Fitness Journey

Contributors:

Ruby Gavagan
Taisiia Sherstiukova
Evan Zachary
Saulo Guzman
Hassan Alanazi

Project Description:

Our user friendly, self led workout routine website helps gym goers easily organize their weekly fitness schedule. This website has a simple, aesthetically pleasing design and an easily operated system to save personal workout routine data. The user starts by registering for an account or signing into an existing account. They can select exercises for a specific day of the week to accomplish their fitness goals such increasing muscle mass or improving cardio. There are a wide variety of types of workouts on this site that target different muscle groups with varying repetition frequencies. This lets the user personalize their fitness journey. The home page displays the workout routines that apply to their fitness goals. The user can also remove or edit exercises from the schedule if desired. These workout plans are a week by week routine so it can be managed easily for the user. Details of each workout are given which includes the type of exercise, duration/repetitions, and description of exercise. You can modify your workout plan to get new routines for a new week. They can change their password and their account data will remain. Finally, selecting Logout saves the users data for the next time they log in and redirects them to the login page.

Project Tracker - GitHub project board: https://github.com/srpg15/CSCI-3308-13-01

Video: **https://youtu.be/wD0-L1KUKiw**

VCS:

https://github.com/srpg15/CSCI-3308-13-01

Contributions:

Ruby Gavagan:

My contributions included the front end development to achieve a bright and elegant user friendly interface. I created the website's navigation bar, header, footer, and conceived the design of the homepage. I coded the back end for the login and register features as well as being responsible for the security of the password hashing. I was the data manager for this project to make sure all of the exercises, duration values, and workout descriptions were accurate, accessible, and correctly launching on our website, which was a crucial aspect of bringing this project to life. I also was incharge of a lot of the administrative actions such as booking weekly meetings, delegating tasks, contriving the ideas for the project, updating the readme each week, and leading communications.

Taisiia Sherstiukova:

On my end, I helped with front-end stuff. First, I did the initial set-up for the project, such as creating the directory and the code for basic functionalities that we used in other labs, such as log in, register pages and homepage.

Second, I created the design CSS for the login and register pages by using internal and external CSS. I helped trace some other bugs like login form giving an error when entering the password, incorrect destruction of the session when logging out, finding errors in the SQL files and adjusting the buttons.

Evan Zachary:

My contributions mostly came in the form of the backend and middleware linking the two. I was responsible for creating the database structure and contributed to the

I assisted in the creation of the middleware between the database and the frontend, helping out with making the queries inside the api calls in order to display

the information and make sure that each user has the appropriate dataset for their account. As a minor role I helped out with displaying the information on the frontend

homepage in association with just general bug fixing and team consulting.

Saulo Guzman:

My contributions would mostly come from the Frontend and middleware implementations.
- I created the main design for the homepage with HTML by adding table layout, dropdown bars, and buttons for both add and remove.
- I created an API that got both data from the exercise table and user table which presents the information when you click on the drop bar.
- Create a middleware and alerts which require the user to have information for login, and everytime they add or remove an exercise, would put an alert to users.

Hassan Alanazi:

Most of my contributions to this project were in the middleware.

Technologies worked on: NodeJs, Ejes.

I was responsible for adding functionality to the features in the home and profile pages.

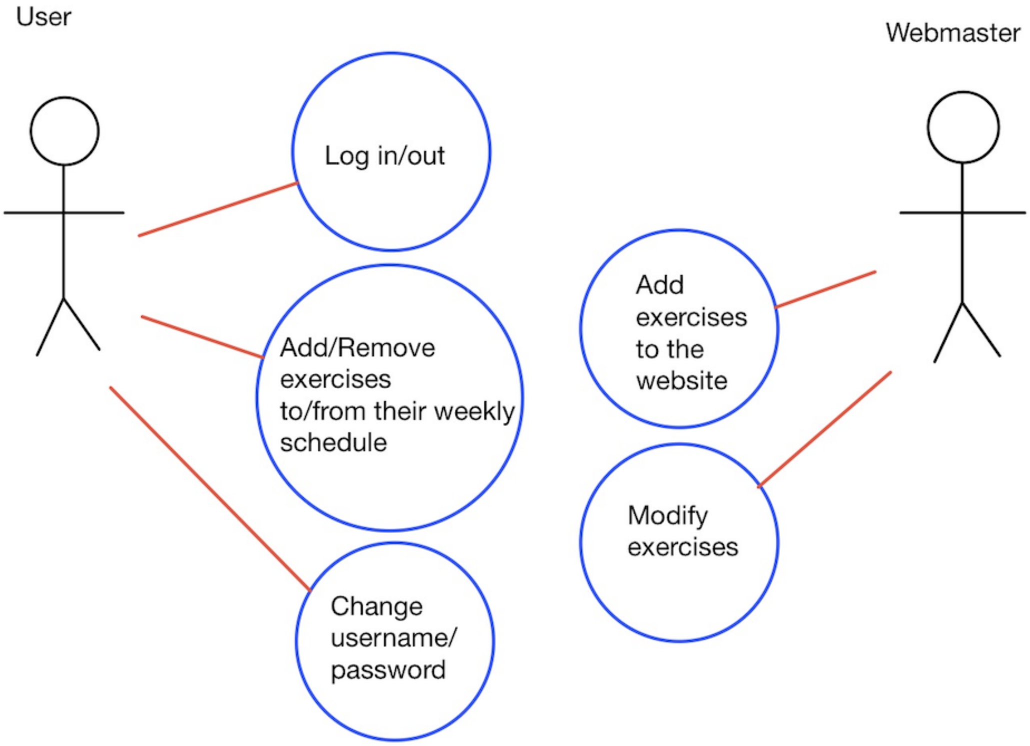
Homepage features that I worked on:

- Adding exercises to the user’s weekly schedule which is displayed in the homepage.
- Removing exercises from the user’s weekly schedule.

Profile page features that I worked on:

- Option for the user to change the username of his account.
- Option to the user to Change the password of his account.
- Connecting both these pages to the database.
- Updating the database.

Use Case Diagram:


User
Webmaster
Log in/out
Add/Remove exercises to/from their weekly schedule
Change username/ password
Add exercises to the website
Modify exercises

Test results:

Use cases:

1. When a new user tries logging in to the website.
2. When an existing user tries to login. But enter an incorrect password.
3. When an existing user tries to login. and enters a correct password.
4. When a user tries to add/remove an exercise, but leaves the fields empty.

Test Cases:

1. When a new user tries to login, the website redirects the user to the register page and prompts the user that he has to create an account, because he doesn't have an account.
2. When the user enters his username correctly, but his password is incorrect, the website prompts him that the entered password is incorrect.
3. When the user enters both the username and the password correctly, the website redirects him to the home page.

4. When the user leaves the input fields empty, the website won’t add/remove any exercises, and it will send a message to the user.

Observations:

When testing the use cases we noticed that the most performed tasks by the website were adding or removing exercises. The reason for that is because the users were managing their workout routine. They weren’t leaving the input fields empty, because of the clear design of the add/remove areas in the homepage. We also noticed that the user behavior was consistent with the third use case.

Deployment:

User needs to have docker installed on the computer.
Then, they must go into the application directory and type docker compose up.
Lastly, they type localhost3000 in the browser of choice.