|  |
| --- |
| Climbing and General Training Tracker |
|  |

Table of Contents

[Analysis: 1](#_Toc155177499)

[The Problem: 1](#_Toc155177500)

[The stakeholders: 2](#_Toc155177501)

[Solving the problem computationally: 2](#_Toc155177502)

[Abstraction: 2](#_Toc155177503)

[Decomposition: 3](#_Toc155177504)

[Thinking Ahead: 3](#_Toc155177505)

[Logistics: 4](#_Toc155177506)

[Software and Hardware Requirements: 4](#_Toc155177507)

[Limiting Factors: 4](#_Toc155177508)

[Research: 4](#_Toc155177509)

[Existing Solutions: 4](#_Toc155177510)

[1 – Crimpd: 4](#_Toc155177511)

[2 – Gym Tracker: 5](#_Toc155177512)

[Questionnaire: 6](#_Toc155177513)

[DESIGN: 8](#_Toc155177514)

[Website Arrival: 8](#_Toc155177515)

[The program will include: 9](#_Toc155177516)

# Analysis:

## The Problem:

For my project, I am designing a website where a user can create an account with to track Climbing and Gym exercises. This will be similar to existing programs such as Crimpd (https://www.crimpd.com) and other general gym tracking apps (https://www.strong.app). The website can analyze and suggest better ways to train for the user’s needs. On creating an account, the user can specify their goals and the website will tailor their climbing and training to their needs.

The overall goal of the website is to enhance a climber’s training structure and exercise efficiency. This will allow climbers to use the resource to develop their skills faster and with less injury.

The reason I am making this website is that climbing training plans and information is hard to understand and incoherent to implement without paying money for professional trainers, this makes climbing training hard to begin and understand. My resource will allow all levels of climbers to train with consistency and understand which exercises to do and why.

For this to function the website must have an easy user interface for users to quickly navigate to what they need. The website will also need a database to store users’ login details to allow personalized experiences.

On arrival to the site, users will be prompted to log in or signup, after signup users will have an optional quiz that will tailor the website’s suggestions to the user. Then users will be shown options such as account, logbook, exercise finder, and my plan. Users can follow these options to navigate to which feature they wish to use.

## The stakeholders:

The project will be accessible to all internet users who use a web browser, this means this won’t limit most potential users. The main user base will be experienced climbers who are looking to structure their training to improve or maintain strength. However, the website will also be useful to other sporting athletes who want to track and organize their training.

The Main users will be climbers looking for a free method to further their training so mostly experienced climbers.

## Solving the problem computationally:

My problem is well suited to computational methods of solving problems since computers are quick and optimized for algorithms such as recommending exercises to users and well as providing an easy and quick method for users to access the resource.

### Abstraction:

Abstraction is a method of removing elements of code or programs that aren’t relevant or important to the main purpose of the program. For my problem, I’ll ignore the styling and appearance of the website while I focus on the basic functionality such as the database and algorithms working. I will also focus on more important functions such as login and sign-up services and the climbing logbook.

Abstraction is also hiding data useless to users and code from the user this involves a fluid experience for the user through a GUI. Users won’t be able to see code and how the website works such as the code to optimize training plans and suggest exercises. Users will only see the outputs from the code.

By coding the most important functional aspects of the site first I’ll ensure that I follow a direct order of programming and be able to test later website functions without limitations as the backbone programming is already there.

In my program, I will abstract the following and more:

* Simplistic layout only shows items relevant to the current page
* Hamburger menu to allow navigation
* Showing progress in a simple graphical form
* Plan will be designed on a weekly rotation
* Personalization algorithm will work of simple answers (multichoice, yes/no, scale)

### Decomposition:

I will need to use decomposition to help me effectively complete the website. Decomposition is the method of breaking down problems into simpler, smaller tasks. This helps make problems easier to approach and complete as its easier to focus on smaller manageable parts than trying to do it all at once.

For example, when coding algorithms for login I will focus on individual elements then put them together such as visuals then database then input. This will make it easier to focus on the purpose of the code I am writing and help keep code modular.

When decomposing the development of the home page I’ll separate it into these main parts:

* Home page design
* Login feature
* Additional Links e.g. forgot password button
* Username and password storage
* Username and password login checker

For the home page design ill need to consider the useability and appearance the page should be visually interesting but also easy to interact with.

I plan to store usernames and passwords in a database. I will hash the password before storage and then to check the password I will hash the inputted password and compare the two hashed passwords.

I’m adding a forgot password button to make the site more accessible, if users forget their password, they will still be able to gain access to their account after proving their identity.

For all navigation on the home page, I will use JavaScript to provide functionality without having to reload the website. This will make the page more appealing to users especially people with slower internet.

### Thinking Ahead:

Thinking ahead is a way of planning a method of producing a program such as deciding on features and methods beforehand.

* I will program as a website as I have experience with creating functional websites and it provides access to the program without download.
* I will use Python as a backend language as I have made algorithms using this language before and it’s a more efficient language for repetitive data processing than alternatives.
* The site will make noises notifying users of when to start exercises such as weighted hangs, this makes it easier for users to complete workouts effectively without looking at their phones for timers.

### Logistics:

Navigation on the website will be mostly mouse based with large buttons to click on with short descriptions more detailed information will be displayed after navigating to specific pages. During workouts, the site will use sounds to present time, reps, and sets to allow users to complete workouts without watching their phones.

## Software and Hardware Requirements:

Due to the nature of websites the requirements for using the website are minimal. Users will need an internet browser and an internet connection. Minimal hardware specs are necessary. The website is using mostly JavaScript rather than many webpages this reduces the need for the website to reload constantly.

## Limiting Factors:

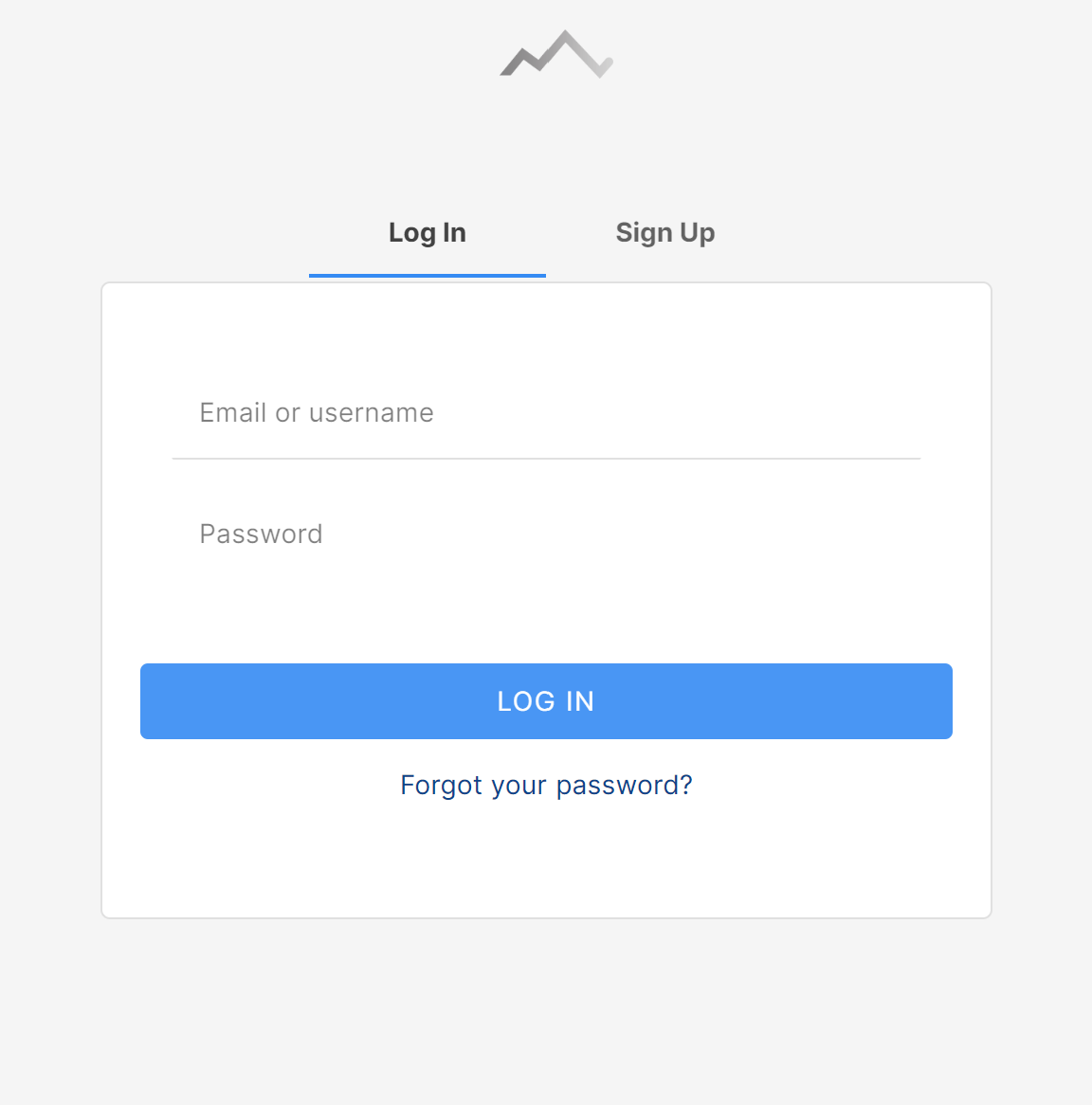
A limiting factor is the school internet, I will primarily be coding on visual studio and the internet blocks many features that make coding easier such as certain extensions.

To get around this challenge I will install all extensions I need at home.

## Research:

### Existing Solutions:

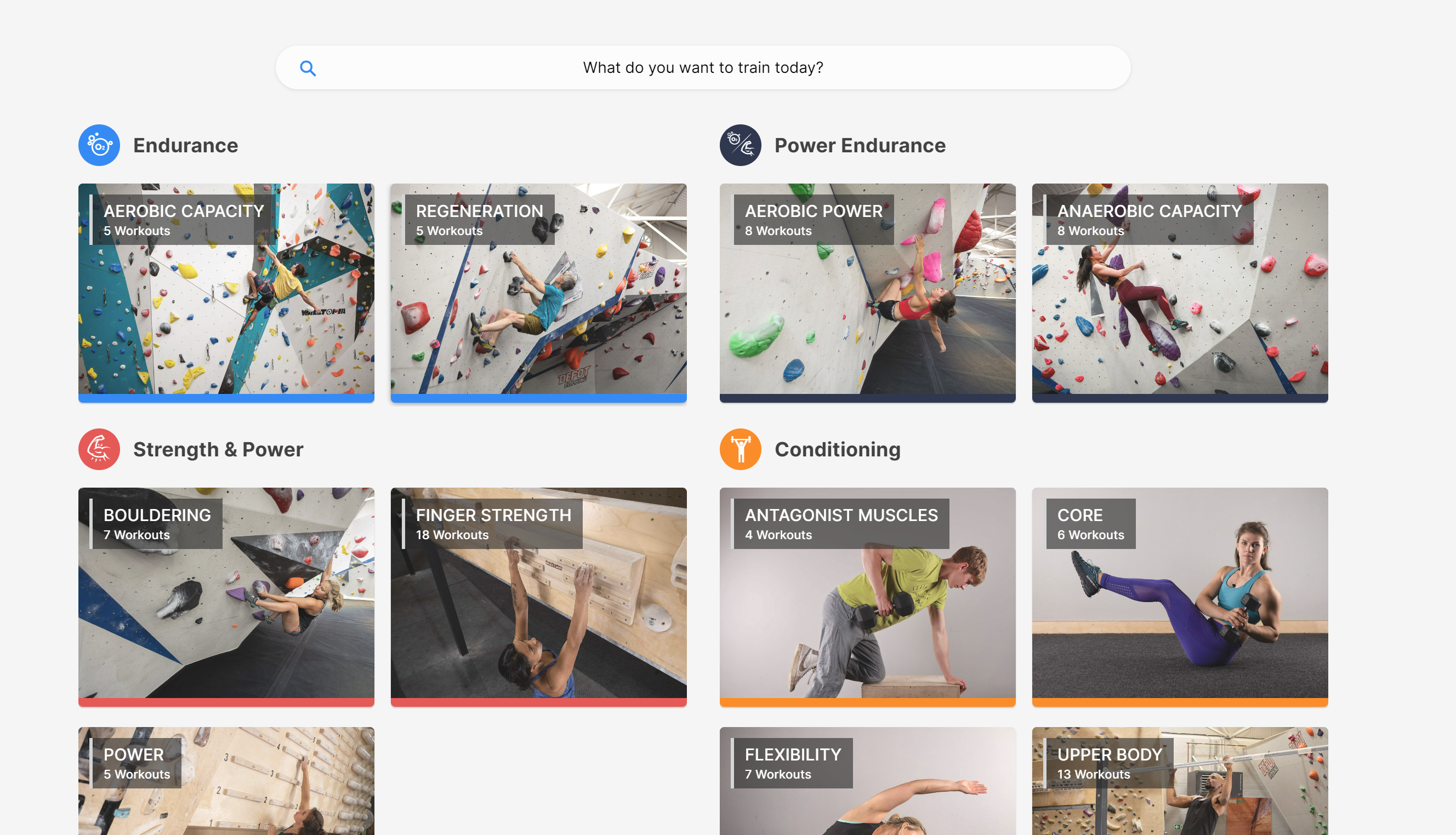
#### 1 – Crimpd:

Crimpd is an existing climbing-specific training app. Crimpd provides users with a selection of workout movements and short workouts, Users use multiple of these in conjunction to create a full workout session. Crimpd also provides a detailed graphical display of progression and a point system to show users how much work they have done in each aspect of climbing.

This is Crimpd’s login/signup page it’s very simple and directs users’ attention towards where to input data this makes login and signup quick and easy. In my site ill use a simpler design on the signup page.

Having a simple signup page with limited questions upon initial site arrival will present the program as simple to use which makes it more approachable to potential users.

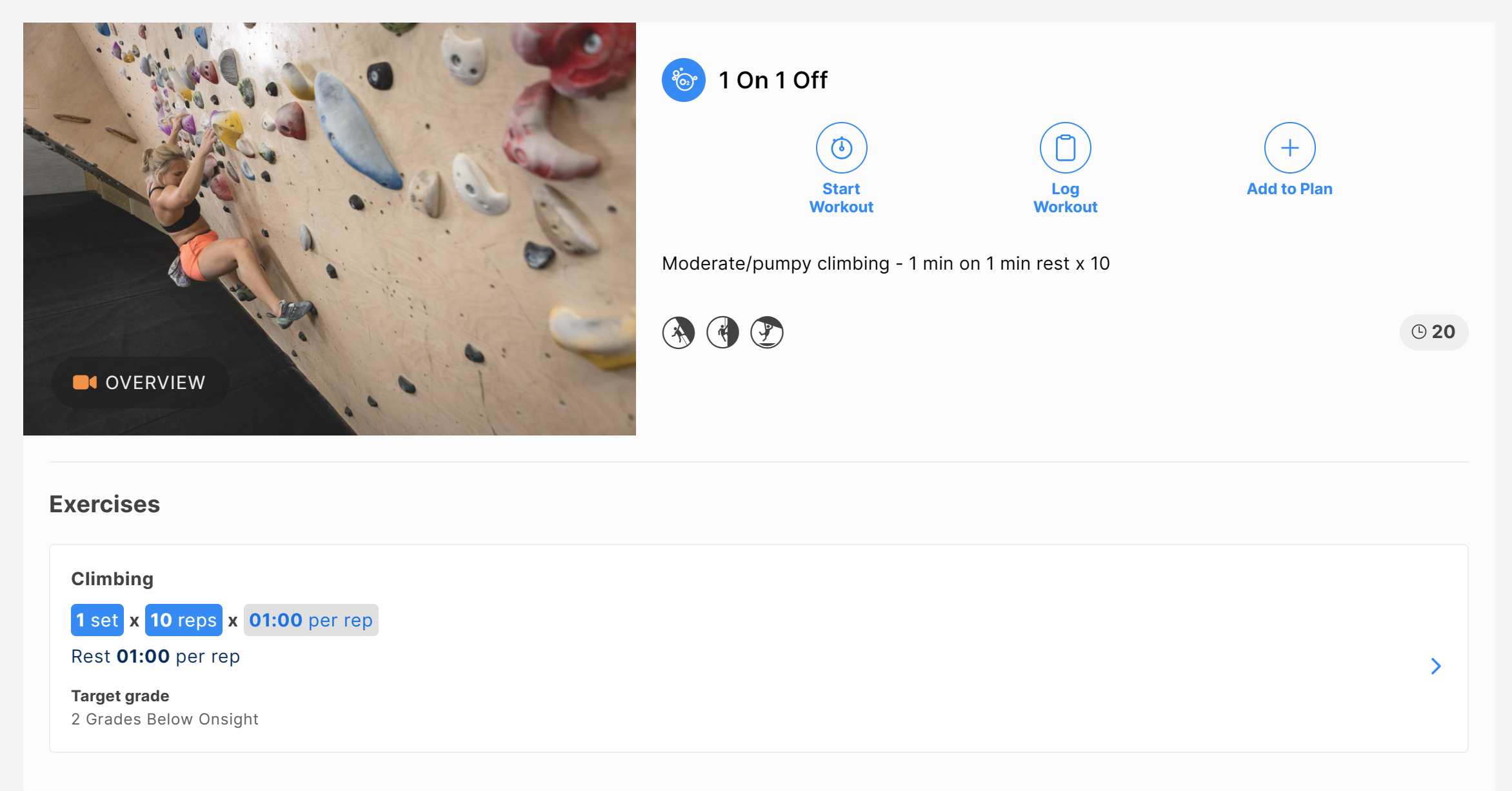
They also have an easy to find “Forgot your password?” which improves verification of user via email confirmation.

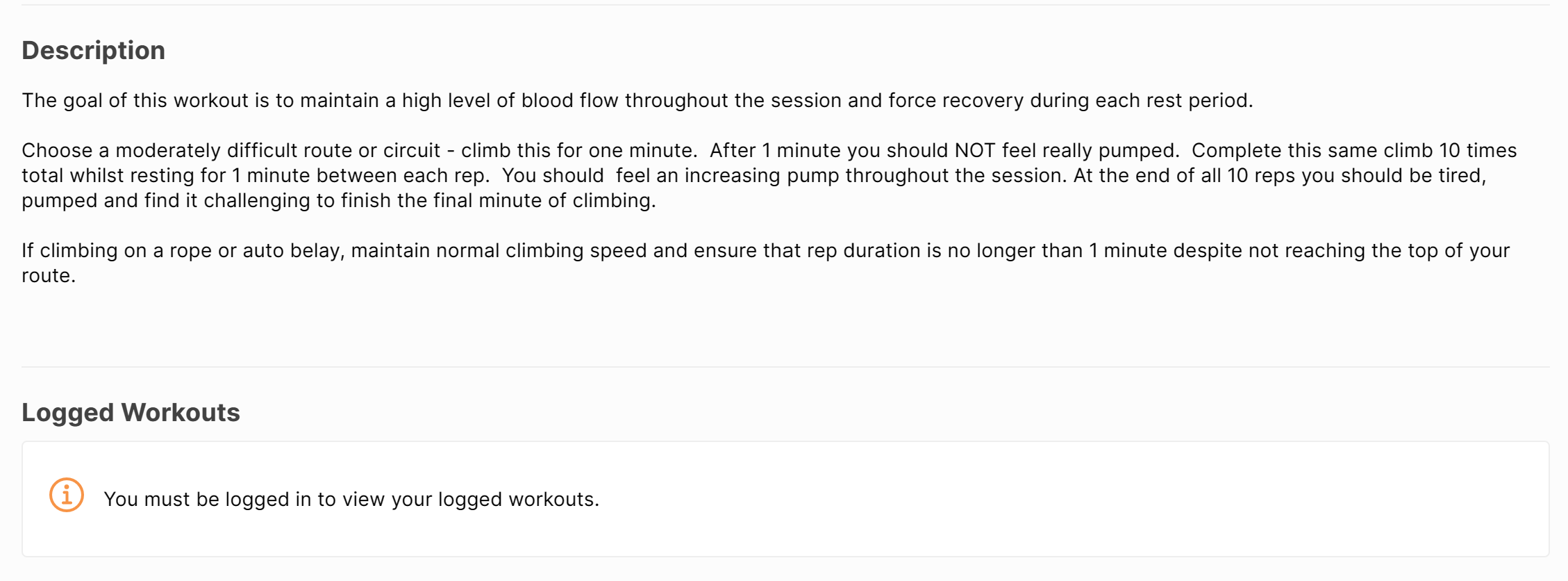
This is Crimpd’s home page, it is divided into types of workouts with simple names and logo coordination. This makes it simple to find a specific workout to complete.

Color coordination makes it easy to spot what component of climbing your training and helps locating specific components when in different exercise views.

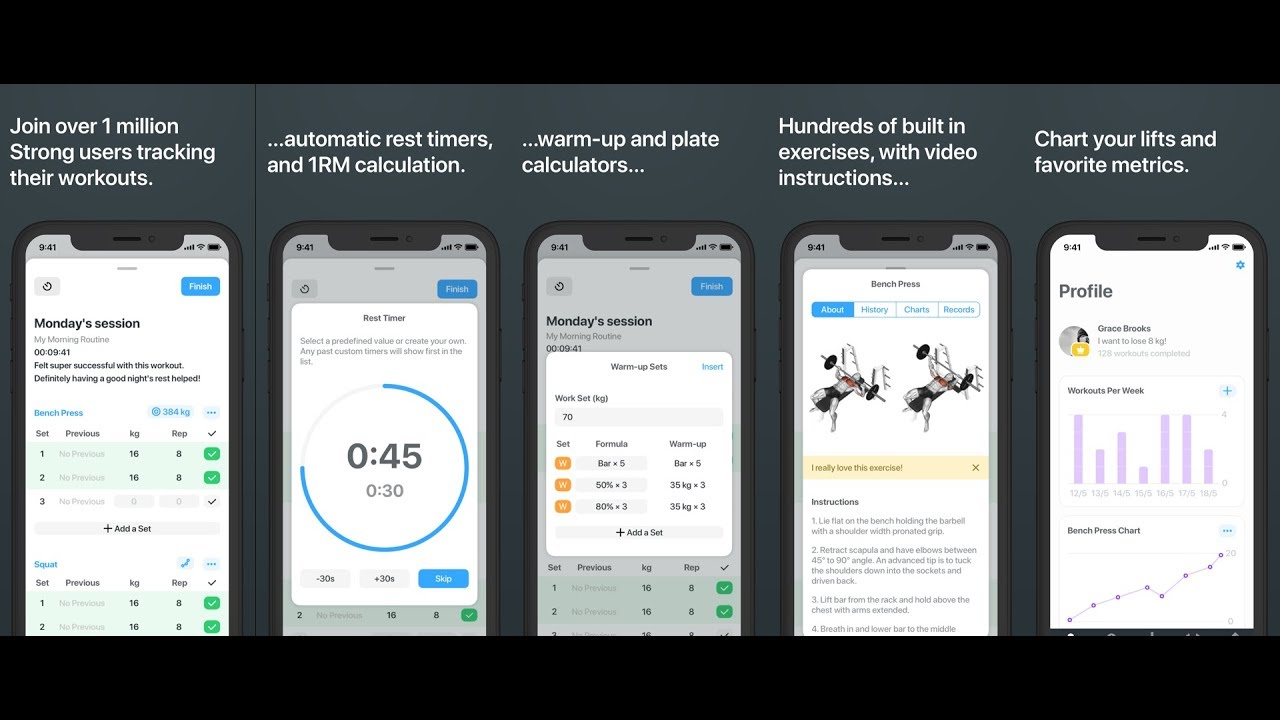
By having simple photos of exercises as backgrounds it makes it easy to understand what an exercise is and allows beginners to understand what the name of an exercise means.

They use abstraction to hide unneeded details such as workout description and equipment since it isn’t needed when briefly looking at what area of climbing to work on.

This is the page shown when clicking on a specific exercise, in here they show all the details to the user since it’s no longer to abstract and hide data. There are 3 main buttons clearly labeled and visible. This makes it easy for users to know where they can navigate off the page.

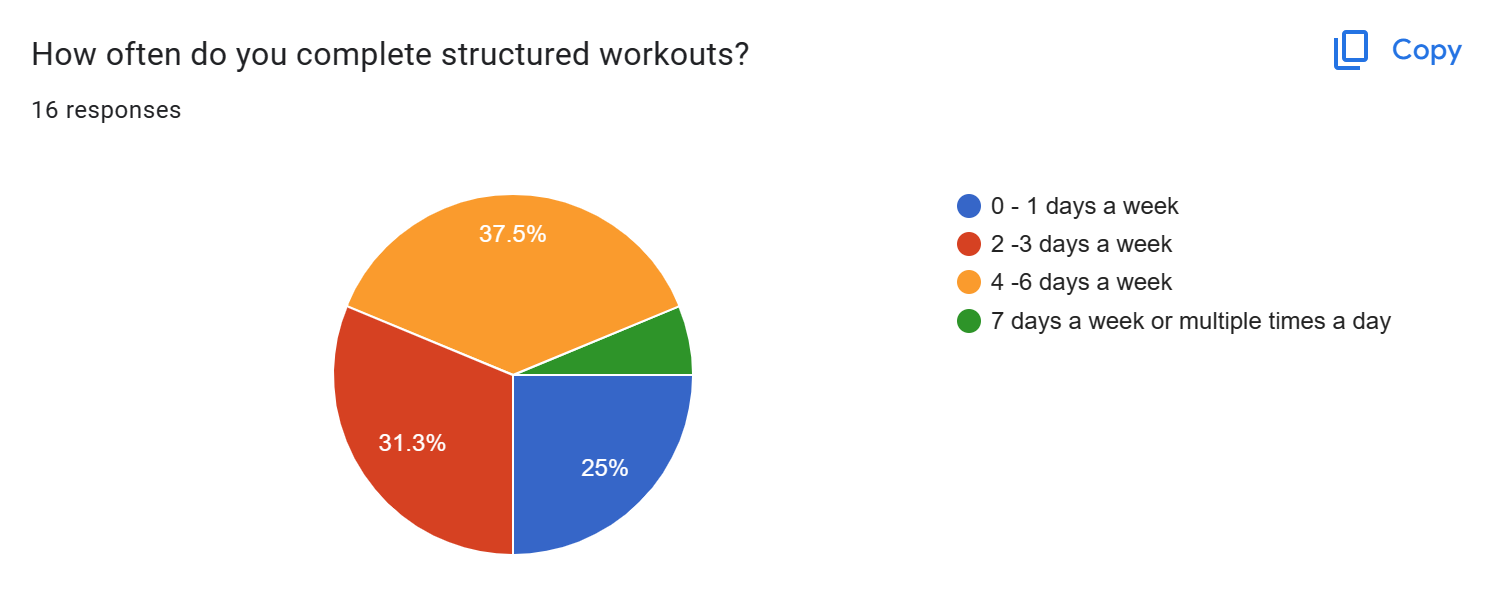
It also has a demonstration video available to watch, this makes it easier to understand for users. They also have a basic description of sets, reps and times. This makes it obvious how long a user will spend on an exercise.   
They also provide a brief description of what the exercise achieves and how to perform it. This makes it clear for anyone who is new to training or prefers words over the video.

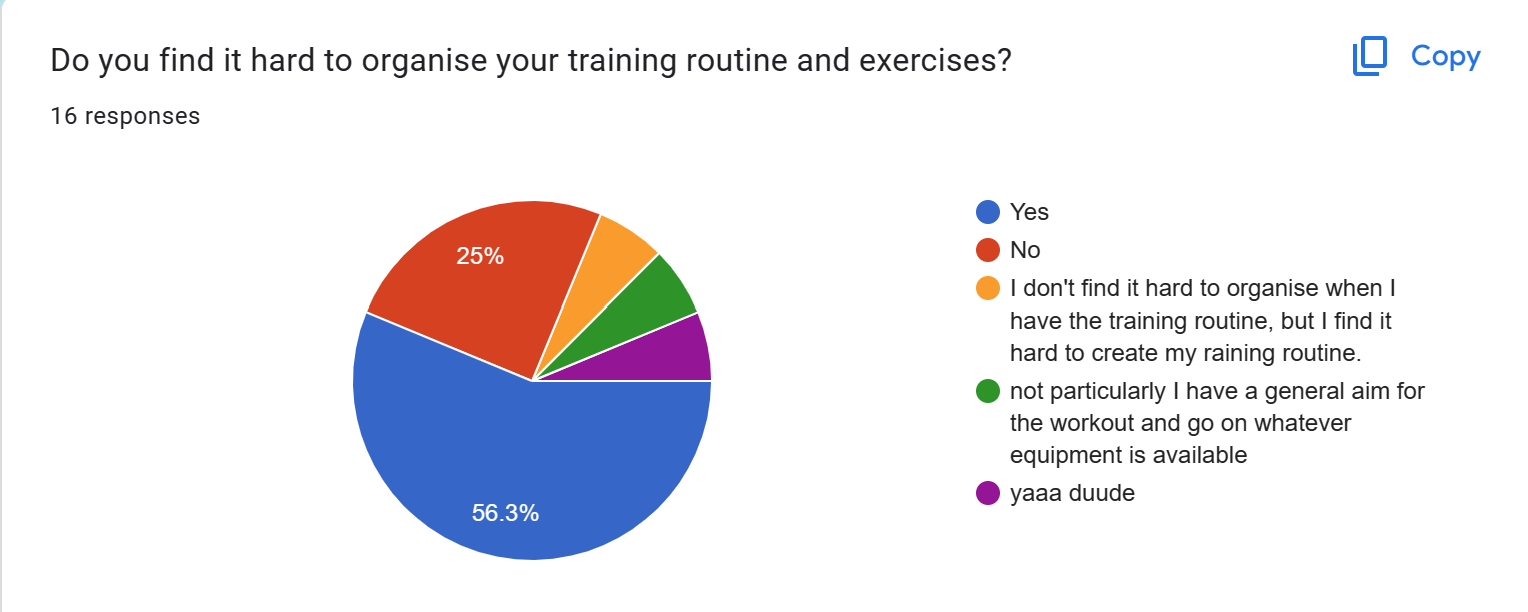
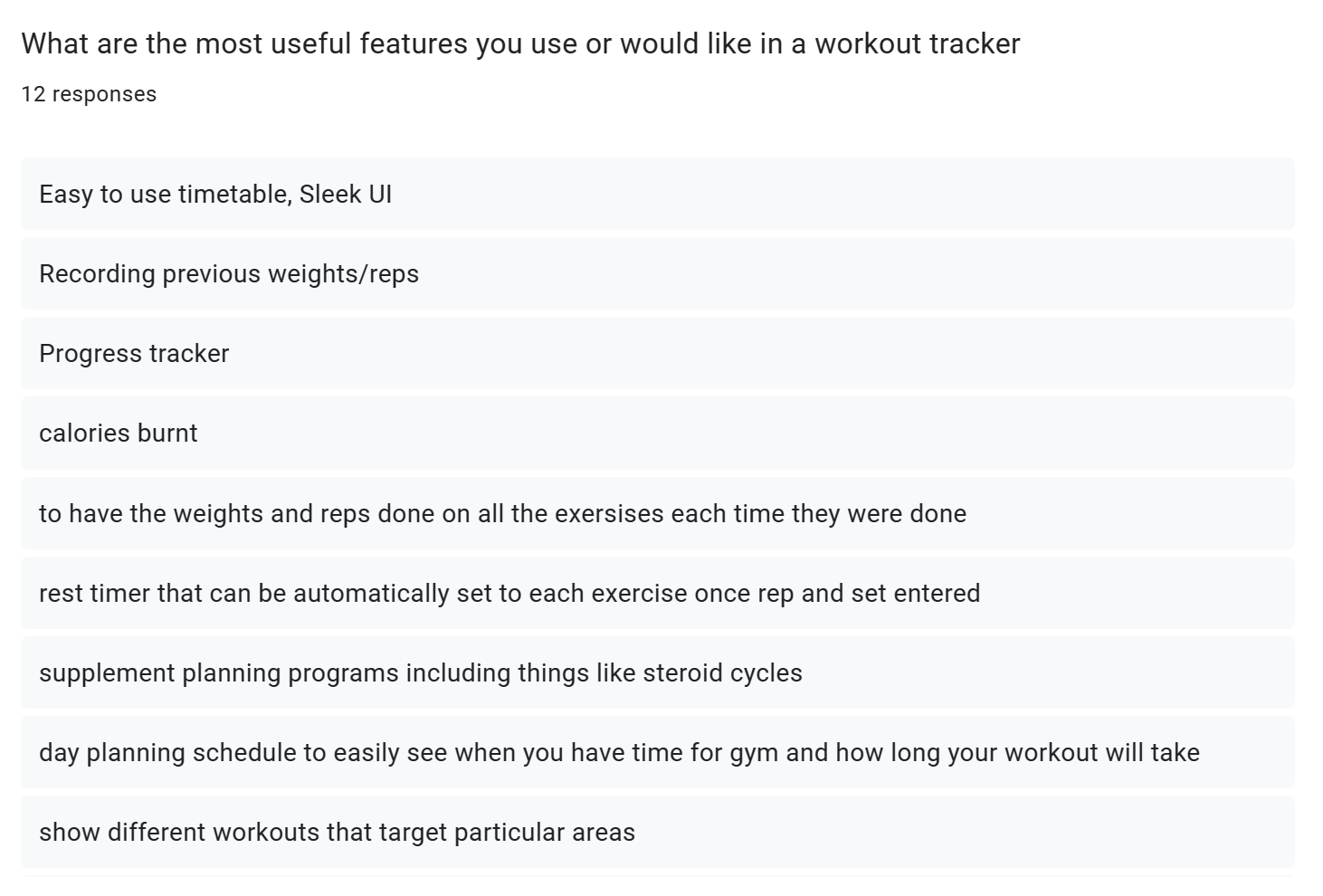
#### 2 – Gym Tracker:

Strong gym tracker is a non-specific training app tailored to Apple and IOS. It doesn’t have a website of desktop app but has android and IOS apps on their designated stores. Strong also improves its accessibility by making it work on Apple Watches which for some people can make it easier and quicker to view and interact with the service.

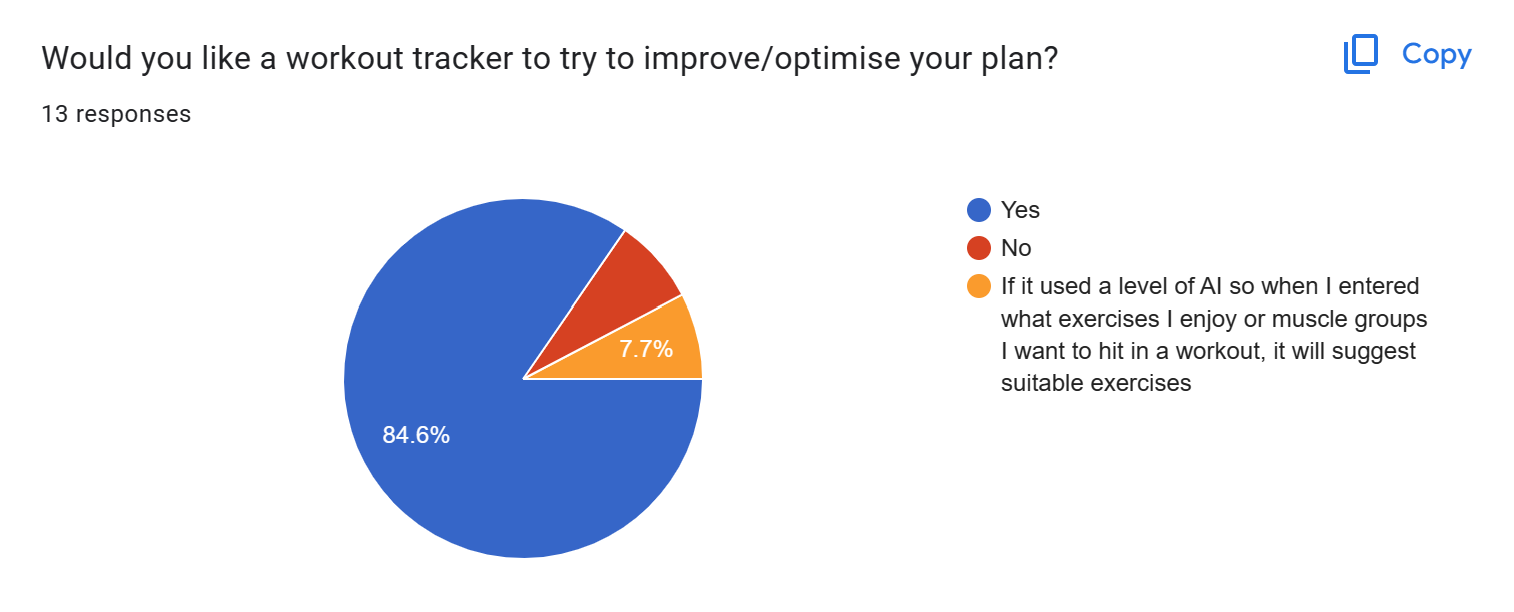
Strong workout provides a plethora of features with a sleek and clean user interface which promotes ease of use.

### Questionnaire:

I conducted a structured google form questionnaire and collected responses from my class and other students. This will help me understand my markets needs and wants. I’ll use this information to improve my website design.

This shows that that there are many potential candidates for my program. However, this is off a small sample so the results may be unrepresentative of the whole population.

These are some of the feature suggestions for the website. I will try to incorporate as many of these as possible however these ones I won’t:

1. Calories Burnt – I won’t incorporate this since the website can be used on any internet device and most likely won’t be a wearable. Collected data won’t be specific enough to provide users an accurate calorie figure.
2. Progress Tracker – I’ll use occasional testing to track progress for users and normal workouts can be used to track progress. Data will be shown on graphs since it is more visual for users.
3. The website will suggest reps and sets for all exercises and will help with rep and rest timings to help usability.
4. Supplement and diet programs won’t be included in the app as my knowledge in these areas is limited.

The responses to this show there is reasonable demand for a product like mine.

# DESIGN:

## Website Arrival:

On arrival to the website, users will be asked to log in using their username and password or signup. During signup, users will be asked questions in a multiple-choice style.

Usernames, question answers and the hashed password will be stored in a database.

Questions will include clients:

* Time/Availability
* Goals
* Experience

This data will be stored in the client’s profile and will be editable at any time. After the data has been collected it will be used to create a custom training program for the client. This program will be customizable manually also.

The login page will be simplistic and will have a forgot password button to improve accessibility.

## The program will include:

All plans will include the users unique:

* Workout days
* Exercises with suggested weight
* Sets, reps and rest

The website will also collect workout data to track progress. This data will be used to target weaknesses and optimize training for progressive overload.

The program will also give a detailed explanation of why this structure was chosen and why certain exercises are picked. This helps users understand our choices and what they are working on. This helps users persevere with their program and not skip exercises.

Each exercise will have a unique description that helps users perfect their form and learn new exercises.

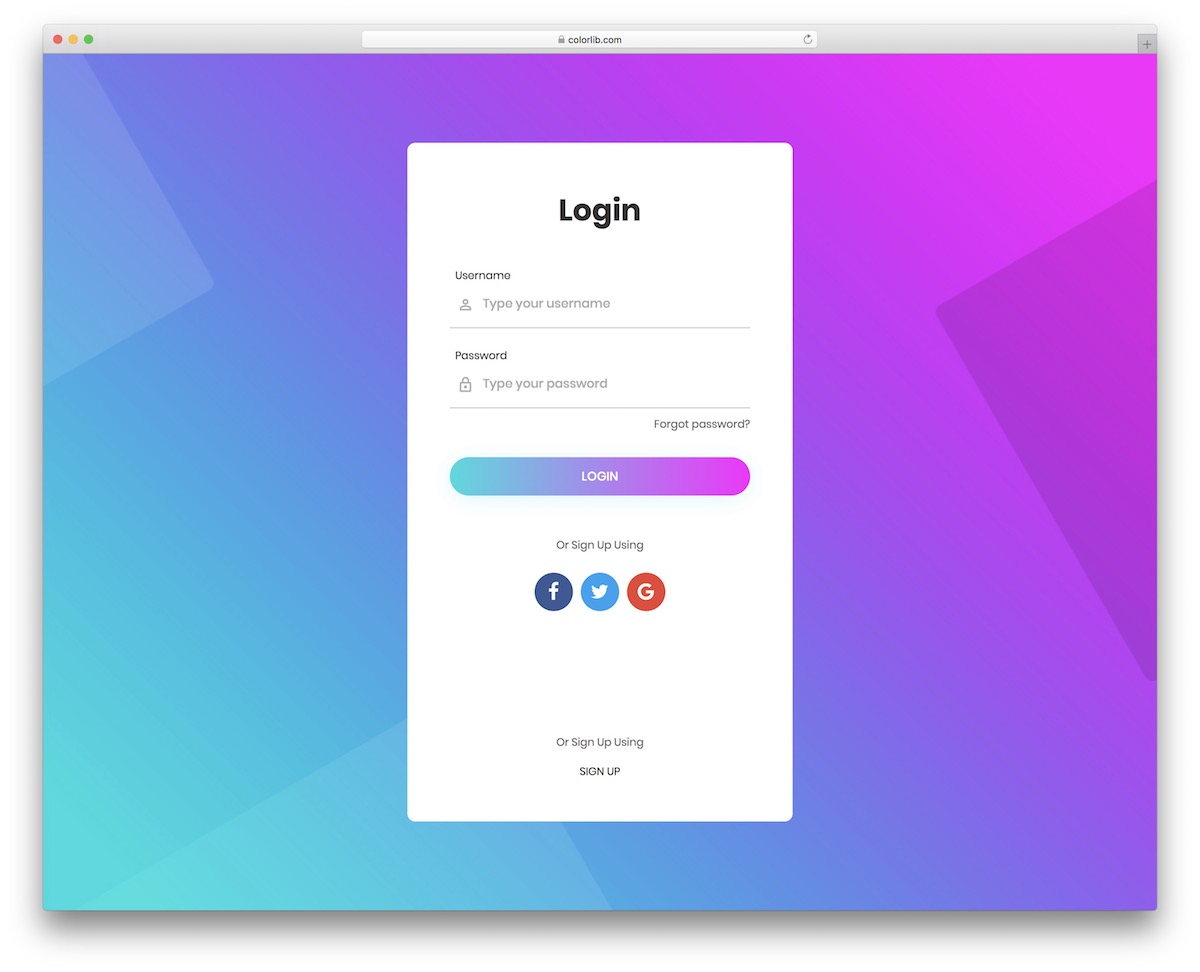
## Login:

The login page will be the index page for my website as users will need to login or signup before having access to features. This is because the features of the website require data about the user to be useful.

If users have an account they will be able to login by entering a username or password. If the user forgets their username or password their data will be recoverable if they can enter either the username or password and answer security questions they enter upon creating an account.

When a new user arrives on the login index page they will be able to switch to the Signup page by clicking a button. This will bring them to a similar page where they can enter a username and password plus a password confirm field is required.

The username and password have a few validation requirements. The username must be composed of only letters and numbers and have no spaces. The password should be at least 8 characters in length and should include at least one upper case letter, one number, and one special character. This not only makes the password more secure but allows the data to be handled easier as it has already been validated before being stored the a database.

All passwords and security questions will be stored using hashing for security, if someone accesses the database unauthorised then they will not have access to personal data. To check that an inputted password matches the hashed password the input will be hashed also then the two values compared. For security questions the inputted value will be converted to all lower case and remove any punctuation before being hashed. This means if the person trying to input their security questions types a slightly different grammar the database will still match the hashes and grant them access.

The image above in an example of a login page I will used as a template for designing my own page. My own version will not have options to login using other methods. This template is good as the fields are obvious and the UI is easy to understand will still being visually interesting.

When a user creates an account they will be prompted with a variety of questions witch will be stored in a database. The answers to the questions will be used by an algorithm to make a tailored plan for the user. Information provided by the user will be editable at anytime via the account section later. Some possible question are below:

* How many days a week are you available?
* How many hours on each day are you available?
* What is your current training focus/goals?
* How much training experience to you currently have?
* What equipment do you have available?

Users will also be prompted to input data such as current weight, height and also current PBs and records they have is certain lifts/exercises.

## 

Above is my website walk through. Users will also be able to logout at any stage of the website. This will redirect you to the login page. The green boxes show what must happen upon loading the page.

## Home Page:

On the homepage, users have access to many activities including:

* My Plan
* My progress
* Start Exercise
* Timer
* Logout

Each of these items will be accessible via a button which will lead the user to a different page.

### My Plan:

Users will have access to their plan to view or edit details. In this area, they are presented with a calendar on which are events for workouts. When selecting an event, they can edit manually or run the workout manually which directs them to the start exercise page.

In this area options selected for the plan creator can be edited also.

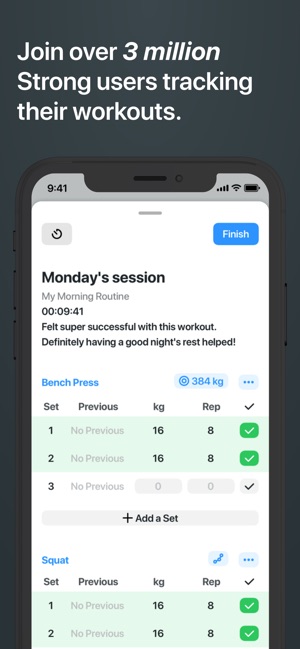
### My Progress:

This area will present statistics in the form of various charts. These can display work out details such as workouts per week and details such as workouts for strength or endurance. Progressions can also be displayed using data collected at the end of workouts and on monthly standardized evaluation days.

This area is useful for users to visually see their progress which is important for motivation. This data is also useful to the program to continually update the plan and suggest improvements.

### Start Exercise / Today:

This page will show the exercises that will be done today and allow the users to log the difficulty the exercises where completed at, this can be stored and pulled up later in the progress section.

This area will also give more in-depth explanations on each exercise including a description, sets and reps for each exercise. This will be displayed in a table where each exercise can be expanded for more details.

To submit the exercise difficulty the user can click a button that will submit the individual exercise once submitted and sent to be stored the option to submit will be greyed out and users will not be able to submit and more of the same exercise unless it is planned to be in the day more than once.

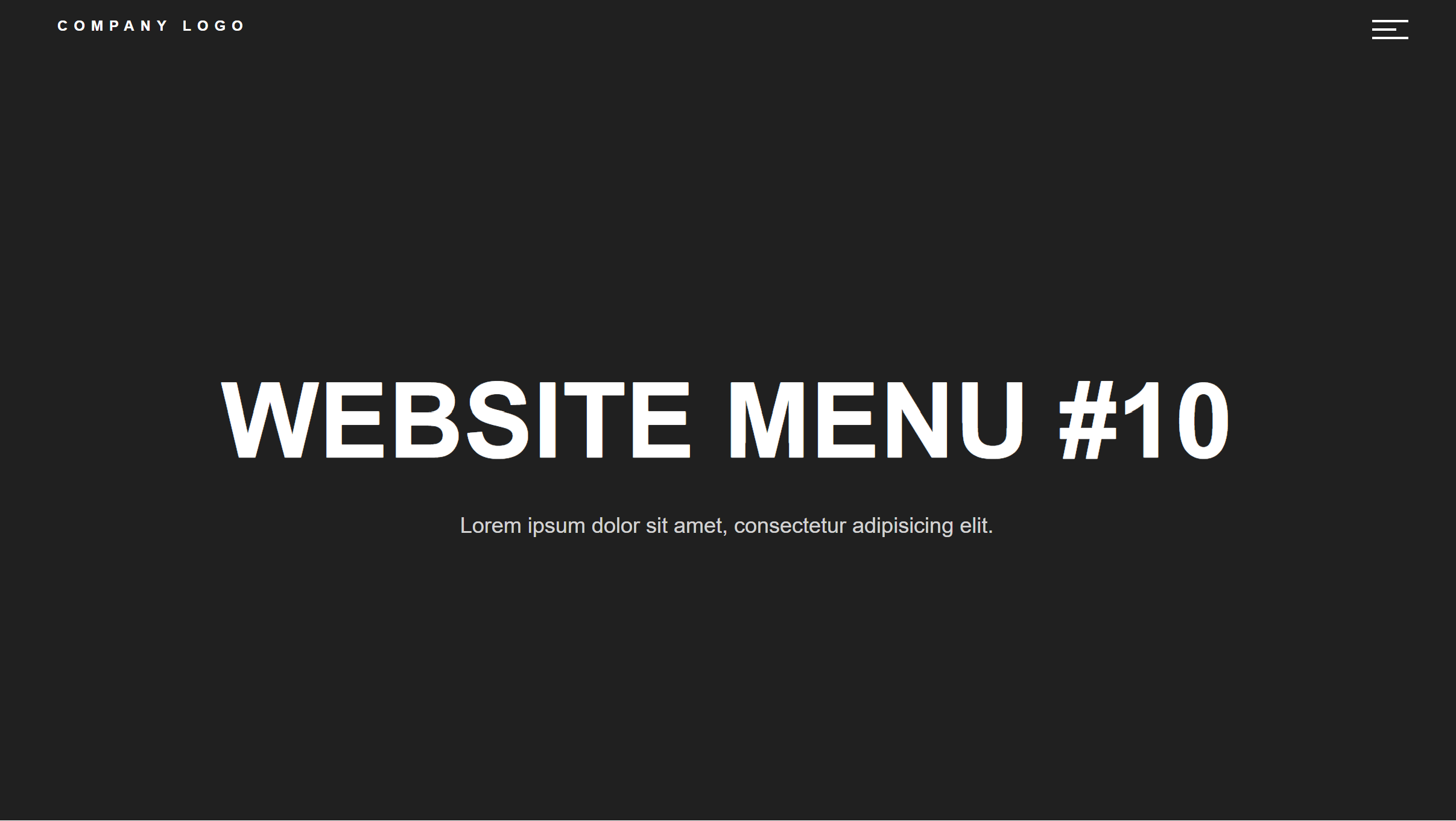
On the ‘My Plan’ page the user will be greeted with a day by day breakdown of their plan in a list style layout. This is simular to the strong app example on the right. Strong allows users to log workout info and change sets and weight in this area, however I will have this functionality on my ‘Today’ page. The ‘My Plan’ page is purely an area to view the overall plan and decide if you like it. Upon loading this page for the first time the users plan will be generated and saved. If the user doesn’t like their plan they can choose to regenerate the plan via a button at the bottom. Alternatvley will be able to manually edit exercises or move days around to fit their needs.

### 

### Stopwatch:

There will also be a simple stop watch page which the users can use to assist in tracking exercise performance. The stop watch will have large numbers for readability, there will be a start, stop and reset button to control the functions on the stopwatch.

### Navigating The Website:

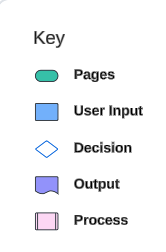
The website will be navigated via a hamburger navigation bar which will open to the center of the screen and cover hide the rest of the website behind. Page navigation options will be similar to the presentation below. The logout option will also be available here.

When hovering on the hamburger icon the icon will change colour to show the user that it is a clickable button. The users username will also be on the navigation bar the user knows what account they are logged into. When clicking on the users name it will relocate the user to the main ‘My Plan’ page.

### Processes in the website:

Below is a more in-depth map / flowchart for the processes inside the website this shows decisions and subprograms that will be run at different points throughout the website. Users will start at the index page that is the login page. Here users can login or navigate to the signup page.

Each of the subprograms and decisions are explained in further flowcharts below.

On each page a subprogram will be called that checks if the user is already logged in by checking is a session variable is set. If the users session variable is set on the login or signup pages the users will be redirected to either the questions page or the my plan page depending on if the questions have been answered before. If the session variable is set on any other pages no redirects will happen. If the login or signup page is loaded while the session variable is unset no redirects will happen but on any other pages if it is unset the user should be redirected to the login page. Upon logging in the session variable is set to store the users username which can be used throughout the program to interact with the database where the primary key is the users username.

# 

## The Plan algorithm / Class:

### General Rules:

* How many days a week are you available?

Users will be planned to do something in every day they have available up to a maximum of 5 days a week of training. They will also only be allowed to be 3 strength training days in a week and they cannot be in succession.

* What is your current training focus/goals?

Focuses can be on strength or technique. People who wish to focus on technique will be planned more on wall climbing and less strength and conditioning work. Users who select a strengths focus will be programmed more strength training and less on wall training.

* How much training experience to you currently have?

Users with more training experience will be programmed more volume as their body’s are more adapted and their recovery capacity will be higher.

Experienced climbers will also be programmed more difficult training techniques that beginners may not be capable of completing with correct form.

All training plans will be tailored to how the user answer the questions above. Certain days a week will also include optional conditioning exercises.

All Exercises will be stored in an exercise .json file with information on the exercise.

"name": "Pull-up",

      "equipment": "Pull-up bar",

      "difficulty": 3,

      "type": "strength",

      "reps": 10,

      "sets": 3,

      "measurement": "weight"

The Plan generated by my algorithm will be stored in a json files named the same as the username which can be read and displayed to the user where necessary. Once a user’s plan is created for the first time a new one should not be generated there after.

When the plan is displayed it will be as a table with days numbers as titles. Rest days will be displayed as a strong stating “Rest Day”.

# Implimentation:

## Index Page (Login):