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

Table of Contents

[Analysis 2](#_Toc162829946)

[The Problem: 2](#_Toc162829947)

[The stakeholders: 3](#_Toc162829948)

[Solving the problem computationally: 3](#_Toc162829949)

[Abstraction: 3](#_Toc162829950)

[Decomposition: 4](#_Toc162829951)

[Thinking Ahead: 4](#_Toc162829952)

[Logistics: 5](#_Toc162829953)

[Procedural and Object Oriented Thinking: 5](#_Toc162829954)

[Conclusion: 5](#_Toc162829955)

[Software and Hardware Requirements: 5](#_Toc162829956)

[Limiting Factors: 7](#_Toc162829957)

[Research: 7](#_Toc162829958)

[Interview: 7](#_Toc162829959)

[Existing Solutions: 9](#_Toc162829960)

[1 – Crimpd: 9](#_Toc162829961)

[2 – Gym Tracker: 10](#_Toc162829962)

[Questionnaire: 11](#_Toc162829963)

[Proposed Solution: 13](#_Toc162829964)

[Success Criteria 14](#_Toc162829965)

[DESIGN: 17](#_Toc162829966)

[Website Arrival: 17](#_Toc162829967)

[Login: 17](#_Toc162829968)

[The Website Flow Chart: 19](#_Toc162829969)

[19](#_Toc162829970)

[Question Pages: 19](#_Toc162829971)

[My Plan: 20](#_Toc162829972)

[My Progress: 20](#_Toc162829973)

[Start Exercise / Today: 20](#_Toc162829974)

[21](#_Toc162829975)

[Stopwatch: 21](#_Toc162829976)

[Navigating The Website: 21](#_Toc162829977)

[IMPLIMENTATION STAGE 22](#_Toc162829978)

[Processes in the website: 23](#_Toc162829979)

[Subprograms: 24](#_Toc162829980)

[Session Check: 24](#_Toc162829981)

[End Session: 24](#_Toc162829982)

[The Plan algorithm / Class: 24](#_Toc162829983)

[General Rules: 24](#_Toc162829984)

# Analysis

## The Problem:

For my project, I am designing a website where a user can create an account and the website will develop a climbing training plan. 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 target audience is climbers looking for a free method to further their training. This will mean my main audience will be experienced and/or dedicated climbers with an interest in improving.

There should be few language and other interaction complications since being a website the language can be translated by the browser. This mean the limit to my audience is the number of interested climbers.

I’ve selected a potential user to represent my target audience, Ben Cook. He’s a 17-year-old climber with a few years’ experience in climbing training. He has a large interest in climbing training but has never used structured plans due to the accessibility limits such as money and time. Due to this ben has struggled with numerous injuries and plateaus during his climbing. Using a free climbing plan maker Ben will be able to organize his training and avoid injuries and maintain stable progress.

I have frequent contact with Ben since we climb together on a regular basis this will allow me to update and adapt my product as its being developed to make sure it is tailored to the potential audience.

## 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.
* For the languages I will use the normal network stack, HTML, CSS and JS. However, I will also use PHP since I have a lot of experience with PHP and it is very effective at modelling data and interacting with databases.
* The outputs from the site will be visual and inputs will be mostly buttons.

### 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.

### Procedural and Object-Oriented Thinking:

The method of thinking about code can either be procedural, object oriented or both.

Procedural think involves breaking down a problem into smaller problems as described in decomposition and coding each of these mini problems as callable functions. This limits the need for code to be repeated and makes debugging easier since each function does a specific task.

OOPs model organizes design around data as objects and defined how objects interact with each other rather than functions and logic. Each object has its own attributes and behavior.

For my website I will use a mainly procedural approach but incorporating OOP in certain algorithms such as connecting to the database and creating the user’s plan. Using OOP to create the users plan makes sure the user’s data is handled safely and is encapsulated. It also means that multiple plans can be created as different objects without having to repeat code or use while loops.

### Conclusion:

The examples above demonstrate how solving this problem via computational methods is appropriate. Using computers to solve this problem is highly effective and efficient, making it much easier than alternative methods such a pen and paper.

If my solution is effective users and stakeholders will be able to use my website to generate a tailored training plan to help them improve their climbing while reducing the risk of injury.

## 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.

Hardware requirements:

|  |  |
| --- | --- |
| Hardware Requirement | Justification |
| Monitor or other digital display | To display the website |
| Mouse or Touchscreen | To interact with website |
| Pentium 4 processor or better | To meet standard web browser requirements |
| 100MB Secondary storage space | To meet standard web browser requirements |
| 128MB RAM | To meet standard web browser requirements |
| Windows 7 or Higher (IOS 10.5.6, Linux released after 2010) | To meet standard web browser requirements |

Software Requirements:

|  |  |
| --- | --- |
| Software Requirement | Justification |
| For general usage of website, no additional software is needed that is not included in the default web browser. | Web Browser to access and load website |
| For Designing and Coding the Website | |
| XAMPP | To Host webserver and database |
| Apache | Included in XAMPP, webserver host |
| MySQL | For the Database |
| Visual Studio or alternative editor | To write and edit code. Additionally provides debugging and other IDE extras |
| Web Browser | To Open webserver |
| Git (optional) | For version control |
| PHP | So that PHP can be run |
| Optional extensions to be used during development | |
| CSS Peak | Easy of coding CSS |
| Git Graph | Easy of version control |
| HTML & CSS Support | Additional html and CSS debugging tools |
| IntelliSense for CSS | Easy of CSS coding |
| Jupyter and extras | All around ease of coding improvements |
| Live SASS compiler | To convert SASS to CSS where needed |
| Path IntelliSense | Ease of PHP coding |

## 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:

### Interview:

The interview will be with my stakeholder and will allow him to influence the features that could end up in the final product. I’ll be asking him simple questions that will allow me to understand what is necessary for the website. Ben’s answers are shown in a different colour and have been condensed to bullet points for simplicity.

**What would you like the website to take into account when designing your plan?**

* **My Availability**
* **My Goals**
* **My experience**
* **My Current Weaknesses**

**Is tracking individual exercises and performance overtime important to you?**

* **Yes**
* **Tracking progress of exercises is important**
* **Progress should be easy to see by putting it on a graph or table**

**What are your favorite features of other solutions you have used previously?**

* **A Timer**
* **Lots of exercise choice**
* **Descriptions for exercises**

**Is it useful to still have options to manually edit your plan?**

* **Yes**
* **Being able to swap days is useful**
* **Being able to change exercises is essential**

**Is the presentation of the training plan important to you?**

* **Sometimes**
* **Functionality is more important**
* **Display should be simple and to the point**

**What is the most important feature for the training tool besides making the plan?**

* **Being able to view the individual day I am on so I don’t have to think too much when I want to start training.**

**How many exercises should be the max per day?**

* **7**

**How many days should be the limit for a week?**

* **5**

**How many strength training days should be the max?**

* **3**

**Is a login system important?**

* **Yes**

**Is the login screen presentation important?**

* **Yes**

**Should the login screen be simple or have lots going on?**

* **Simple with a relaxed design**

**How should the plan be displayed?**

* **Day by day**
* **As a table or list**

**Timer or Stopwatch?**

* **Either, both are useful**

**Collect history or all exercises?**

* **Yes**

**Is a password confirming box important for the signup page?**

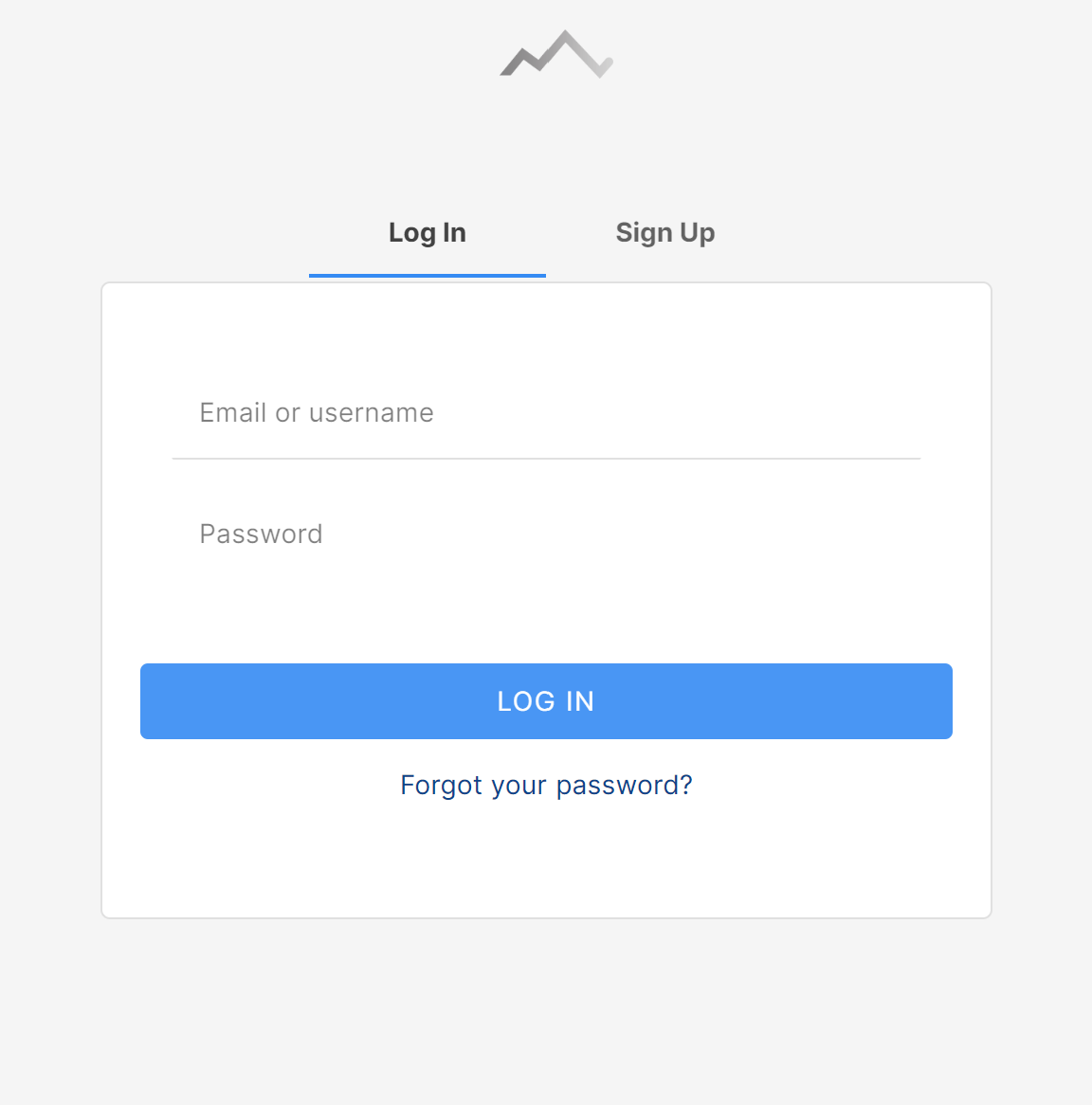
* **Yes**

**Should the signup page look similar to the login page?**

* **Yes**

### Existing Solutions:

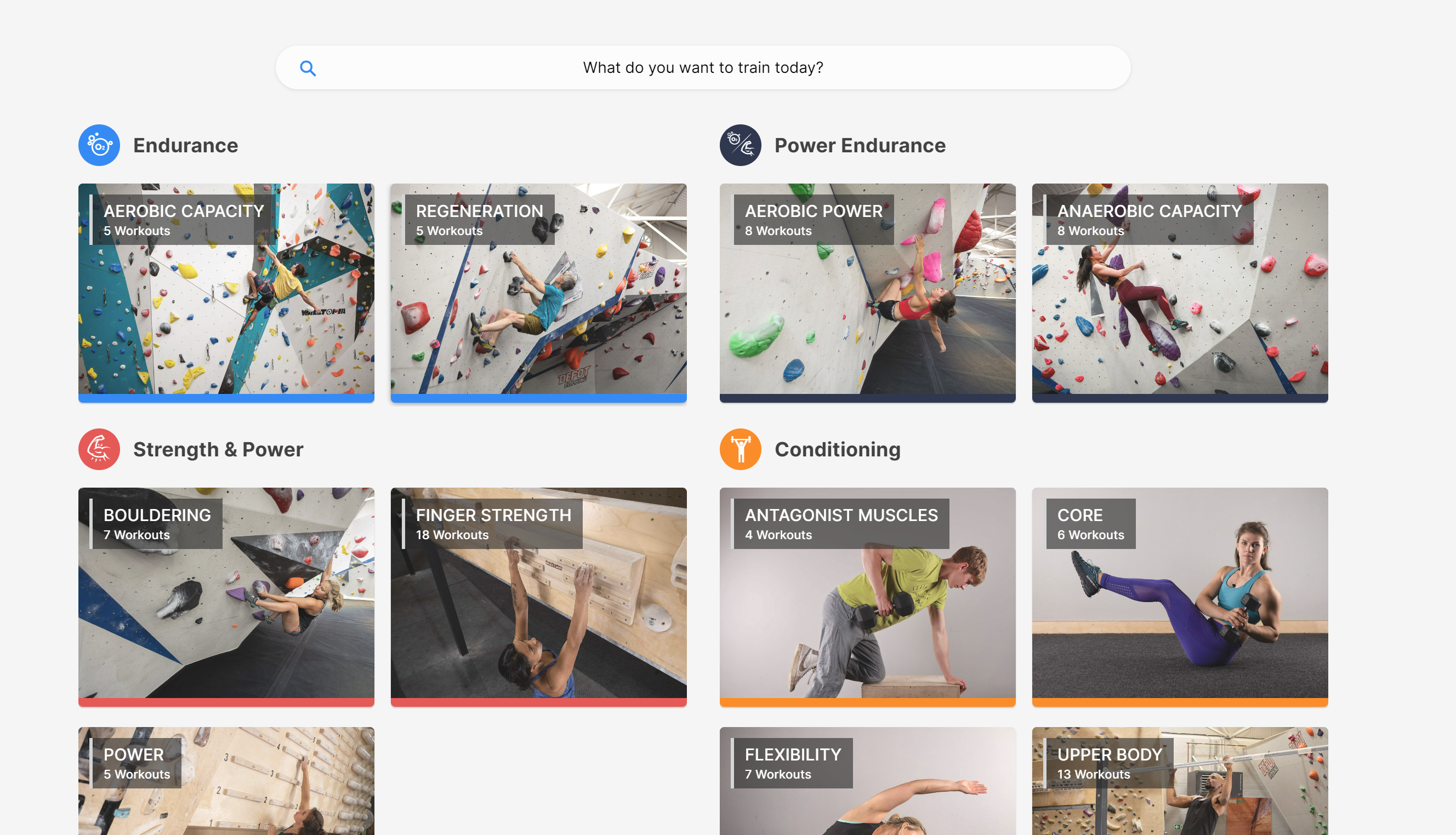
#### 1 – Crimpd:

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 I will 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.

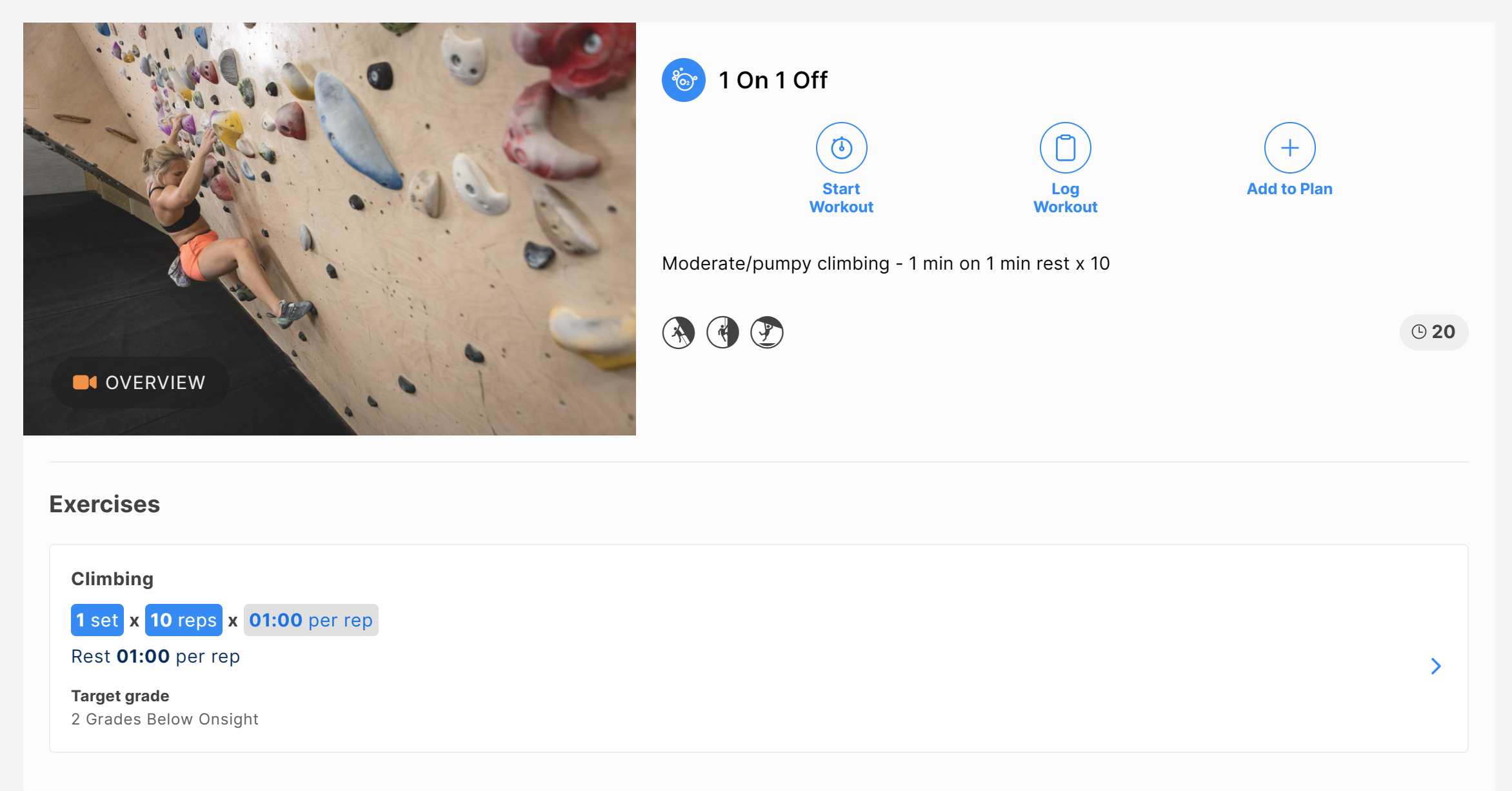
They also have an easy to find “Forgot your password?” which improves verification of user via email confirmation.

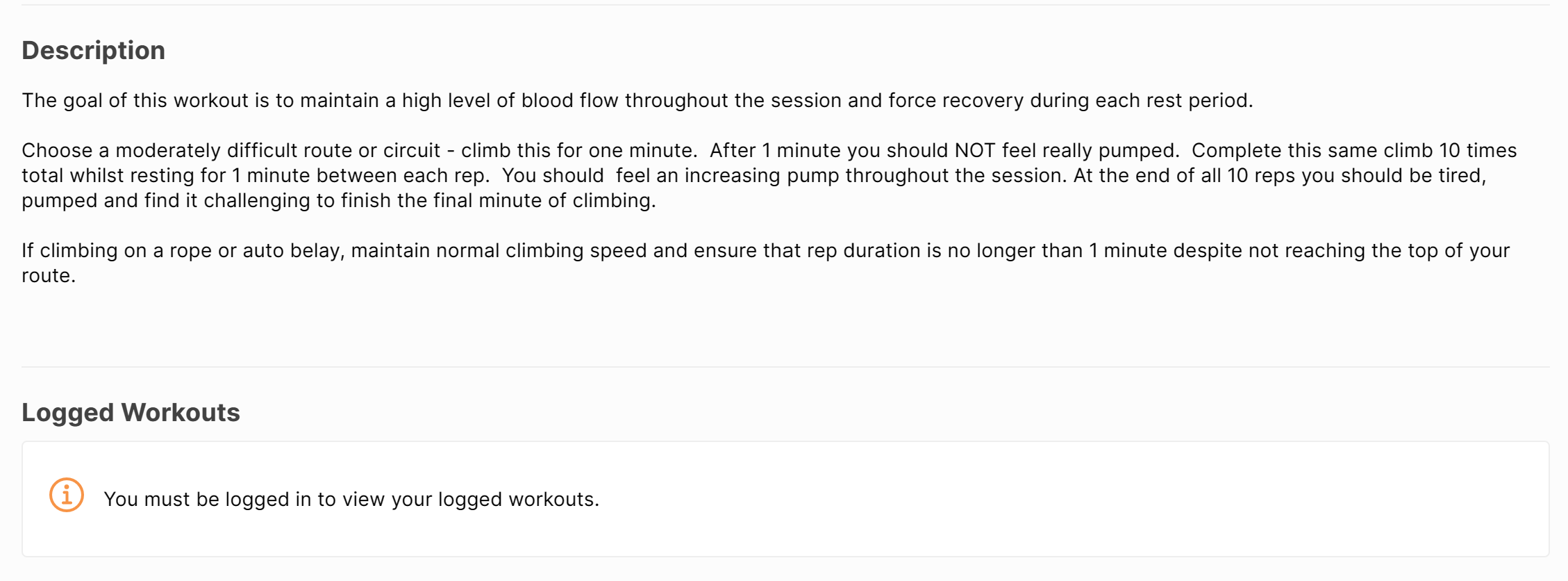
This is Crimpd’s home page, it is divided into types of workouts with simple names and logo coordination. This makes it simple to find a specific workout to complete.

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

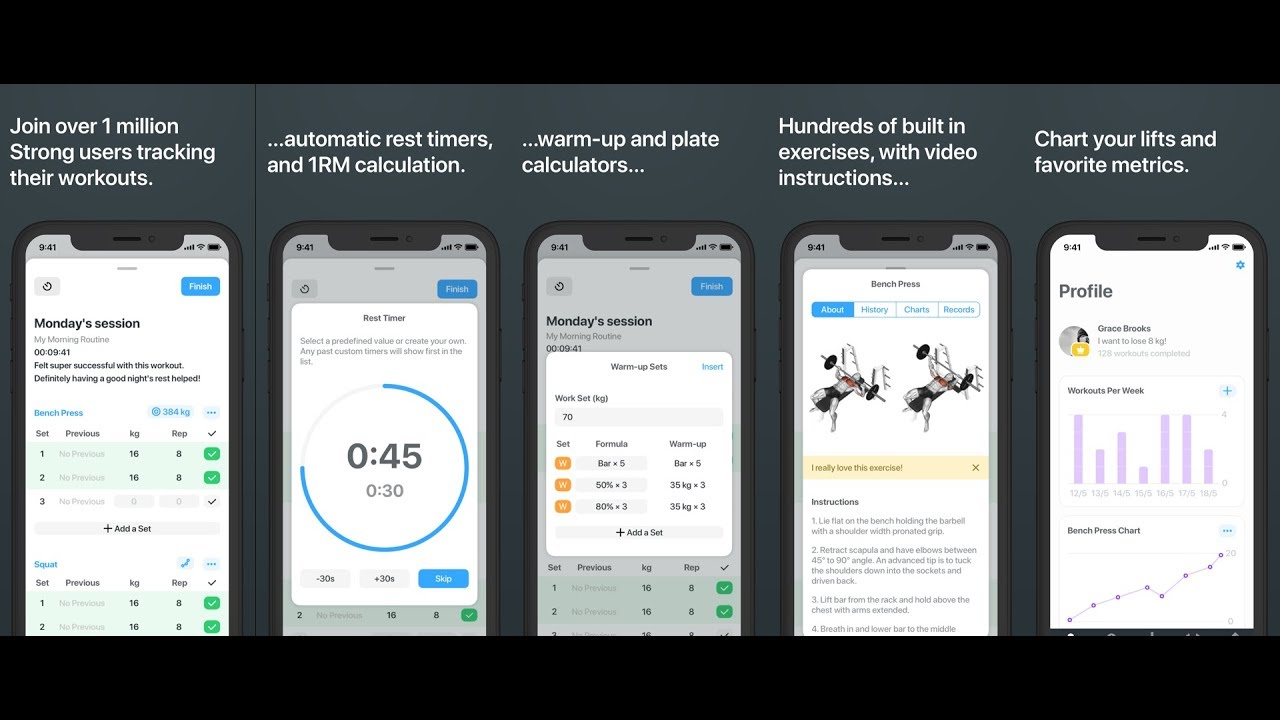
By having simple photos of exercises as backgrounds it makes it easy to understand what an exercise is and allows beginners to understand what the name of an exercise means.

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.

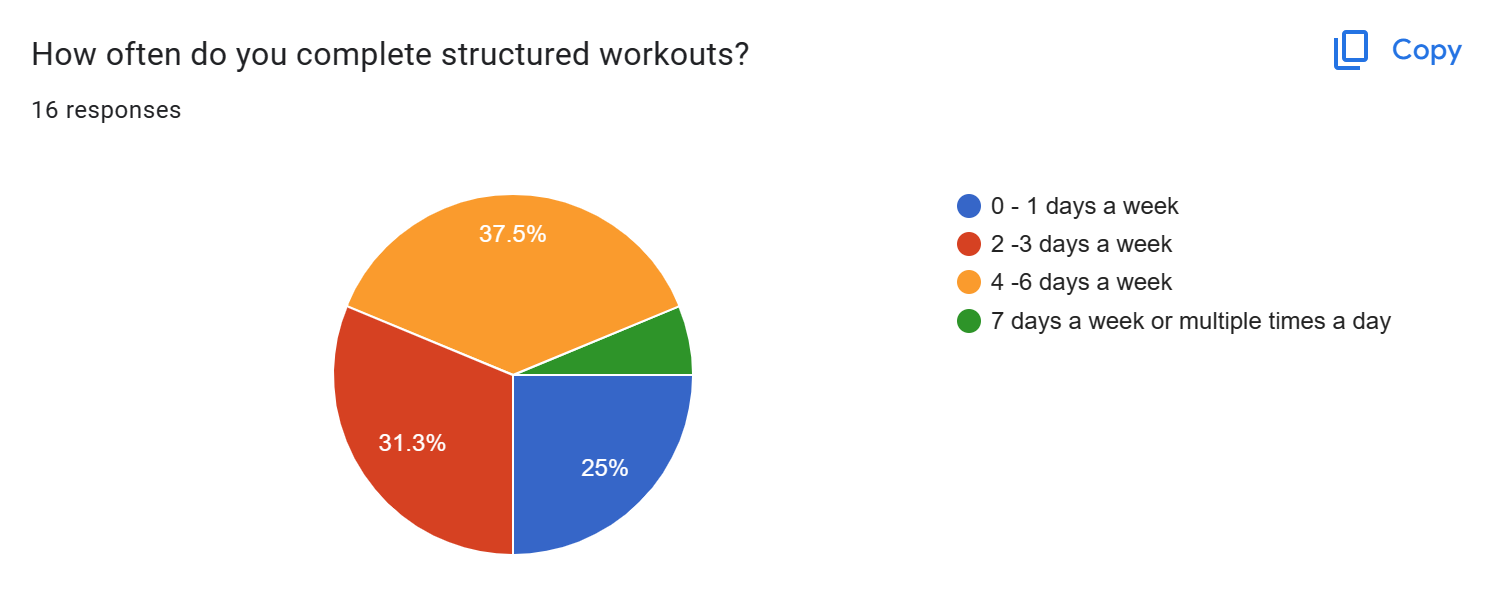
It also has a demonstration video available to watch, this makes it easier to understand for users. They also have a basic description of sets, reps and times. This makes it obvious how long a user will spend on an exercise.   
They also provide a brief description of what the exercise achieves and how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

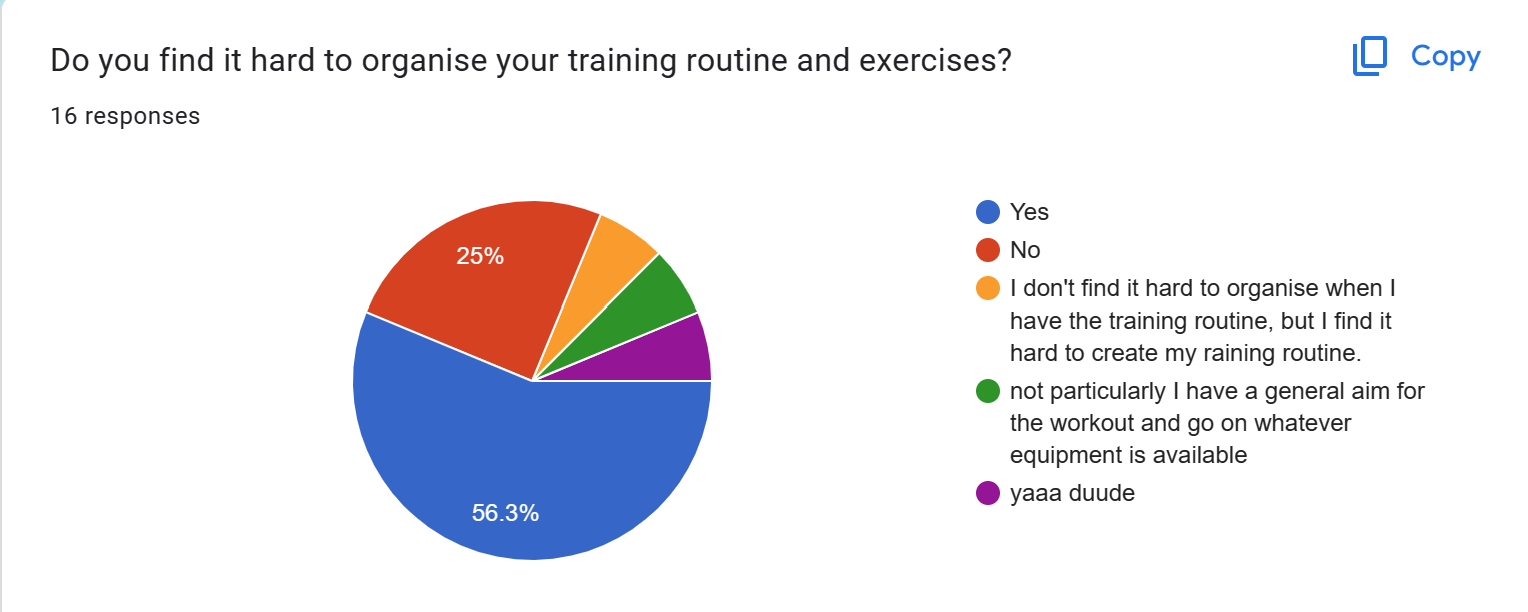
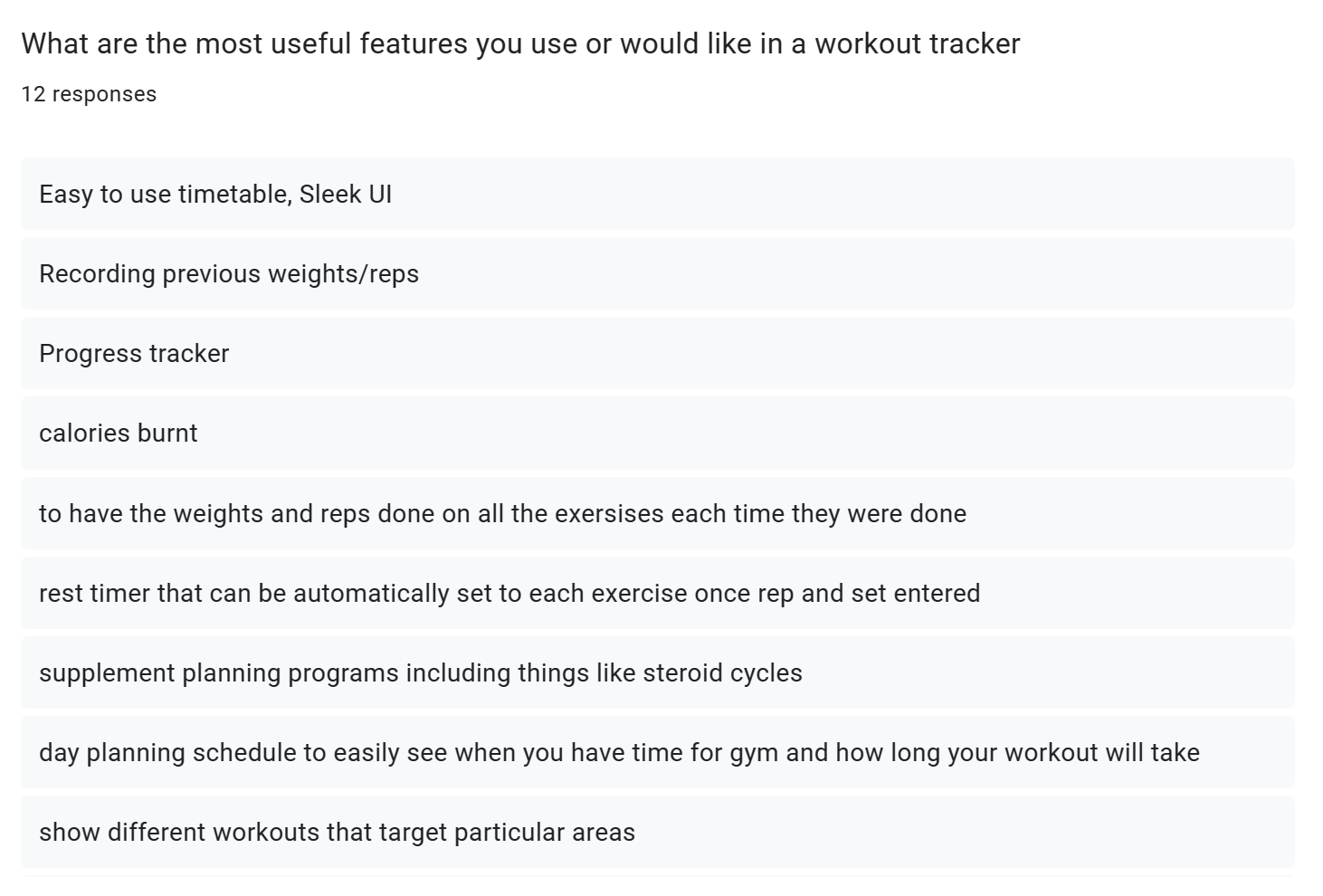
#### 2 – Gym Tracker:

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.

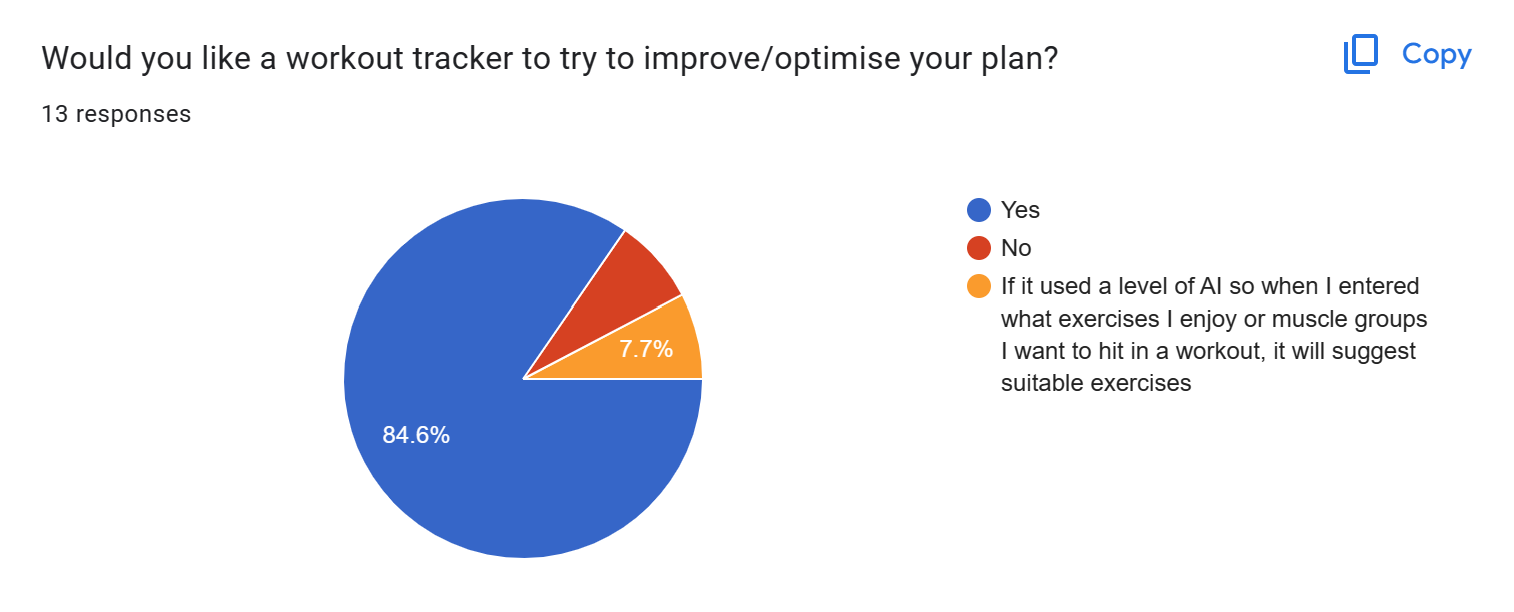
Strong workout provides a plethora of features with a sleek and clean user interface which promotes ease of use.

### 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 are reasonable demand for a product like mine.

## Proposed Solution:

These are the main features proposed for the website as learned from the questionnaire, interview and previous solutions.

|  |  |
| --- | --- |
| Login Page | Simple, maybe a picture background. Focus on the login area no clutter. |
| Signup Page | Similar style to login page. Password confirm is important. |
| Question Pages | Simple, questions: Availability, goals, experience, areas to improve. |
| Plan Algorithm | 3 strength days max, 5 days total max, should tailor using questions. 7 exercises per day. Lots of exercise variation. |
| My Plan page | Simple display, table or list. Should allow manual editing of plan. Include exercise descriptions. |
| My Day page | Shows individual page, allows tracking of each exercise. Stopwatch or timer here? Simple display like My Plan page. |
| My Progress page | Graphical display of progress for all exercises |
| Stopwatch Page | Stopwatch and timer |

## Success Criteria

|  |  |  |
| --- | --- | --- |
| Criteria | Justification | Reference |
| Index Page | A landing page for the website | Feature of all websites |
| Login Page | An area for users to login to their personal account | Interview, Feature from existing solutions |
| Signup Page | An area for new users to create an account | Feature existing solutions |
| Questions Page | Place for users to answer questions needed for the plan generator to function | Interview |
| Max 3 strength training days in plan | Avoids overtraining | Interview |
| Max 5 total days in plan | Avoids overtraining | Interview |
| Max 7 exercises per day | Avoids overtraining | Interview |
| Question about availability | Allows algorithm to know the user’s availability | Interview, questionnaire |
| Question about goals | Allows algorithm to know the users focus | Interview, existing solutions, Questionnaire |
| Question about experience | Allows algorithm to know the users experience | Interview, Questionnaire |
| Logout option throughout website | Let’s user’s logout of their account at any point in the website | Interview, Existing Solutions |
| My Plan Page | Page to display users training plan | Interview, Existing solutions |
| Plan displayed as table or list | Makes plan easy to read | Interview |
| Plan separated by days | Makes plan easy to understand | Interview |
| Simple Login and signup page | Makes process easy to follow when logging in of signing up | Interview |
| Manual editing of exercises in plan | Means users can manually tailor their plan to their preferred exercises | Interview |
| Manually editing of days in plan | Users can rearrange plan days to suit their needs | Interview |
| Each exercise has a description | Helps used understand how each exercise should be performed | Interview |
| Individual day page | Shows the user the training scheduled for today | Interview, Existing solutions |
| Save data from workout | Allows user to track progress and avoid plateaus | Interview, Existing Solutions |
| Progress Page | Displays the users progress over time | Interview, Existing solutions |
| Progress displayed as list or graphically | Makes understanding the data collected from exercises easier | Interview |
| Stopwatch page | Allows user to measure exercises | Interview |

# 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.

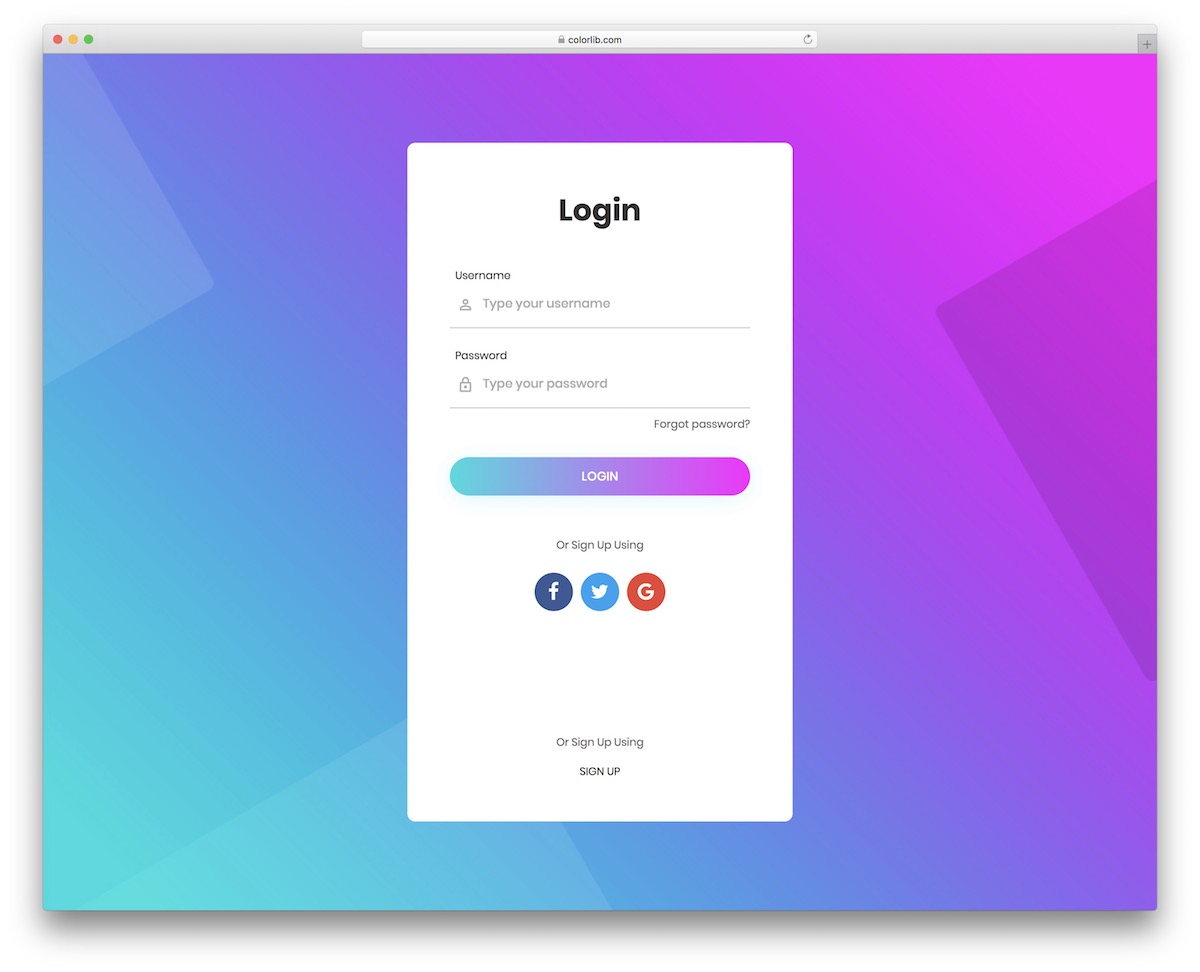
## Login:

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 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 which 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 questions 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?

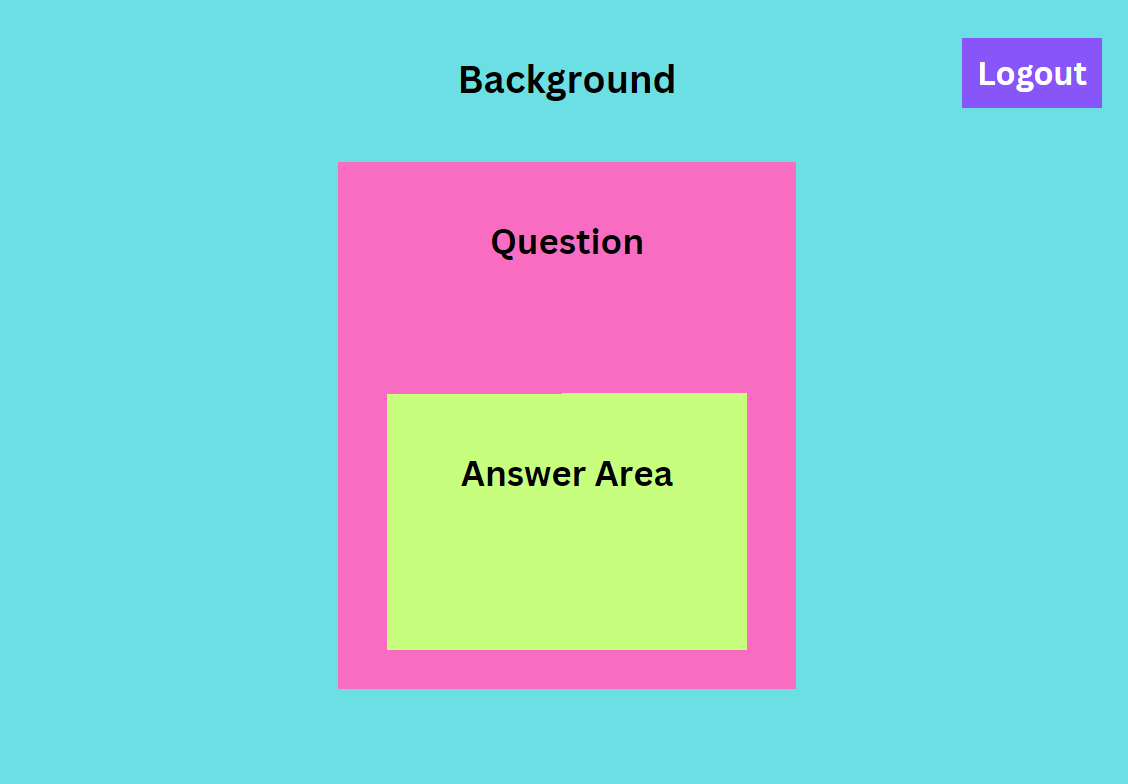
Users will also be prompted to input data such as current weight, height and also current PBs and records they have is certain lifts/exercises.

## The Website Flow Chart:

## 

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.

## Question Pages:

The question pages will be very simple with the focus being on the question. To apply this the question pages will have similar appearance to the login page with a simple background and a content section in the center where the question and an answer area are displayed.

A logout button will also be in the corner so that users can logout during the question answering process an come back to it later. If the user clicks this button they will be redirected to the login page.

## Fitness App Review: My Workout Plan by Sosis Apps - Dammit Karen

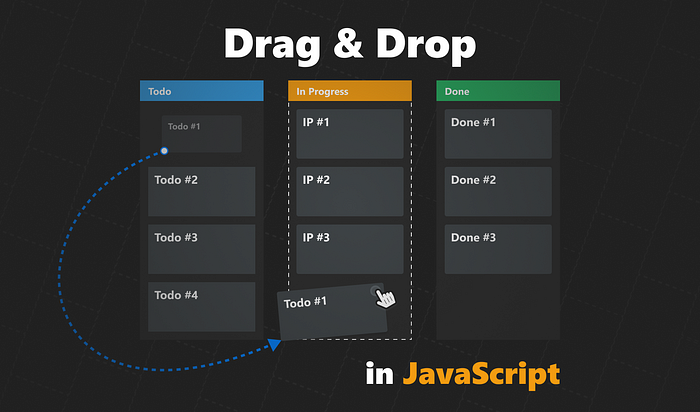
## My Plan:

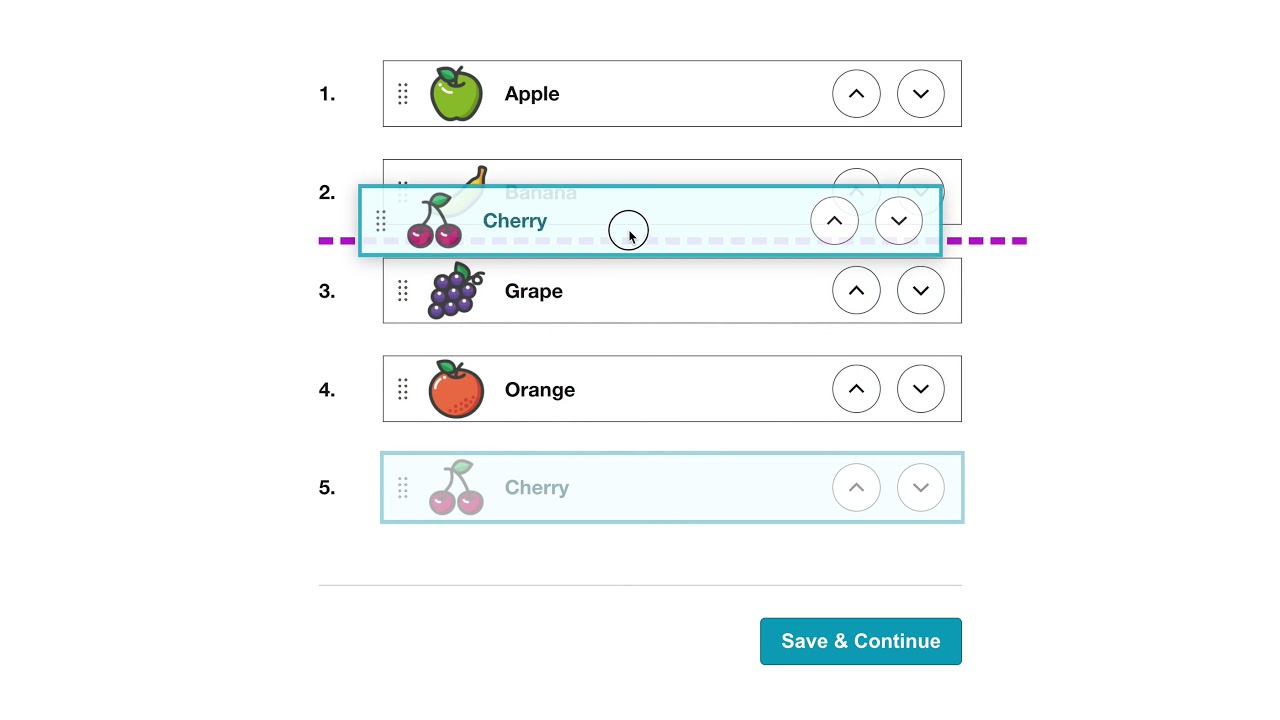
Users will have access to their plan to view or edit details. In this area, they are presented with a calendar on which are events for workouts. When selecting an event, they can edit manually or run the workout manually which directs them to the start exercise page.

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

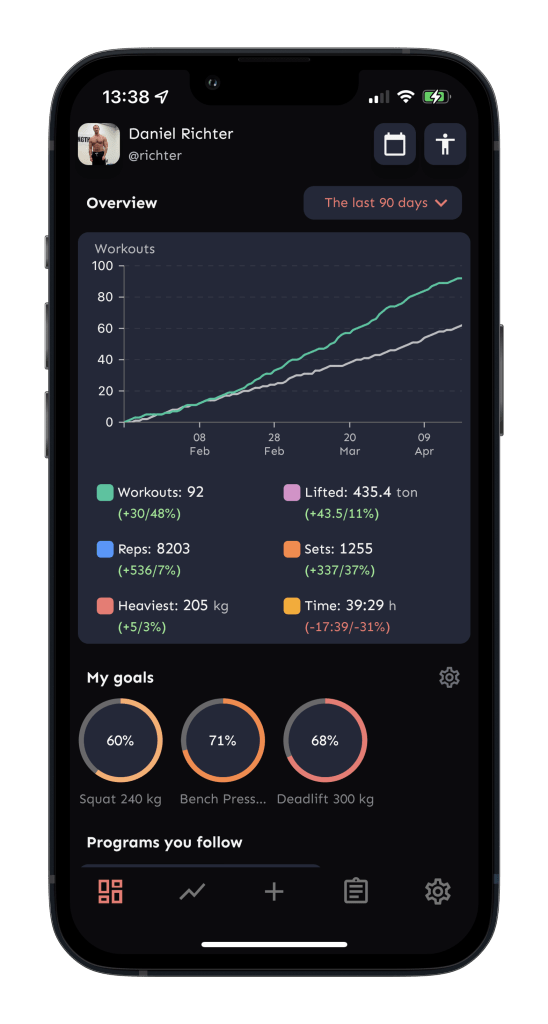
Similar to the example on the right where each day is displayed in a list style for users to see. I will design a similar solution for this page that will allow the user to easily understand their plan.

In this area users will also be able to expand exercises to get a description and more detail. An example of how this will work is in the lorem Ipsum example on the left. Using an accordion style bar for each exercise the user can click on an exercise and then any additional information can be displayed in the expanded accordion.

When a user edits their plan manually the json file containing the plan will have to save these changes so that they will be kept upon a refresh. If the user changes two days around they can be swapped in the json file. If the user swaps out an exercise or adds a new on the exercise must be located in the exercise directory json file and then all its details can be added to the training plan in the correct place.

To move exercises and items around the plan users ideally will be able to drag and drop individual items and move them to the desired location. However, the implementation of this will be difficult and if this becomes too difficult to code another solution will be for users to select two exercises or two days and select a swap button to swap their positions. Similarly, to add exercises users can select a location and select an add exercise button. The idealistic solutions are shown in the pictures above where users can drag and drop items to rearrange them.

## 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.

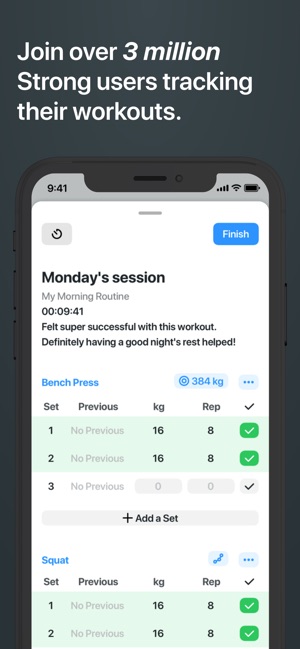
This area is useful for users to visually see them progress which is important for motivation. This data is also useful to the program to continually update the plan and suggest improvements.

My design will be similar to the solution on the left however my solution will additionally show the progress users make on individual exercises.

Simularly to the My Plan page each exercise will be expandable via a accordian menu. Each exercise will have an accorian item and when clicked on will open to display the progress information about the exapanded exercise. The expanded info will contain the graphical data and or a list alternative. An accordian menu is the best solution for this since over the course of training plans the user will collect information about many exerices and displaying all exercises graphs and data at once will be overwhelming and hard to understand for users.

Another solution would be to display the most frequently erformed exercises and add a search bar for users to find their desired exercises if they are no aready displayed.

## Start Exercise / Today:

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

The example solution on the right is from a generic gym workout tracker this will be similar to my solution.

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.

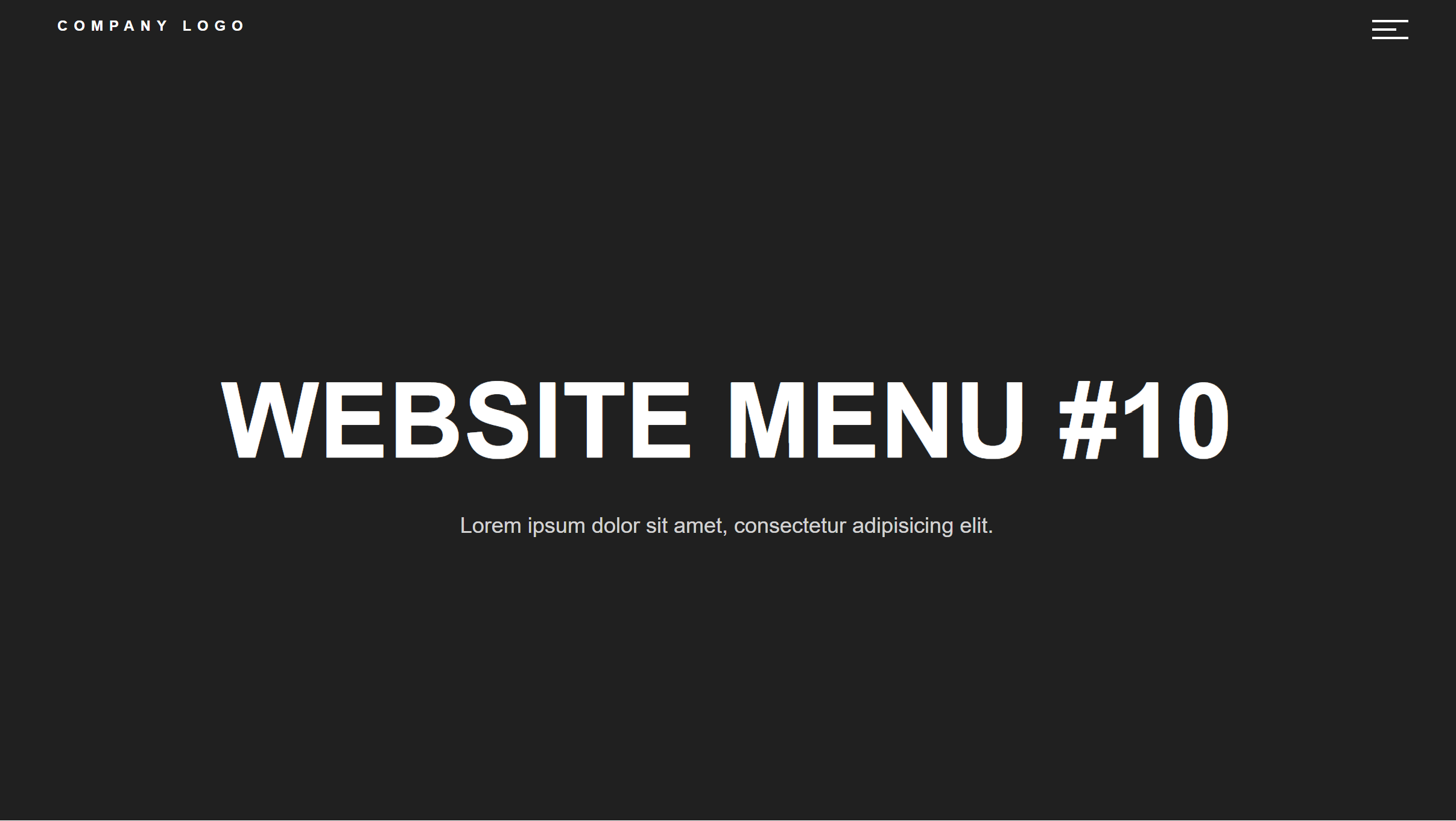
On the ‘My Plan’ page the user will be greeted with a day by day breakdown of their plan in a list style layout. This is simular to the strong app example on the right. Strong allows users to log workout info and change sets and weight in this area, however I will have this functionality on my ‘Today’ 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 user’s 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.

## Logout Button:

On every page in the website apart from the login and signup page there will be a logout button or option. This will allow users to end their session, this is important for the user’s data protection.

On the question pages a logout button such as on the left will be displayed in the corner, this can be clicked to log the user out.

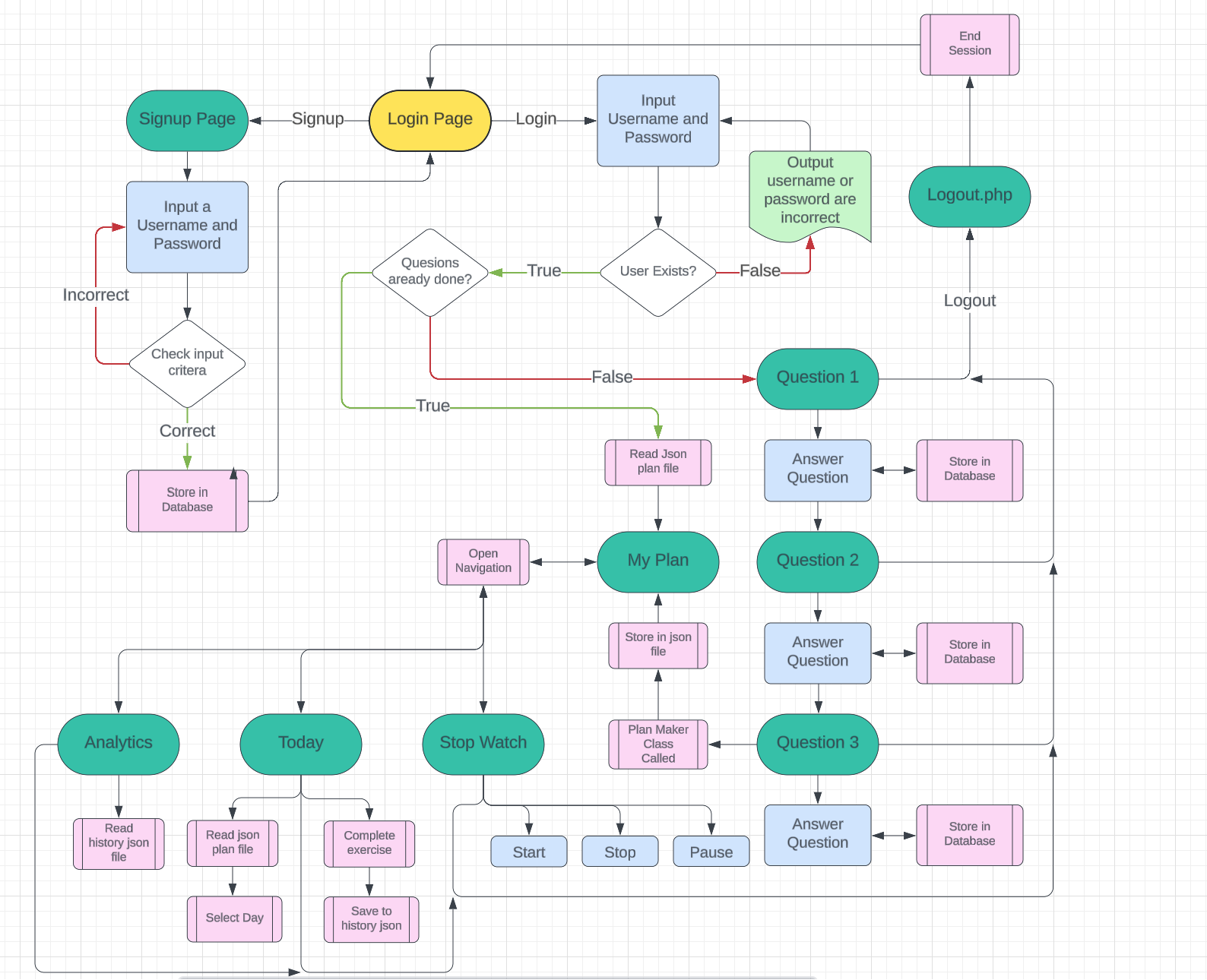
On other pages such as ‘My Plan’ the user can logout via the navigation bar. The last option in the navigation bar will be a logout option that will end the user’s session.

After logging out the user will be redirected to the login page.

# 

# IMPLIMENTATION STAGE

## Processes in the website:

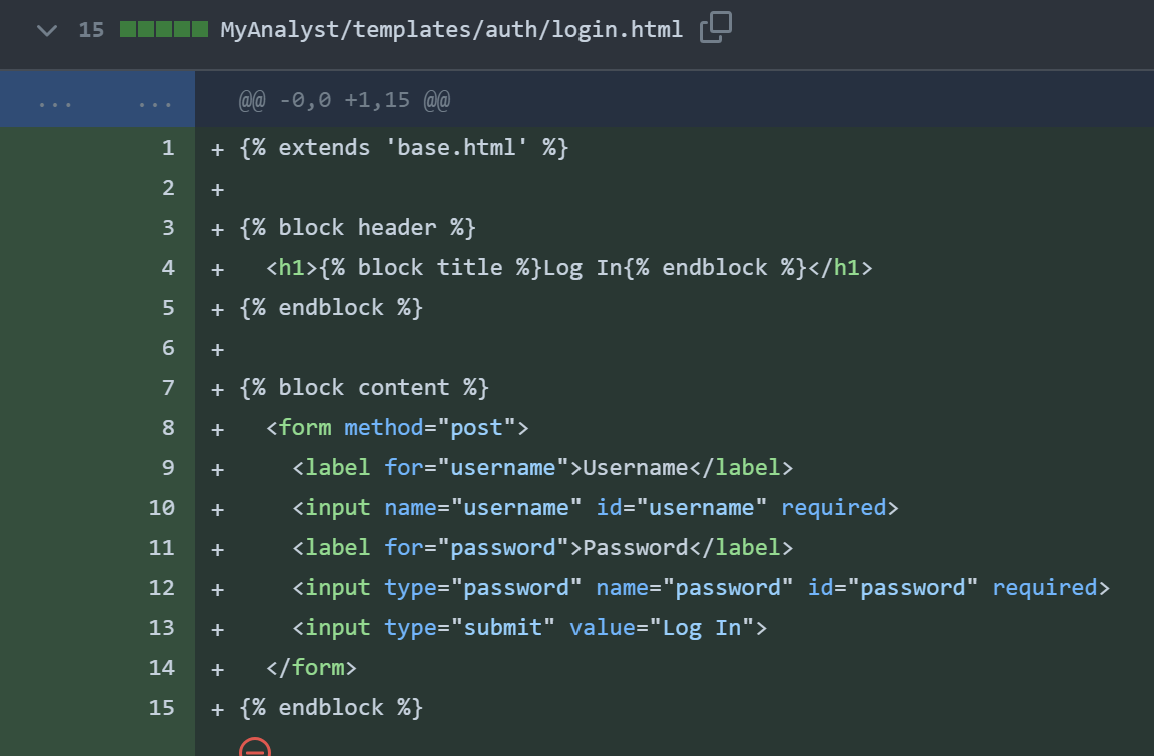
Below is a more in-depth map / flowchart for the processes inside the website this shows decisions and subprograms that will be run at different points throughout the website. Users will start at the index page that is the login page. Here users can login or navigate to the signup page.

## Git Repository:

For my project I decided to use git as a way of managing versions and being able to keep track of my decisions and progress over time. I created a GitHub Repository that stores all the fields from my project. This also allowed me to roll back to previous versions easily if things break during coding.

## XAMPP:

## Initial Approach:

My initial solution for my website was to use flash which is a web framework built around python. I chose this since I had experience and knowledge of python and would not take much learning to adapt to flask.

After experimenting with flask and building some basic pages to experiment with flask I decided that my original approach discussed in the analysis was better and I switched back to the normal web stack.

On the left was my basic login page created using flask. This worked as intended but lacked functionality. After this experiment and also creating a signup page is decided php and normal web stack was more suited to my design needs.

## Login Page:

After switching back to html, css, js and php I created a new template for the login pag

## Common Subprograms:

Since I’m using a mainly procedural approach in my programming my code is bult up of many subprograms that each perform a single task. These can be called when they are needed. This means my code has less repetition and is easier to debug.

### Session Check:

On each page a subprogram is called that checks if the user is already logged in by checking is a session variable is set. If the user’s session variable is set on the login or signup pages the users will be redirected to either the questions page or 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 user’s username which can be used throughout the program to interact with the database where the primary key is the user’s username.

### End Session:

This subprogram is called when the user logs out at any point while using the website

## 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 bodies 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 thereafter.

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”.