The GymGoers App Proposal

MVP:

An app that people i.e. me and my friends can use to track our workouts – Tracking your progression is a very important part of the gym that many people miss out on. Being an online application means that all the data is stored securely and can be accessed at any time.

Title:

The GymGoers

Description:

In essence I am creating an application that can be used to track and visualise the progression of gym goers however there is also a social aspect to this application where as part of the application, users can create workout group chats where the progression of individual group members can be shared.

Context:

Tracking your workouts is an incredibly important part of going to the gym, it helps you track progression, helps you identify where you might be going wrong and if any changes need to be made to your workouts.

The gym has always been a social activity for me and my friends and due to our competitive nature, we are always trying to outdo each other. However, when we went to different universities, we found it difficult to track each other's progress and keep that competitive nature. That is why I wanted to create an application that me, my friends could use to log our workouts and see our individual progression but also where we could share our workout and progression with friends and keep that social aspect that I find very important for the gym. I then realised that this could be used as a tool to help personal trainers manage their clients/class members. During the pandemic there was a rise in online personal trainers as well and this application would help them provide better service to their clients.

Features:

To use the application, you must either login or create an account.
Personal User: this is for the average person who just wants to track their workouts and share it with their friends.

The user can register an account by using an email and password, they can also edit the password to update it

Once logged in they have the following features

NavBar:

- Dashboard – that contains the last 7 days of workout logs
- View workout entries – that contains the last months workout logs with the option to delete logs
- Make a new workout entry
- Make a group chat – where you can invite other users to
- View group chats

Dashboard:

- Shows the workouts in the last 7 days
- Maybe - Has different graph showing the time spent and weights pushed on each day

Group chat:

- Allow users to create group chats and have the feature to add and remove friends from their group chats

Business User: this is for users that need to use the application as a tool to manage multiple clients and to provide better feedback to them.

With this account the business user will not have the features to create workout entries it will be only used to manage personal users.

Navbar:

- Create a group chat – invite members
- View group chat – add members, view members workouts, and delete members from group

Create workout groups – personal users can be invited to workout groups, here their workouts can be seen by the admin (The business user) and they have the feature to remove personal users from the group

User interface:

On the applications start-up the user is shown a log in page with a header that has an option for the register page.

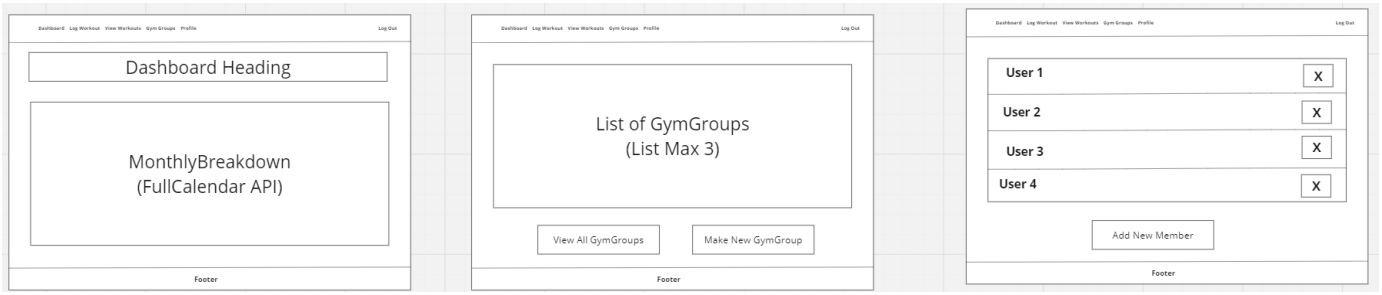
On log in, the dashboard is shown which contains the workout logs in the last 7 days, there is also a navigation menu that has the following options; dashboard, log workout, view workouts, gym groups, and profile.

On the log workout page, the user is shown a form that is used to log workouts, here the user must enter the type of exercise, the weight, number of reptations and sets and any additional notes they would like to add about the exercise. There will be a button to submit the workout log to the database.

When viewing the workout logs, the user is sent to a new page that contains the workout of the last month, the user will be able to click on a workout to see all the details of the workout.

Clicking Gym Groups will take the user to a new page that will show at maximum 3 of the groups they are in with two options, to create a new group or to view all their groups.

When a specific Gym group is accessed by the group admin/admins, they will have the ability to delete members from the group and also, they have access to view induvial members workout logs.



Architecture:

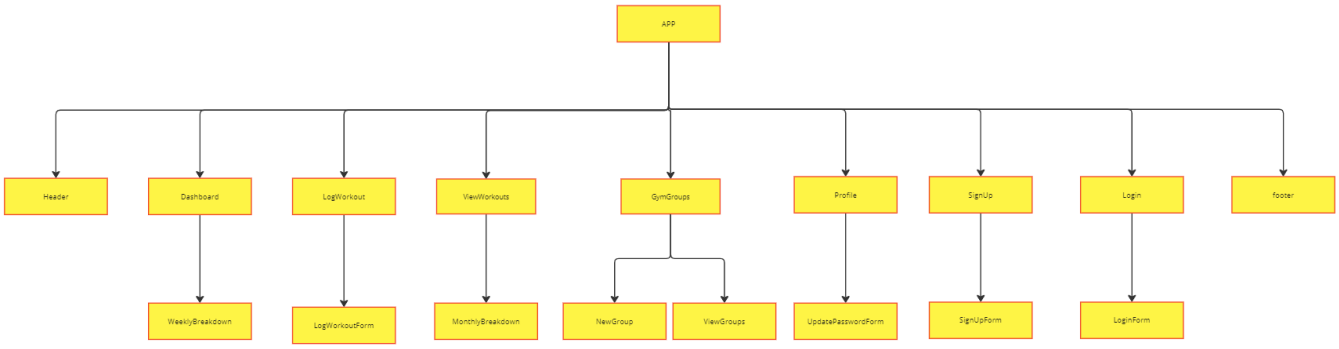
My application will be split into 3 parts.

The first part is the actual the client-side application, this will be created using react and will contain the user interface and also will contain the logic on how the user interacts with the application.

The second part is the database, for this I will be using the MongoDB Atlas service and it will hold all the data that is stored which includes users details as well as the workout logs and the GymGroups.

To complete the application, I will need a server-side application and I will use node.JS for this. The server-side application will connect the client-side with the database and help handle the requests between the two.

Component Hierarchy:



RESTful Routing:

I will need a set of routes to handle requests made while using the applications. A user must be able to register an account and also log in to an existing account with the option to edit their password once they are logged in. This requires a POST request to register an account, a GET request to check that the details used to log in match the records in the database, and a PUT request that updates a user's password in order to update the password.

Once logged in a user is able to make workout logs and view workout logs. This will require a POST request to add workout logs to the database and also a GET request to get the workout logs of a specific user.

One of the features of this application is to create GymGroups, this will also require a POST request to the database that create a GymGroup model which contains the members of the GymGroup.

Technologies Used:

frontend

- React
- JavaScript

Backend

- Node
- Express

Database

- MongoDB – atlas
- Mongoose

For testing

- Mocha
- Chai

APIs

Potentially I will make use of the FullCalendar API to display the workout logs in a calendar format. The reason I have chosen the FullCalendar APi is because I am using JavaScript and the API is an open-source JavaScript library. By using this API integrating the calendar functionality will be easy as it is also highly customisable.

Deployment:

I will be making use of cloud services for the deployment of this project. The objective of this project is to create an application that my friends and I can use at any time, this means that the application must be deployed in a way that allows access at all times.

MongoDB Atlas - So that the database can be accessed at all time, I will be using the MongoDB Atlas cloud service to host the database.

Netlify – I will use Netlify to host the server-side application, Netlify supports continuous development from Git repositories. As I am creating a react app and using a Git repository for this project this service will integrate seamlessly with my application.

Render – This application will be created using nodeJS, Render is a tool that makes deploying applications created using NodeJS.