**Description:**

A interactive website that can record, plan, calculate, and give suggestion to users for a healthy living style. The type of stuff that can be recorded our like calories intake to reps done during a workout. The plan is like a calendar where you can schedule and have a consistent way of knowing what you will be doing and what you have done. As for the calculate it will do calculation from how much calories you burned to how much reps you made during a set period of time. Finally the suggestion will look over your record in our database and will give appropriate feedback to you like what kind of meal you should eat.

**Targeted audience and objective:**

Targeted audience is for any users that can use our website in a daily basis.

The objective is to helps users keep track of their healthy lifestyle.

**List of features:**

Route Tracker

Assigned feature: Christopher Maeda

Brief description: This feature will take in 2 different marker on a map (address) and will draw up a route connecting the 2 marker (like google maps). The route will calculate the distance and tell you an approximate time it will take you to complete it. The time would be calculated by the distance and a speed that user would be able to input or take a worldwide average speed. You will then be able to save the route on to a database to be able to pick that same route like a daily morning run route.

Programming logic: There are 2 different ways of using this feature. The first way is a user will pass (input) in 2 address and the feature will create a route from the roads that user can walk/run on connecting the 2 address. The output will show the route on a map which shows the distance of the route and a time which can be taken as another input from the user otherwise use a average walk/run speed. The other method of using this feature is accessing the database where it will grab all saved routes and list them out for the user to choose from. The output will be the same from the other method when the user has chosen one of the routes given.

Calories Calculator

Assigned feature: Christopher Maeda

Brief description: This feature will tell you how much calories you have accumulated on that day. It will also tell you how much calories you have burned on that same day. The calculation is done by default information on the account (height and weight), the workout you have logged, and meal you have ate. The calories data would be stored daily in a database.

Programming logic: The input are 4 integer values (form). 2 of which are default value (height and weight) found in user account (account database). The other 2 are the calories intake that day and calories burned that day. The 2 calories integer value can be found by different ways like user manually input it or have it look up the users database of meals (calories intake) and workout done (calories burned). After having obtained the 4 input value there will be a formula that calculate how much calories that was accumulated that day (formula can be found online). The output will be a integer value that prints to the user how much calories they have accumulated.

Events Calendar

Assigned feature: Christopher Maeda

Brief description: A feature that would notify the user any upcoming events that will happen locally like a local marathon next week. The calendar can also be use to log any events the user wants to put like a gym trainer meeting this Wednesday.

Programming logic: This feature can have 2 different input methods. One is where user fills out a form and add any events they want to log in the calendar. The process is done by reading the form they submit like the name of the events with times (start and end). When the form is submitted a output will be shown where they can see a event in their calendar (the view of the calendar is depended on how they set it like weeks and months). The other method is where they can see a list events locally (where their address is set in their account database or have a address be inputted). The listed events are within a certain date the user can input or by default be a month of the current date. The list of events the user can see can then choose which events they want to add to their calendar. The output will be the same as the other method where they can see the added event.

Track record of weight exercises

Assigned feature: Christopher Jones

Brief description: CRUD of data for the most common weightlifting exercises, with the weights/reps stored for a workout on a particular date. Ability to add exercises to the user’s personal list of exercises if it does not already exist. Exercise options are divided by major muscle group and equipment (barbell, dumbbell, machine, etc.)

Programming logic: Initial basic database CRUD for weight exercises, according to sets/reps/weight for each exercise per each workout. Initial build will have basic CRUD functionality and a list feature for the workouts on a calendar day, with more advanced functionality to allow workouts to be viewed across a different time scale.

Suggestion of what exercises

Assigned feature: Christopher Jones

Brief description: This feature on the MVP level will take one of a set number of common fitness goals, and suggest the proper weight exercises to do to accomplish these goals. On the increased complexity of this feature, the system will analyze what exercises the user is doing, and will suggest exercises in areas that are being neglected (i.e. the user is only logging upper body exercises, the feature will suggest that the user incorporates squats/deadlifts to their routine)

Programming logic: On the MVP level, there will be a dropdown menu with common fitness goals, and selecting an option will display good exercises to meet those goals. On the more complex state, the exercises that are logged will be associated with a major muscle group movement (upper body push/pull, lower body squat/lever, core), and the feature will look over the past workouts of the user, and suggest exercises in the major weight training exercise groups that are lacking, or more reps in the neglected exercises to even out the user.

Calculate how much you lifted

Assigned feature: Christopher Jones

Brief description: This feature will take the total tonnage of the weight lifted that you have logged, and will compile the total weight of all your reps/sets over a given variable period of time (day/week/month/year), and will display the total tonnage, as well as a motivational comparator of how much you’ve lifted (i.e. you’ve lifted the equivalent of 2 sedans this week), with a possible visual representation of the tonnage as well.

Programming logic: This feature will basically use a function to sum the tonnage of the weight lifted given a particular period of time (weight \* reps \* sets), and will then use a comparator chart with the given weight of some impressive objects, and will deduce a proper comparison from that chart.

Meal Suggestion

Assigned feature: Sam Bebenek

Brief description: Users can pick a diet type they are interested in and a list of meal suggestions will be presented based on their desired diet type. Some diet types may include low calorie, low carb, high protein, vegan, and more. Users can click on a meal they’re interested in, and will be presented with a new page containing an image of the meal, a short description, and a link to a recipe on an external site (potentially allrecipes.com)

Programming logic: It will contain simple CRUD functionality on the admin side to edit the database of meals. Users will select their diet type from a drop down menu, and a list will populate from a SELECT command on the database. The individual view pages will also be generated from a SELECT command. Three database tables will be needed. One for diet types, one for meals, and a bridging table between the two. This is because a meal can potentially be a part of many different diets.

Find a bro

Assigned feature: Sam Bebenek

Brief description: Users will be able to find a friend that is interested in going to the gym with them. In a user account, users can select if they want to be discoverable or not to other users. If they have chosen to be findable, then a list of other users in their city will be displayed on the find a bro page. If not, they will be prompted to change their settings on the find a bro page. A user can click on another user on the list, and will be brought to another page. If time allows for it, a messaging system can be implemented to facilitate contact between the two users. If not, a simple contact form can be filled out and will send the form data as an email to the proper recipient’s given email address.

Programming logic: Columns will need to be added to the users table to store address and ‘findable’ boolean data. If the user has set themselves to be findable by other users, a list will be generated from all database users that are both findable and are in the same city. If the messaging system is used, an extra independent messaging feature may be needed. If not, when the user completes the email form, an email will be generated and sent to the email address of the recipient, including the email of the sender so that further contact can be done through those user’s email.

Reward system

Assigned feature: Sam Bebenek

Brief description: As users continue to interact with the website and log their completed workouts, they will contribute towards a personal progression reward system. This progression system will emulate a progression system in video games, such that for each completed workout they will receive “experience”, and will “level up” once a certain amount of experience is received. A rewards page will show the user’s current experience and how much they need to reach the next level. Levels will be represented in 9 tiers: 3 tiers of bronze, 3 tiers of silver, and 3 tiers of gold. Each tier will have an associated medal icon, and that icon will be presented next to their usernames throughout the website and will be presented next to their username in the “Find a Bro” page. Other rewards can be presented on this page as well.

Programming logic: A class will be created to retrieve a user’s “experience” points and store all the information relevant to the progression system, such as the name of the medal images and the amount of experience required for each tier. A database can also be used to store this information, however it may not be necessary since this information will likely be static. This class will also contain a method to print the username and medal icon, and ones to retrieve a user’s experience and current tier. This class will be used to generate the rewards page.

Workout planner

Assigned feature: Joshua Silveira

Brief description: Create a weekly/timely planner of exercises. Users can either create their own custom planner or use the pre programmed suggested plans. Planners can be flexible ex.(3 times a week with rest days) or can be programmed to allow for multiple planners to run concurrently ex(A simple 4 day/ week plan but the rest days allow for a 5k run per week).

Programming logic: CRUD functionality for ‘planner’. A Plan class contains a week list array. Each index in the array represents a day of the week. Each day then contains a list of exercises/activities for the day or can be scheduled as rest. The planners can have their week repeated any amount of times, ex. a plan can be followed for 3 months. The planners display a workout for a week at a glance or individual days. Concurrent plans stack on the same day.

Goal Tracker

Assigned feature: Joshua Silveira

Brief description: A user can create a Goal with a specific timeline and intent. ex(Run a 5k in under 30 minutes in 5 weeks ). Users can also set smaller goals at intervals in their main goal timeline ex. (run a mile in 8 minutes in week 2). As users accomplish their goals the app congratulates/encourages them. Users can also pick preprogrammed challenges to accomplish instead of creating their own goals.

Programming logic: Goals have CRUD Functionality. A goal class contains an intent property and an end date. A goal class can also contain a list of goals each which their own intent and goal that are related to the parent goal. Goals are displayed as Lists with checkboxes to indicate their completion. Challenges appear as a dropdown in the Create Goal View as an option. Challenges have their own intent and an end date that is subject to change on the User's input.

Status Account

Assigned feature: Joshua Silveira

Brief description: User Accounts that contain all the main inputs required for the other features and provides updates on how healthy a user is and provides information at a glance. Ex.(Current Weight, height, Goal Weight , Goals, Current max exercise weights/Reps, Planner/Calendar, Email address). If the user is accomplishing their goals/exercises their status is shown to be healthy/consistent. If their goals are not being met their are shown encouragement/advice ex.(Set smaller goals to build up to a bigger goal)

Programming logic: A user can create their own account to hold all their information. A master user table contains all user ids. A user class holds important properties like Name and email address and has foreign keys to other features ex(Goals list). Users Can read their information or choose to update/delete it.

**Time line (Task and schedule):**

Can be found in the “Project Timeline.xlsx” file.