Take a post out of the "posts" collection that is linked to a specific user.
5. Post Modification: Using a special ID, you can change the post's photos and descriptions.

**Fronted**

1. Homepage Feed Display: Showcase all user-submitted posts on the homepage feed.
2. Multi-Media Upload: Allow users to upload up to four photos/videos per post.
3. Description Requirement: Mandate users to include a description before submitting a post.
4. Post Component: Incorporate username, follow status, edit/delete options, description, picture slider, like, comment, share, and comments sections within each post.
5. Edit Functionality: Grant users the ability to edit both picture and description of their posts.
6. Post Removal: Enable users to delete their posts if they regret their content.

**Ease of Sharing Workout Schedules and Status (IT21225024- Bhagya P.S)**

**Backend**

1. Getting Posts: Get each and every item from the "posts" collection in the database.
2. Post Creation: Create a database document object by converting a post.
3. Retrieve Particular Posts: Permit the database to bring up particular posts according to predetermined standards.
4. Post Deletion: Take out a specific post from the database's "posts" collection that is linked to a specific user.
5. Post Modification: Give users the option to edit post descriptions and images by providing a unique ID.

**Frontend**

1. Homepage Feed: Showcase every post made by users on the homepage feed.
2. Image Upload: Permit users to attach a maximum of four images to each post.
3. Before posting, users must make sure to add a description.
4. Post Content: For every post, show the user’s name, status, follower count, edit, delete, description, image slider, like, comment, share, and comments.
5. Editing: Give users the option to change their description and photo.
6. Post Removal: Give people the option to take down posts they've regretted.
7. contribute Media and Add Descriptions: Give users the option to contribute media and descriptions to their postings.

8. Fitness Status Updates: Provide users with ready-made templates to produce and post updates about their level of fitness, including metrics like weight lifted, number of pushups accomplished, and distance run.

**Functionality for Sharing and Organizing of Meal Plans (IT21228094- Mendis A.R. P)**

**Backend**

1. Getting Posts: Extracting every item from the "posts" collection in the database.
2. Post Creation: Creating a database document object from a post.
3. Post Retrieval: This feature permits the database to be queried for certain posts.
4. Post Deletion: Eliminating a certain post from the "posts" collection that is linked to a specific user.
5. Post Modification: Enabling the use of a distinct identifier to change post descriptions and photos.

**Frontend**

1. Post Display: Putting every post made by users in the main feed.
2. Enabling users to attach a maximum of four images to each post.
3. Requiring users to provide a description before to publishing anything.

4. Post Components: Showing choices for editing and deleting each post, along with a picture slider, username, follow status, like, remark, share, and comments.
5. Changing Posts: Enabling users to make changes to their post descriptions and images.
   Post Removal: Enabling people to take down posts they've regrettably posted.

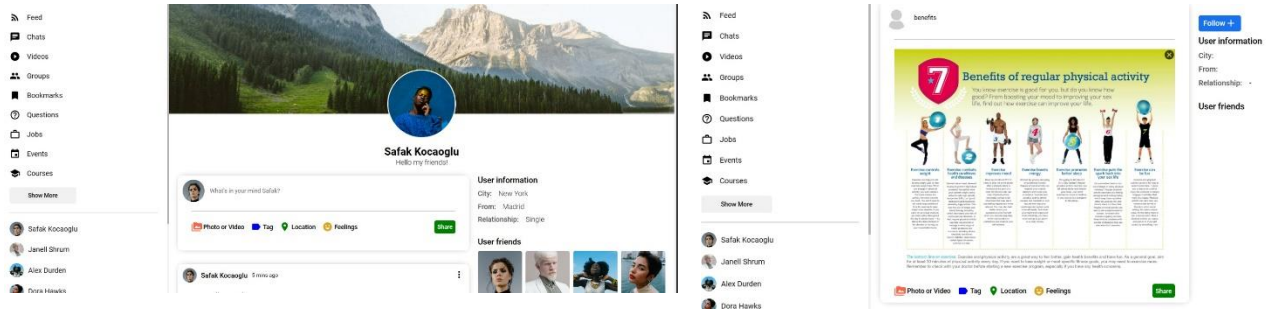
**Interaction and Community Engagement Functionality (IT21215292- Madhusanka J.A. A)**

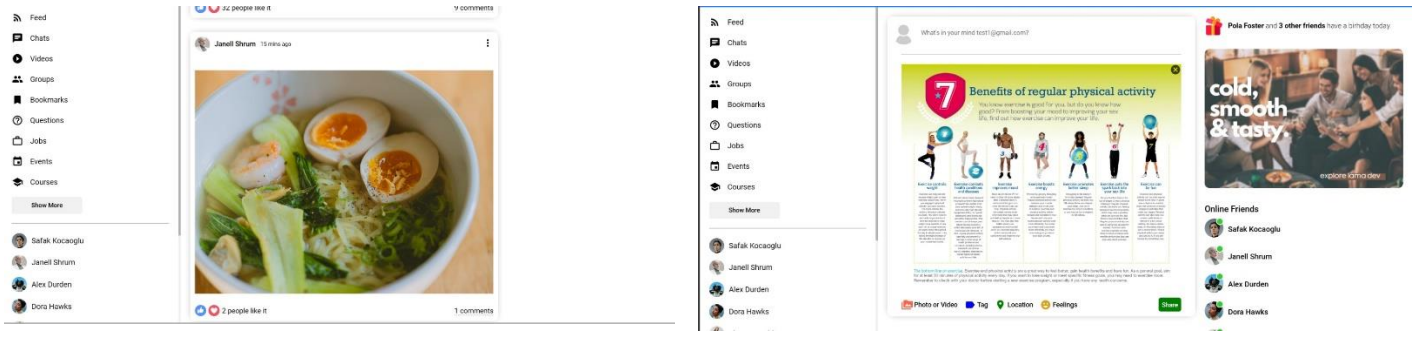
**Backend**

1. Getting Posts: Getting access to records in the "posts" collection of the database, making sure that every post is ready for processing.
2. Post Conversion: To enable effective storage and retrieval, user-submitted posts are transformed into structured database document objects.
3. Post Retrieval: This feature allows posts to be specifically retrieved according to predetermined criteria, improving user experience by presenting pertinent content.
4. Post Deletion: Giving users control over their content and guaranteeing privacy by enabling them to erase their own postings from the database's "posts" collection.
5. Post Modification: Allowing users to update their content as needed by providing the ability to change post descriptions and photos using unique IDs.

**Frontend**

1. Homepage Feed Display: To guarantee visibility and interaction with community material, all user-submitted posts are displayed on the homepage feed.
2. Multi-Image adds: This feature increases the diversity and expression of content by enabling users to add up to four photos per post.
3. Requirement for Description: By requiring users to provide a description with every post, the programme promotes context and user engagement.
4. The unified post layout ensures a streamlined user experience by combining many post features, similar as the username, follow status, edit/delete choices, picture slider, like/remark/share functionalities, and comments, into a single post format.
5. Post Editing: Encouraging flexibility and content control by enabling users to make changes to descriptions and photographs after they have been posted.
6. Post Removal Option: Encouraging content moderation, preserving a pleasant user experience, and giving people the option to remove posts they regret.



# 8.GitHub

https://github.com/PAF-IT3030/paf-assignment-2024-jun_we_158_team