Front end design (Aditi)

# INDEX.HTML

## Main page (activity page)

## Menu for creating a habit

* Input text box for submitting a habit. Upon submission, this will be sent to the mongo DB.
* Select frequency for tracking habit. This could be a select input, with a dropdown showing 1, 2, 3 days.
* Set a requirement for completing the habit.
* Button for ‘creating’ the habit, which will store this information in the database using a post method. Then display this data in a container (like a post).

## Table displaying habits

* The habit ‘post’ will display information, such as: time left to complete requirements, habit streak (number of consecutive days the habit was completed), information on the progress (5000 steps completed out of 10000). There will be a button to mark the habit as completed. There can also be an input field where you can store the progress of the habit.

# Modal containers (on index.html, make some basic containers which will show the modal content)

## Login

* Username
* Password
* Login button

## Register

* Create username
* Create password
* Confirm password
* Email?
* Could place a captcha \*research this if there is time\*?
* Create account button

Functionality of the main page (Sean)

Timeline

Description automatically generated

## Login using login modal container as described on page 1. Register using another modal container.

Processes for logging in:

* If user exists and password matches the one in the database for that user, proceed to show main page. Create a function which sends the data to the back end, for it to be validated. If the user exists, proceed to main page.
* If user does not exist, or password is incorrect, stay on login box and throw an error ‘username is incorrect’ ‘password is incorrect, try again’

Processes for registering:

* If username and password suit the criteria, upon pressing ‘create account’ the data from this form will be sent and stored to the ‘users’ database. This can be done with a fetch post request. The back end will do the password hashing.
* If username already exists (it is already stored in the database) throw an error ‘Name already taken, choose another’.

## Menu displaying options for creating the habit.

* Add form event listener. Once all the fields in the menu are entered and the submit button is pressed, send the data to the database. Then, create a new container displaying this habit information.

## Habit Information

* Display the habit information which was taken from the create habit menu.
* Create a function which will take the current date, and find the different between this and the date set for the habit to be completed. Then display this difference on the container.
* Create a button which will mark the habit as complete.
* For every time the mark complete button is pressed, add to a ‘streak’ counter and display this counter on the container. Only have the button be pressable once, until the next day when the habit ‘resets’.
* For all of the data points, when they are updated in this container, they should be sent to the database to be stored.

File structure

Client

Static

* css
* style.css (for styling index.html)
* modal.css (for styling login + register boxes)
* js
* modal.js (for processing login+register data)
* requests.js (functions for fetching database data)
* layout.js(for general HTML manipulation on index.html)

index.html

tests

For testing the functions on the front end, and the actual front end page itself. Will decide how this will be done on Monday or Tuesday.