SENG2021 21T1

Final Report

Team Half Stack

Timothy Devine	z5316844
Ethan Dickson	z5309251
Lachlan Fraser	z5258840
Michael Agius	z5257081
Guy Chilcott	z5263967
Yueru Duan (Ellen)	z5210986

Table of Contents

Business Part

System Purpose Explanation and Problems Addressed

We originally created our system to be an all in one physical and mental health tracker with an emphasis on having a simple user interface. We explored ideas of drag path map APIs as well as fitness tracking APIs such as REST API and were excited at the possibilities these could open. Brainstorming from here we wanted to expand into speed, distance, elevation and calorie calculations for workouts and even dreamed of going as far as adding goals, badges and friends all into our system. While we did not end up adding everything we originally brainstormed, we improved upon many of these ideas and removed some others we no longer saw as applicable along the way.

We have finished with a system that is not restricted to any age group, it is built for the community and allows anyone, no matter their experience with health and exercise, to push themselves and receive workouts tailored for them. We also ensured we have a consistent and simple user interface so you do not have to be a tech professional just to use our system. We wanted to make a one stop spot for all health tracking needs, removing the issue of having a specific application for each individual need. Finally, the health calculations are all done for you by our system so you do not have to manually figure out those calorie and weight losses.

Our system improves upon its competitors because of its simplicity, versatility and dedication to you.

List of Features

Full Body Grin

- Workout Planning and Tracking
- Get Quality Workout Data, Calories and Pace
- Sleep Tracking, Quality & Time
- Personalised to your goals
    - Receive Workout Recommendations accordingly
- Personalised to your stats
    - Receive Calorie Data specific to you
- Provides users with a set number of free workouts that can be tracked or planned
- Premium Membership for dedicated users
    - Grants unlimited access to the app

Updated User Stories

Sign Up and Log In

Feature 1: Sign Up by Creating an Account

As a fitness enthusiast
I want to make an account on this service
So that I can use Full Body Grin to help me achieve the fitness level I want and am shown only data relevant to me, and not others unsolicited

(1) Scenario: A new user wants to sign up and fills in the registration form correctly
   (a) Given the user is on the landing page
   (b) When the user clicks the 'Sign Up' button
   (c) Then the user is directed to the Sign Up page
   (d) When the user enters all required fields in the correct format (username and email that have not been signed up before, valid date of birth, valid height, valid weight, valid password) and clicks the 'Submit' button
   (e) Then a modal window pops up informing the user they have signed up successfully, they are added to the database, and they are directed to the Log In page

(2) Scenario: A new user wants to sign up and fills in an invalid field
   (a) Given the user is on the Sign Up page
   (b) When the user enters either a username or email that has been signed up with before, an invalid date of birth, invalid height, invaid weight, invaid password or two different passwords when confirming the password and clicks the 'Submit' button
   (c) Then a modal window pops up informing the user that they have failed the registration due to an invalid field, they are not added to the database, and they stay on the Sign Up page

(3) Scenario: A new user changes their mind while filling out the registration form
   (a) Given the user is on the Sign up page
   (b) When the user clicks the back button in their browser
   (c) Then the user is not added to the database, and they are directed to the landing page

Feature 2: User Login with Email Address and Password

As a fitness enthusiast who has previously signed up
I want to log into the website
So that I can access all my information and activities

(1) Scenario: A registered user wants to log into their account and fills in the email address and password correctly
    (a) Given the registered user is on the landing page
    (b) When the user clicks the 'Log In' button
    (c) Then the user is directed to the Log In page
    (d) When the registered user enters a valid email address and a valid password and clicks the 'Log In' button
    (e) Then the registered user is signed into their account, and the user is directed to the Dashboard page
(2) Scenario: A registered user wants to log into their account but provides an invalid email address or password
    (a) Given the registered user is on the Log In page
    (b) When the registered user enters an invalid email address or an invalid password in and clicks the 'Log In' button
    (c) Then a modal window pops up informing the user that the email address/password entered is invalid

Planning Workouts

Feature 1: Plan A New Workout

As a fitness enthusiast
I want to plan new workouts
So that I can have motivation to improve my fitness level

(1) Scenario: A logged-in user wants to plan a new workout and fills in all fields correctly
    (a) Given the user is on the Dashboard page
    (b) When the user clicks 'Plan New Workout' button
    (c) Then they are directed to the Plan New Workout page on which they can fill in workout details
    (d) When the user selects a valid completion date, selects a workout type (either running or cycling) from a dropdown list and draws the path they plan to take on the map
    (e) Then path length, calories burnt and predicted time taken are all calculated automatically and corresponding fields are filled with these values
    (f) When the user clicks the 'Submit' button
    (g) Then a modal window pops up informing the user that the new workout has been created successfully

(2) Scenario: A logged-in user wants to plan a new workout but at least one of the required fields are filled in incorrectly
    (a) Given the user is on the Plan New Workout page
    (b) When the user provides either an empty or invalid completion date, empty workout type or doesn't draw a path on the map
    (c) Then path length, calories burnt and predicted time taken are not calculated and corresponding fields remain empty
    (d) When the user clicks the 'Submit' button
    (e) Then a modal window pops up informing the user that some inputs are invalid
(3) Scenario: A logged-in user changes their mind while planning new workouts and doesn't want to plan a new workout
    (a) Given the user is on the Plan New Workout page
    (b) When the user clicks the 'Back' button
    (c) Then they are directed back to the Dashboard page

Feature 2: View Upcoming Workouts

As a fitness enthusiast
I want to view all upcoming workouts
So I can be reminded and allocate time wisely

(1) Scenario: A logged-in user wants to view all upcoming workouts
    (a) Given the user is on the Dashboard page
    (b) When the user clicks the 'Show All' button in the 'Upcoming Workouts' section
    (c) Then they are directed to the Upcoming Workouts page and general summaries of upcoming planned workouts are listed in chronological order
(2) Scenario: A logged-in user wants to view details of a planned workout
    (a) Given the user is on the Upcoming Workouts page
    (b) When the user clicks on one of the specific planned workouts
    (c) Then they are directed to the workout data page on which they can view details of the selected planned workout

Feature 3: Edit planned workouts

As a fitness enthusiast
I want to edit a planned workout
So the workout becomes more achievable for me given my current fitness level and time availability

(1) Scenario: A logged-in user wants to edit a planned workout's details and all the required fields are valid after the edit is made
   (a) Given the user is on the workout data page of a specific planned workout and has clicked the 'Edit' button
   (b) When the completion date is valid, either running or cycling is selected from the dropdown list as the workout type, and a valid path is drawn on the map after edits are made
   (c) Then path length, calories burnt and predicted time taken are calculated again and corresponding fields are filled with the new values
   (d) When the user clicks the 'Save' button
   (e) Then a modal window pops up informing the user that the workout has been updated and the user stays on the workout data page with the updated workout details

(2) Scenario: A logged-in user wants to edit a planned workout's details but at least one of the required fields are invalid after the edit is made
   (a) Given the user on the workout data page of a specific planned workout and has clicked the 'Edit' button
   (b) When either the completion date is invaid, no workout type is selected from the dropdown list or no path is drawn on the map after edits are made and the user clicks the 'Save' button
   (c) Then a modal window pops up informing the user that some information is invalid and the user stays on the workout data page

(3) Scenario: A logged-in user wants to discard edits made to the workout
   (a) Given the user is on the workout data page of a specific planned workout and has clicked the 'Edit' button
   (b) When the user clicks the 'Cancel' button
   (c) Then a modal window pops up asking the user to confirm that they want to discard the edits made to the workout
   (d) When the user clicks the 'Discard Changes' button
   (e) Then the user stays on the workout data page and workout details remain unchanged

Feature 4: Delete Planned Workouts

As a fitness enthusiast
I want to delete an upcoming workout
So that I can focus on a different workout plan instead

(1) Scenario: A logged-in user wants to delete a workout
    (a) Given the user is on the workout data page of a specific planned workout
    (b) When the user clicks the 'Delete' button
    (c) Then a modal window pops up asking the user to confirm that they want to delete this workout
    (d) When the user clicks 'Yes, I want to delete this workout' button
    (e) Then a modal window pops up informing the user that the workout has been deleted, and the user is directed back to the Upcoming Workouts page
(2) Scenario: A logged-in user changes their mind when trying to delete a workout
    (a) Given the user is on the workout data page of a specific planned workout and has clicked the 'Delete' button
    (b) When the user clicks the 'No, I don't want to delete this workout' button in the modal window
    (c) Then the user stays on the workout data page of that workout

Feature 5: Filter Upcoming Workouts

As a fitness enthusiast
I want to filter upcoming workouts based a specific time or workout type
So only workouts I am interested in are shown to me

(1) Scenario: A logged-in user wants to view planned workouts in a specific time period and enters a valid start date and/or end date
    (a) Given the user is on the Upcoming Workouts page
    (b) When the user enters the start date and/or end date correctly and clicks the 'Search' button
    (c) Then only workouts with a completion date in the time range are displayed in chronological order
(2) Scenario: A logged-in user wants to view planned workouts in a specific time period but enters an invalid start/end date
    (a) Given the user is on the Upcoming Workouts page
    (b) When the user enters the start date or end date incorrectly and clicks the 'Search' button
    (c) Then a modal window pops up informing the user they have entered an invalid start/end date

(3) Scenario: A logged-in user wants to filter planned workouts by workout type
    (a) Given the user is on the Upcoming Workouts page
    (b) When the user selects one workout type from the dropdown list and clicks the 'Search' button
    (c) Then only workouts of the select workout type are displayed on the Upcoming Workouts page in chronological order
(4) Scenario: A logged-in user wants to clear any filters that have been applied
    (a) Given the user is on the Upcoming Workouts page
    (b) When the user clicks the 'Clear' button
    (c) Then all upcoming workouts are displayed in chronological order

Feature 6: Mark a Planned Workout as Completed

As a fitness enthusiast
I want to add a planned workout to the completed workout logs automatically
So I don't need to log it again manually

(1) Scenario: A logged-in user wants to log a planned workout automatically
    (a) Given the user is on the workout data page of a specific planned workout
    (b) When the user clicks the 'Complete' button
    (c) Then a modal window pops up informing the user that this workout has been added to the workout log, and the workout is removed from the list of upcoming workouts

Track Workouts

Feature 1: Log Previous Workouts

As a fitness enthusiast
I want to log my previous workouts
So I can have an idea of what I have completed and add new data for the application to analyse

(1) Scenario: A logged-in user wants to log a previous workout and enter all required fields correctly
    (a) Given the user is on the Dashboard page
    (b) When the user clicks the 'Log Workout' button
    (c) Then the user is directed to the Log Workout page where they can enter workout details
    (d) When the user selects a valid date, enters a positive integer for the time spent field, selects a workout type from the dropdown list and draws the path they travelled on the map
    (e) Then path length, calories burnt and pace are calculated and corresponding fields are filled with the new values.
    (f) When the user clicks the 'Submit' button
    (g) Then a modal window pops up informing the user that the workout has been logged successfully

(2) Scenario: A logged-in user wants to log a previous workout and enters at least one required field incorrectly
    (a) Given the user is on the Log Workout page
    (b) When the user either selects an invalid or empty date, enters an invalid input for the time spent field, doesn't select a workout type from the dropdown list or doesn't draw a path on the map
    (c) Then path length, calories burnt and pace are not calculated and corresponding fields remain empty
    (d) When the user clicks the 'Submit' button
    (e) Then a modal window pops up informing the user that some inputs are invalid
(3) Scenario: A logged-in user changes their mind while logging a previous workout
    (a) Given the user is on the Log Workout page
    (b) When the user clicks the 'Back' button
    (c) Then the user is directed back to the Dashboard page

Feature 2: View Recently Logged Workouts

As a fitness enthusiast
I want to view my logged workouts
So I can know if I have improved

(1) Scenario: A logged-in user wants to view all previous logged workouts
    (a) Given the user is on the Dashboard page
    (b) When the user clicks on the 'Show All' button in the 'Previous Workouts' section
    (c) Then they are directed to the Old Workouts page and general summaries of their recently logged workouts are listed in chronological order
(2) Scenario: A logged-in user wants to view details of a previous logged workouts
    (a) Given the user is on the Old Workouts page
    (b) When the user clicks on one of the specific workouts
    (c) Then they are directed to the workout data page on which they can view details of the selected previous workout

Feature 3: Filter Previous Workouts

As a fitness enthusiast
I want to filter logged workouts based on completion date or workout type
So only workouts I am interested in are shown to me

(1) Scenario: A logged-in user wants to sort previous workouts by calories burnt
    (a) Given the user is on the Old Workouts page
    (b) When the user selects 'Low to High'/'High to Low' from the 'Sort by Calories burnt' dropdown list
    (c) Then the Old Workouts page is refreshed and previously logged workouts are shown in the specified order

(2) Scenario: A logged-in user wants to view previous workouts in the specific time range and enters a start date and/or end date correctly
    (a) Given the user is on the Old Workouts page
    (b) When the user enters the start date and/or end date correctly and clicks the 'Search' button
    (c) Then the Old Workouts page refreshes and only workouts that were logged in the time range are displayed in chronological order
(3) Scenario: A logged-in user wants to view workouts that were logged in the specific time range but enters start date or end date incorrectly
    (a) Given the user is on the Old Workouts page
    (b) When the user enters start date or end date incorrectly and clicks the 'Search' button
    (c) Then a modal window pops up informing the user they have entered an invalid start/end date
(4) Scenario: A logged-in user wants to filter logged workouts by workout type
    (a) Given the user is on the Old Workouts page
    (b) When the user selects one workout type from the dropdown list and clicks the 'Search' button
    (c) Then only workouts of the select workout type are displayed on the Workout log page in chronological order
(5) Scenario: A logged-in user wants to clear any filters currently applied
    (a) Given the user is on the Old Workouts page
    (b) When the user clicks the 'Clear' button
    (c) Then all previous workouts are displayed in chronological order

Feature 4: Reuse A Previous Workout

As a fitness enthusiast
I want to reuse a previous workout as one of my upcoming workouts automatically
So that I don't need to enter the same details again while planning a new workout

(1) Scenario: A logged-in user wants to reuse a previous workout.
    (a) Given the user is on the workout data page for a specific previous workout
    (b) When the user clicks the 'Reuse' button
    (c) Then a modal window pops up informing the user that this workout has been added to the upcoming workouts

Goal Setting

Feature 1: Add New Goals

As a fitness enthusiast
I want to create new goals
So that I can gain motivation and achieve the fitness level I desire

(1) Scenario: A logged-in user wants to add a new goal and all fields are entered correctly
   (a) Given that the user is on the Old Workouts page
   (b) When the user clicks on the 'Create Goal' button
   (c) Then the user is directed to the Create New Goal page on which they can enter details of a new goal
   (d) When the user selects a valid completion date, enters a positive integer for at least one of these fields: total calories burnt, total time spent or total distance travelled, optionally selects a workout type from the dropdown list and clicks the 'Submit' button
   (e) Then a modal page pops up informing the user that the goal has been created successfully, and the user is directed to the Previous Workouts page

(2) Scenario: A logged-in user wants to add a new goal and some fields in the Goal form are entered incorrectly
   (a) Given the user is on the Create New Goal page
   (b) When the user either selects an invalid completion date, enters an invaid value for at least one of these fields: total calories burnt, total time spent and total distance travelled and clicks the 'Submit' button
   (c) Then a modal window pops up informing the user that they have entered invalid input

(3) Scenario: A logged-in user changes their mind while creating a new goal
   (a) Given the user is on the Create New Goal page
   (b) When the user clicks the 'Back' button
   (c) Then the user is directed to the Previous Workouts page

Feature 2: View Goals and Workout Recommendations

As a fitness enthusiast
I want to view my goals and workout recommendations
So I can see my goal progression and find out how best to achieve them

(1) Scenario: A logged-in user wishes to view their goals and workout recommendations
   (a) Given that the user is on the Old Workouts page
   (b) When the user clicks the 'View Recommendations' button
   (c) Then the user is directed to the Goals and Recommendations page on which goals with different status (completed, in progress, uncompleted) are shown in different colors and workout recommendations are displayed

Feature 3: Delete Existing Goals

As a fitness enthusiast
I want to delete an existing goal
So that I can focus on an easier goal for now

(1) Scenario:
   (a) Given that I am on the Goals and Recommendations page
   (b) When I click the 'Delete' button for a specific goal
   (c) Then that goal is removed from the Goals and Recommendations page

Feature 4: Use A Recommendation

As a fitness enthusiast
I want to use a workout recommendation and add it to my upcoming workouts automatically
So that I don't need to enter the same details again while planning a workout

(1) Scenario: A logged-in user wants to add a recommendation to the upcoming workouts
   (a) Given the user is on the Goals and Recommendations page
   (b) When the user clicks the 'Use' button of a specific recommendation
   (c) Then a modal window pops up informing the user that the recommendation has been added to upcoming workouts successfully, and this recommendation is removed from the Goals and Recommendation page

Sleep and Mental Health

Feature 1: Log A New Sleep and Mental Health Entry

As a mindful user
I want to log my sleep and mental health
So that I can understand how my sleep and mood affect my exercise

(1) Scenario: A logged-in user wants to record new sleep and mental health data and fills in all fields correctly
   (a) Given the user is on the Dashboard page
   (b) When the user clicks the 'Log Sleep and Mental Health' button
   (c) Then they are directed to the Log Sleep page on which they can fill in sleep details
   (d) When the user selects a valid date, enters a positive number for the time asleep and selects exactly one checkbox for their mental status and clicks the 'Submit' button
   (e) Then a modal window pops up informing the user that the new sleep has been logged successfully

(2) Scenario: A logged-in user wants to record new sleep and mental health data but at least one of the required fields are filled in incorrectly
    (a) Given the user is on the Log Sleep page
    (b) When the user provides either an empty or invalid date, empty or invalid time or doesn't select exactly one checkbox and clicks the 'Submit' button
    (c) Then a modal window pops up informing the user that some inputs are invalid
(3) Scenario: A logged-in user changes their mind while logging new sleep and mental health data
    (a) Given the user is on the Log Sleep page
    (b) When the user clicks the 'Back' button
    (c) Then they are directed back to the Dashboard page

Feature 2: View Previous Sleep and Mental Health Data

As a mindful user
I want to view previously logged sleep and mental health data
So I can understand my long term sleep and mood patterns

(1) Scenario: A logged-in user wants to view all previous sleep and mental health data
    (a) Given the user is on the Dashboard page
    (b) When the user clicks the 'Show All' button in the 'Sleep and Mental Health' section
    (c) Then they are directed to the Sleep Logs page and general summaries of previous sleeps are listed in chronological order along with an appropriate bar chart showing sleep data from the past five days
(2) Scenario: A logged-in user wants to view details of a previous sleep
    (a) Given the user is on the Sleep Logs page
    (b) When the user clicks on one of the specific logged sleeps
    (c) Then they are directed to the sleep details page on which they can view details of the selected sleep

Update User Information

Feature 1: Change Personal Statistics

As a fitness enthusiast
I want to change my personal statistics such as weight and height
So that calories burnt can be calculated correctly

(1) Scenario: A logged-in user wants to update personal statistics and fills in all required fields correctly
   (a) Given the user is on the Dashboard page
   (b) When the user clicks the 'Settings' button
   (c) Then the user is directed to the Update User Info page
   (d) When the user enters valid positive numbers for height and weight and clicks the 'Change Height' or 'Change Weight' button
   (e) Then a modal window pops up informing the user that the personal statistics have been updated successfully

(2) Scenario: A logged-in user wants to update personal statistics but fills in at least one required field incorrectly
   (a) Given the user is on the Update User Info page
   (b) When the user enters invalid input for height or weight, and clicks the 'Change Height' or 'Change Weight' button
   (c) Then a modal window pops up informing the user that at least one input is invalid

Feature 2: Update Email and Password

As a fitness enthusiast
I want to change my email and password
So that my account can be more secure and up to date

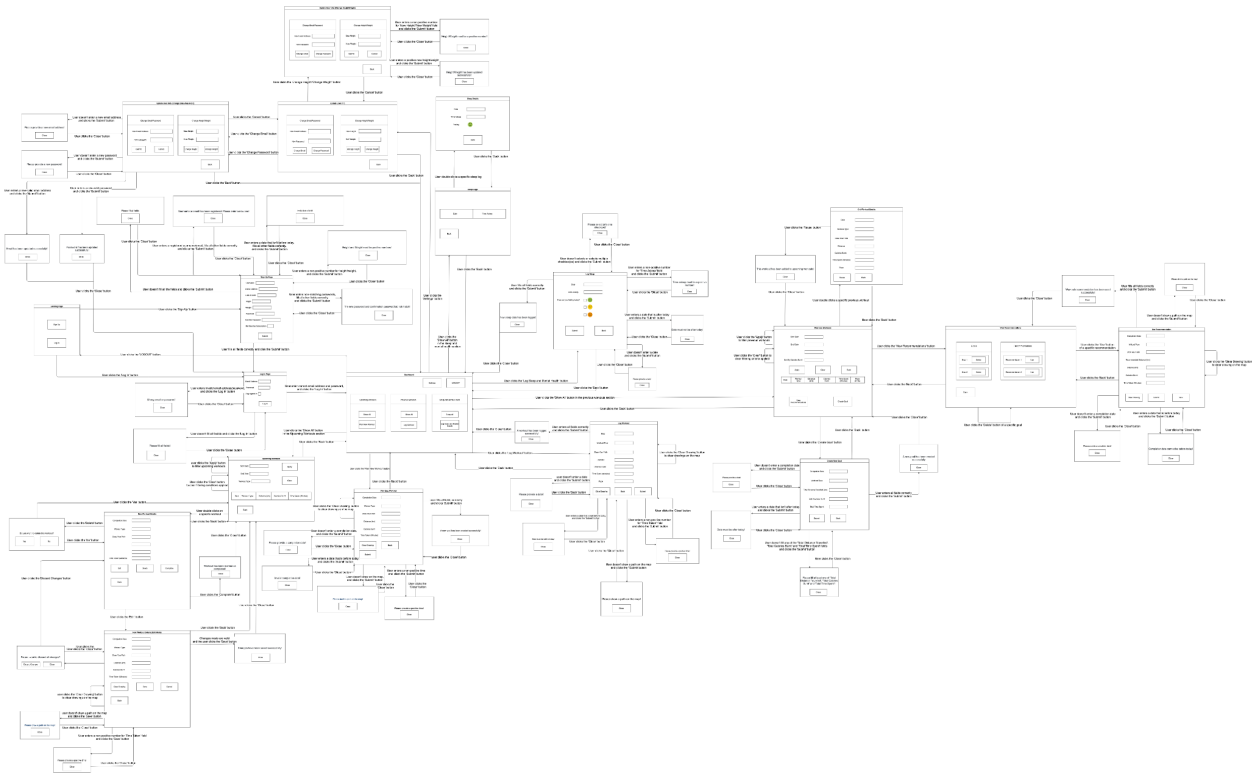
(1) Scenario: A logged-in user wants to update their email/password and fills in all required fields correctly
   (a) Given the user is on the Dashboard
   (b) When the user clicks the 'Settings' button
   (c) Then the user is directed to the Update User Info page
   (d) When the user enters a valid new email and/or a valid new password and clicks the 'Change Email' or 'Change Password' button
   (e) Then a modal window pops up informing the user that their email and password have been updated successfully

(2) Scenario: A logged-in user wants to update their email/password and fills in at least one of the required fields incorrectly
   (a) Given the user is on the Update User Info page
   (b) When the user either enters an invalid email or password and clicks the 'Change Email' or 'Change Password' button
   (c) Then a modal window pops up informing the user that at least one input is invalid

Design Part

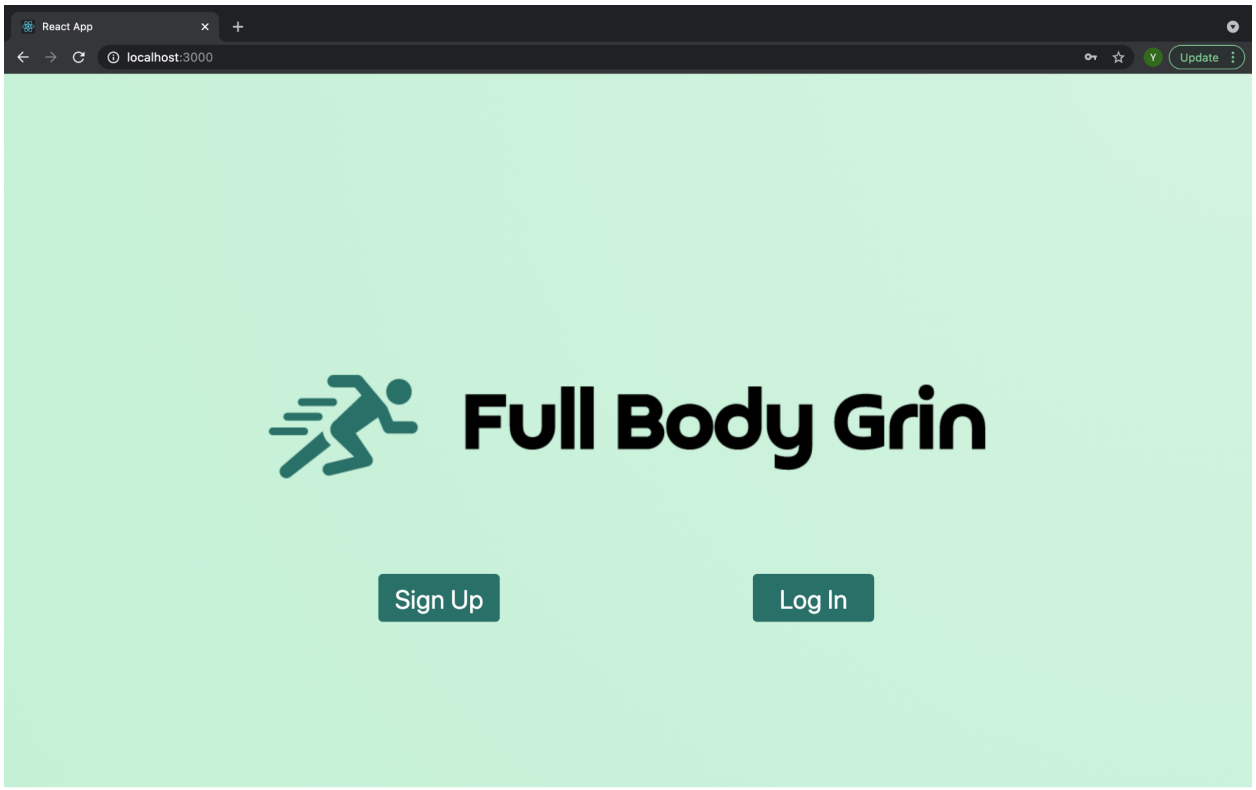
Updated Design and Information on Prototype Implementation

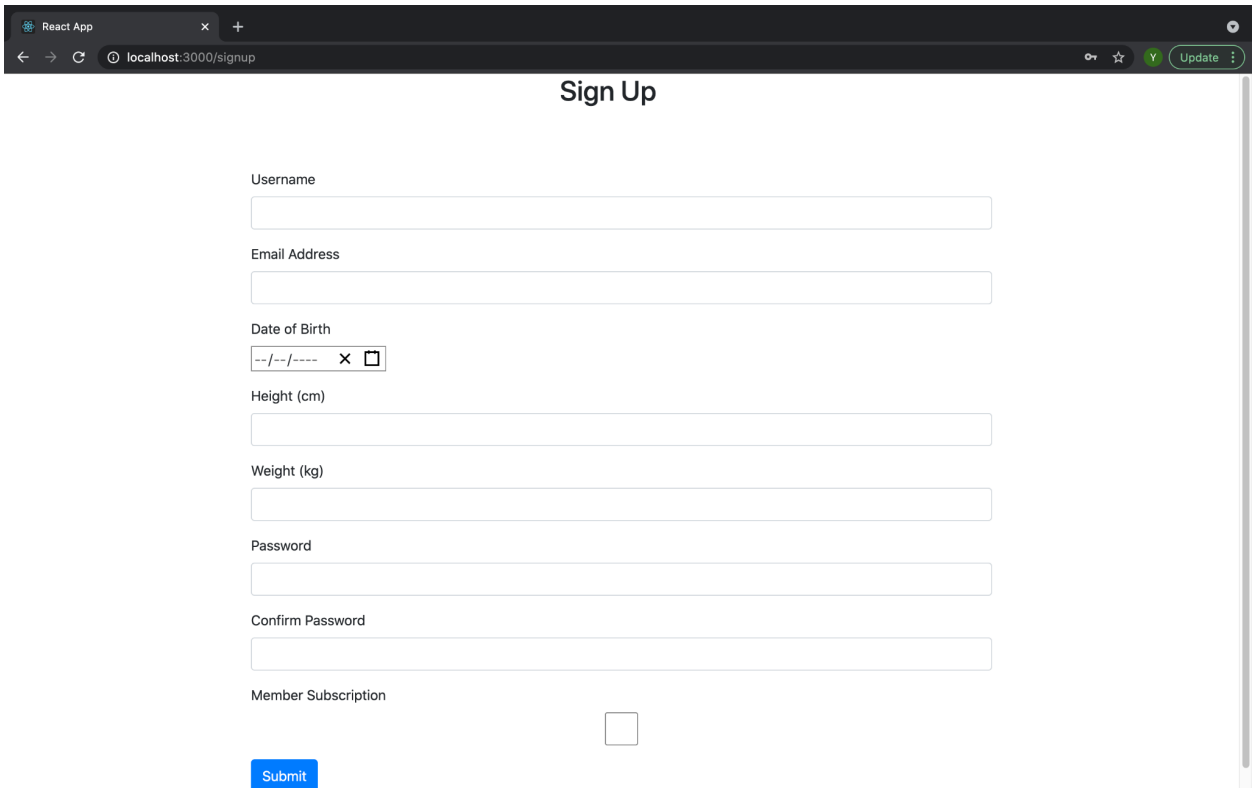
Low-Fidelity Prototype

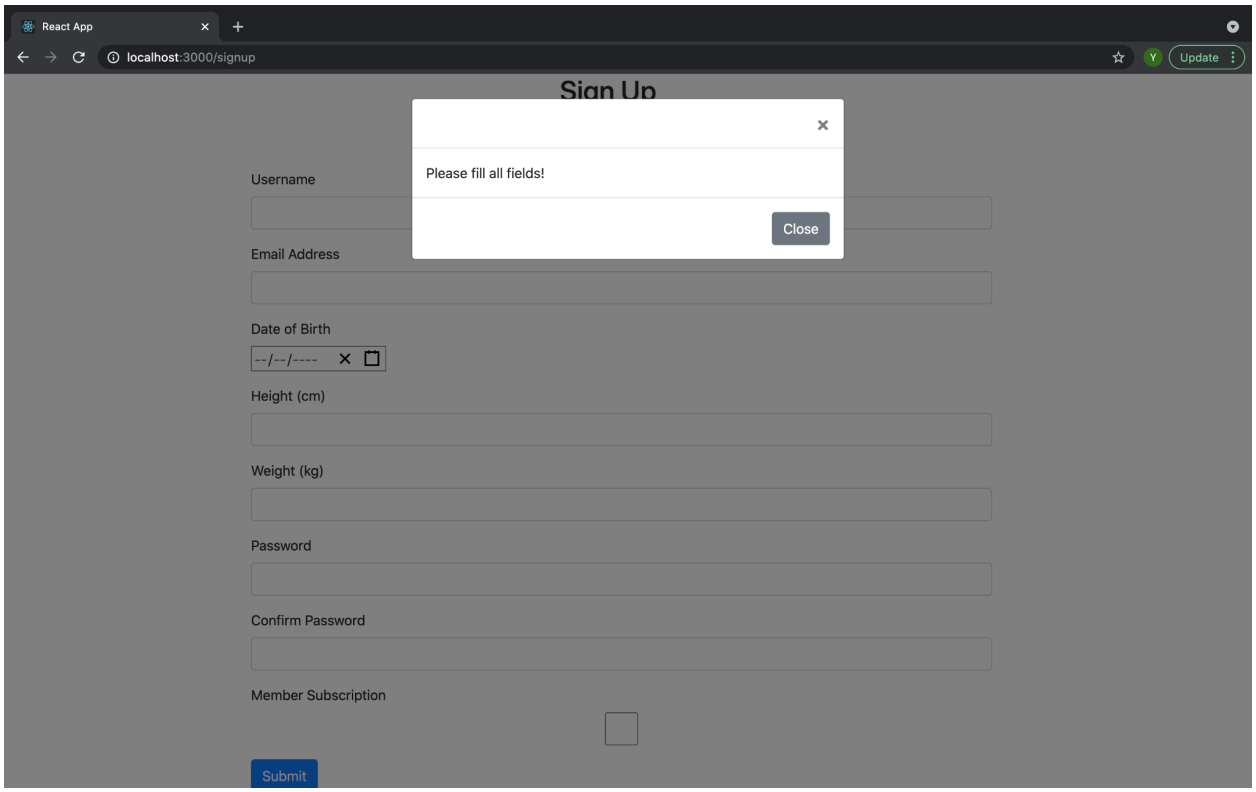


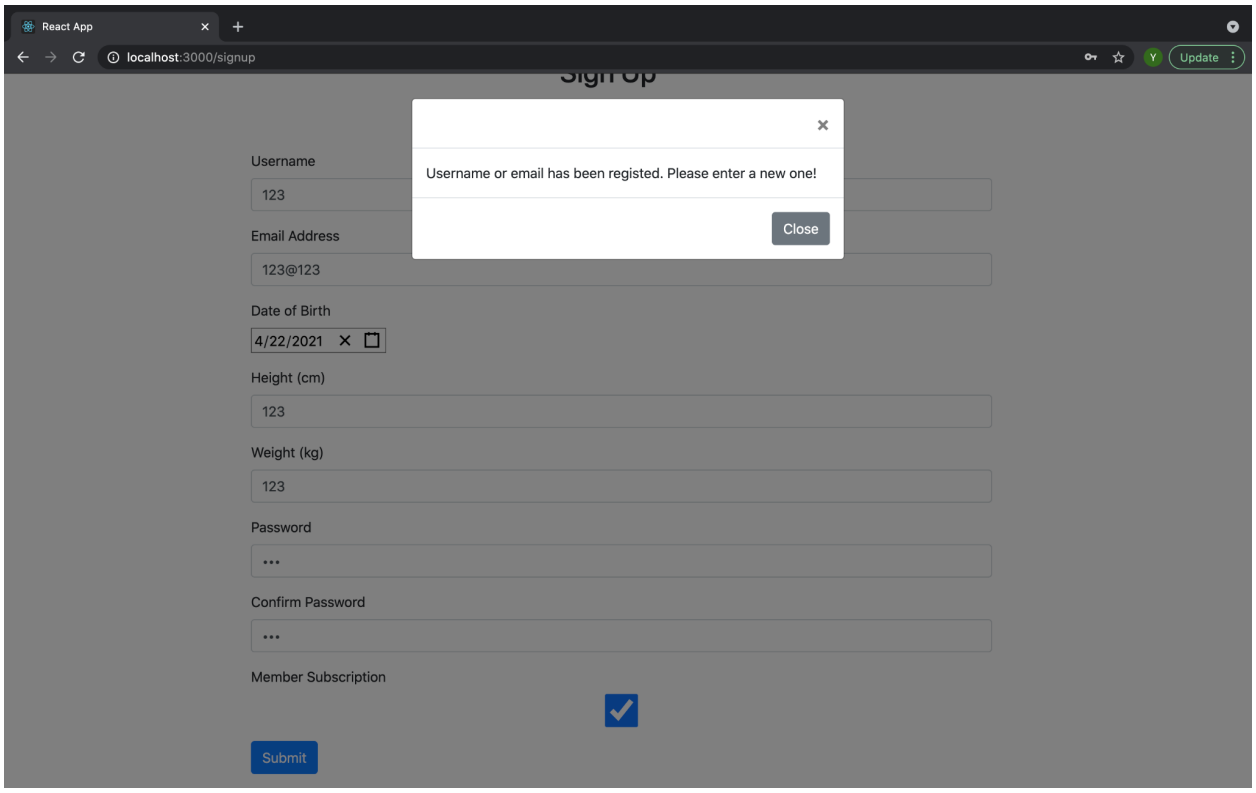
See https://drive.google.com/file/d/1ffxiQgpP4BOwdWfLv-iNTXg-3_J_Oyyj/view?usp=sharing for higher resolution.

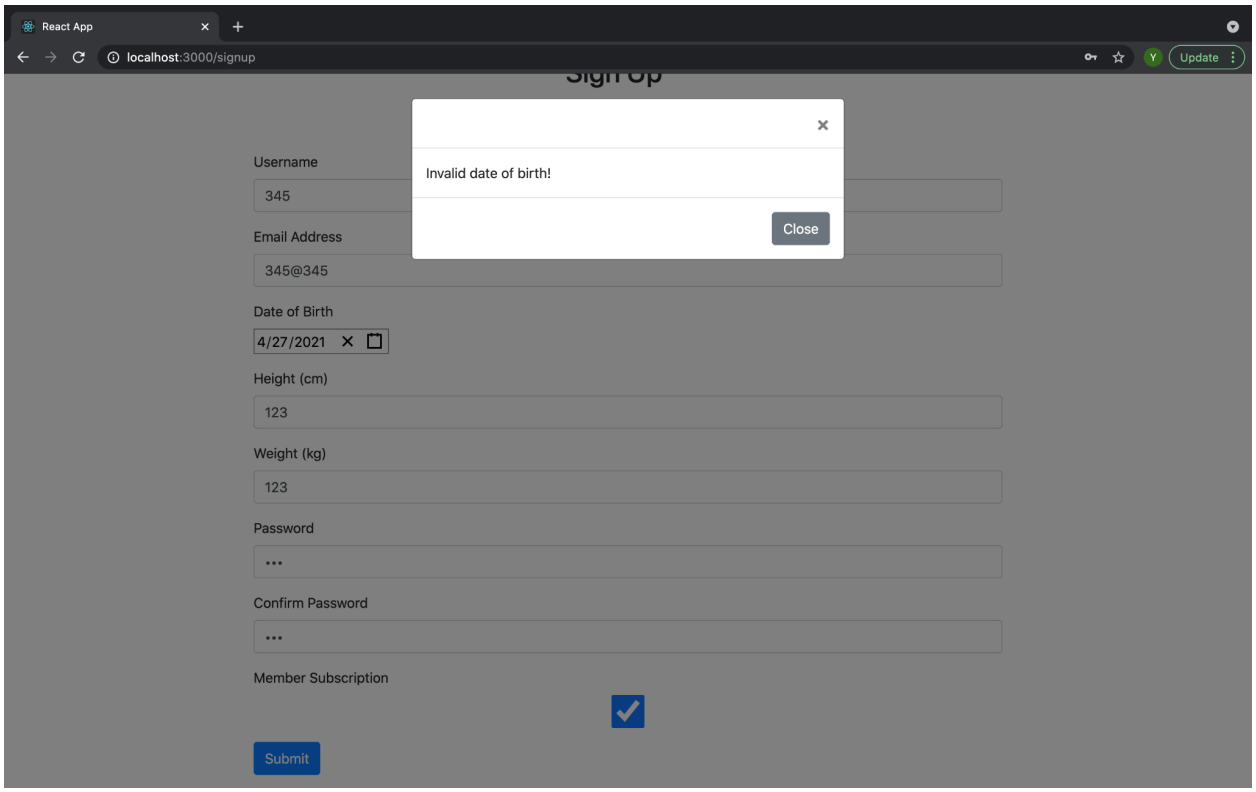
Final Prototype

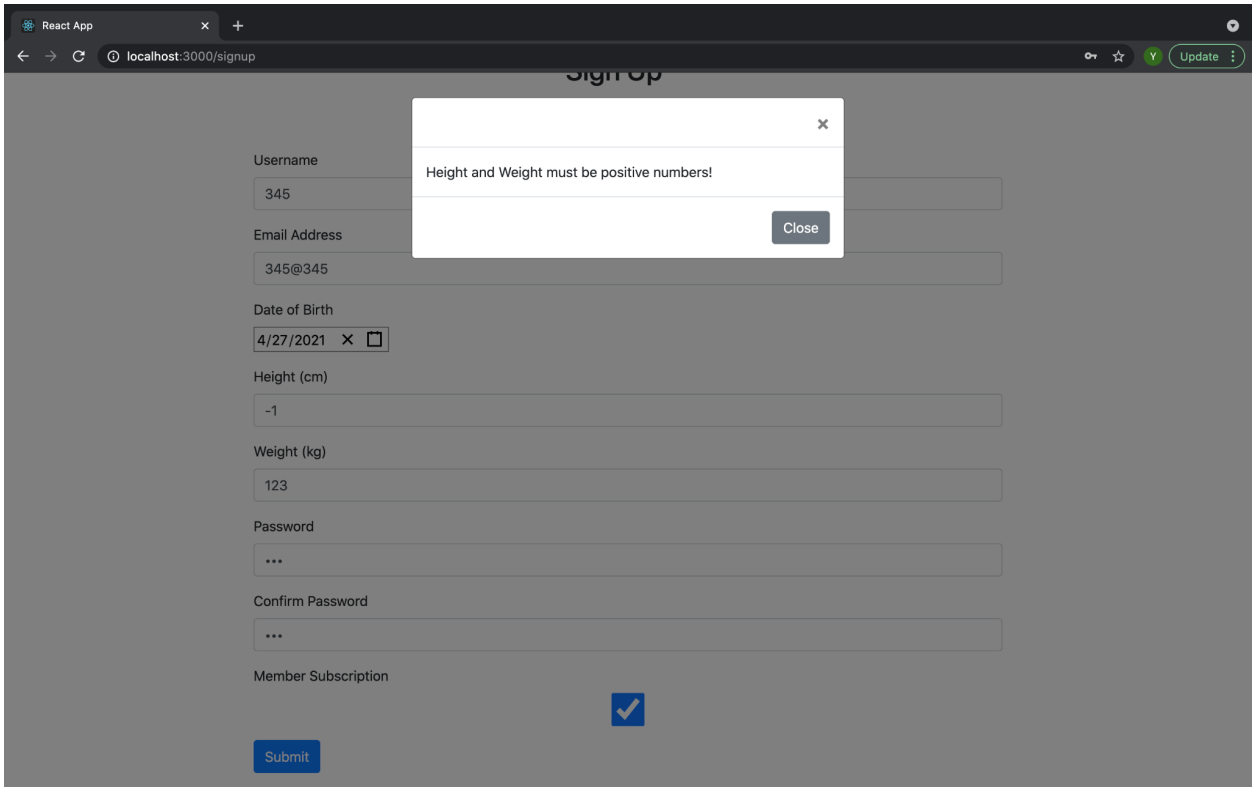


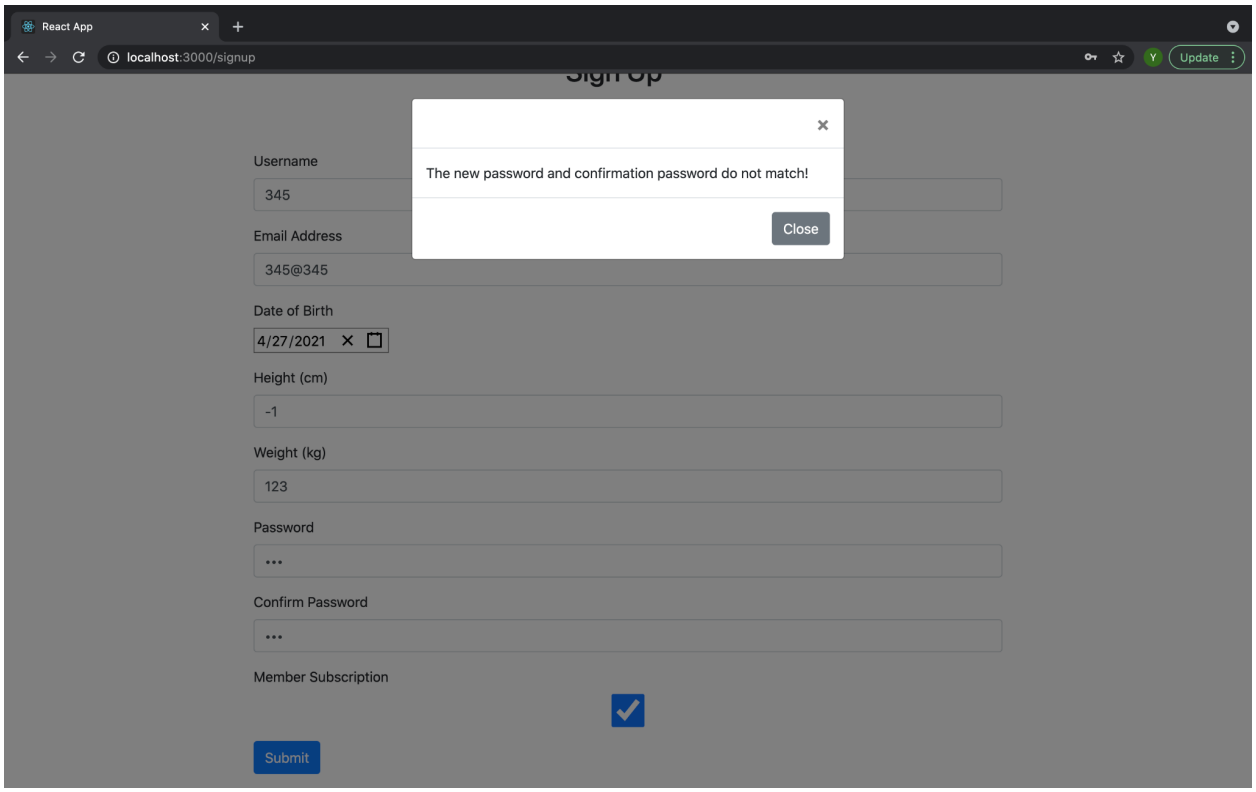


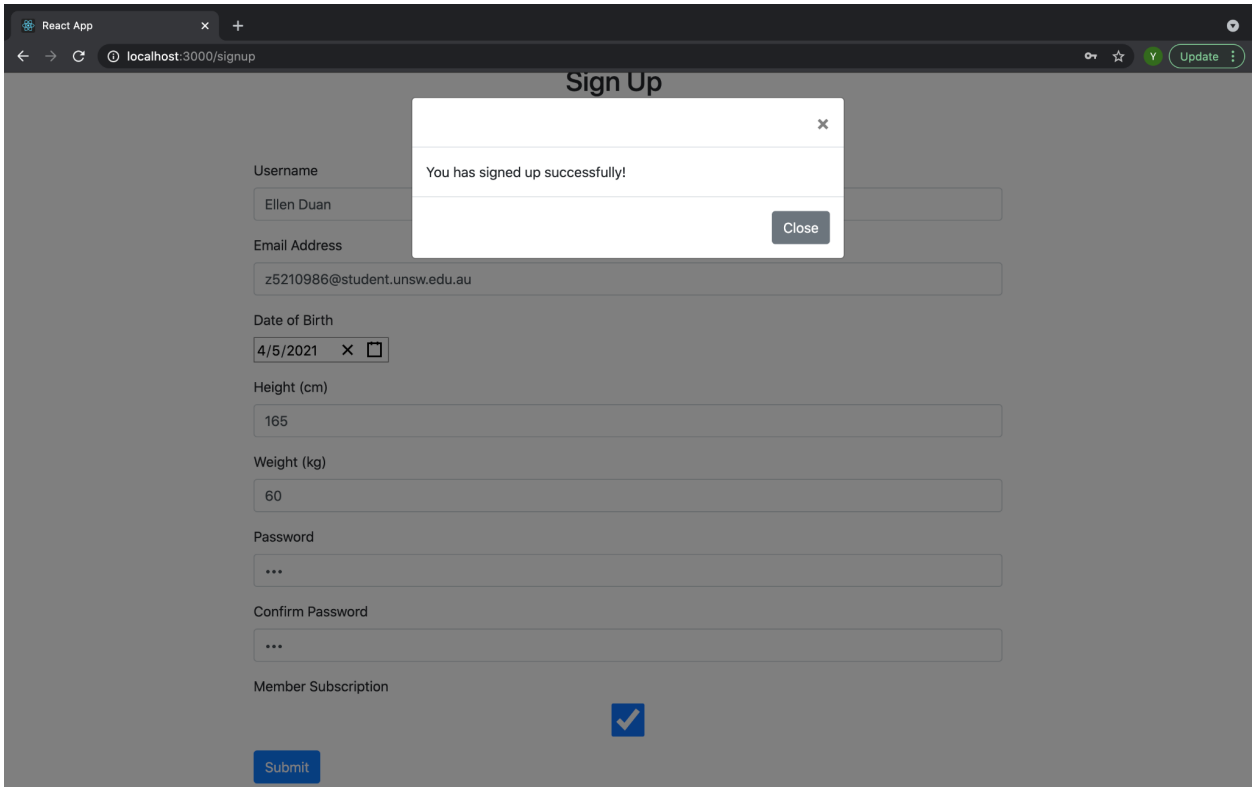

React App
localhost:3000/signup
Update
Sign Up
Please fill all fields!
Close
Username
Email Address
Date of Birth
--/--/----
Height (cm)
Weight (kg)
Password
Confirm Password
Member Subscription
Submit

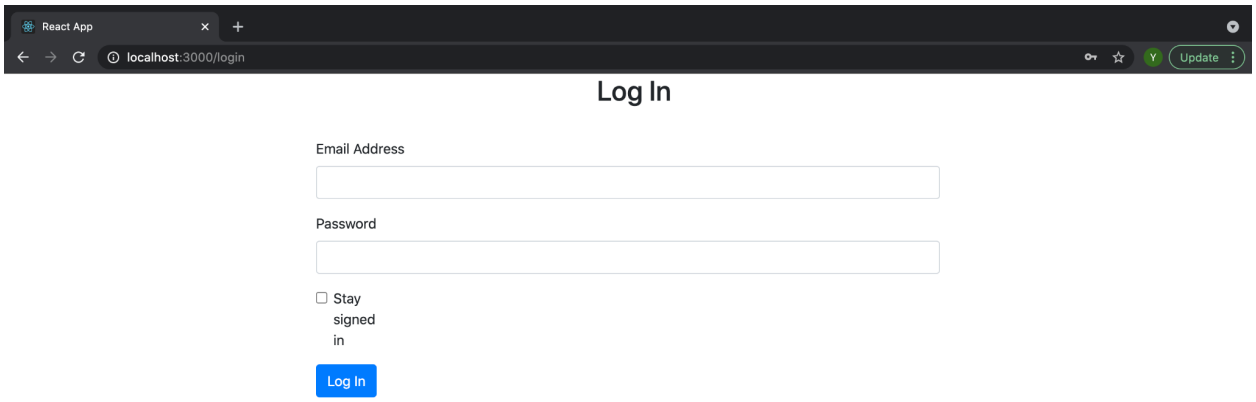

React App
localhost:3000/signup
Update
Username or email has been registed. Please enter a new one!
Close
Username
123
Email Address
123@123
Date of Birth
4/22/2021
Height (cm)
123
Weight (kg)
123
Password
Confirm Password
Member Subscription
Submit

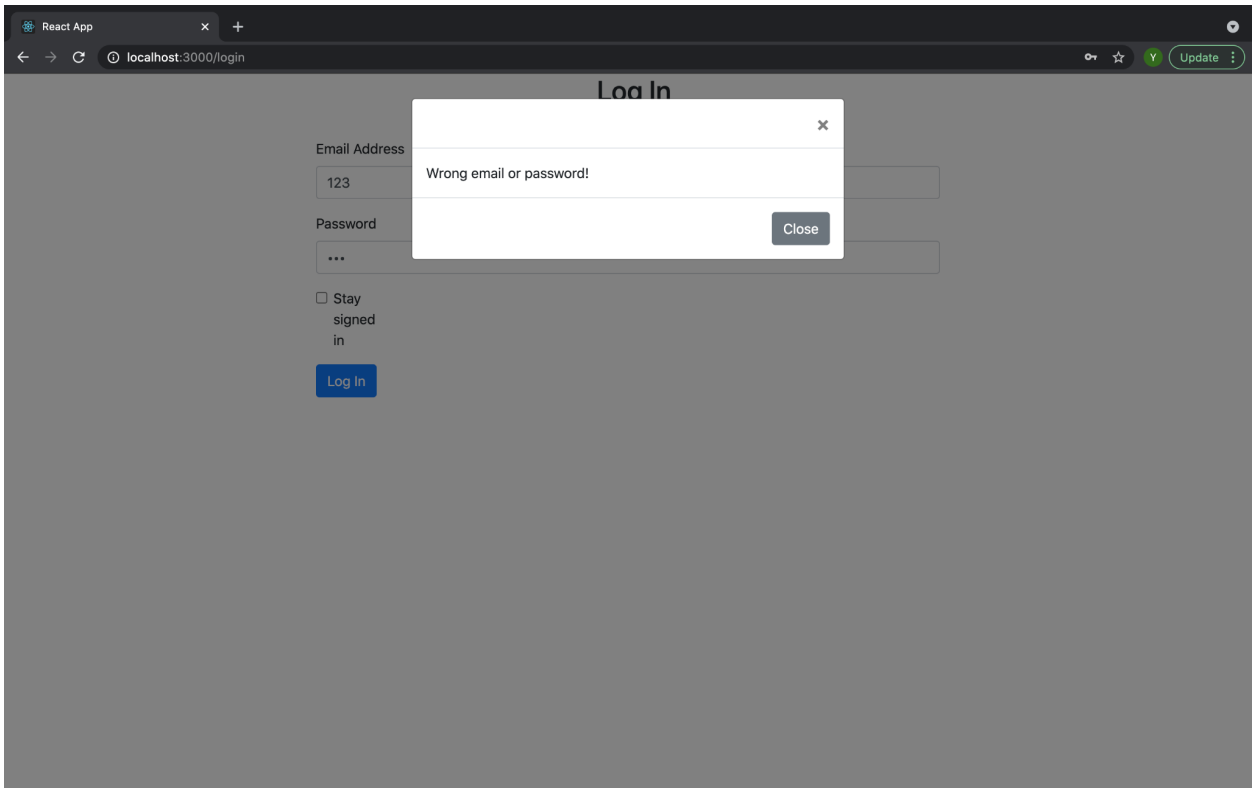

React App
localhost:3000/signup
Update
Invalid date of birth!
Close
Username
345
Email Address
345@345
Date of Birth
4/27/2021
Height (cm)
123
Weight (kg)
123
Password
Confirm Password
Member Subscription
Submit

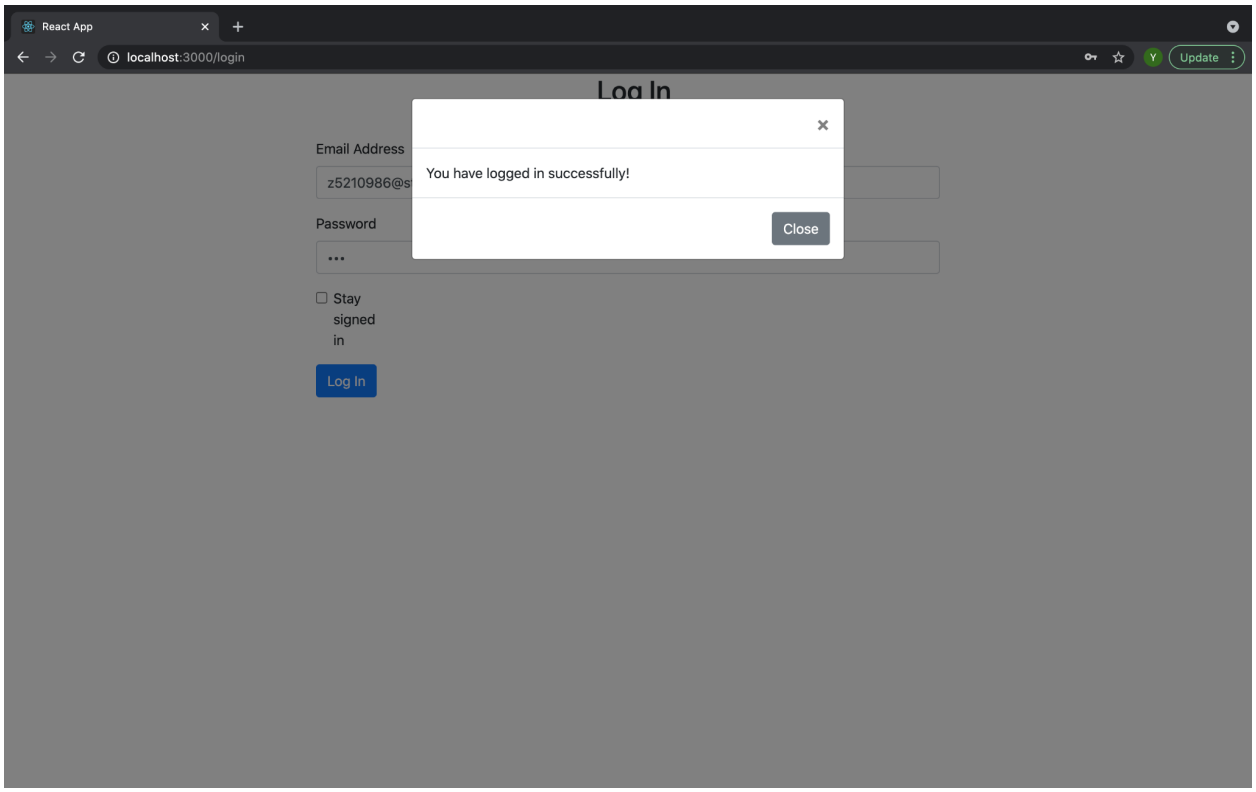

React App
localhost:3000/signup
Update
Height and Weight must be positive numbers!
Close
Username
345
Email Address
345@345
Date of Birth
4/27/2021
Height (cm)
-1
Weight (kg)
123
Password
Confirm Password
Member Subscription
Submit

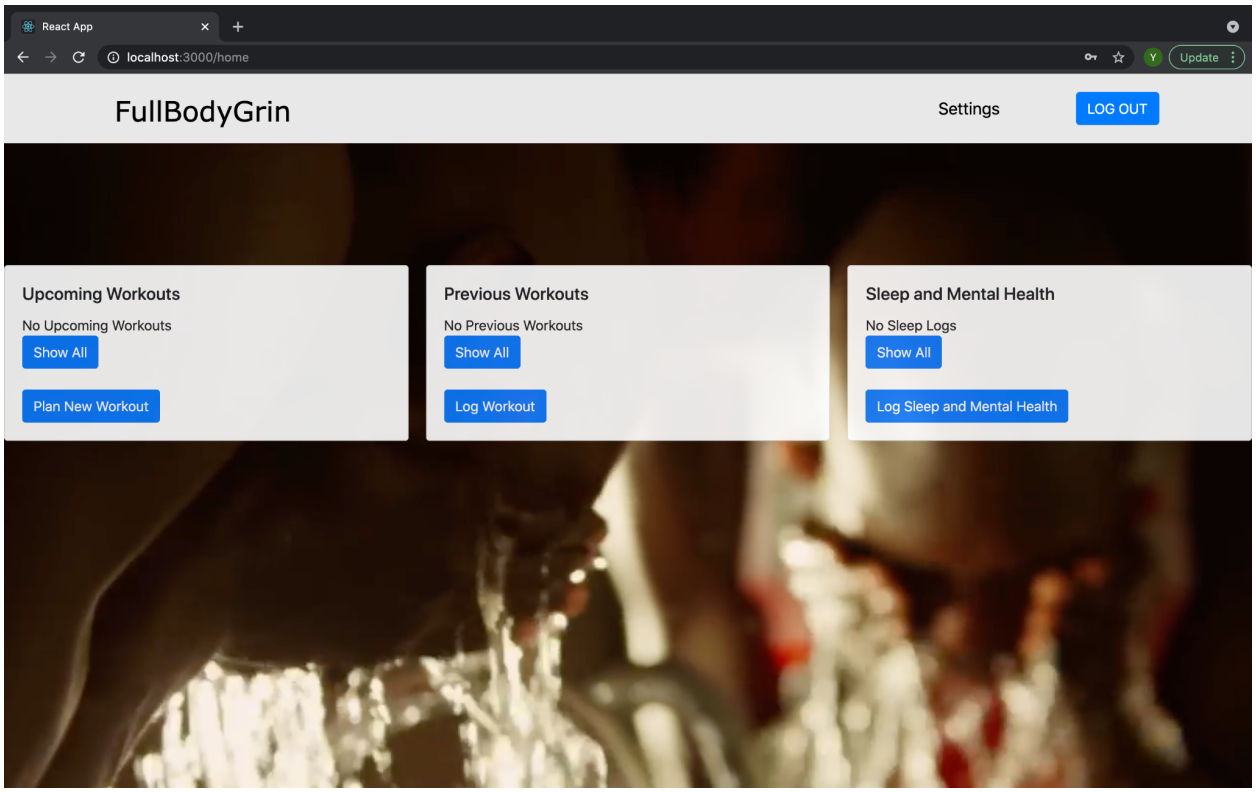

React App
localhost:3000/signup
Update
Sign Up
The new password and confirmation password do not match!
Close
Username
345
Email Address
345@345
Date of Birth
4/27/2021
Height (cm)
-1
Weight (kg)
123
Password
Confirm Password
Member Subscription
Submit

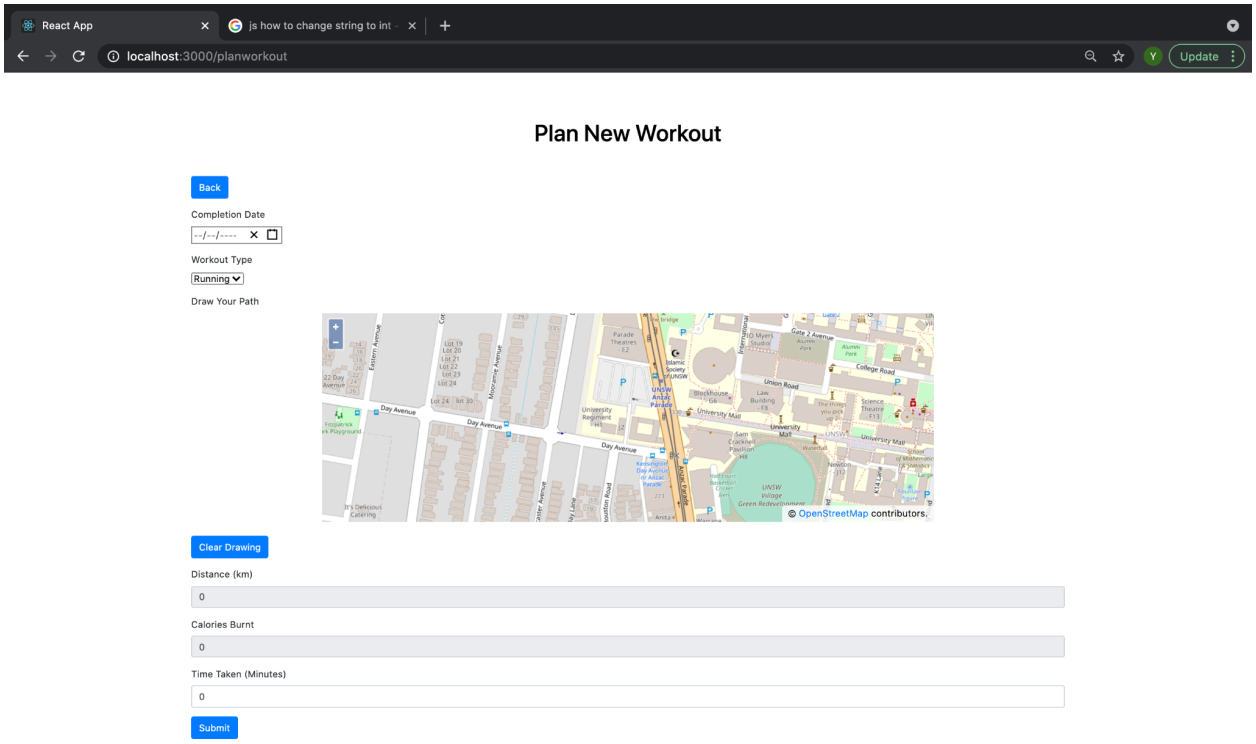

React App
localhost:3000/signup
Update
Sign Up
You has signed up successfully!
Close
Username
Ellen Duan
Email Address
z5210986@student.unsw.edu.au
Date of Birth
4/5/2021
Height (cm)
165
Weight (kg)
60
Password
Confirm Password
Member Subscription
Submit

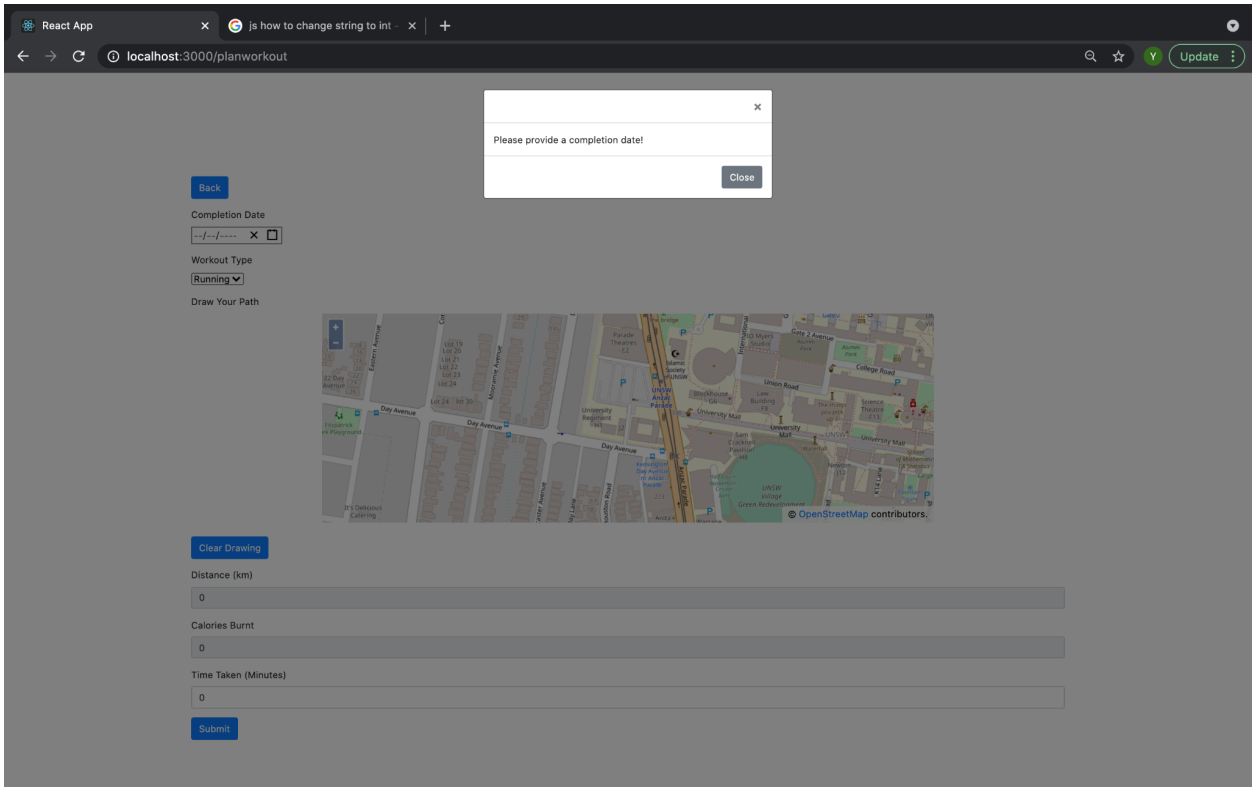

React App
localhost:3000/login
Update
Log In
Email Address
Password
Stay signed in
Log In

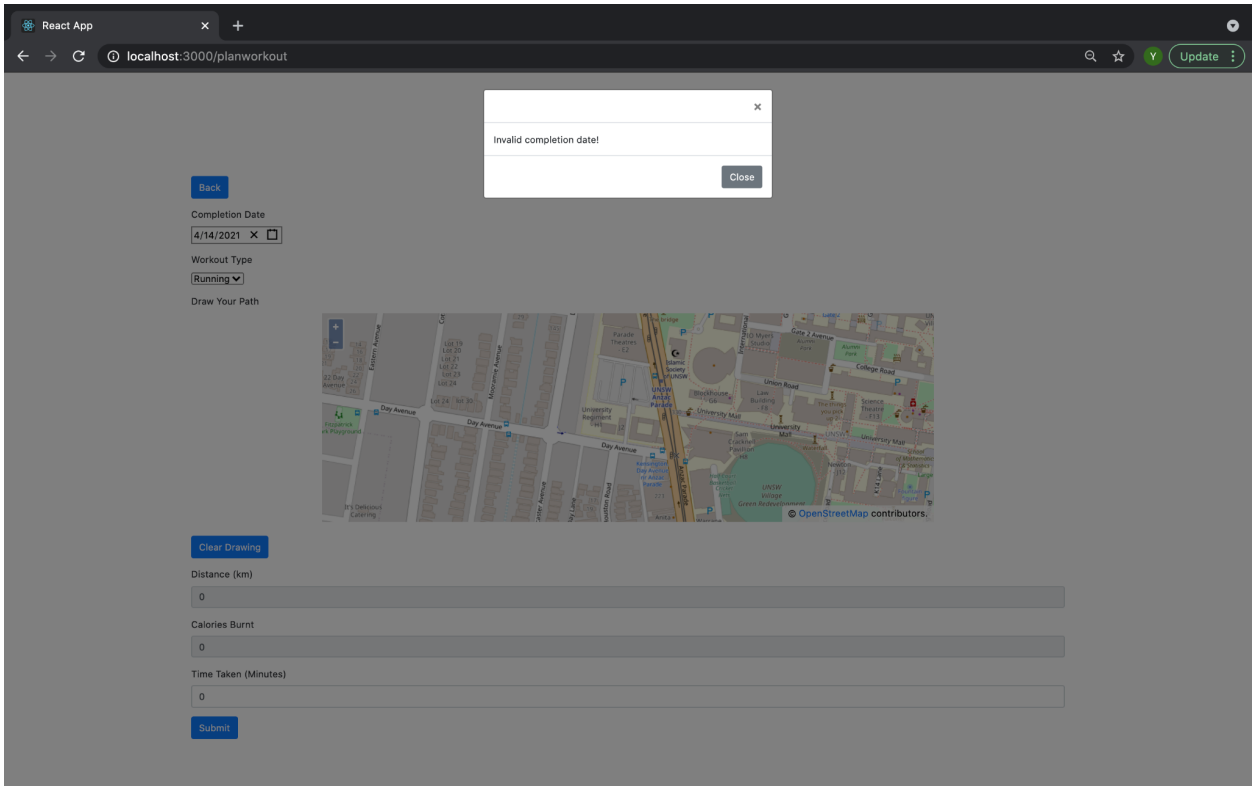

React App
localhost:3000/login
Update
Log In
Email Address
123
Password
Stay signed in
Log In
Wrong email or password!
Close

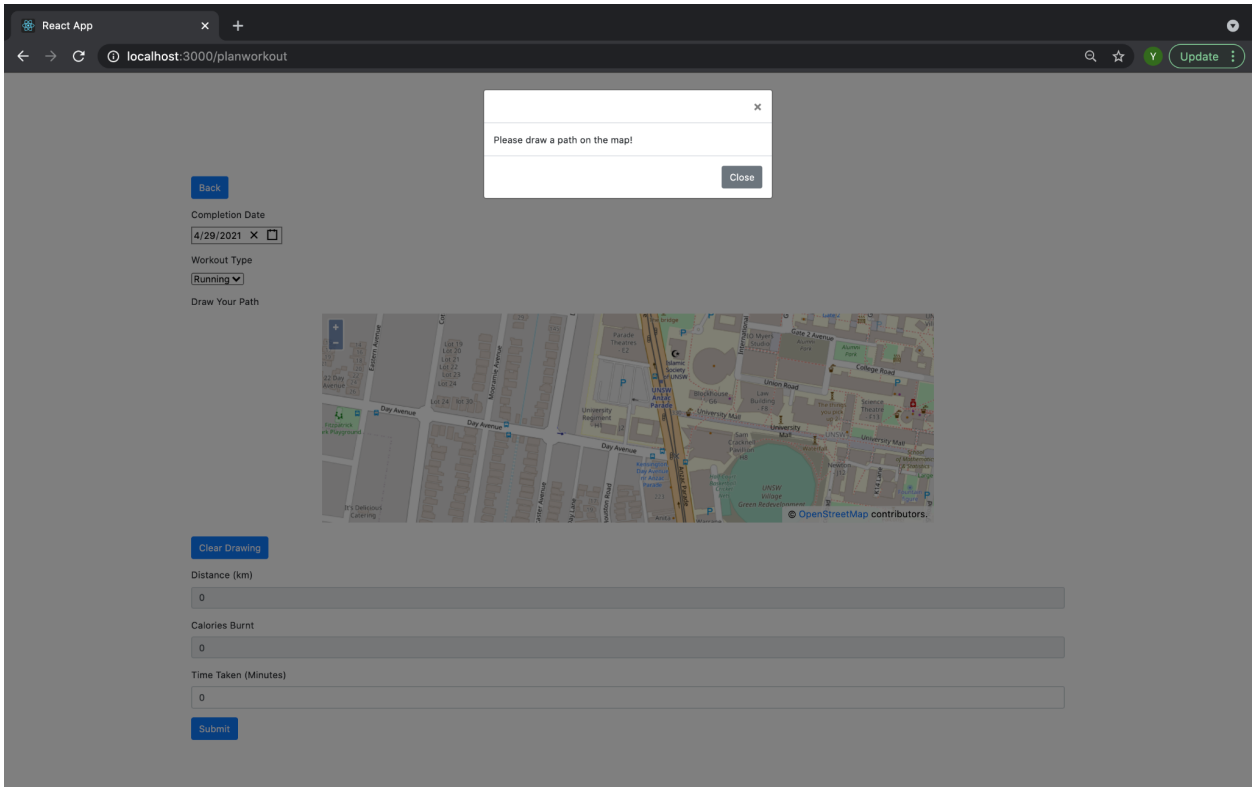

React App
localhost:3000/login
Update
Log In
Email Address
z5210986@s
Password
Stay signed in
Log In
You have logged in successfully!
Close

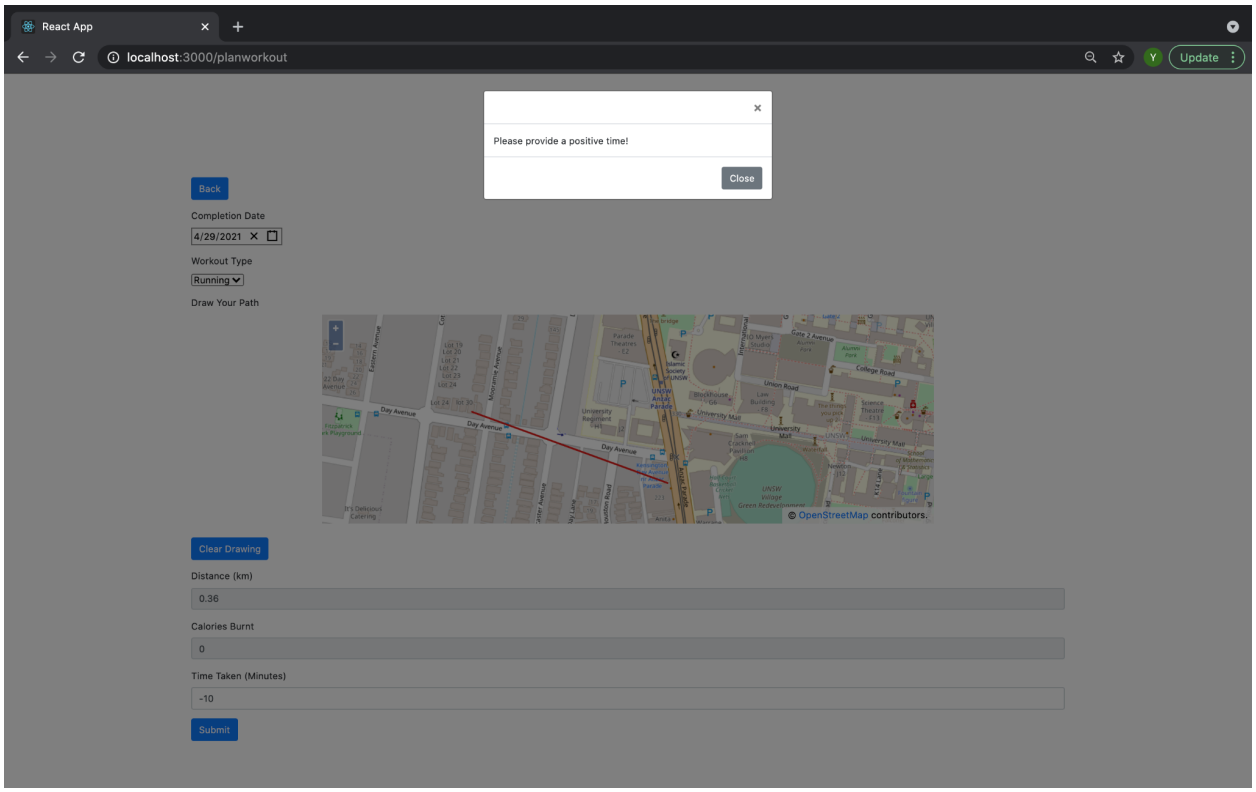

React App
localhost:3000/home
Update
FullBodyGrin
Settings
LOG OUT
Upcoming Workouts
No Upcoming Workouts
Show All
Plan New Workout
Previous Workouts
No Previous Workouts
Show All
Log Workout
Sleep and Mental Health
No Sleep Logs
Show All
Log Sleep and Mental Health

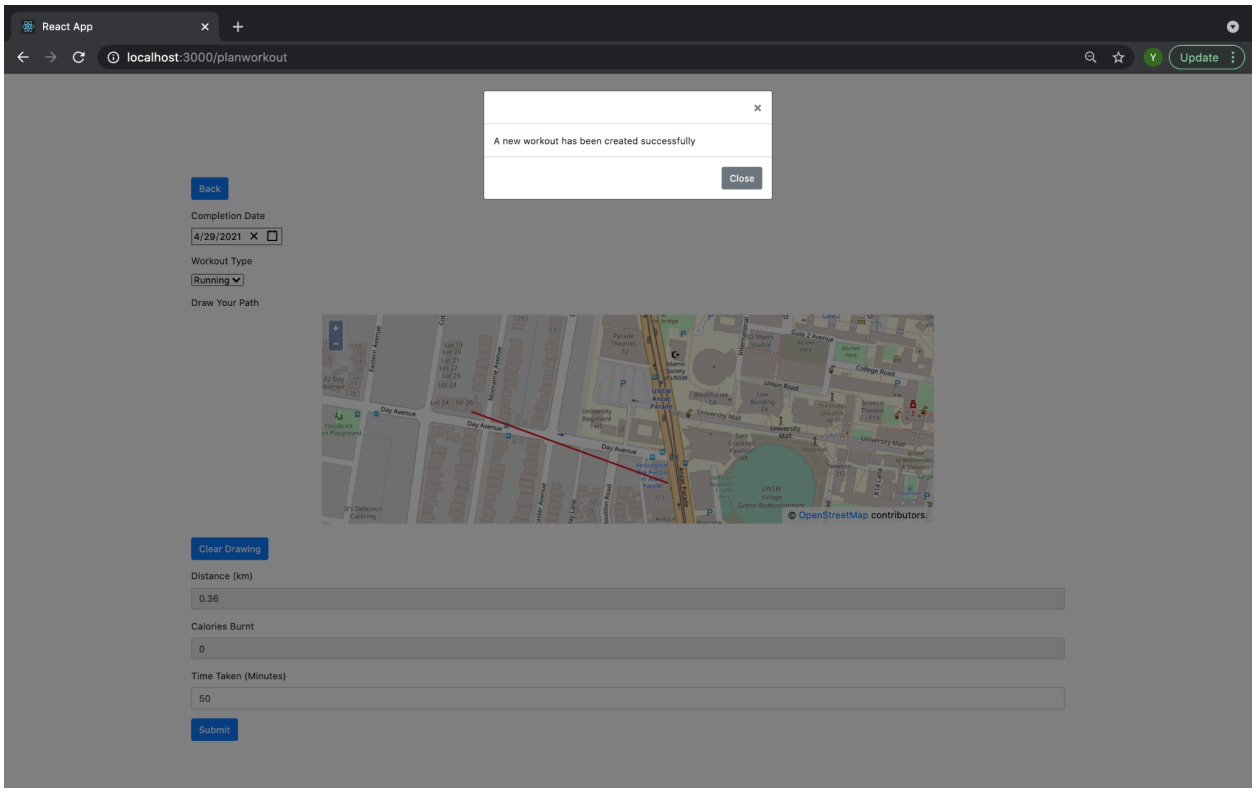

React App
js how to change string to int
localhost:3000/planworkout
Update
Plan New Workout
Back
Completion Date
--/--/----
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0
Calories Burnt
0
Time Taken (Minutes)
0
Submit


React App
js how to change string to int
localhost:3000/planworkout
Update
Please provide a completion date!
Close
Back
Completion Date
--/--/----
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0
Calories Burnt
0
Time Taken (Minutes)
0
Submit


React App
localhost:3000/planworkout
Update
Invalid completion date!
Close
Back
Completion Date
4/14/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0
Calories Burnt
0
Time Taken (Minutes)
0
Submit


React App
localhost:3000/planworkout
Update
Please draw a path on the map!
Close
Back
Completion Date
4/29/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0
Calories Burnt
0
Time Taken (Minutes)
0
Submit


React App
localhost:3000/planworkout
Update
Please provide a positive time!
Close
Back
Completion Date
4/29/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0.36
Calories Burnt
0
Time Taken (Minutes)
-10
Submit


React App
localhost:3000/planworkout
Update
A new workout has been created successfully
Close
Back
Completion Date
4/29/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0.36
Calories Burnt
0
Time Taken (Minutes)
50
Submit

Upcoming Workouts

Back

Start Date --/--/---- End Date --/--/---- Workout Type All Apply Clear

Completion Date	Workout Type	Distance (km)	Calories burnt	Time Taken (Minutes)
2021-04-29	Running	0.36	820	50

1

New Workout Details

Back

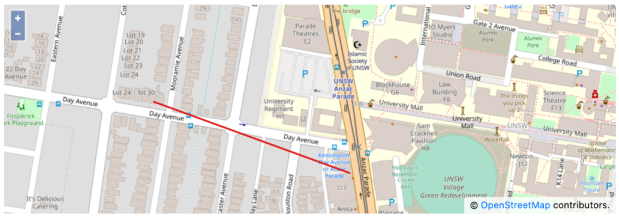
Completion Date

4/29/2021

Workout Type

Running

Draw Your Path



Distance (km)

0.36

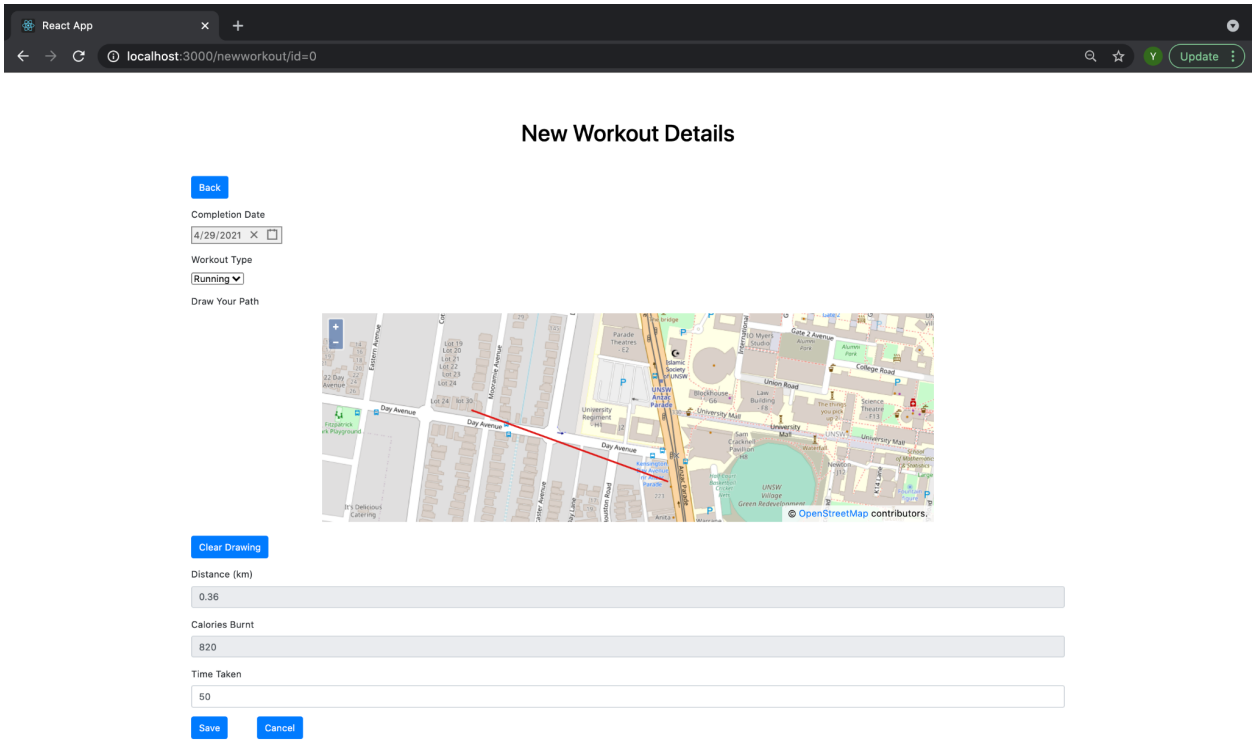
Calories Burnt

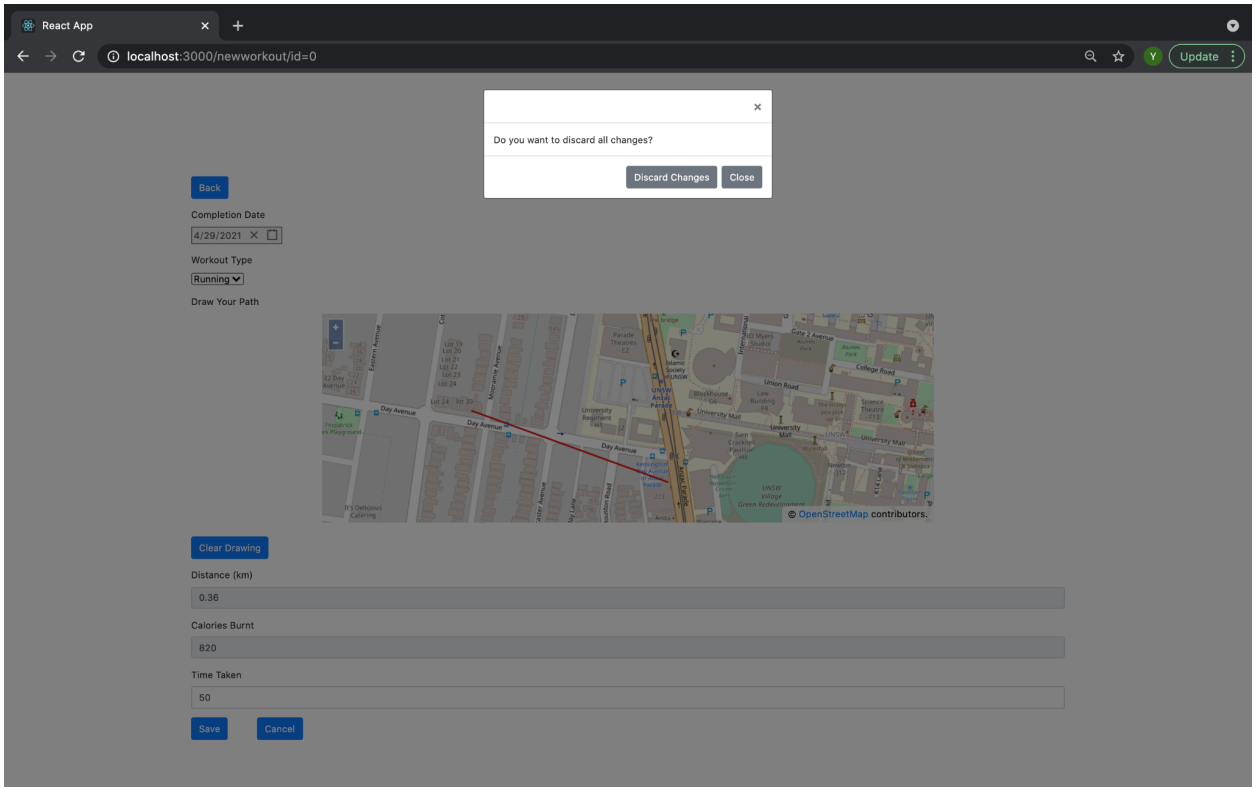
820

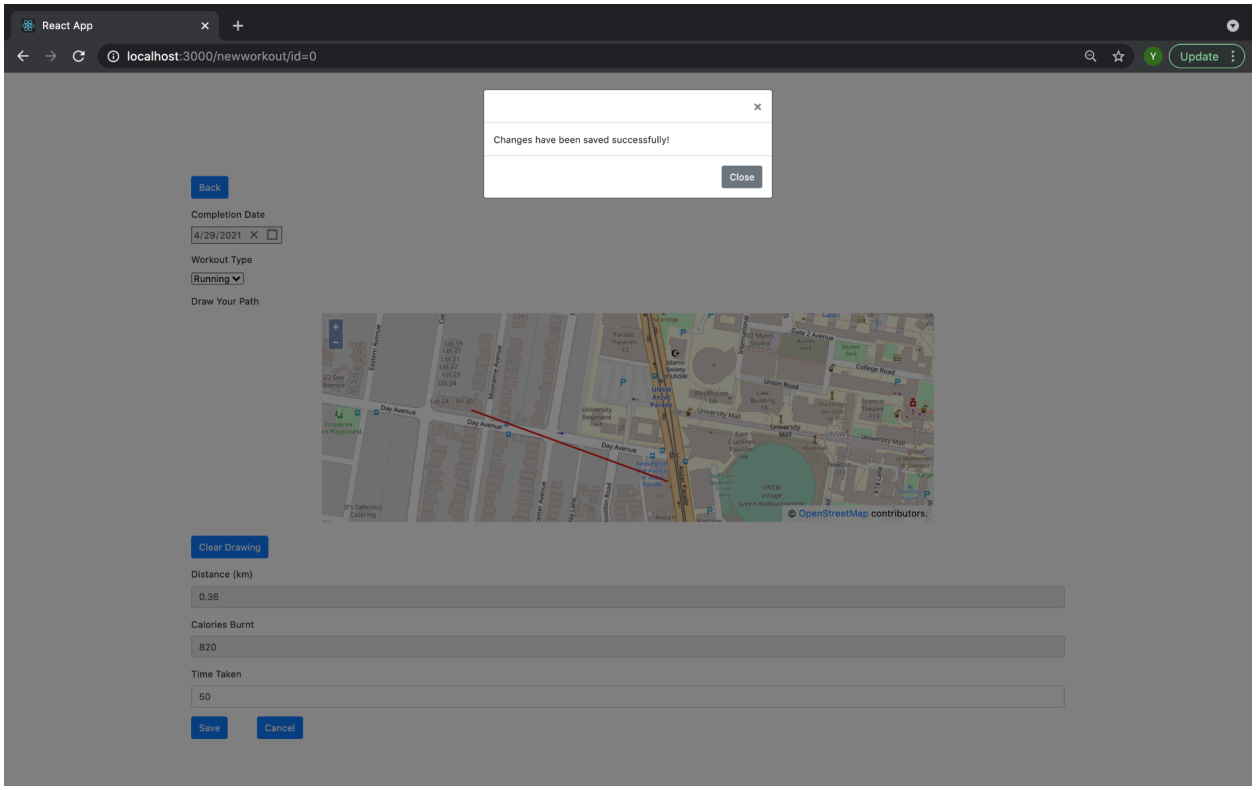
Time Taken

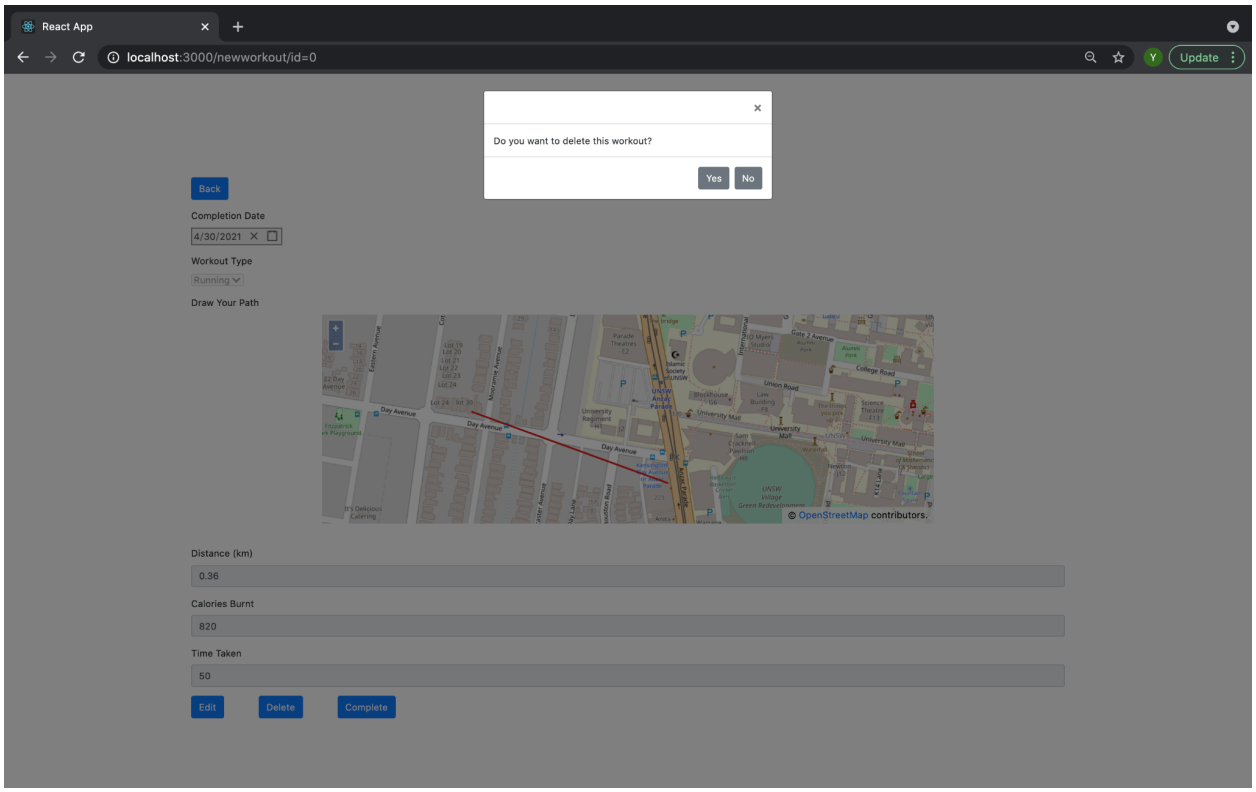
50

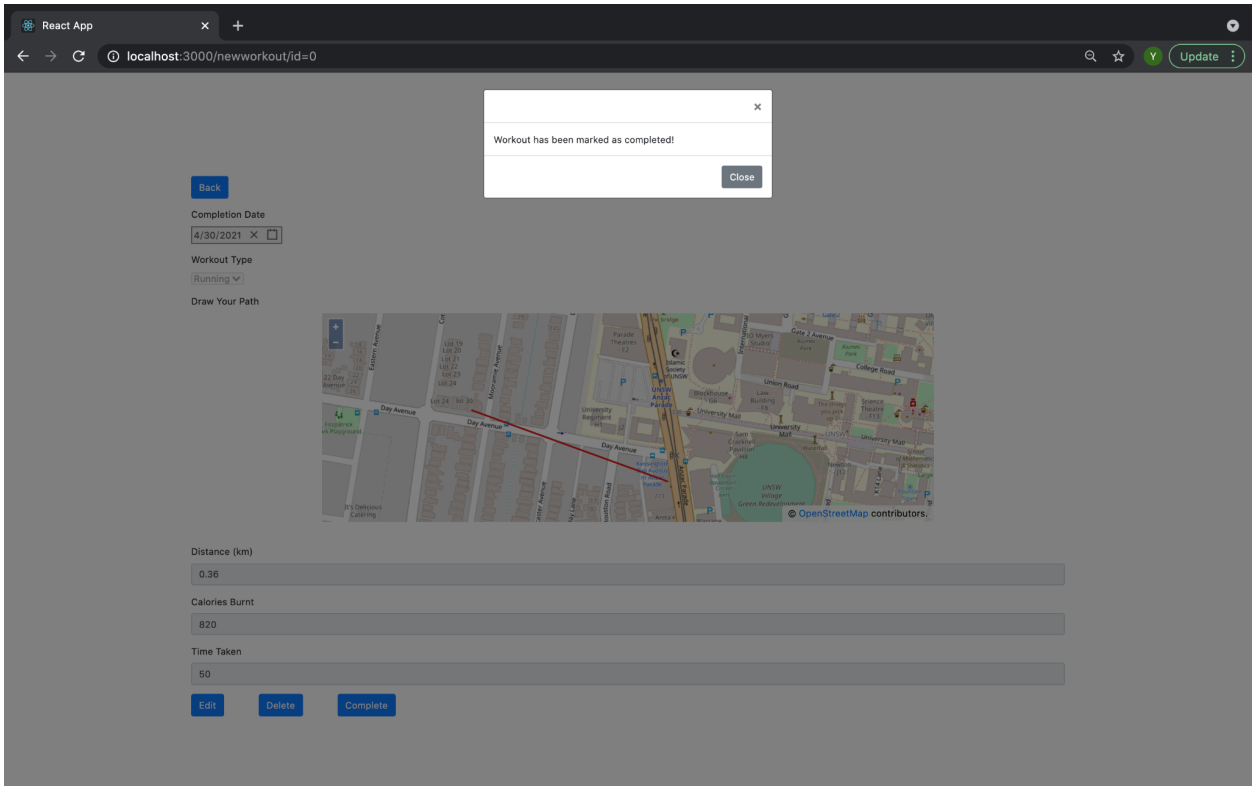
Edit Delete Complete

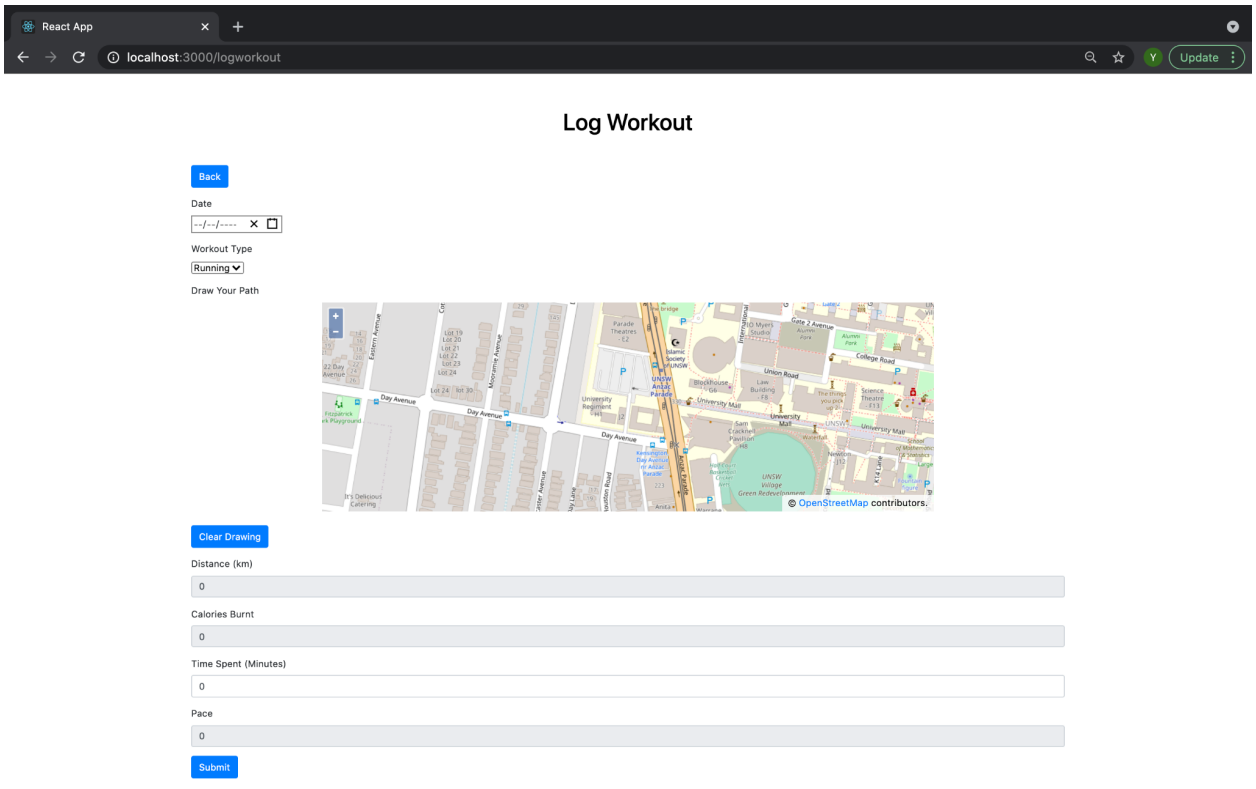

React App
localhost:3000/newworkout/id=0
Update
New Workout Details
Back
Completion Date
4/29/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0.36
Calories Burnt
820
Time Taken
50
Save
Cancel

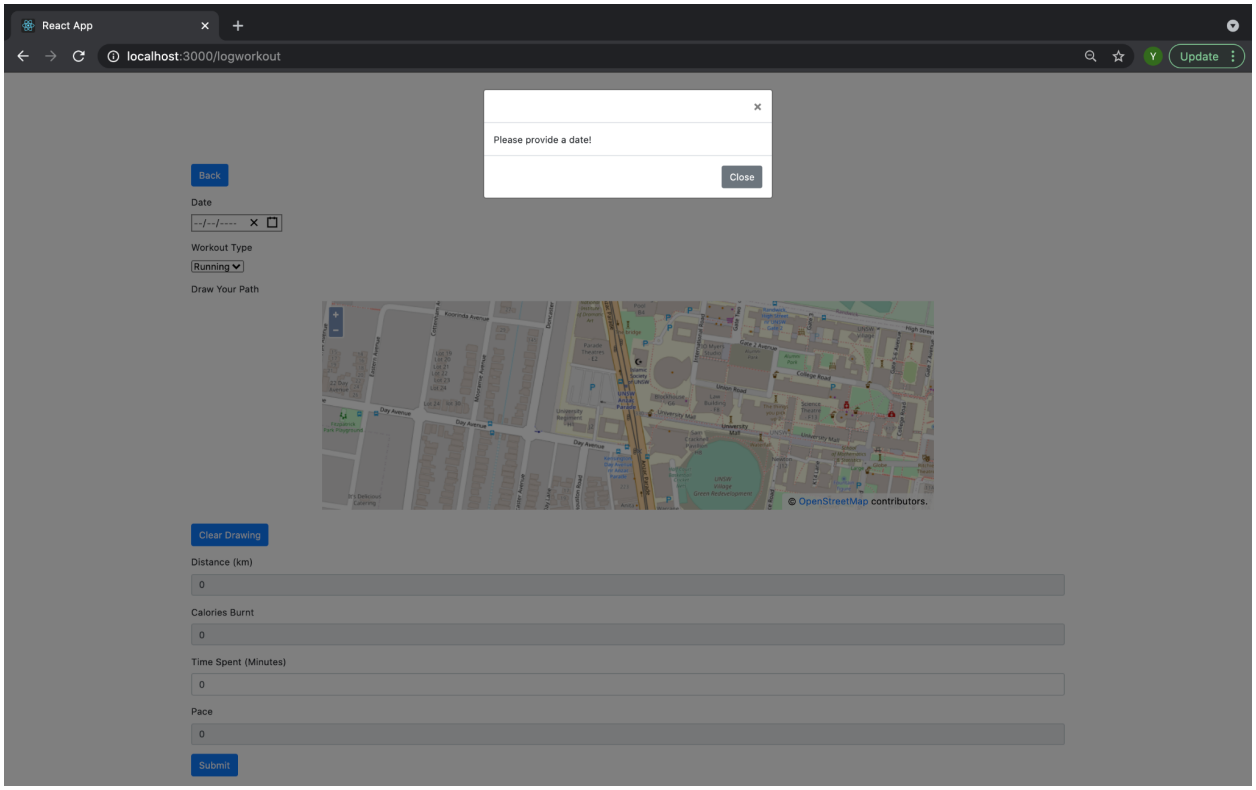

React App
localhost:3000/newworkout/id=0
Update
Do you want to discard all changes?
Discard Changes
Close
Back
Completion Date
4/29/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0.36
Calories Burnt
820
Time Taken
50
Save
Cancel

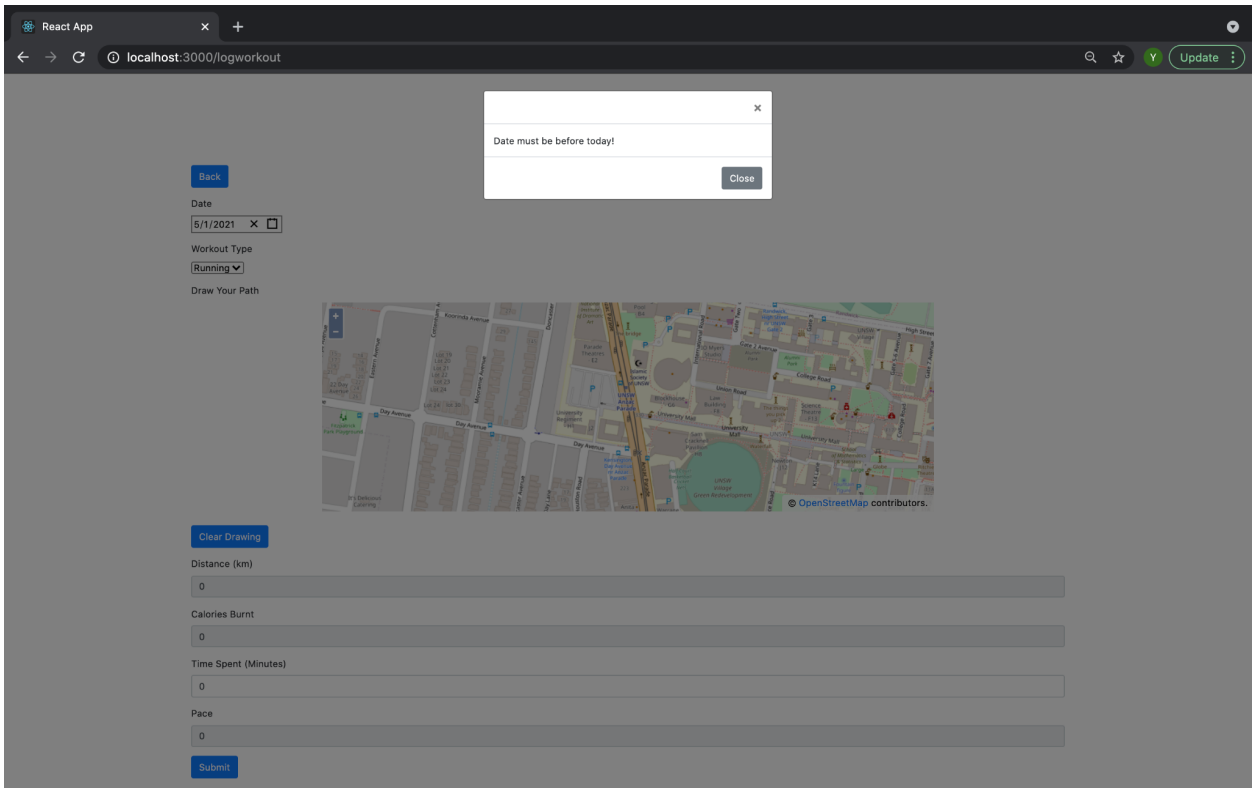

React App
localhost:3000/newworkout/id=0
Update
Changes have been saved successfully!
Close
Back
Completion Date
4/29/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0.36
Calories Burnt
820
Time Taken
50
Save
Cancel

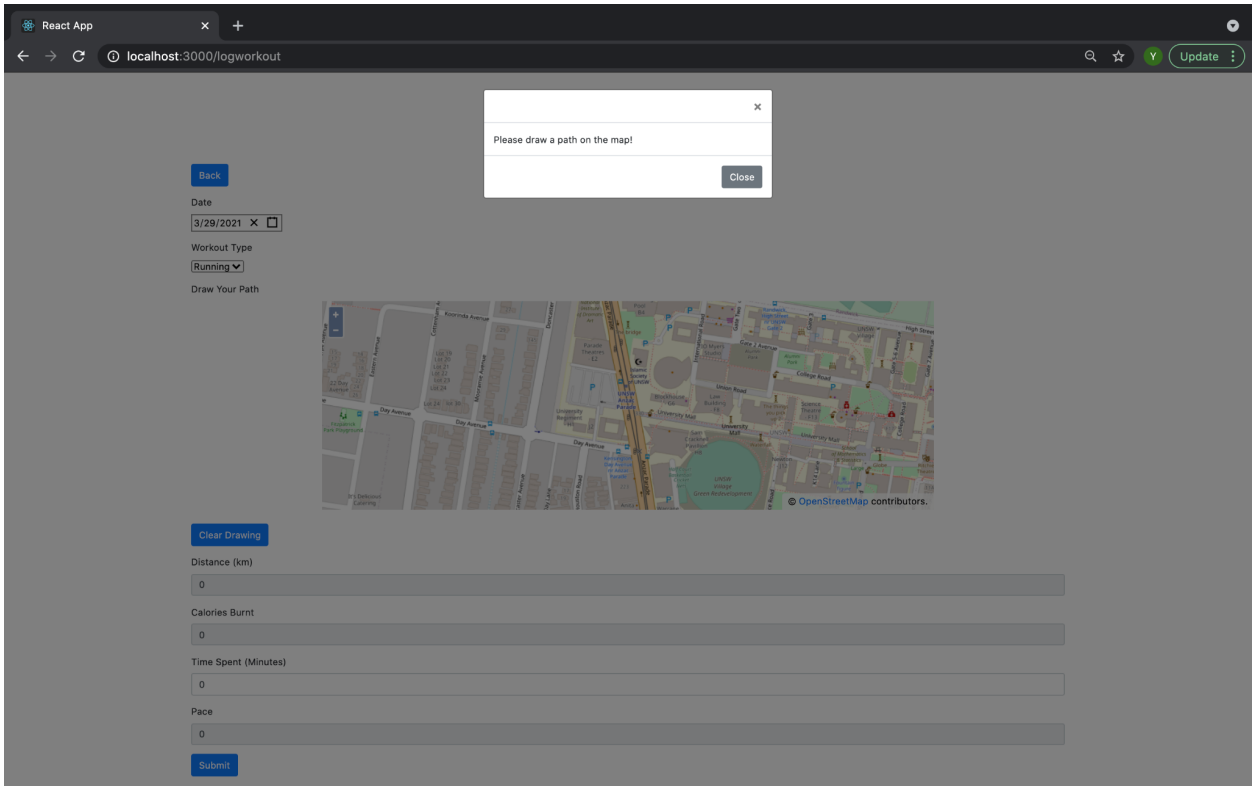

React App
localhost:3000/newworkout/id=0
Update
Do you want to delete this workout?
Yes
No
Back
Completion Date
4/30/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Distance (km)
0.36
Calories Burnt
820
Time Taken
50
Edit
Delete
Complete

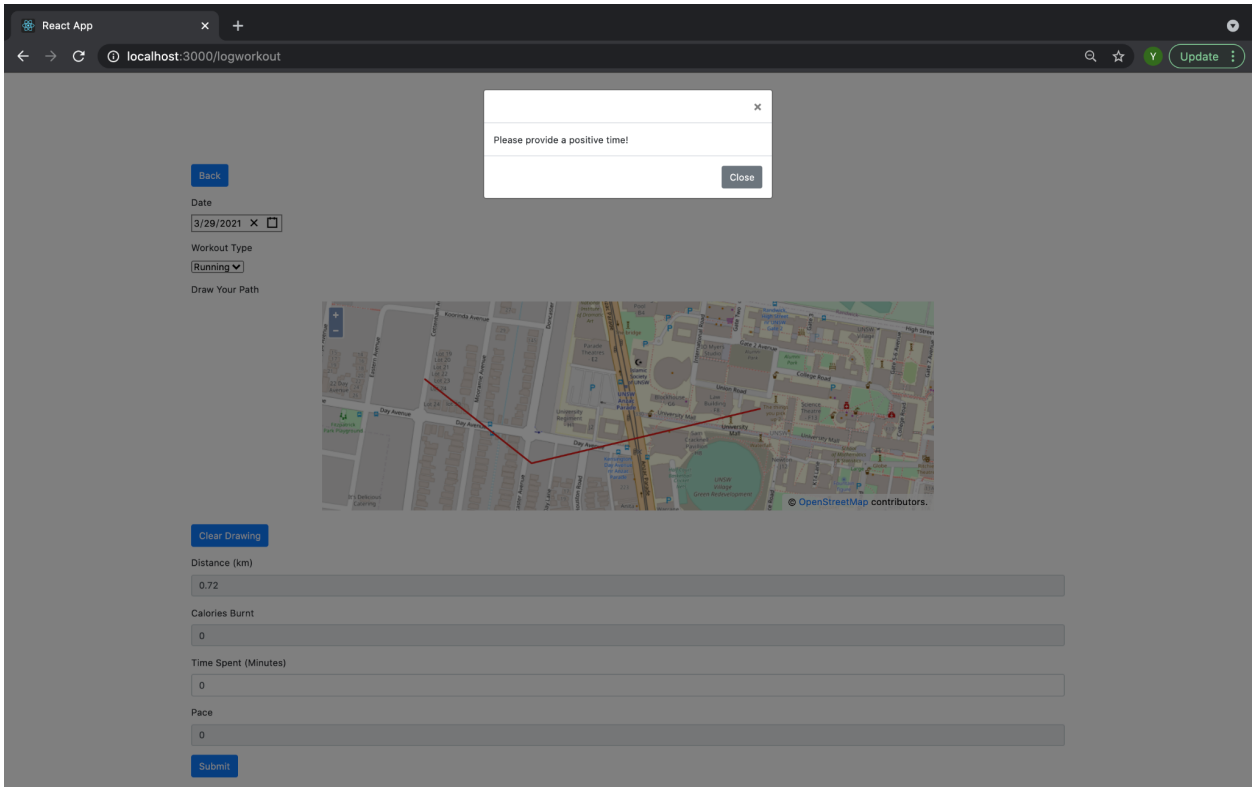

React App
localhost:3000/newworkout/id=0
Update
Workout has been marked as completed!
Close
Back
Completion Date
4/30/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Distance (km)
0.36
Calories Burnt
820
Time Taken
50
Edit
Delete
Complete

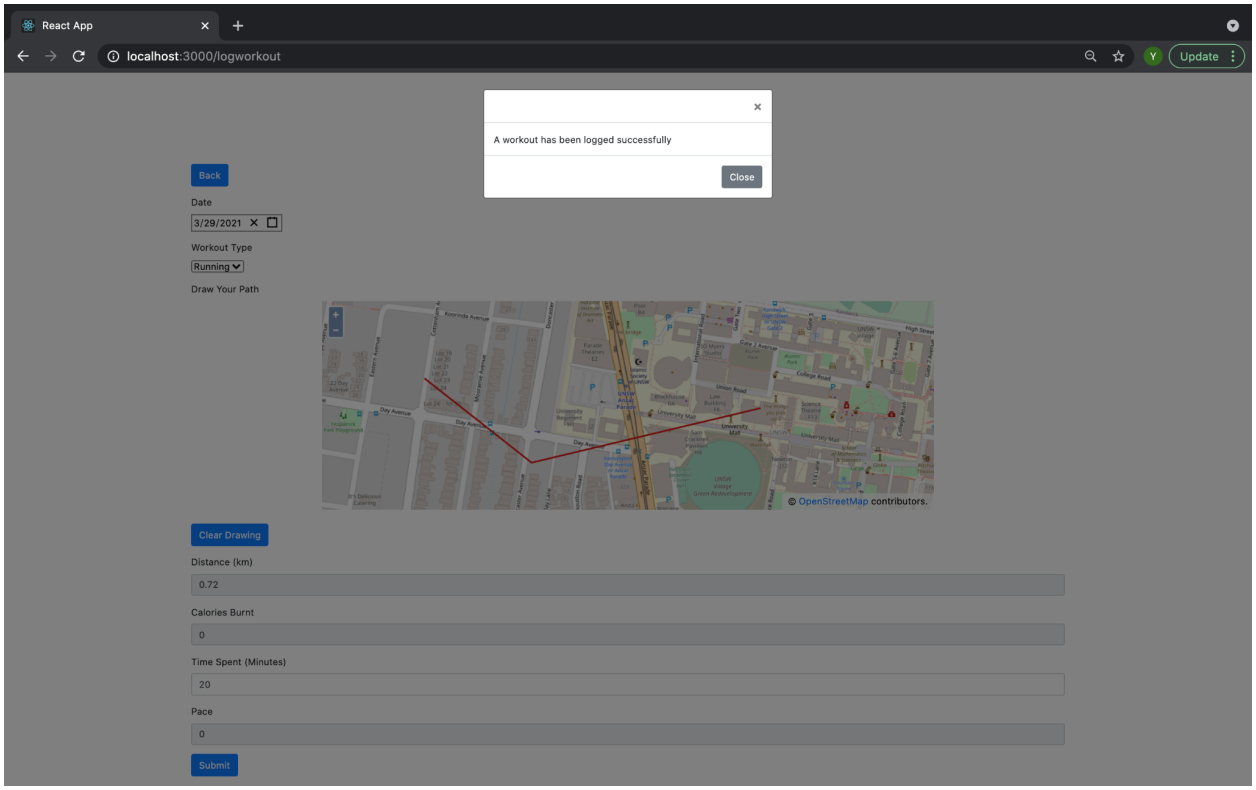

React App
localhost:3000/logworkout
Update
Log Workout
Back
Date
--/--/----
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0
Calories Burnt
0
Time Spent (Minutes)
0
Pace
0
Submit

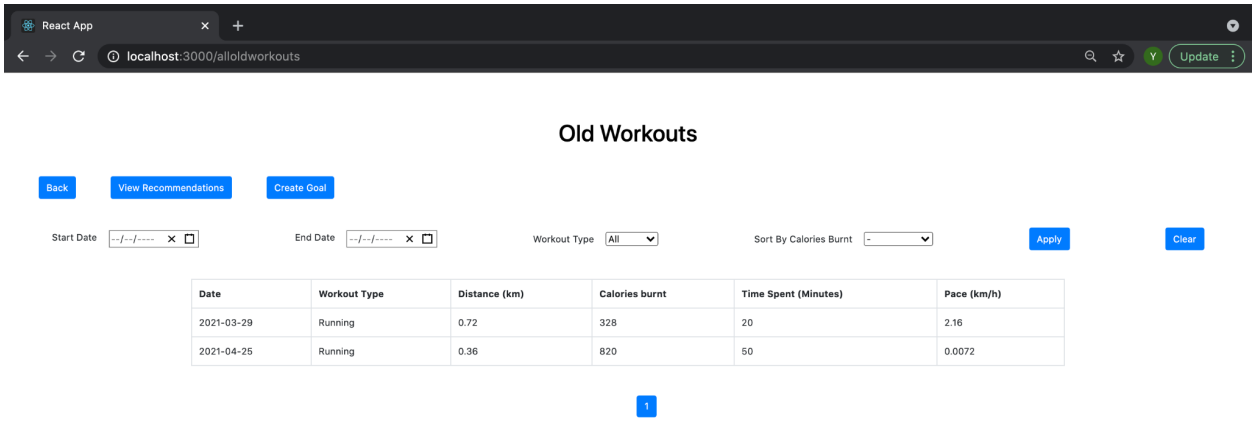

React App
localhost:3000/logworkout
Update
Please provide a date!
Close
Back
Date
--/--/----
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0
Calories Burnt
0
Time Spent (Minutes)
0
Pace
0
Submit

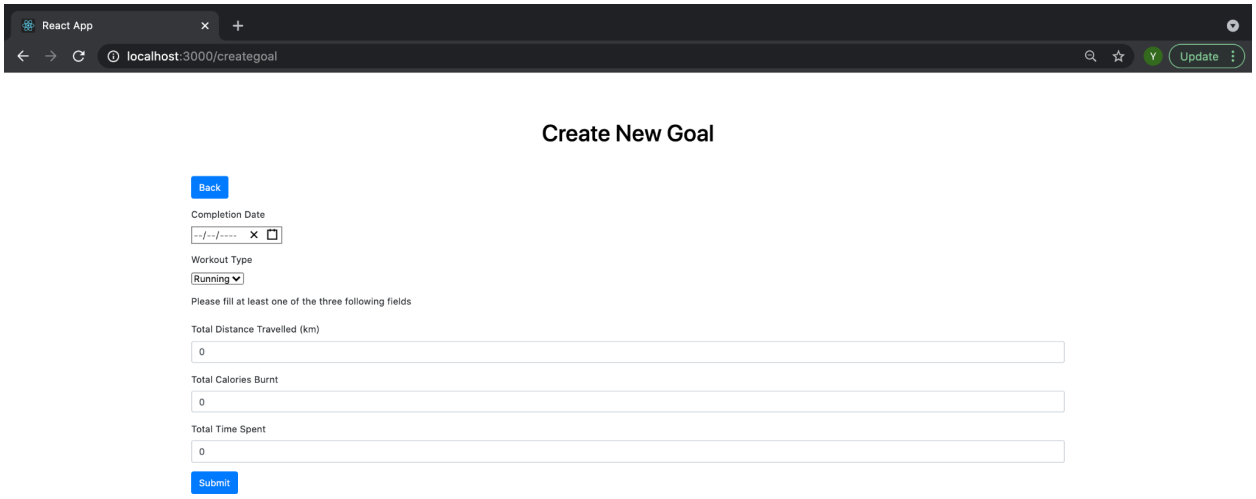

React App
localhost:3000/logworkout
Update
Date must be before today!
Close
Back
Date
5/1/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0
Calories Burnt
0
Time Spent (Minutes)
0
Pace
0
Submit

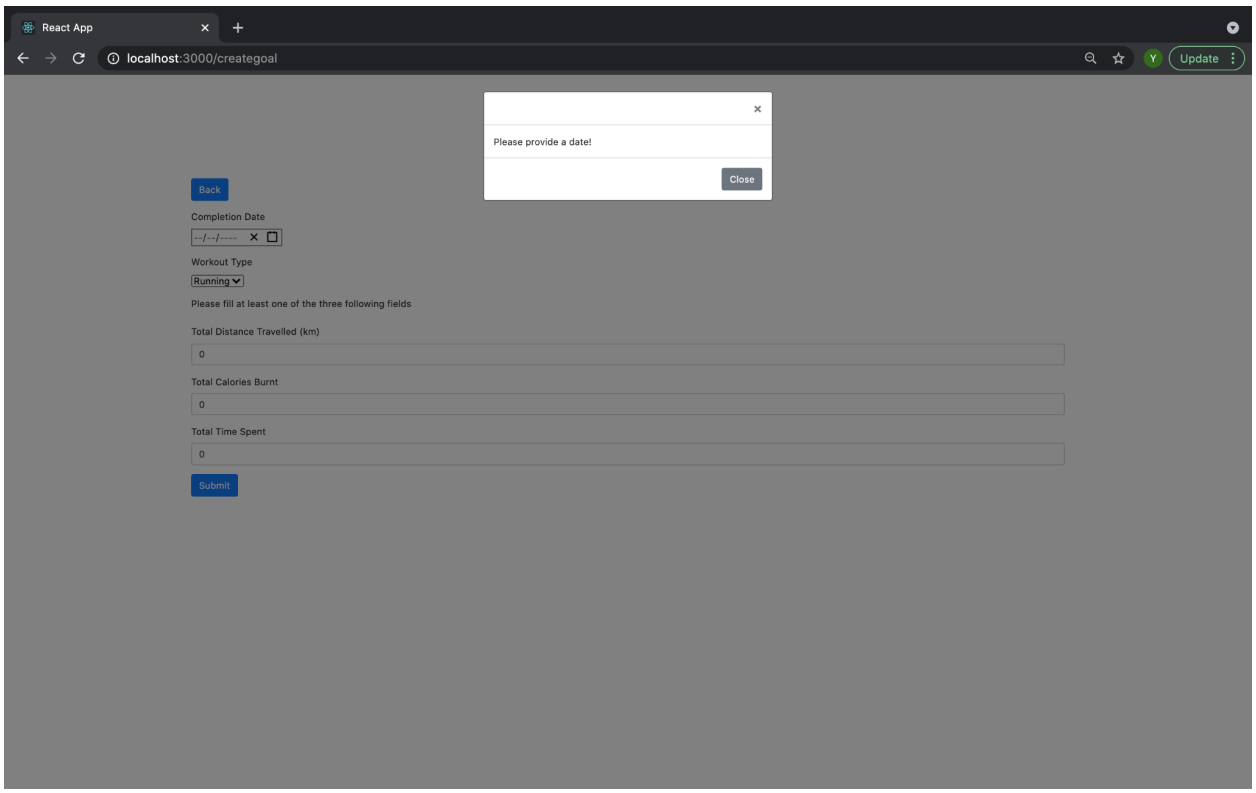

React App
localhost:3000/logworkout
Update
Please draw a path on the map!
Close
Back
Date
3/29/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0
Calories Burnt
0
Time Spent (Minutes)
0
Pace
0
Submit

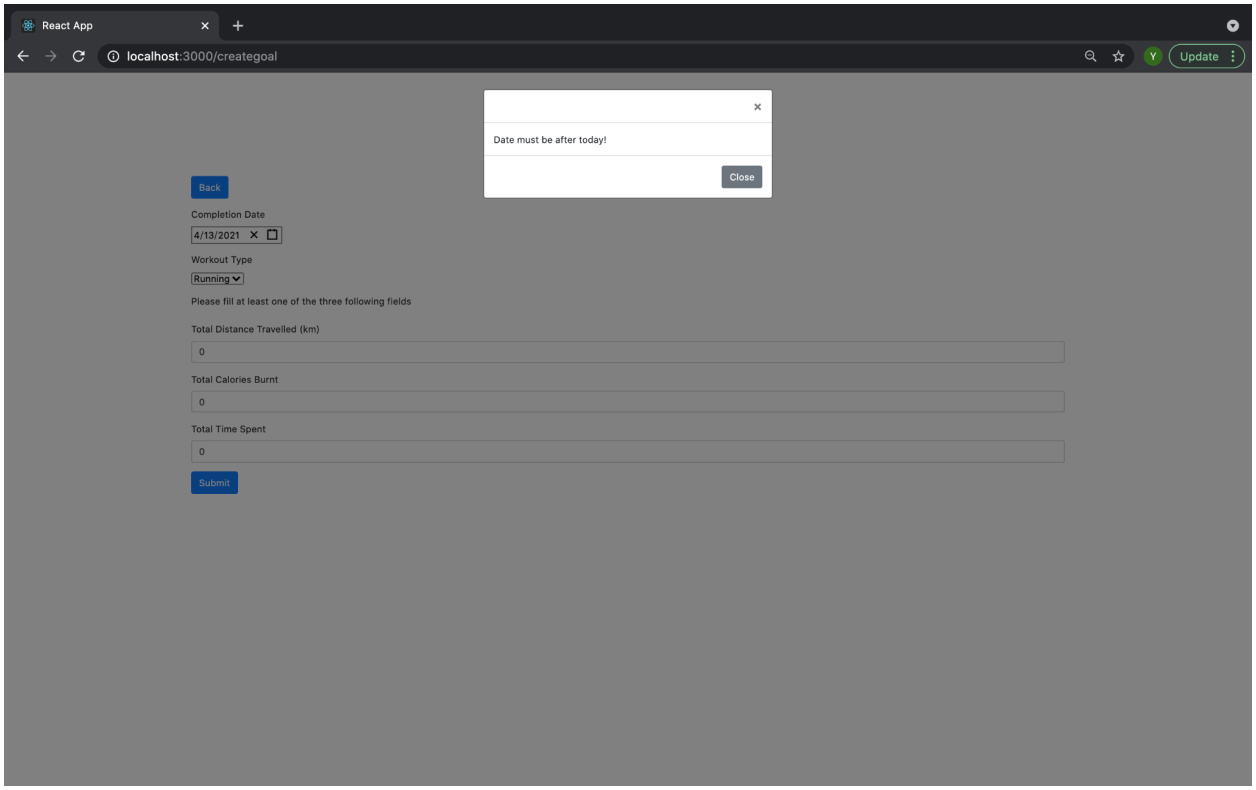

React App
localhost:3000/logworkout
Update
Please provide a positive time!
Close
Back
Date
3/29/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0.72
Calories Burnt
0
Time Spent (Minutes)
0
Pace
0
Submit

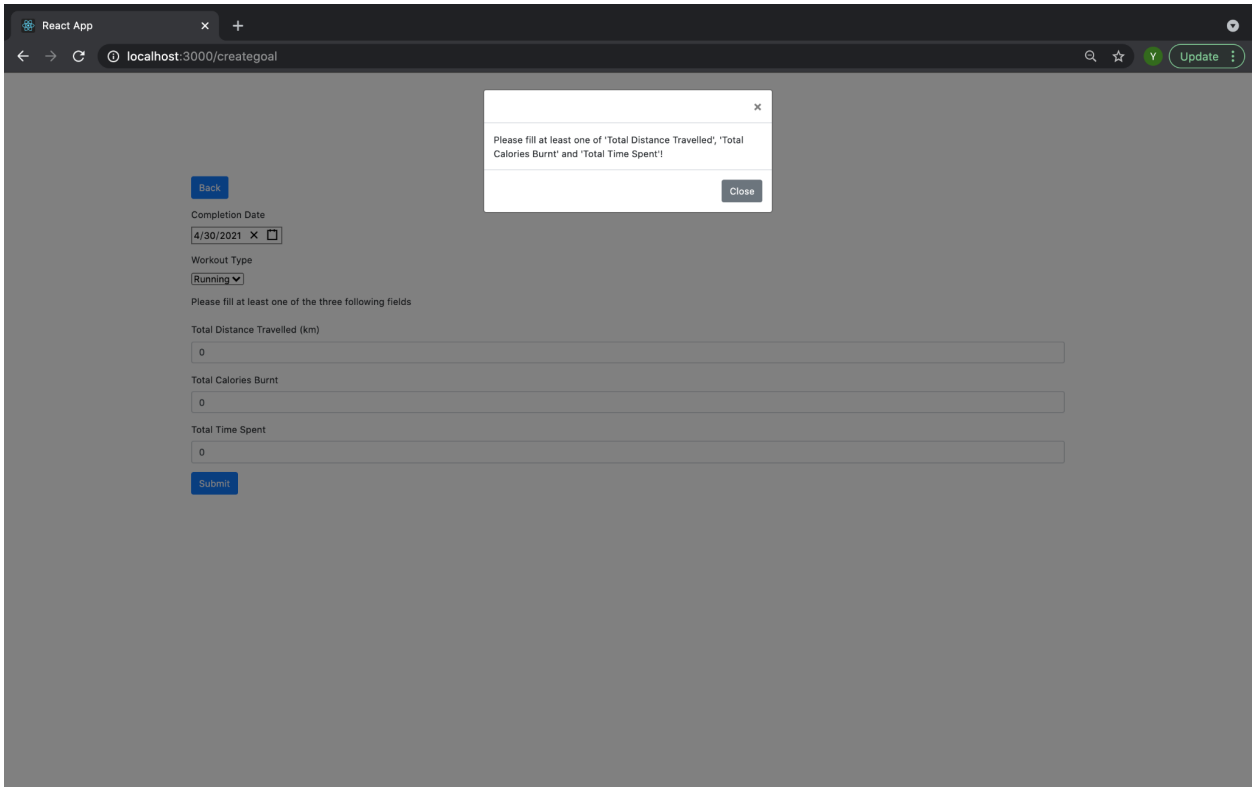

React App
localhost:3000/logworkout
Update
A workout has been logged successfully
Close
Back
Date
3/29/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Distance (km)
0.72
Calories Burnt
0
Time Spent (Minutes)
20
Pace
0
Submit

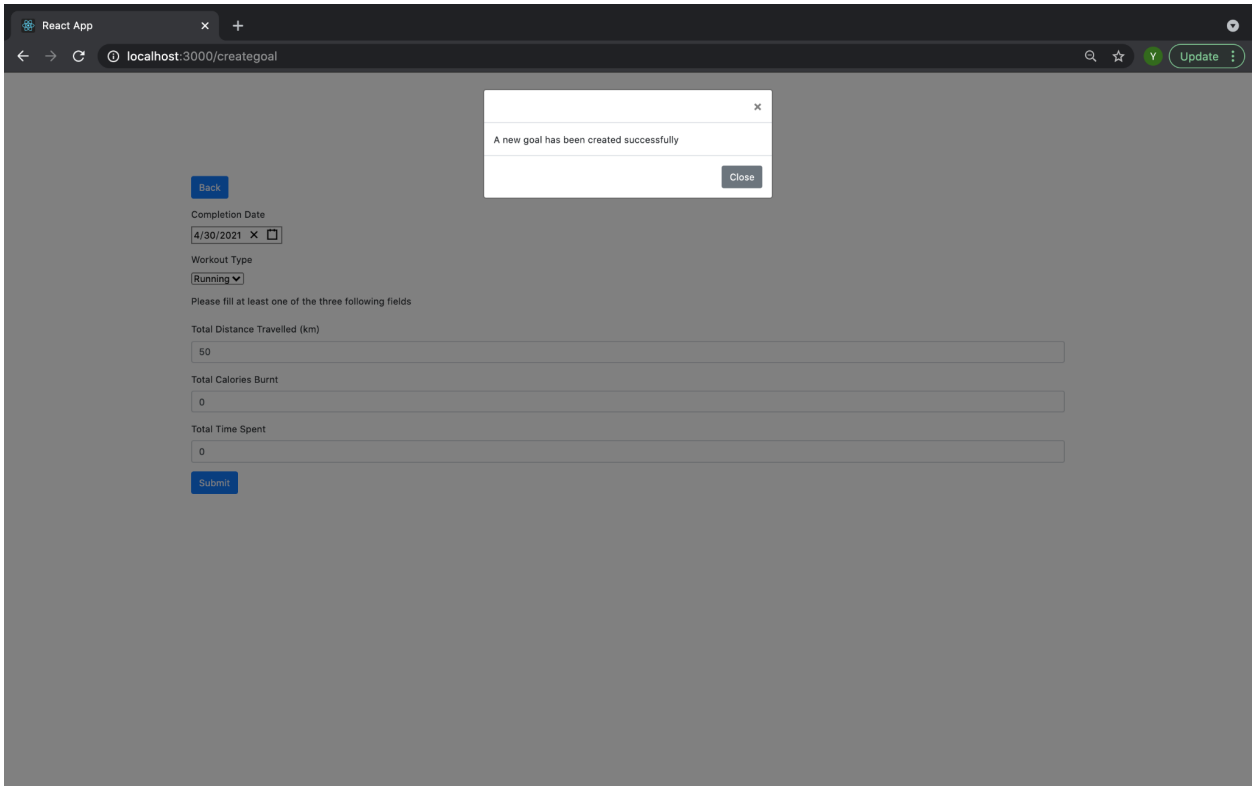

React App
localhost:3000/alloldworkouts
Update
Old Workouts
Back
View Recommendations
Create Goal
Start Date
End Date
Workout Type
All
Sort By Calories Burnt
Apply
Clear
Date
Workout Type
Distance (km)
Calories burnt
Time Spent (Minutes)
Pace (km/h)
2021-03-29
Running
0.72
328
20
2.16
2021-04-25
Running
0.36
820
50
0.0072
1

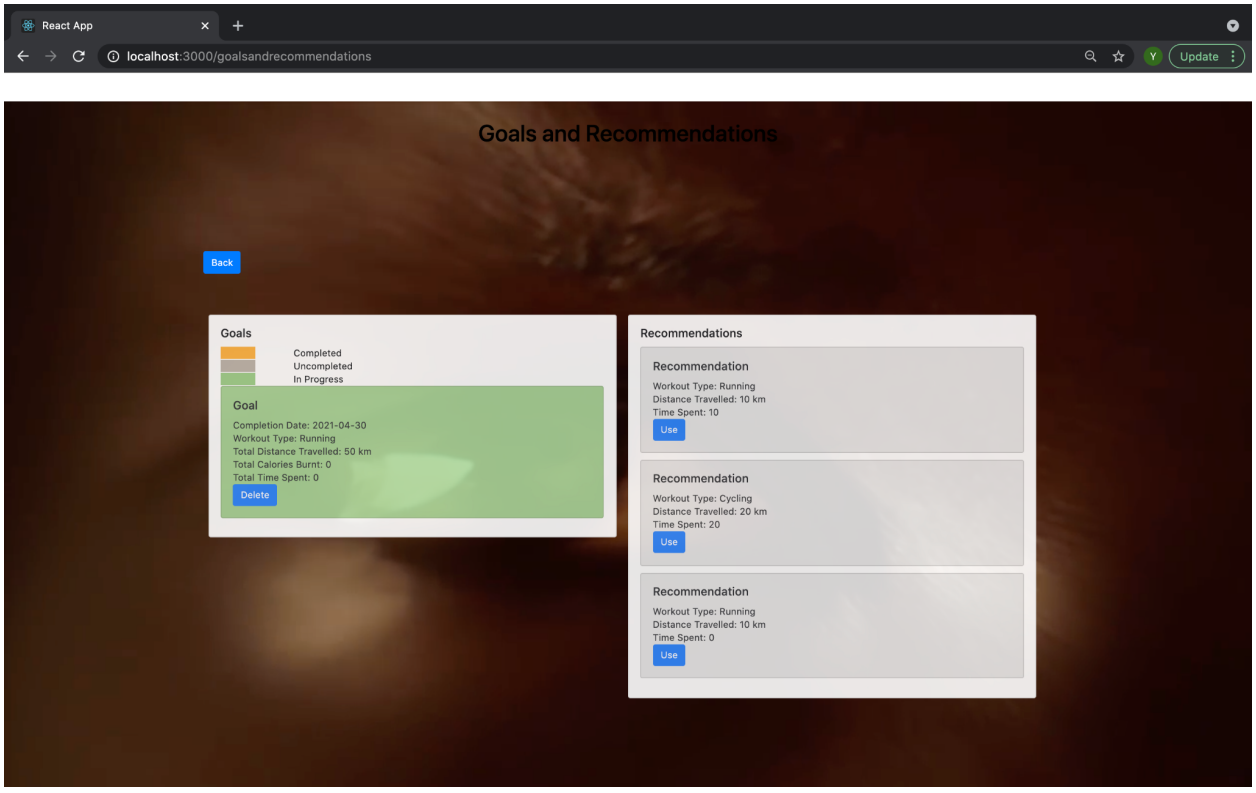

React App
localhost:3000/creategoal
Update
Create New Goal
Back
Completion Date
--/--/----
Workout Type
Running
Please fill at least one of the three following fields
Total Distance Travelled (km)
0
Total Calories Burnt
0
Total Time Spent
0
Submit

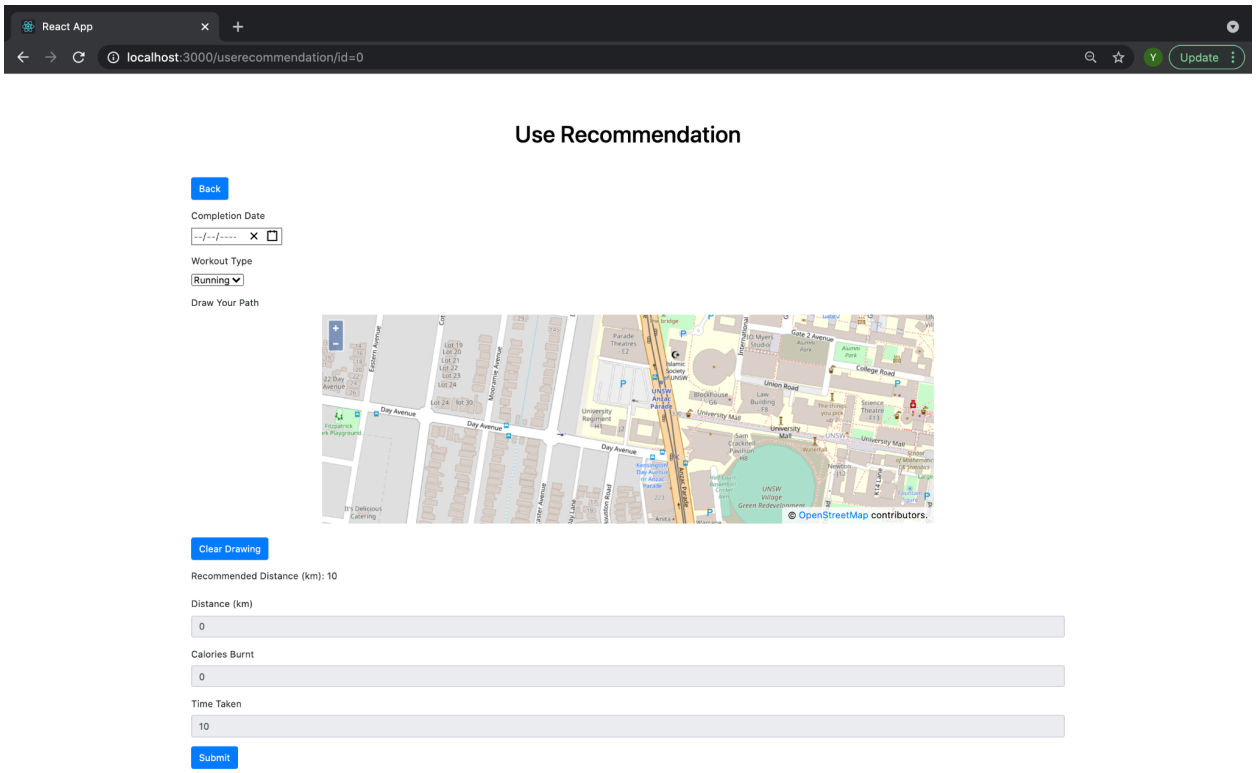

React App
localhost:3000/creategoal
Update
Please provide a date!
Close
Back
Completion Date
--/--/----
Workout Type
Running
Please fill at least one of the three following fields
Total Distance Travelled (km)
0
Total Calories Burnt
0
Total Time Spent
0
Submit

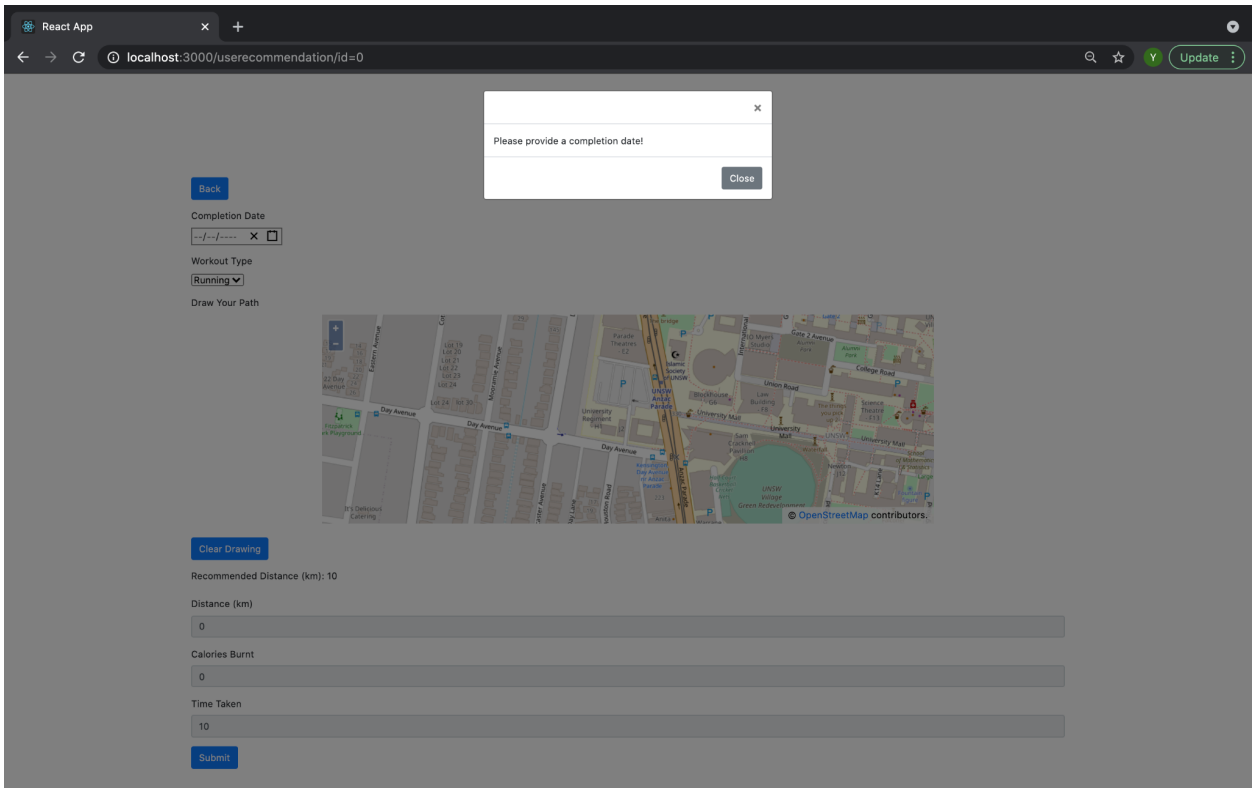

React App
localhost:3000/creategoal
Update
Date must be after today!
Close
Back
Completion Date
4/13/2021
Workout Type
Running
Please fill at least one of the three following fields
Total Distance Travelled (km)
0
Total Calories Burnt
0
Total Time Spent
0
Submit

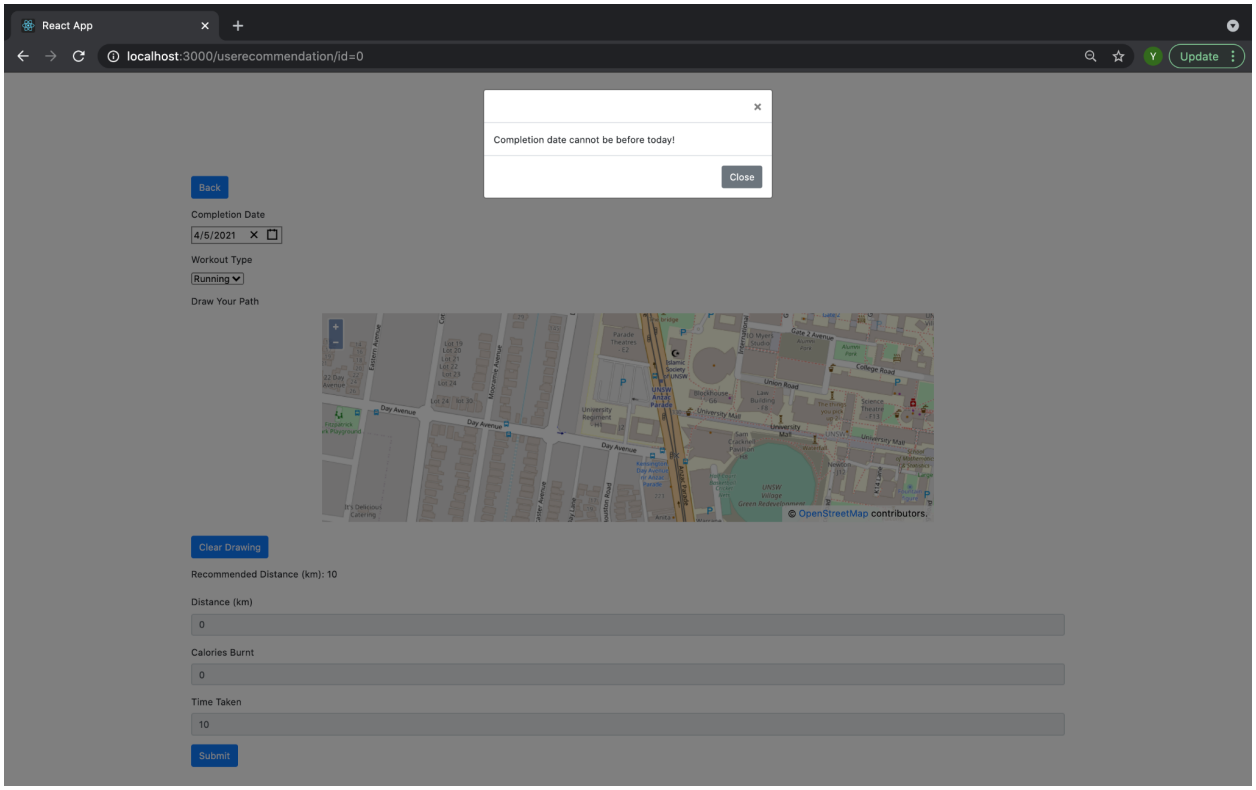

React App
localhost:3000/creategoal
Update
Please fill at least one of 'Total Distance Travelled', 'Total Calories Burnt' and 'Total Time Spent'!
Close
Back
Completion Date
4/30/2021
Workout Type
Running
Please fill at least one of the three following fields
Total Distance Travelled (km)
0
Total Calories Burnt
0
Total Time Spent
0
Submit

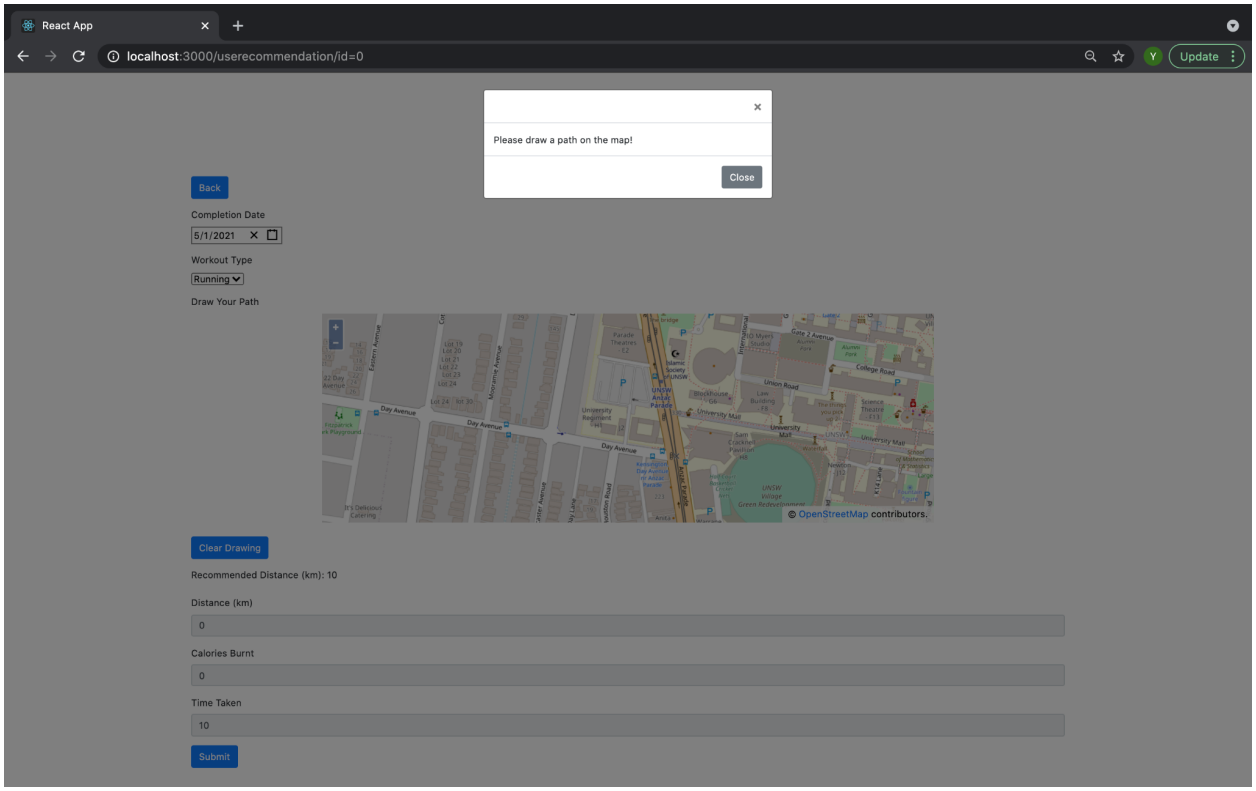

React App
localhost:3000/creategoal
Update
A new goal has been created successfully
Close
Back
Completion Date
4/30/2021
Workout Type
Running
Please fill at least one of the three following fields
Total Distance Travelled (km)
50
Total Calories Burnt
0
Total Time Spent
0
Submit

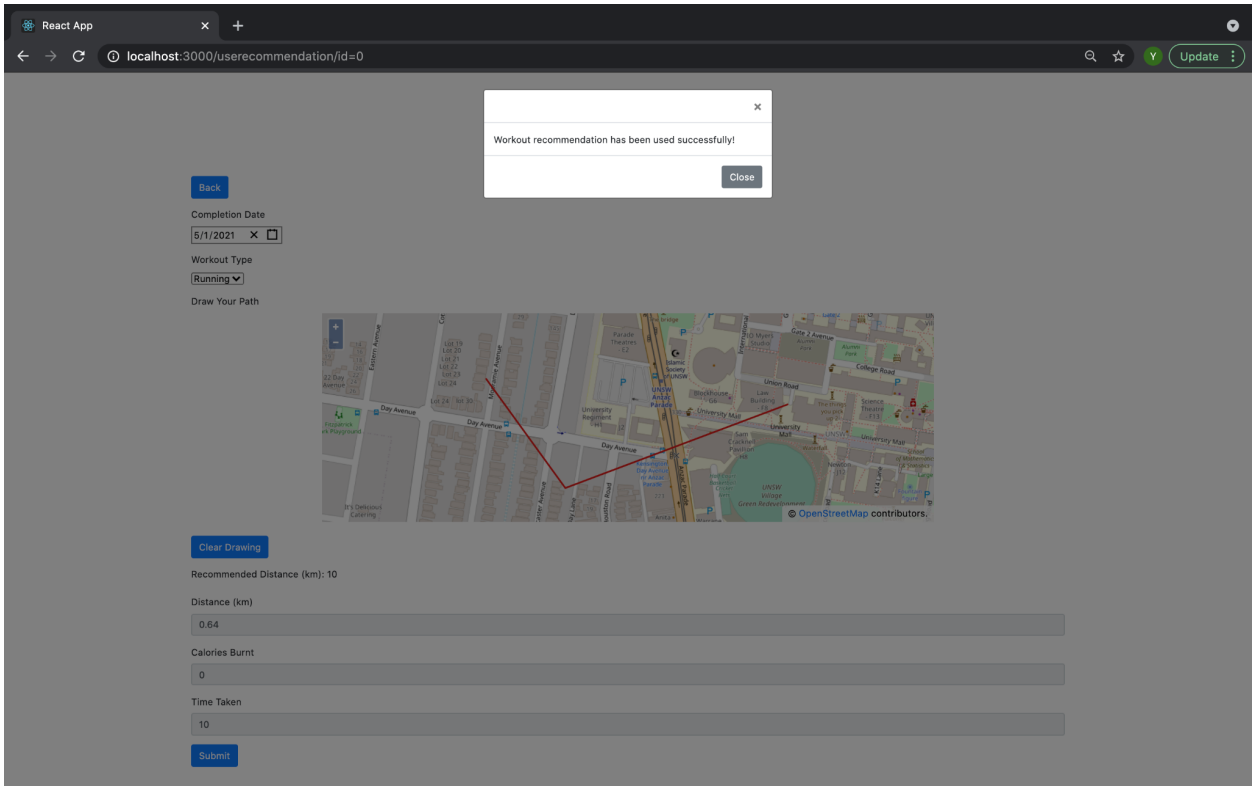

React App
localhost:3000/goalsandrecommendations
Update
Goals and Recommendations
Back
Goals
Completed
Uncompleted
In Progress
Goal
Completion Date: 2021-04-30
Workout Type: Running
Total Distance Travelled: 50 km
Total Calories Burnt: 0
Total Time Spent: 0
Delete
Recommendations
Recommendation
Workout Type: Running
Distance Travelled: 10 km
Time Spent: 10
Use
Recommendation
Workout Type: Cycling
Distance Travelled: 20 km
Time Spent: 20
Use
Recommendation
Workout Type: Running
Distance Travelled: 10 km
Time Spent: 0
Use

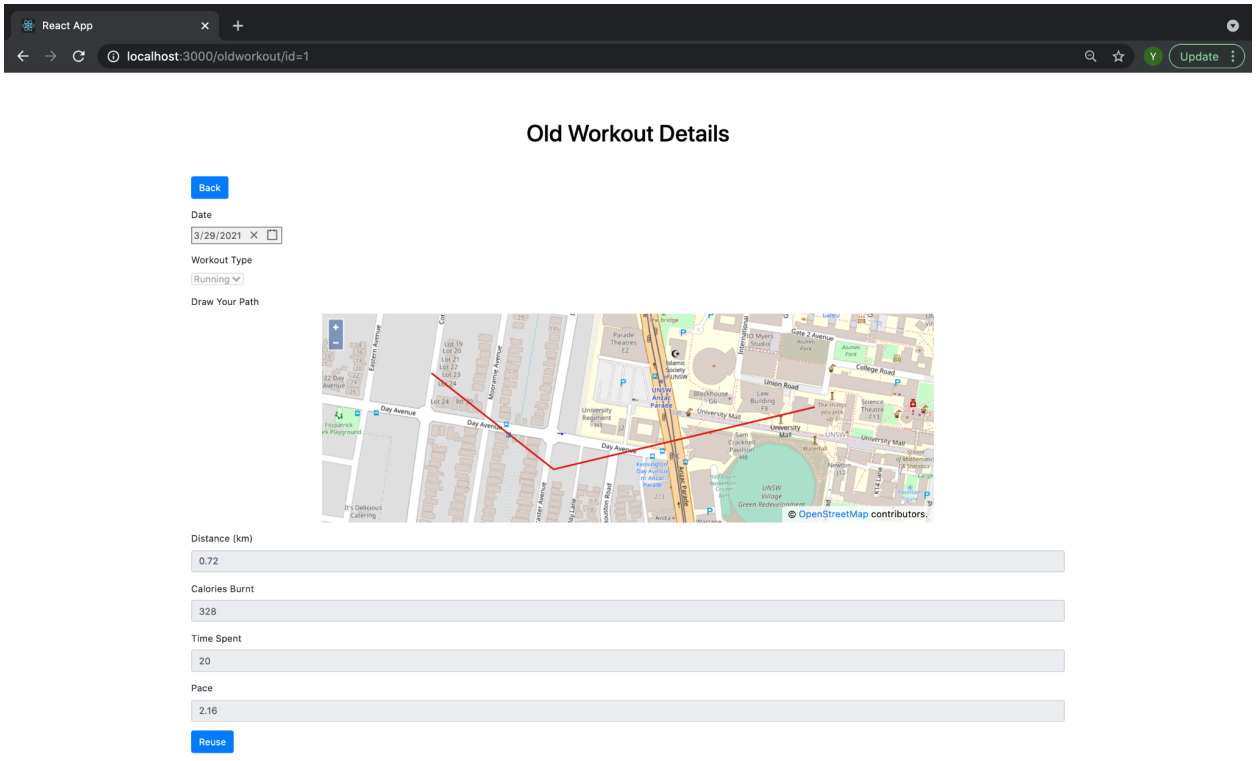

React App
localhost:3000/userecommendation/id=0
Update
Use Recommendation
Back
Completion Date
--/--/----
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Recommended Distance (km): 10
Distance (km)
0
Calories Burnt
0
Time Taken
10
Submit

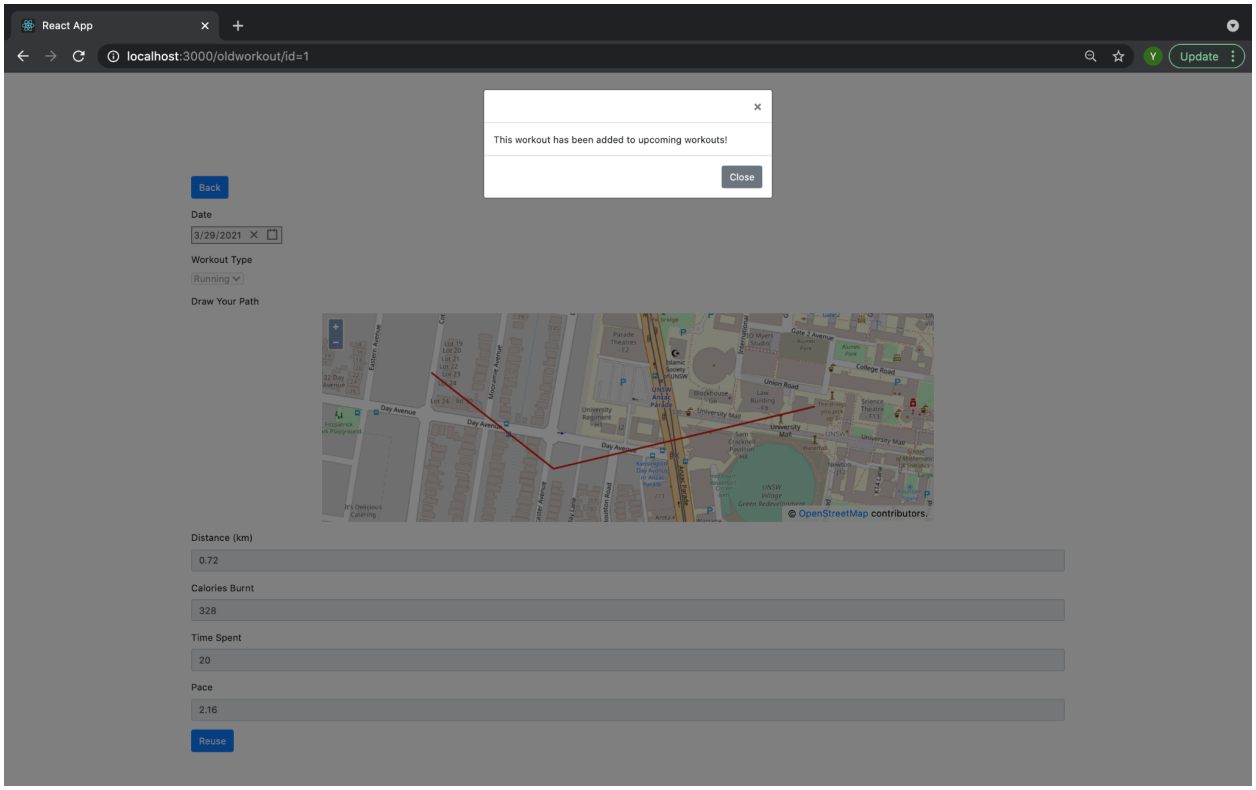

React App
localhost:3000/userecommendation/id=0
Update
Please provide a completion date!
Close
Back
Completion Date
--/--/----
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Recommended Distance (km): 10
Distance (km)
0
Calories Burnt
0
Time Taken
10
Submit

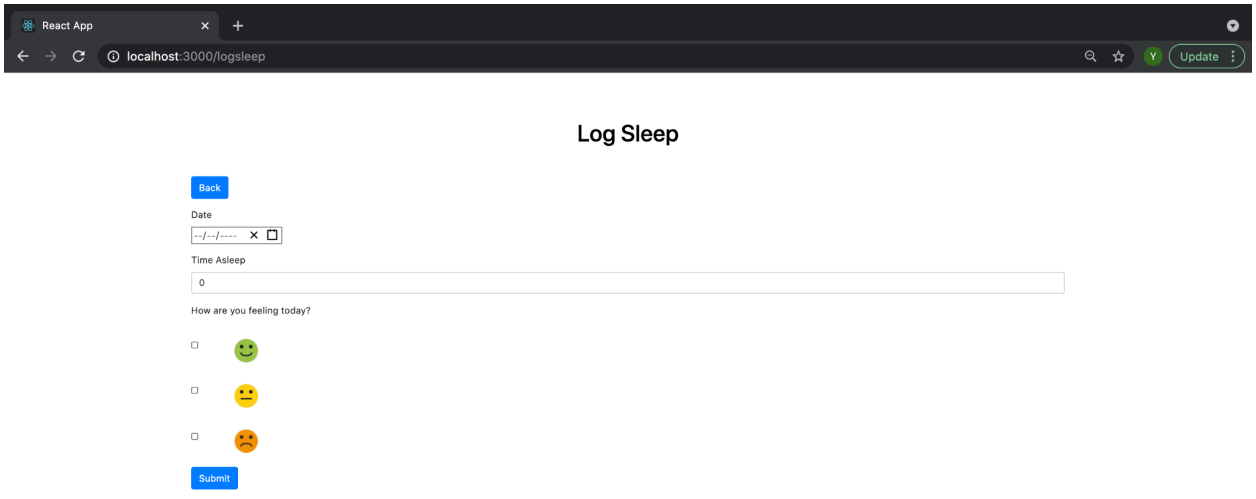

React App
localhost:3000/userecommendation/id=0
Update
Completion date cannot be before today!
Close
Back
Completion Date
4/5/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Recommended Distance (km): 10
Distance (km)
0
Calories Burnt
0
Time Taken
10
Submit

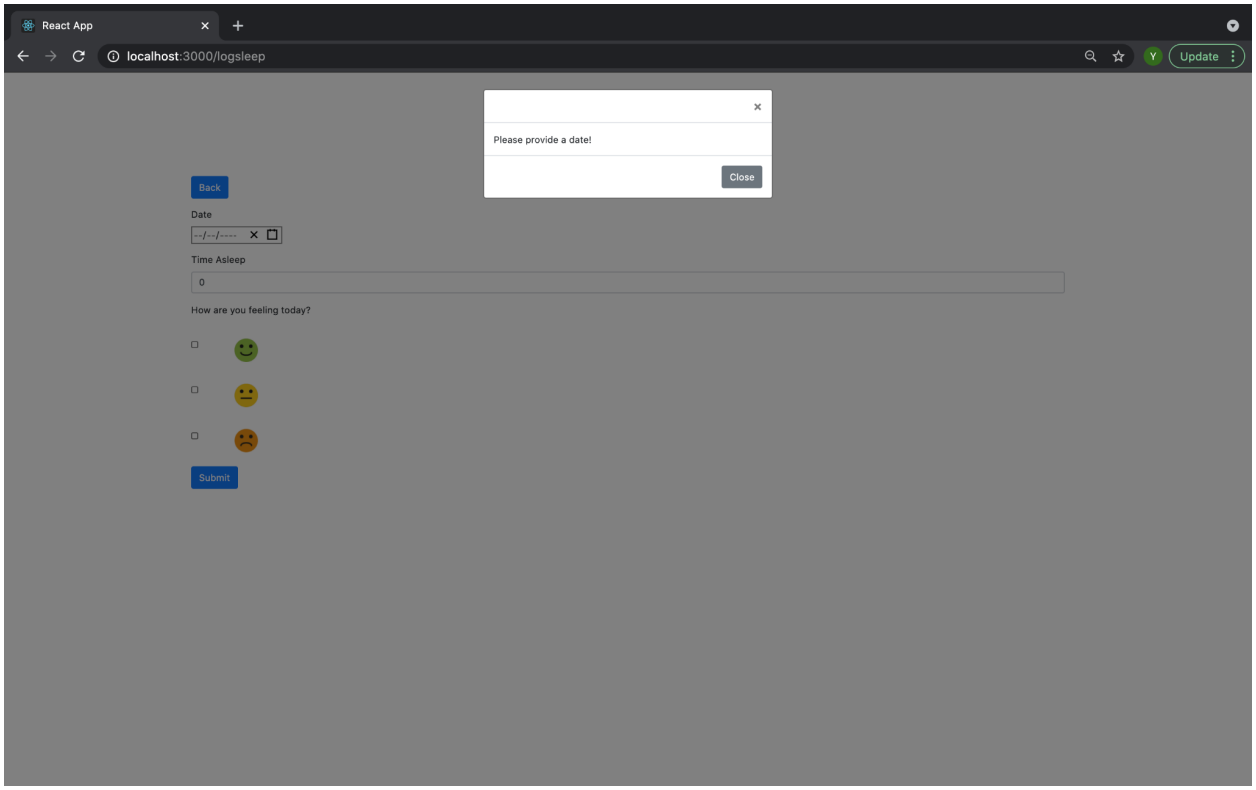

React App
localhost:3000/userecommendation/id=0
Update
Please draw a path on the map!
Close
Back
Completion Date
5/1/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Recommended Distance (km): 10
Distance (km)
0
Calories Burnt
0
Time Taken
10
Submit

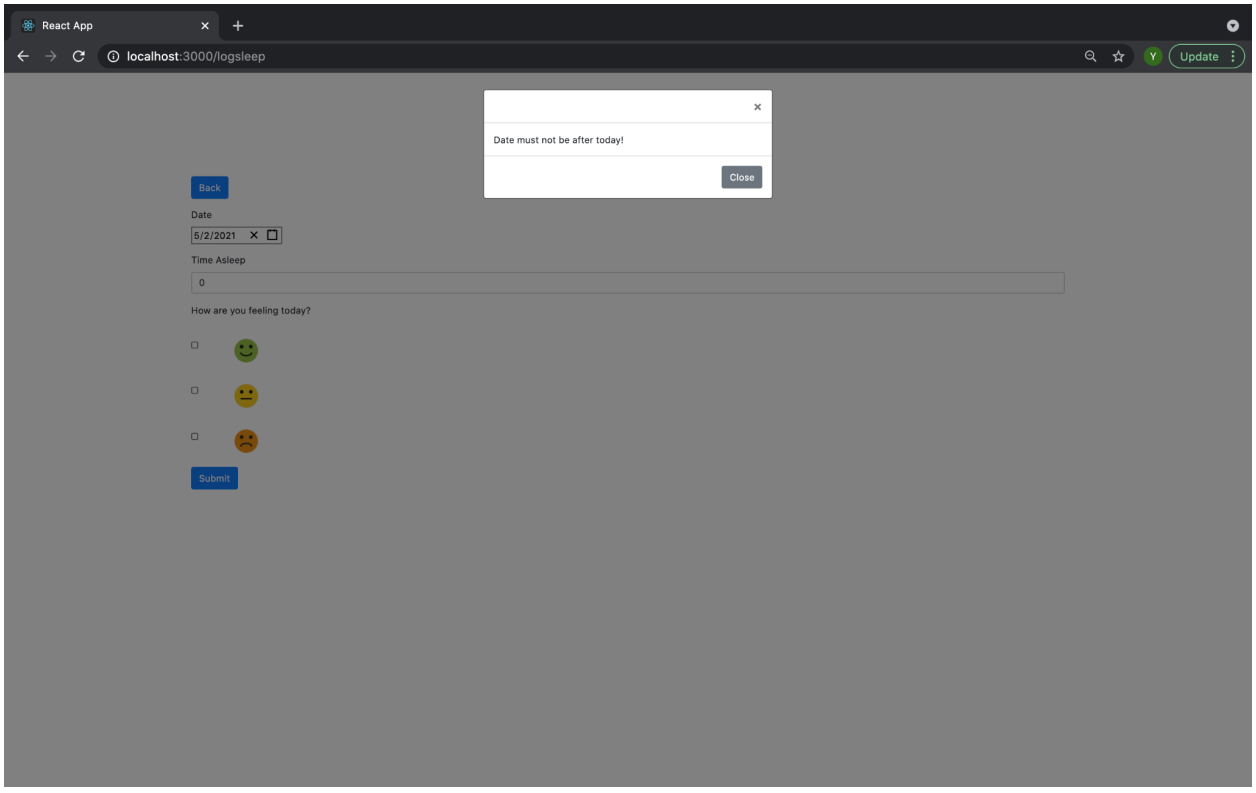

React App
localhost:3000/userecommendation/id=0
Update
Workout recommendation has been used successfully!
Close
Back
Completion Date
5/1/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Clear Drawing
Recommended Distance (km): 10
Distance (km)
0.64
Calories Burnt
0
Time Taken
10
Submit

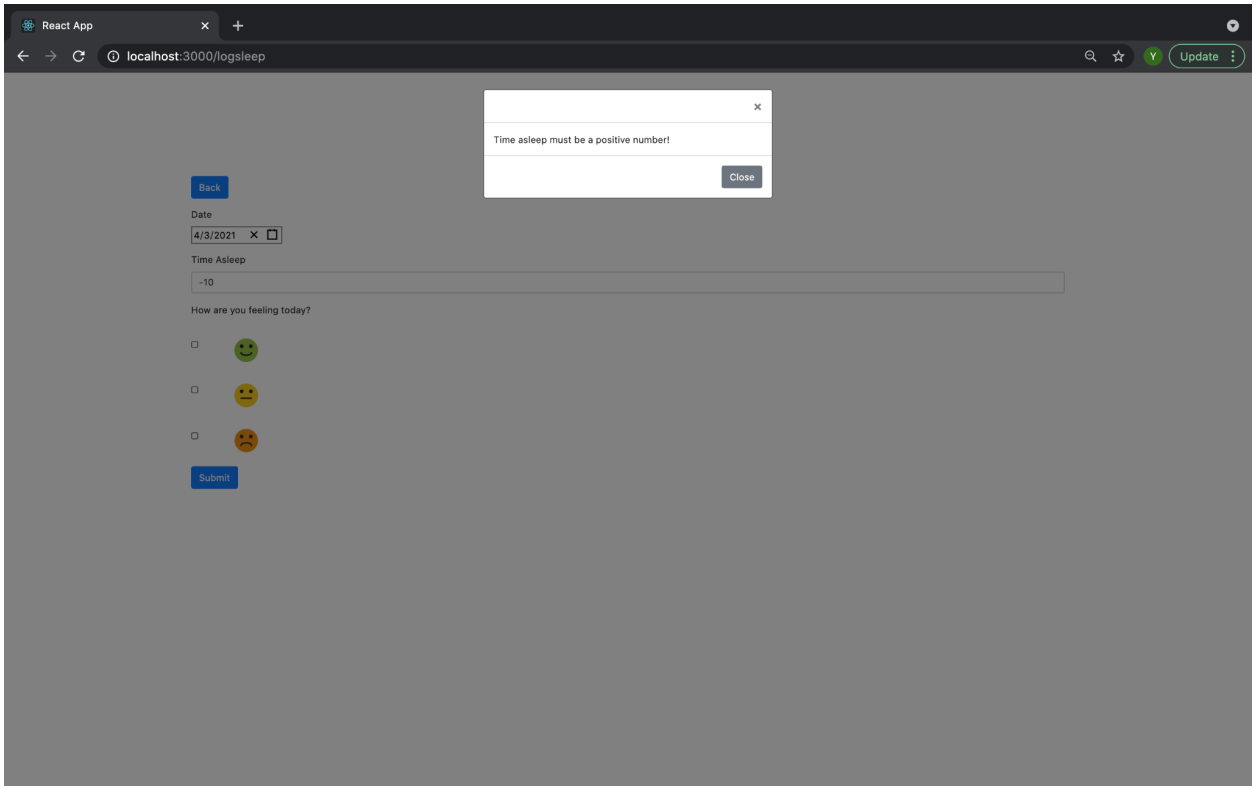

React App
localhost:3000/oldworkout/id=1
Update
Old Workout Details
Back
Date
3/29/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Distance (km)
0.72
Calories Burnt
328
Time Spent
20
Pace
2.16
Reuse

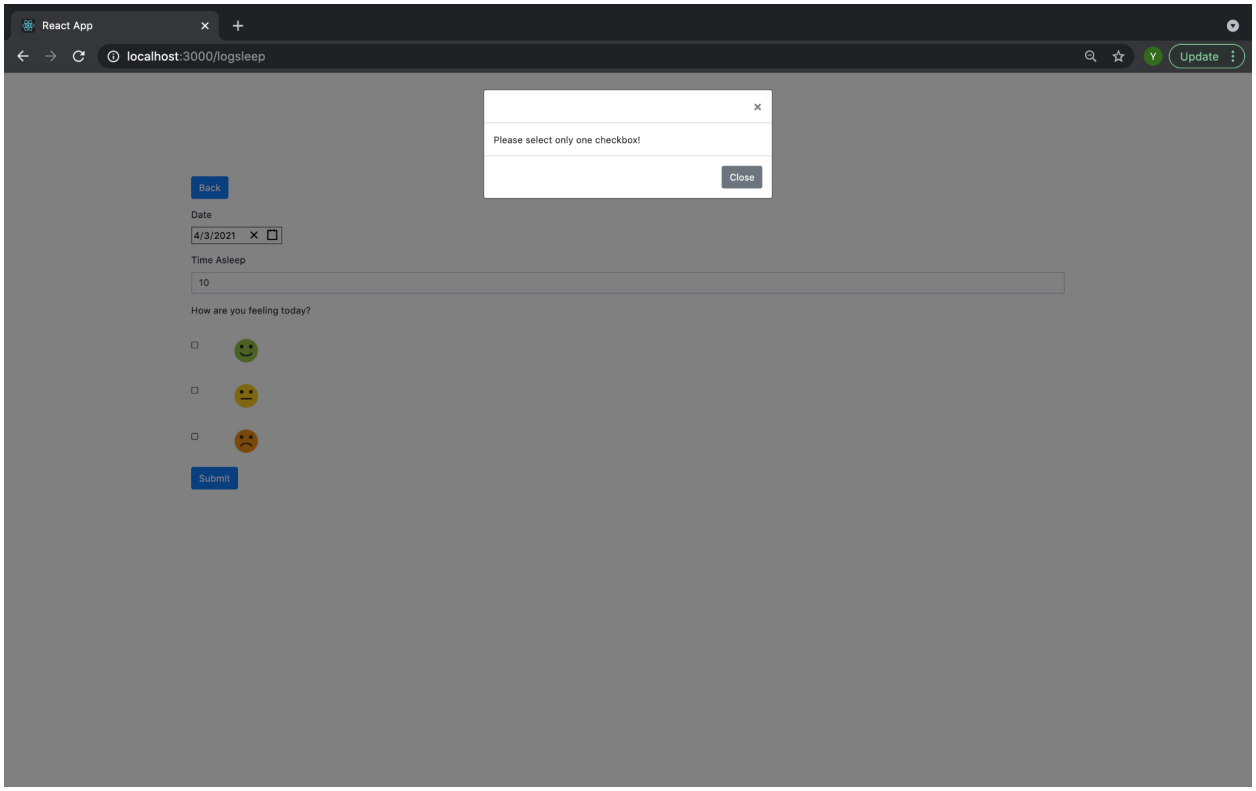

React App
localhost:3000/oldworkout/id=1
Update
This workout has been added to upcoming workouts!
Close
Back
Date
3/29/2021
Workout Type
Running
Draw Your Path
© OpenStreetMap contributors.
Distance (km)
0.72
Calories Burnt
328
Time Spent
20
Pace
2.16
Reuse

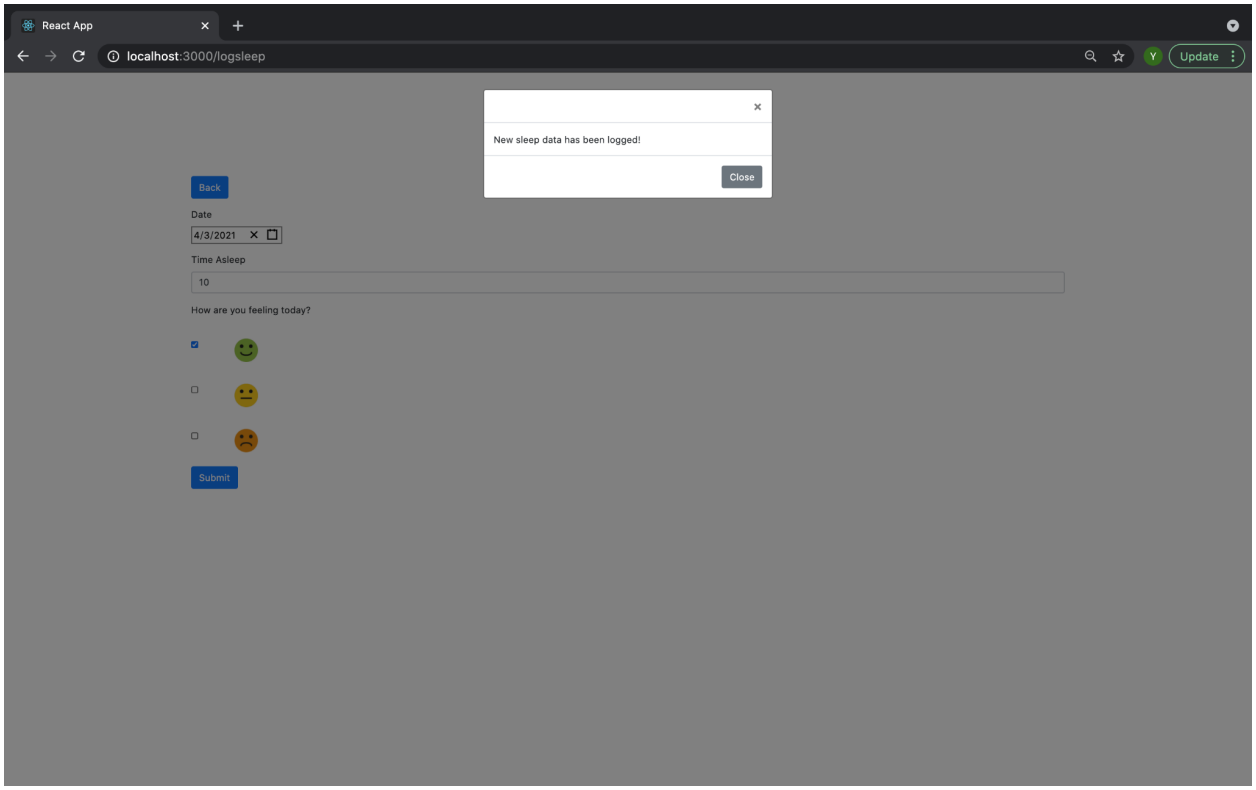

React App
localhost:3000/logsleep
Update
Log Sleep
Back
Date
--/--/----
Time Asleep
0
How are you feeling today?
Submit

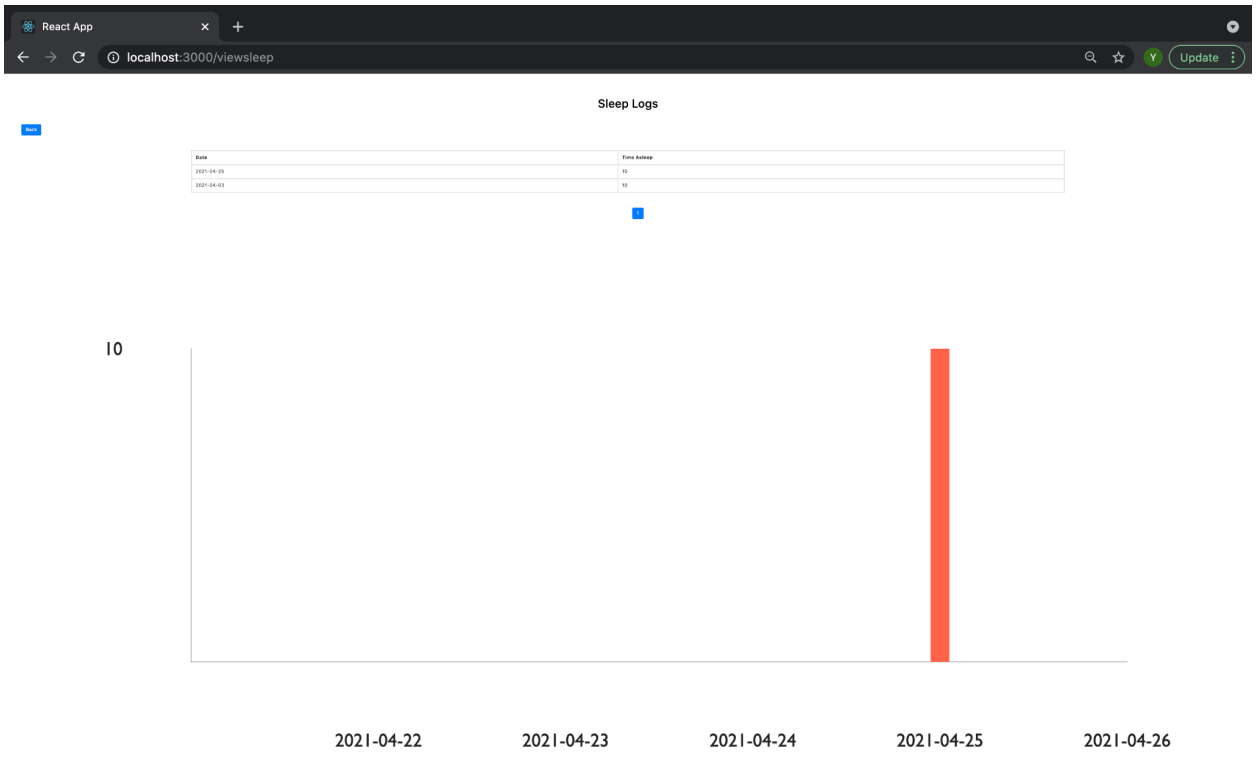

React App
localhost:3000/logsleep
Update
Please provide a date!
Close
Back
Date
--/--/----
Time Asleep
0
How are you feeling today?
Submit

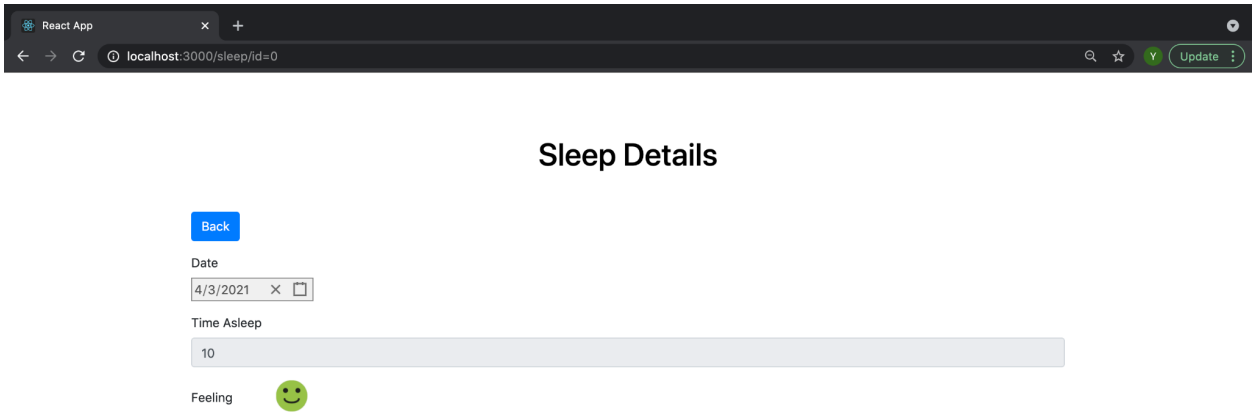

React App
localhost:3000/logsleep
Update
Date must not be after today!
Close
Back
Date
5/2/2021
Time Asleep
0
How are you feeling today?
Submit

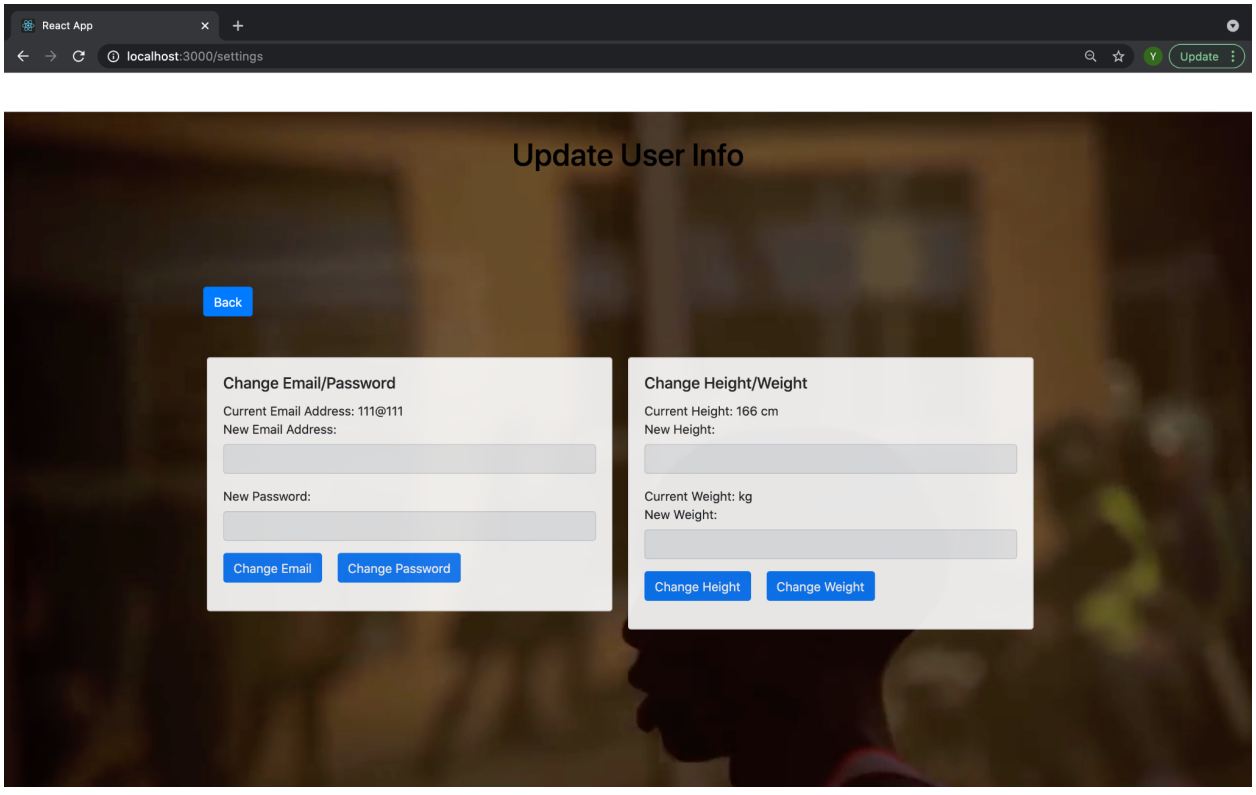

React App
localhost:3000/logsleep
Update
Time asleep must be a positive number!
Close
Back
Date
4/3/2021
Time Asleep
-10
How are you feeling today?
Submit

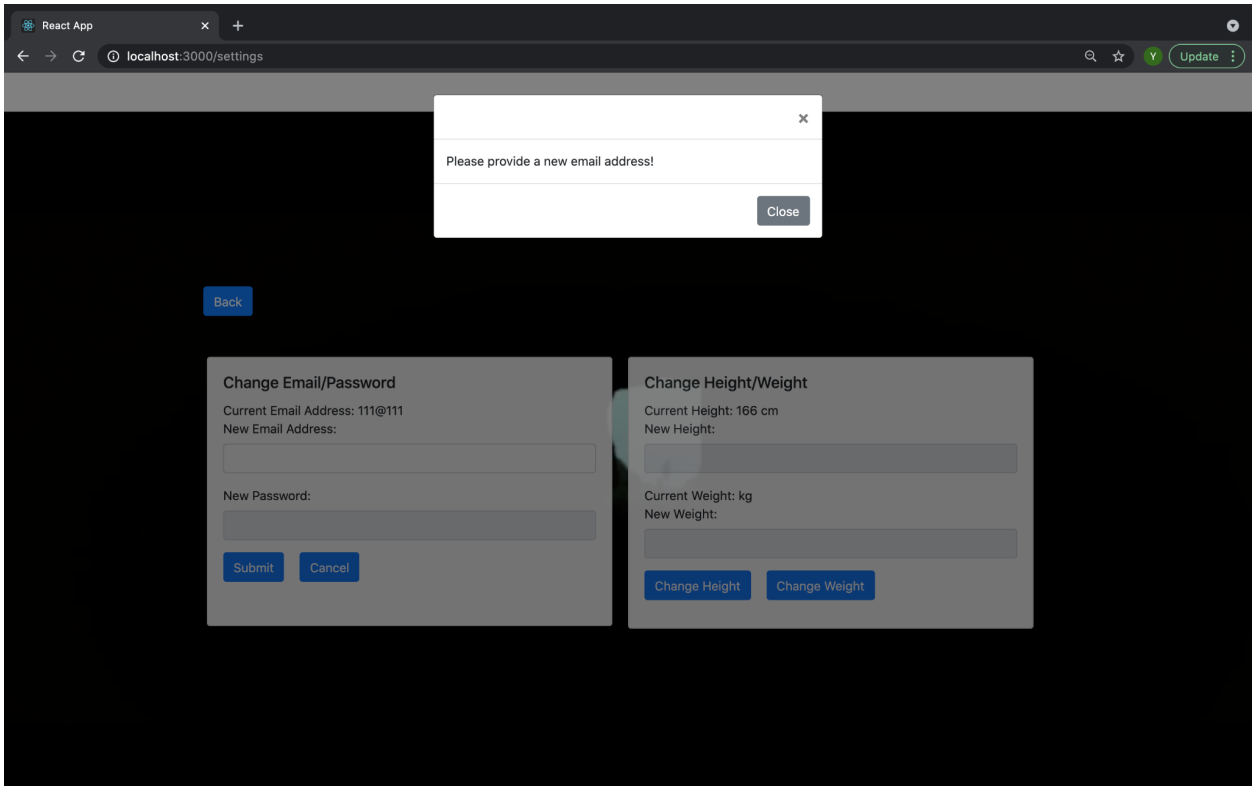

React App
localhost:3000/logsleep
Update
Please select only one checkbox!
Close
Back
Date
4/3/2021
Time Asleep
10
How are you feeling today?
Submit

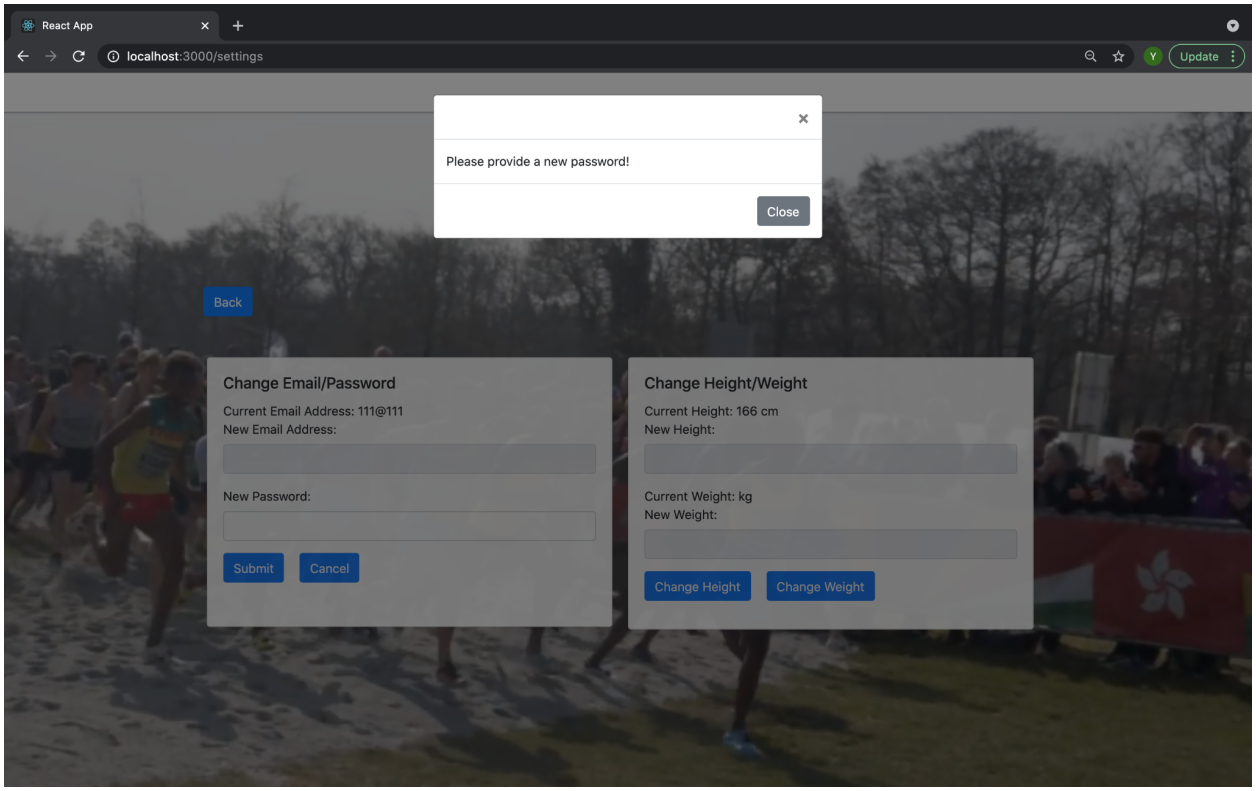

React App
localhost:3000/logsleep
Update
New sleep data has been logged!
Close
Back
Date
4/3/2021
Time Asleep
10
How are you feeling today?
Submit

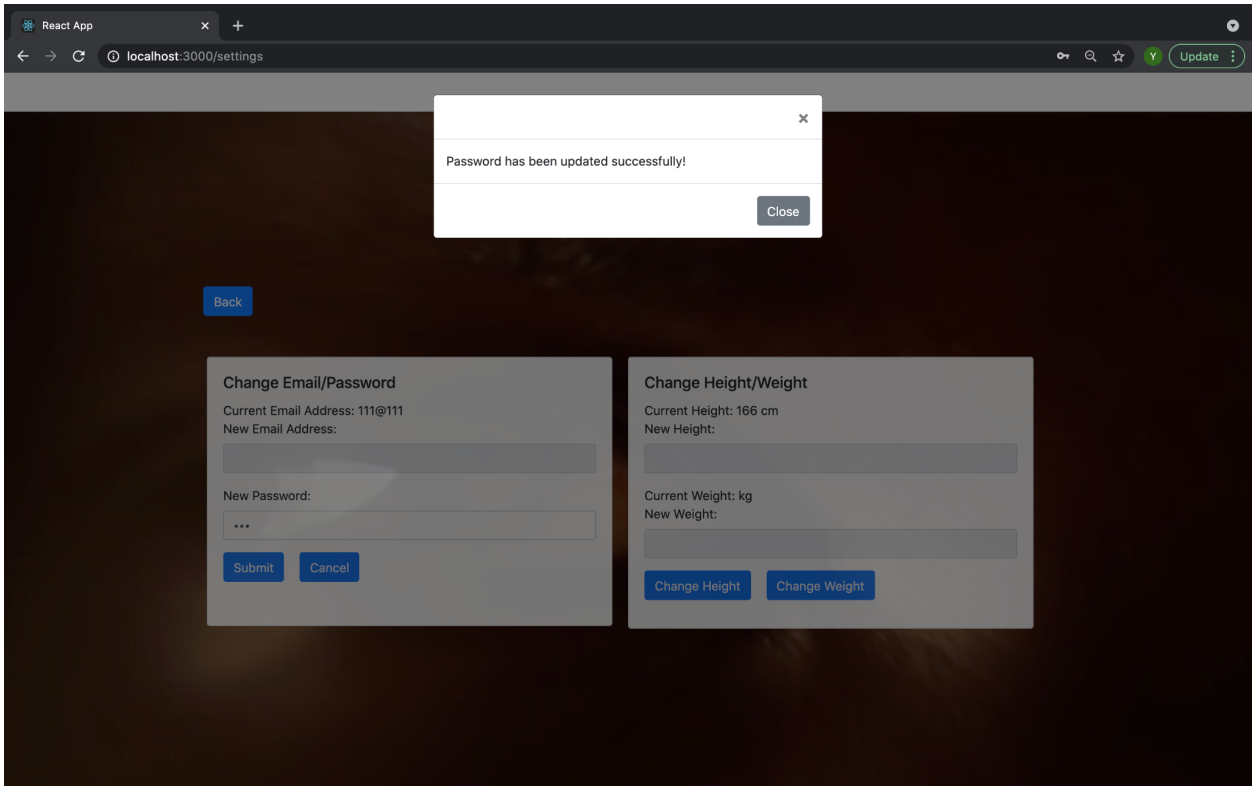

React App
localhost:3000/viewsleep
Update
Sleep Logs
Back
Date
Time Asleep
2021-04-25
10
2021-04-03
10
1
10
2021-04-22
2021-04-23
2021-04-24
2021-04-25
2021-04-26

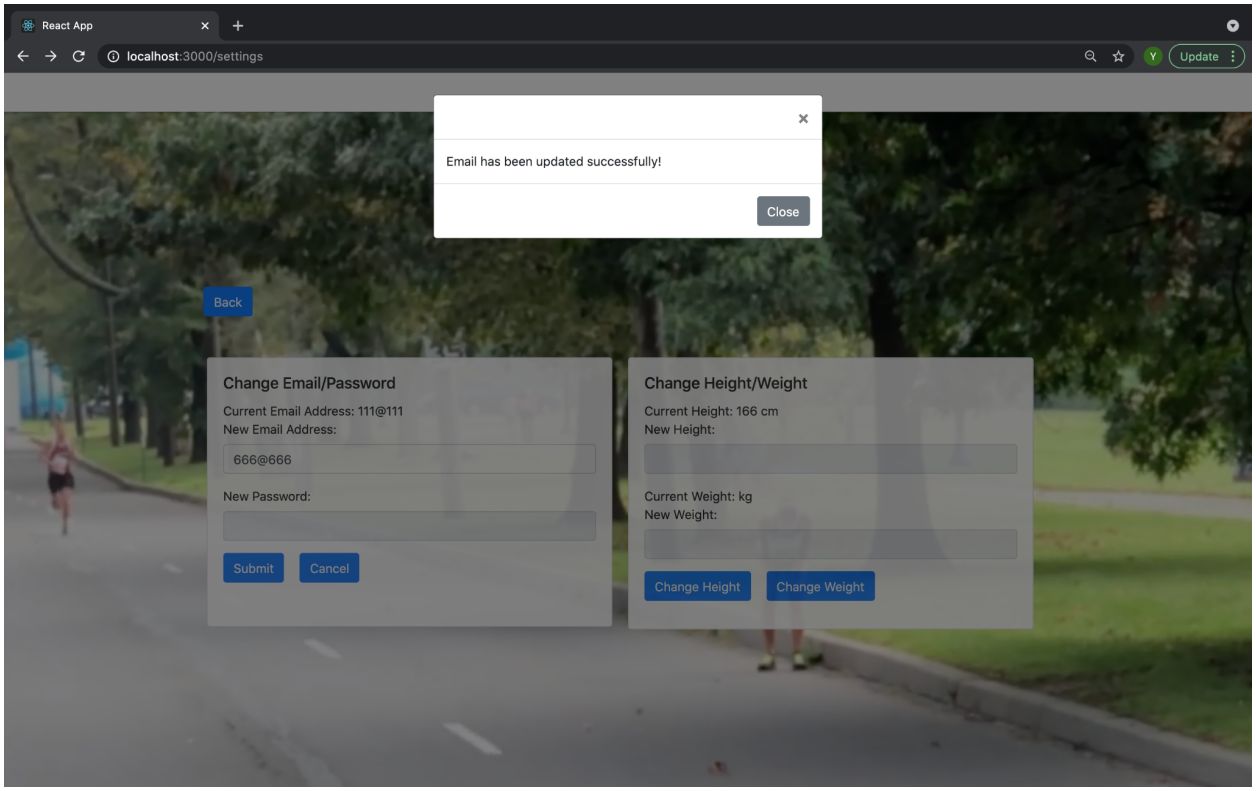

React App
localhost:3000/sleep/id=0
Update
Sleep Details
Back
Date
4/3/2021
Time Asleep
10
Feeling

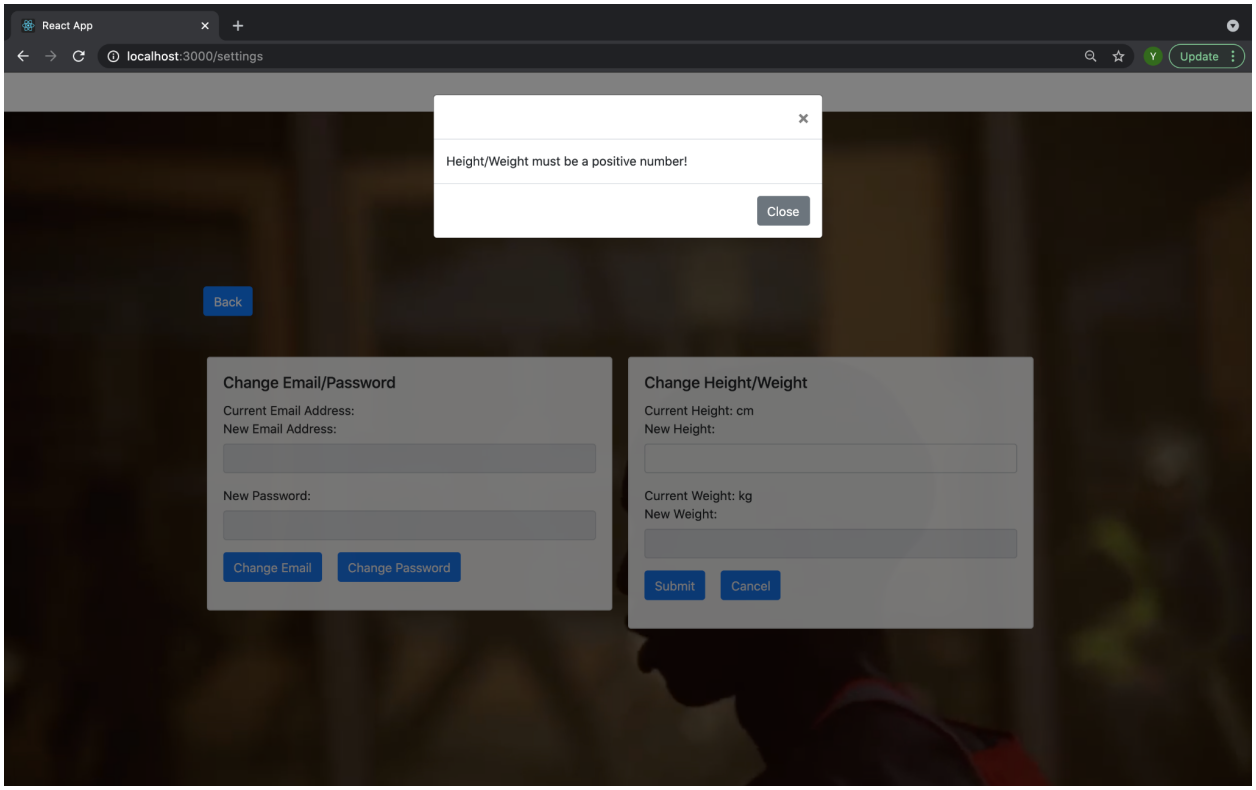

React App
localhost:3000/settings
Update
Update User Info
Back
Change Email/Password
Current Email Address: 111@111
New Email Address:
New Password:
Change Email
Change Password
Change Height/Weight
Current Height: 166 cm
New Height:
Current Weight: kg
New Weight:
Change Height
Change Weight

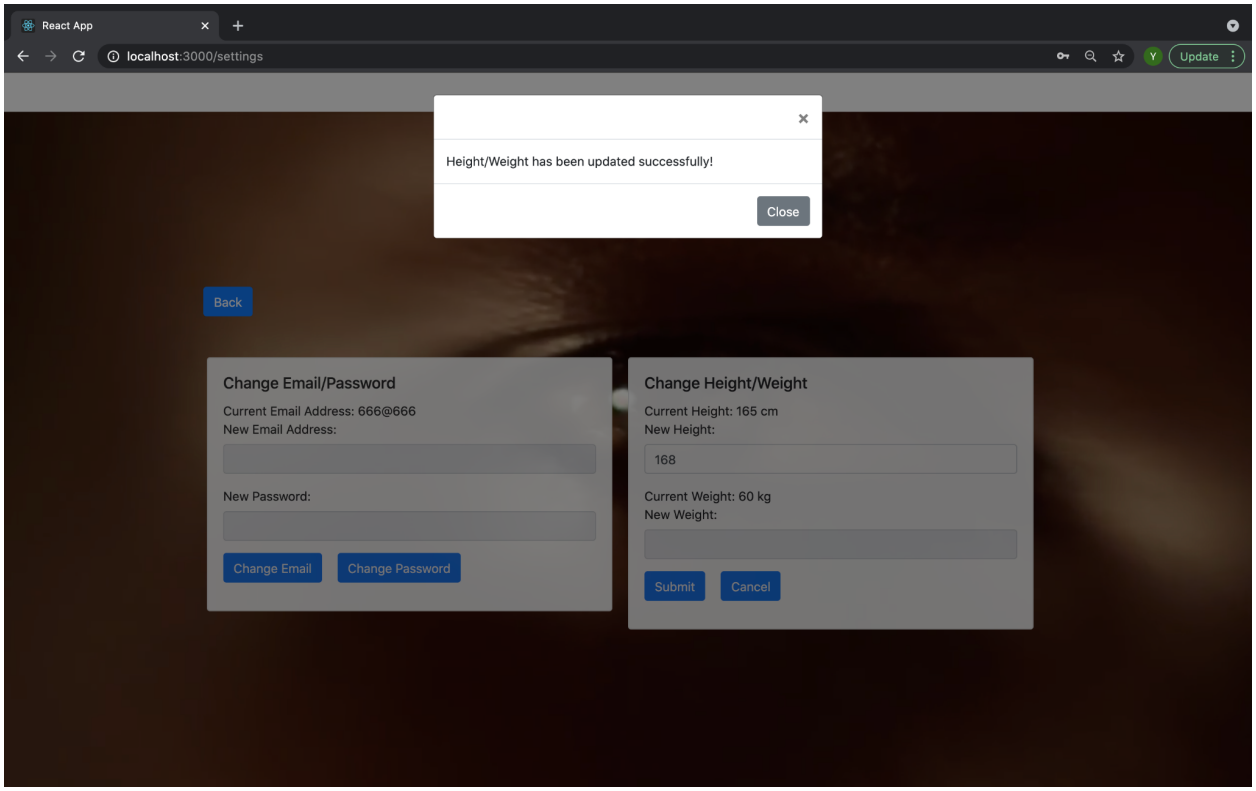

React App
localhost:3000/settings
Update
Please provide a new email address!
Close
Back
Change Email/Password
Current Email Address: 111@111
New Email Address:
New Password:
Submit
Cancel
Change Height/Weight
Current Height: 166 cm
New Height:
Current Weight: kg
New Weight:
Change Height
Change Weight


React App
localhost:3000/settings
Update
Please provide a new password!
Close
Back
Change Email/Password
Current Email Address: 111@111
New Email Address:
New Password:
Submit
Cancel
Change Height/Weight
Current Height: 166 cm
New Height:
Current Weight: kg
New Weight:
Change Height
Change Weight

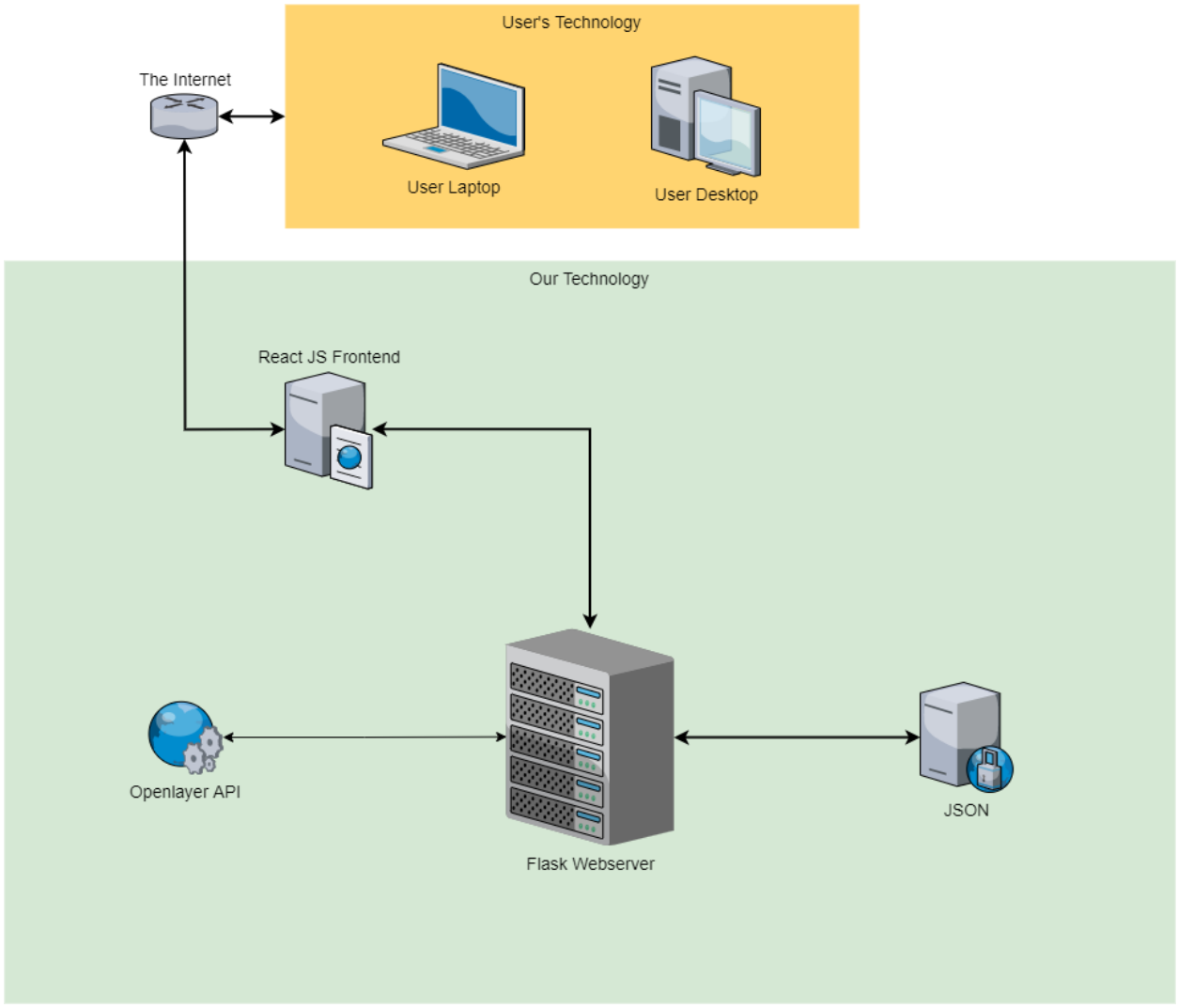

React App
localhost:3000/settings
Update
Password has been updated successfully!
Close
Back
Change Email/Password
Current Email Address: 111@111
New Email Address:
New Password:
Submit
Cancel
Change Height/Weight
Current Height: 166 cm
New Height:
Current Weight: kg
New Weight:
Change Height
Change Weight


React App
localhost:3000/settings
Update
Email has been updated successfully!
Close
Back
Change Email/Password
Current Email Address: 111@111
New Email Address:
666@666
New Password:
Submit
Cancel
Change Height/Weight
Current Height: 166 cm
New Height:
Current Weight: kg
New Weight:
Change Height
Change Weight

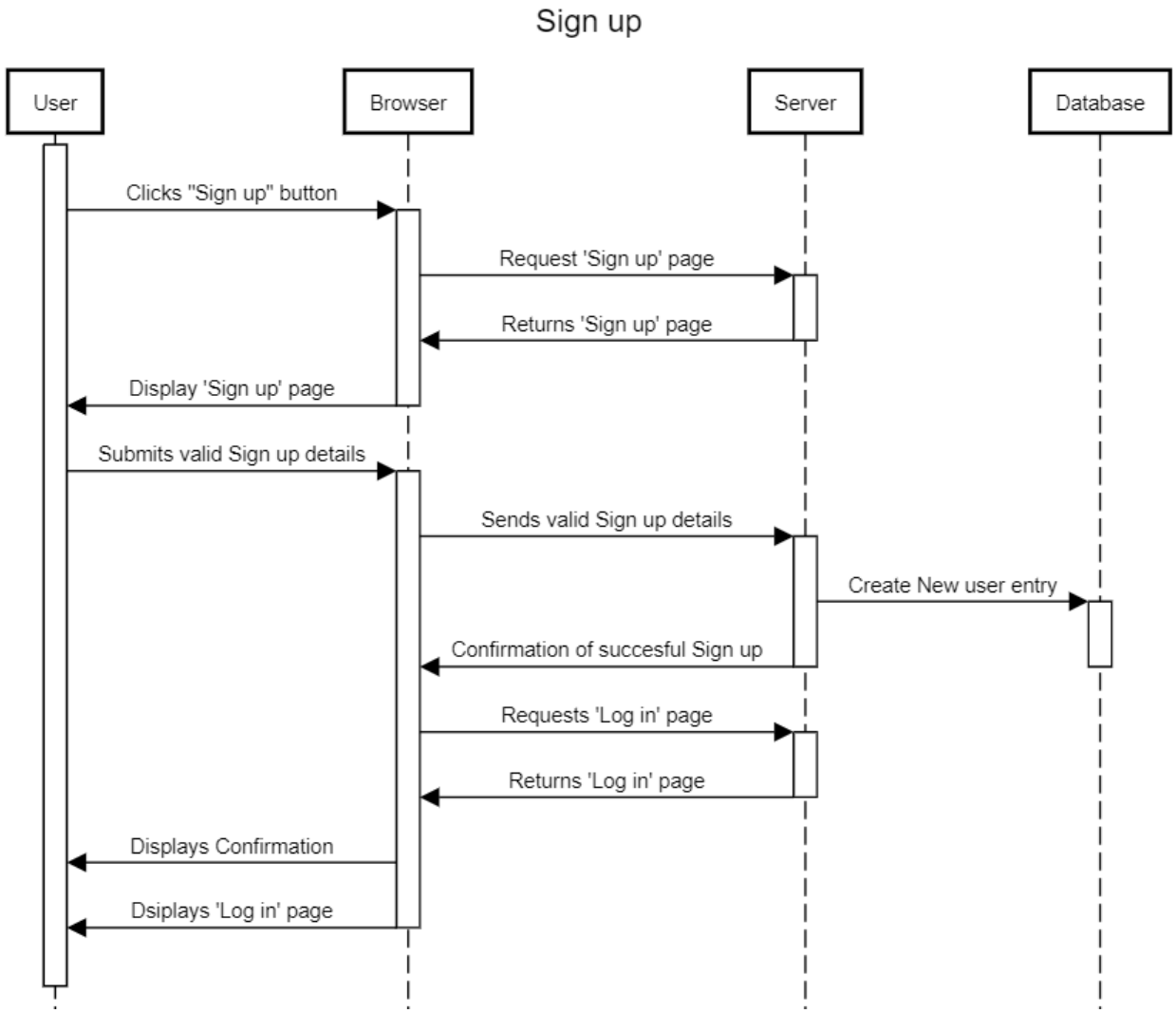

React App
localhost:3000/settings
Update
Height/Weight must be a positive number!
Close
Back
Change Email/Password
Current Email Address:
New Email Address:
New Password:
Change Email
Change Password
Change Height/Weight
Current Height: cm
New Height:
Current Weight: kg
New Weight:
Submit
Cancel


React App
localhost:3000/settings
Update
Height/Weight has been updated successfully!
Close
Back
Change Email/Password
Current Email Address: 666@666
New Email Address:
New Password:
Change Email
Change Password
Change Height/Weight
Current Height: 165 cm
New Height:
168
Current Weight: 60 kg
New Weight:
Submit
Cancel

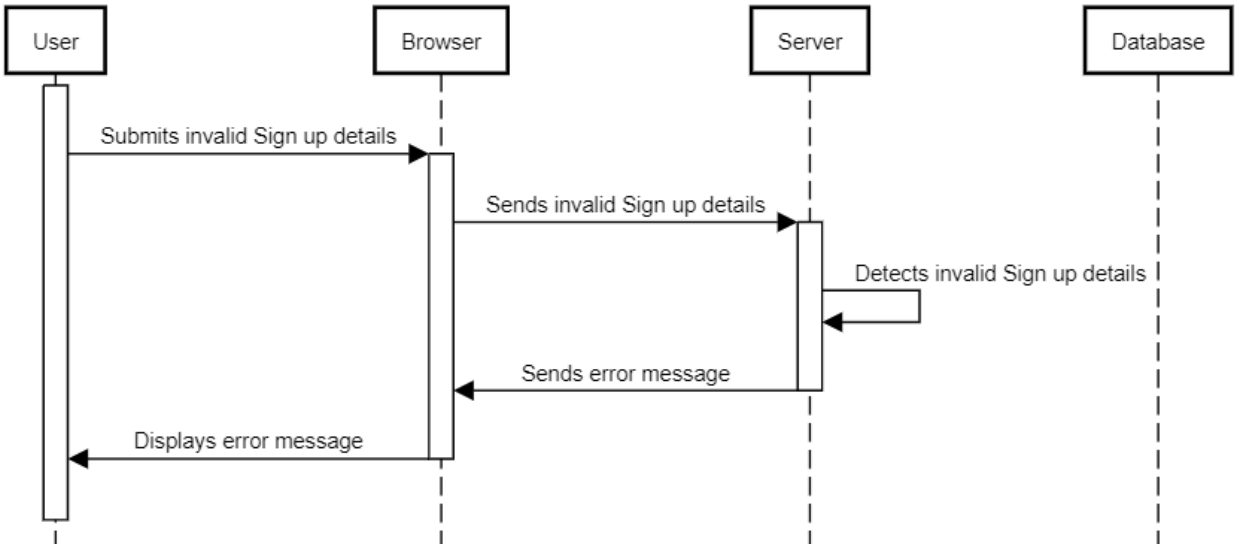
Final Software Architecture



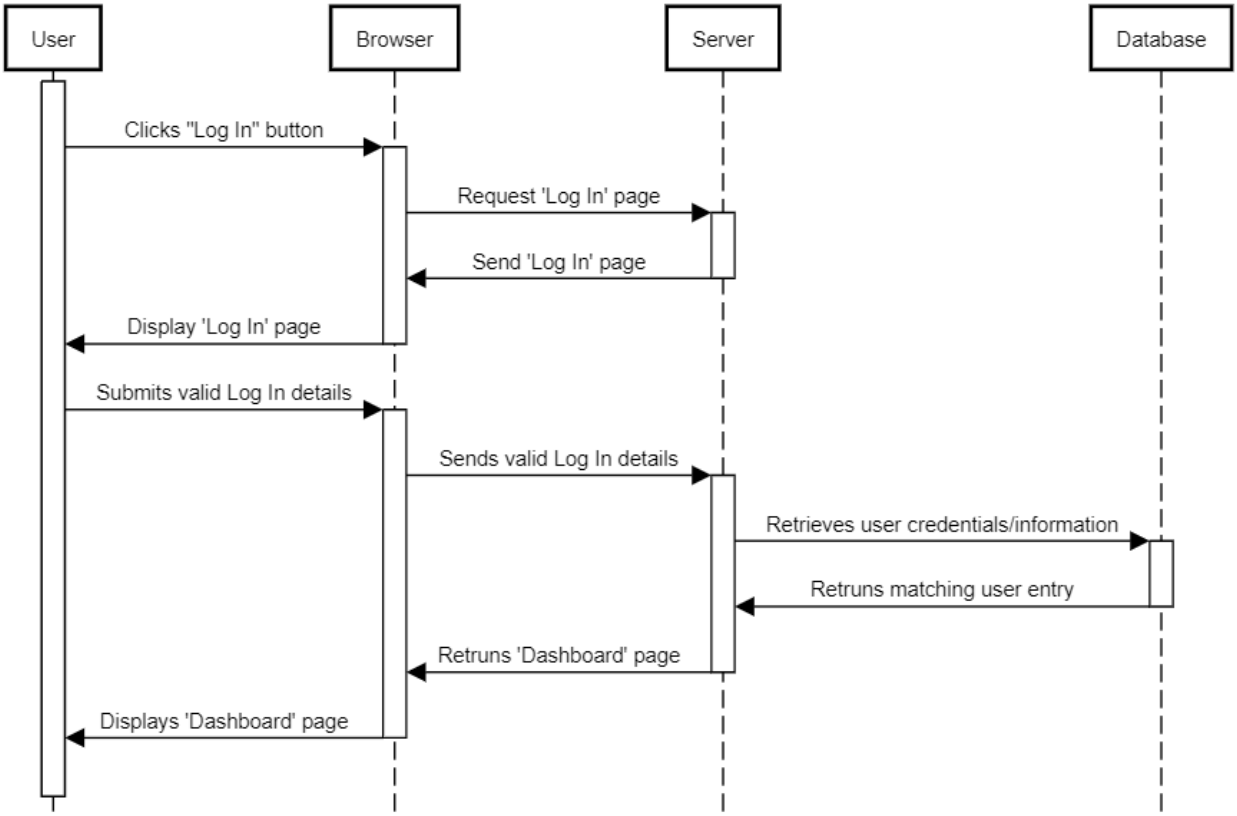
Sequence Diagrams



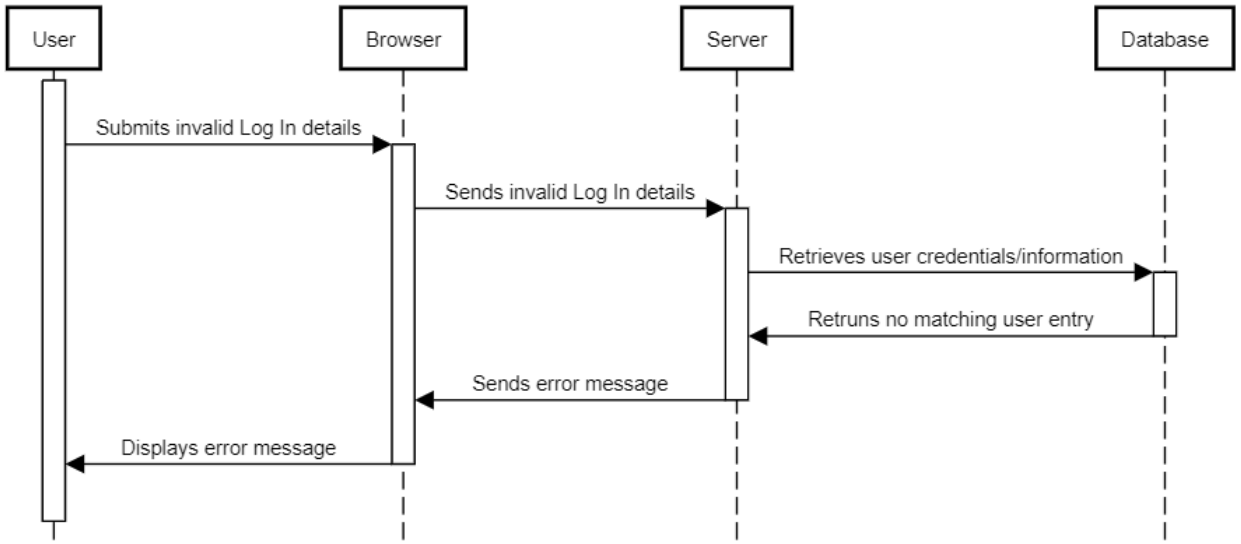
Invalid Sign up



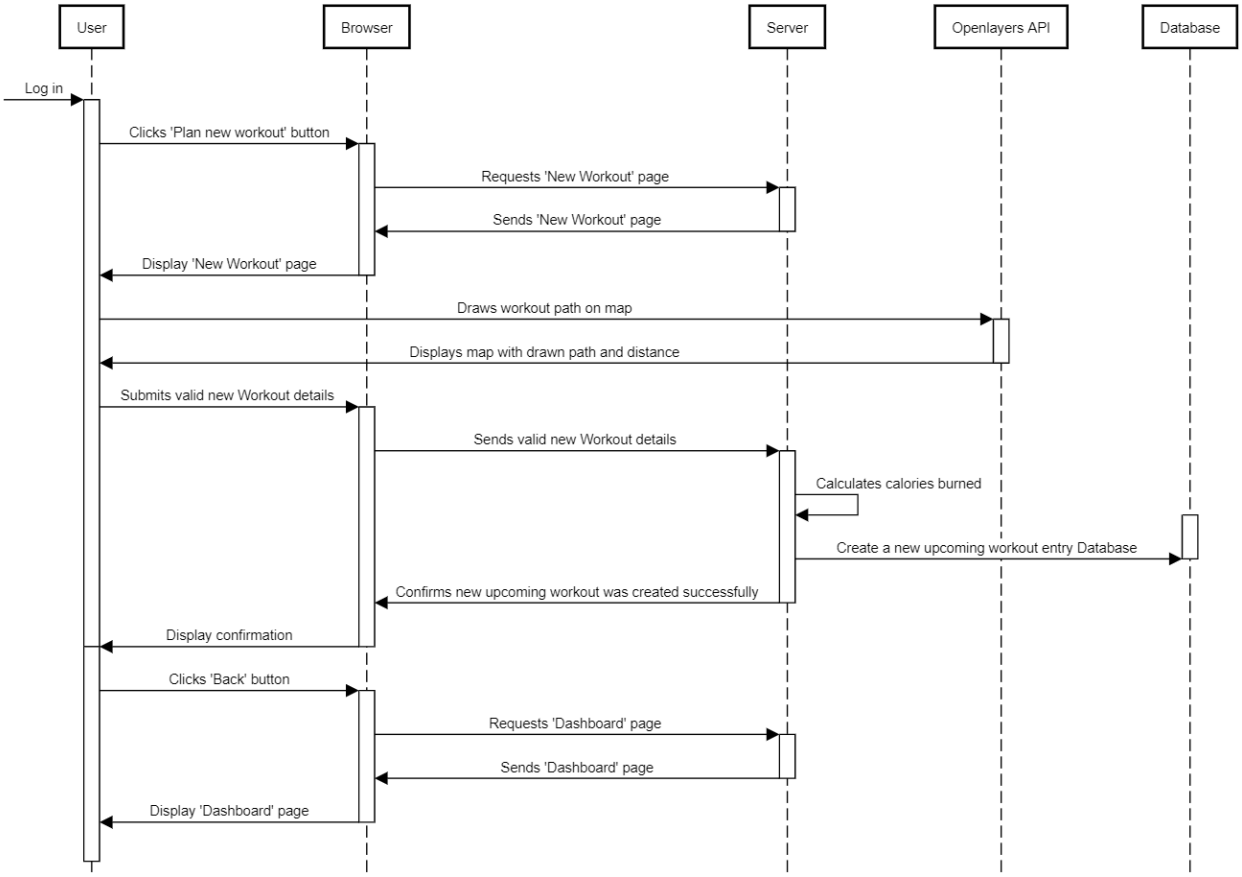
Log in

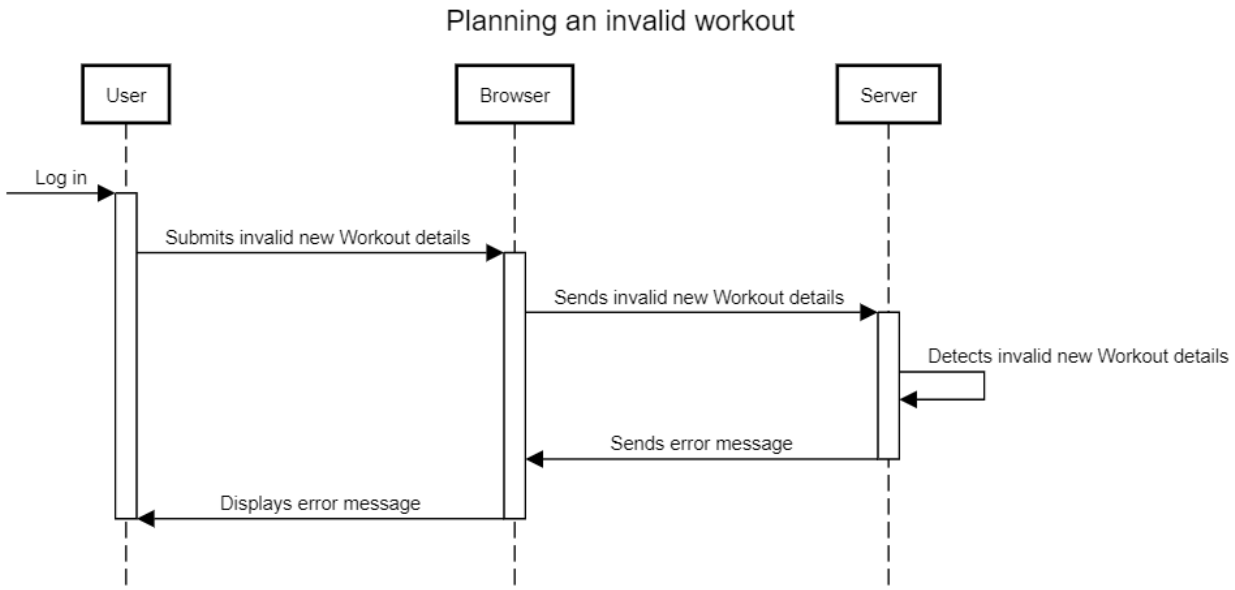


Invalid Log in

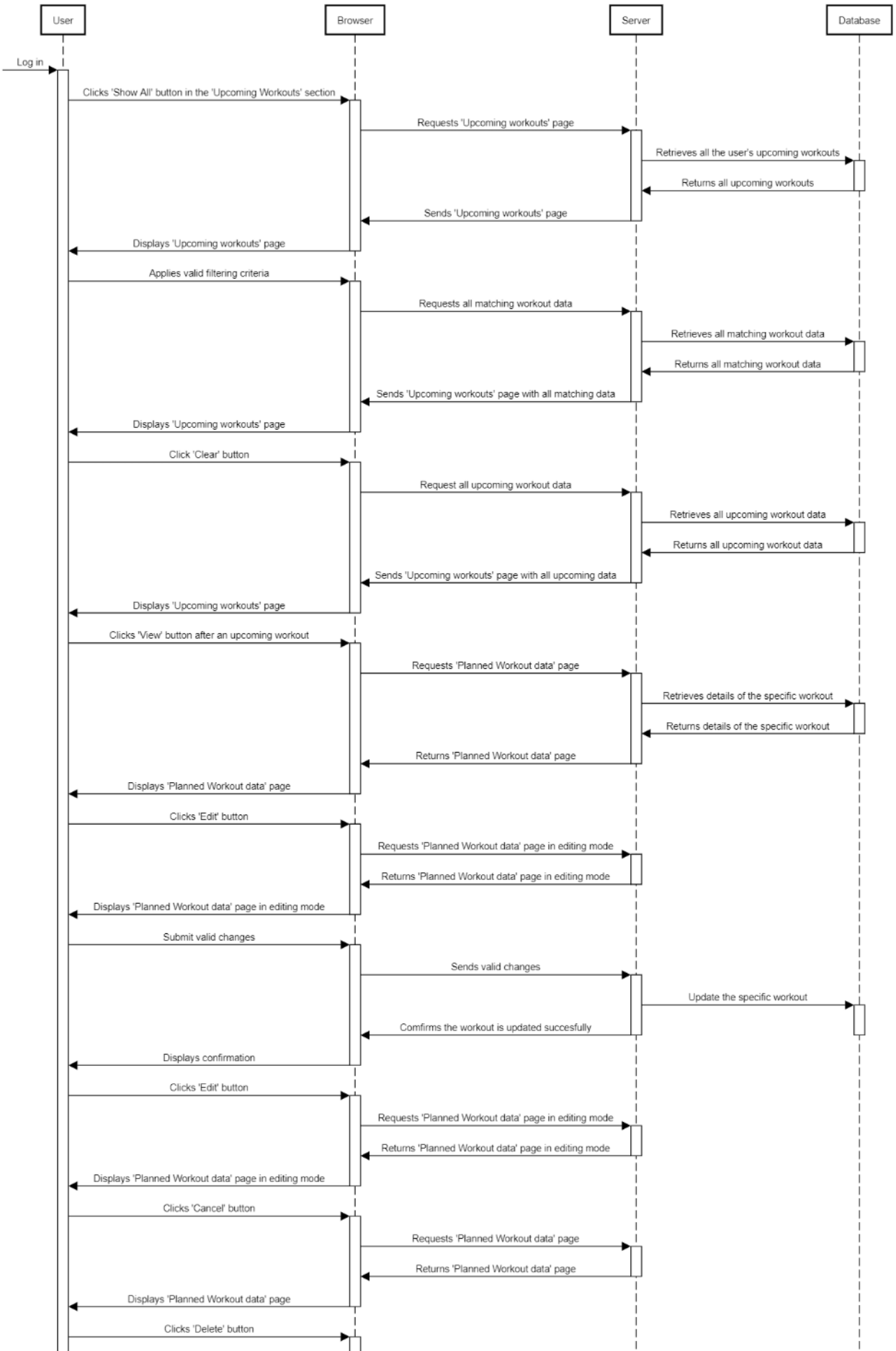


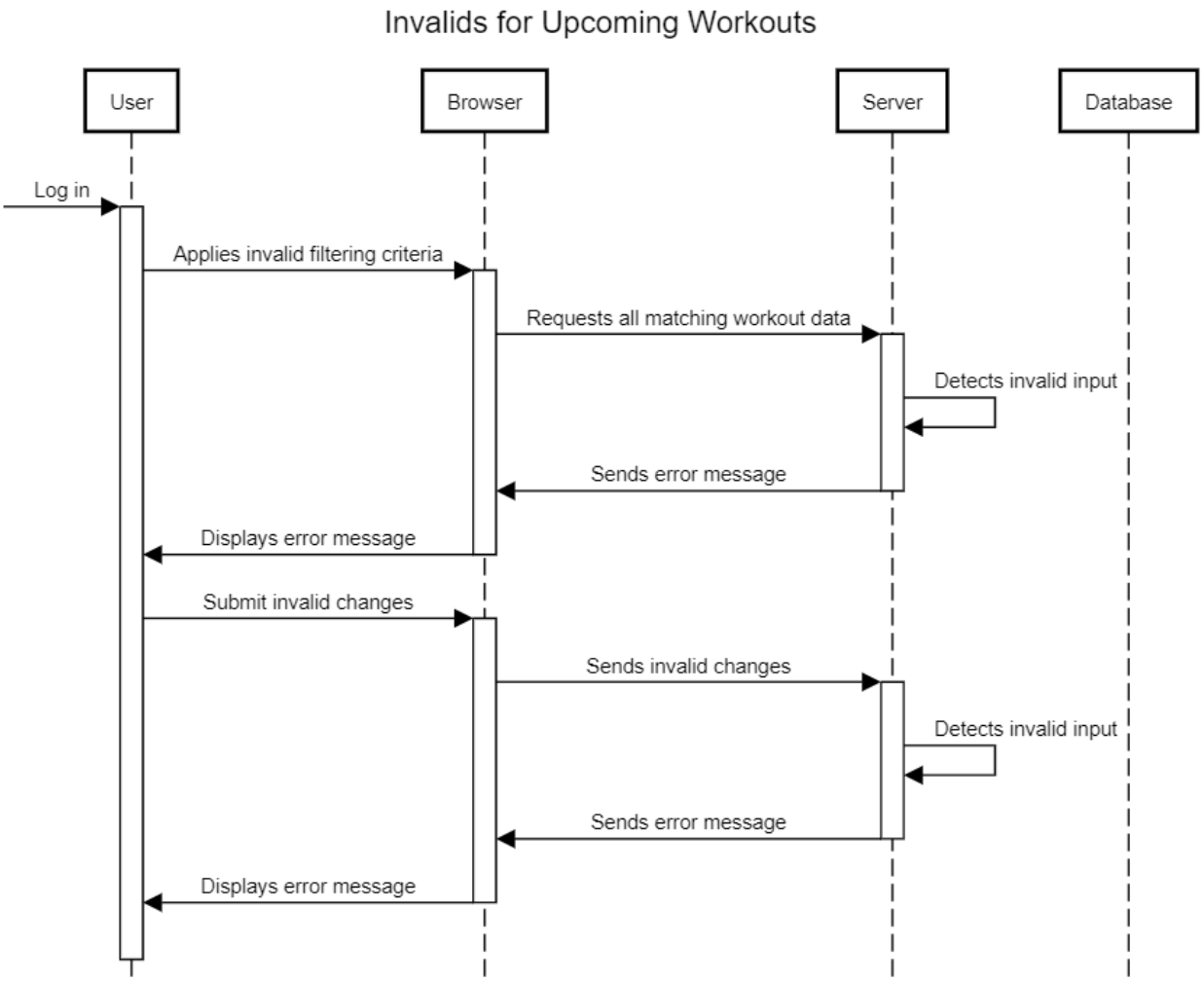
Planning a workout



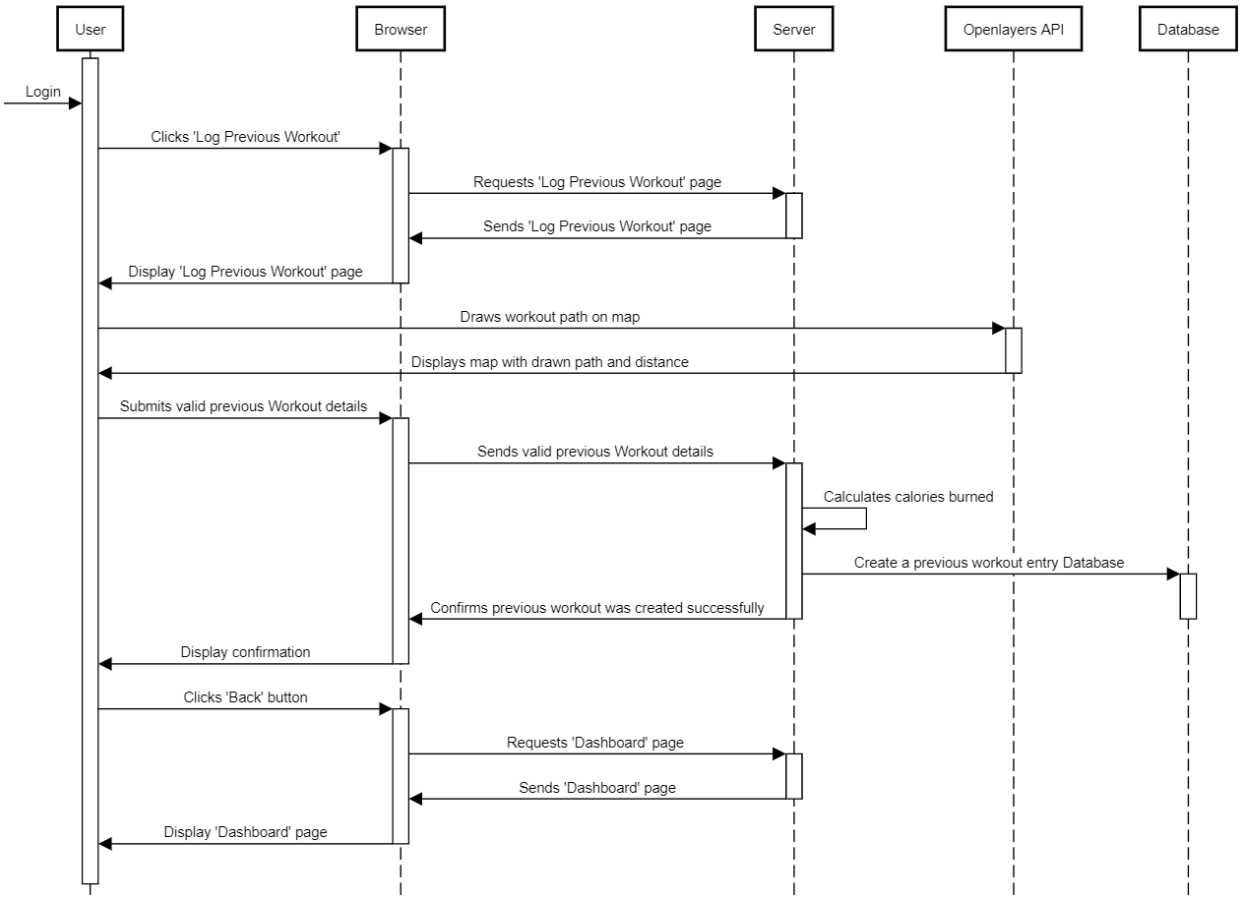

Planning an invalid workout
User
Browser
Server
Log in
Submits invalid new Workout details
Sends invalid new Workout details
Detects invalid new Workout details
Sends error message
Displays error message

Upcoming Workouts

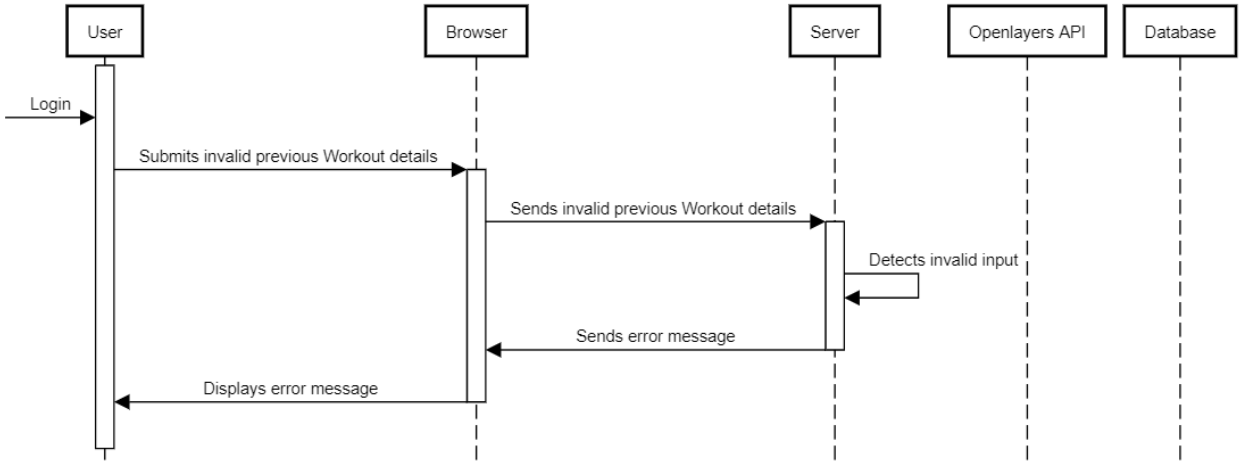


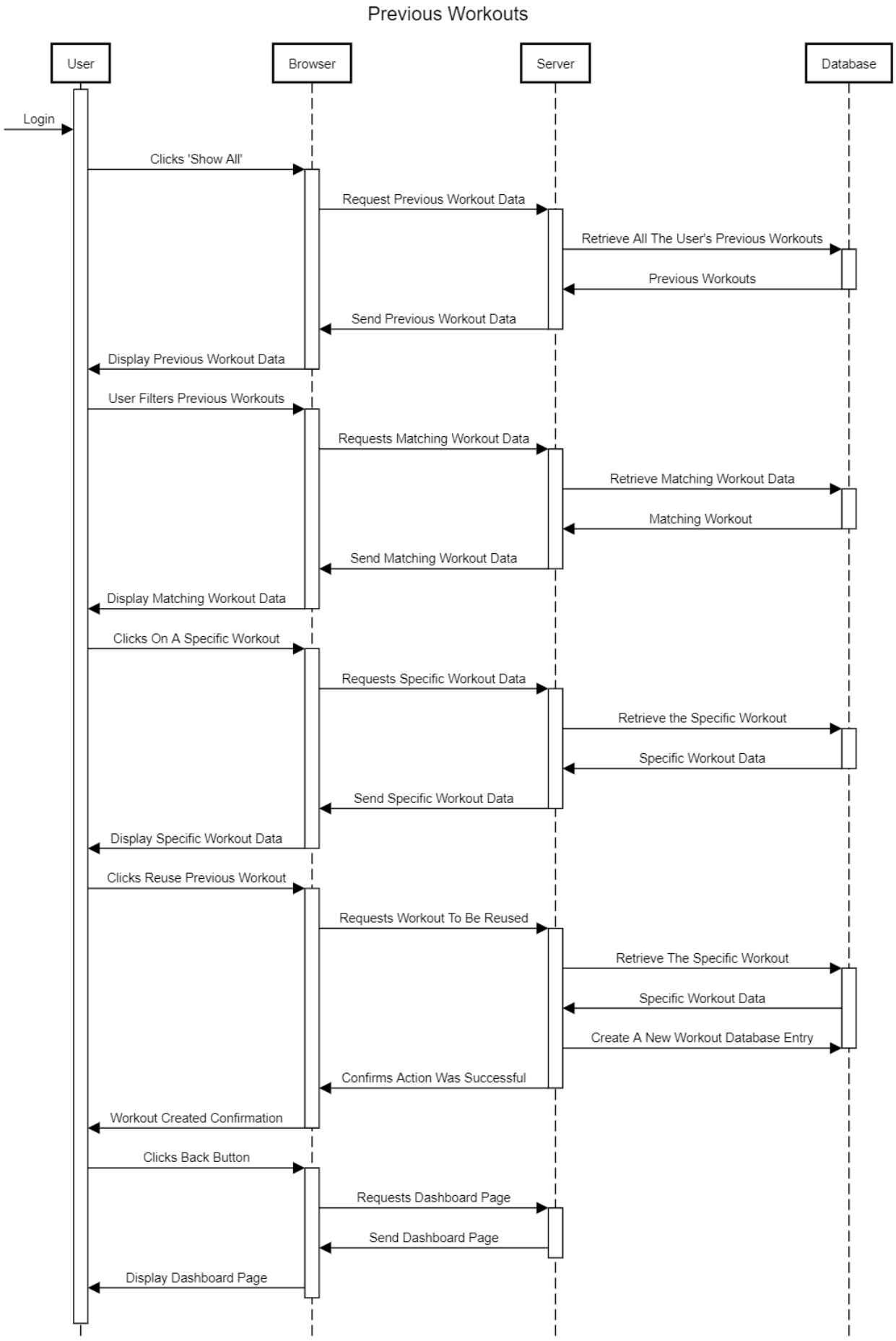

Invalids for Upcoming Workouts
User
Browser
Server
Database
Log in
Applies invalid filtering criteria
Requests all matching workout data
Detects invalid input
Sends error message
Displays error message
Submit invalid changes
Sends invalid changes
Detects invalid input
Sends error message
Displays error message

Log Workout

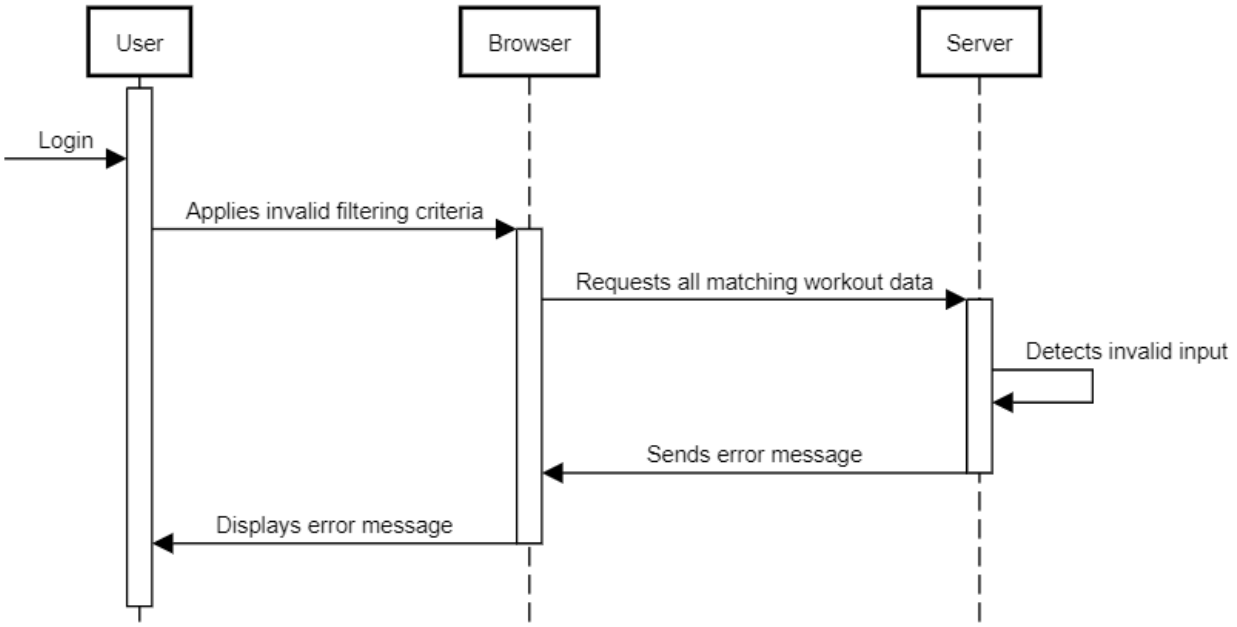


Log Invalid Workout

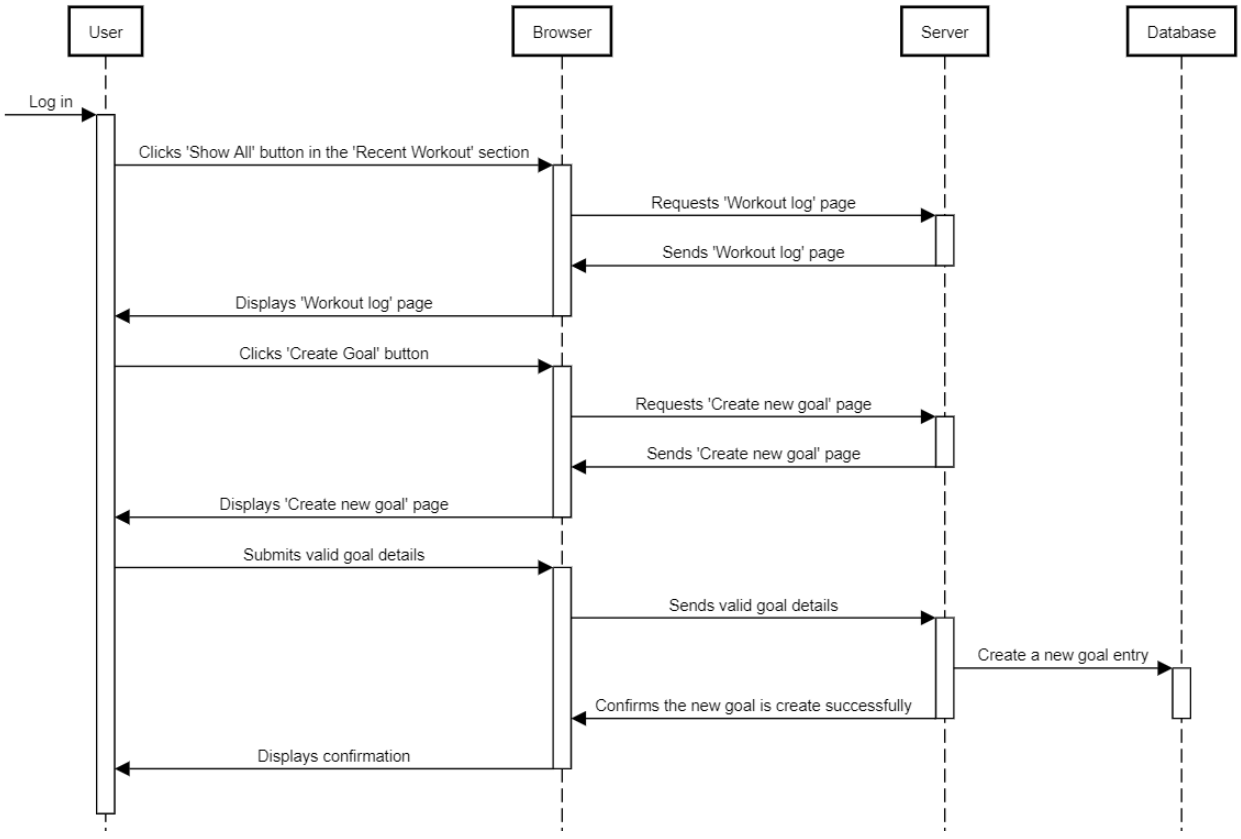


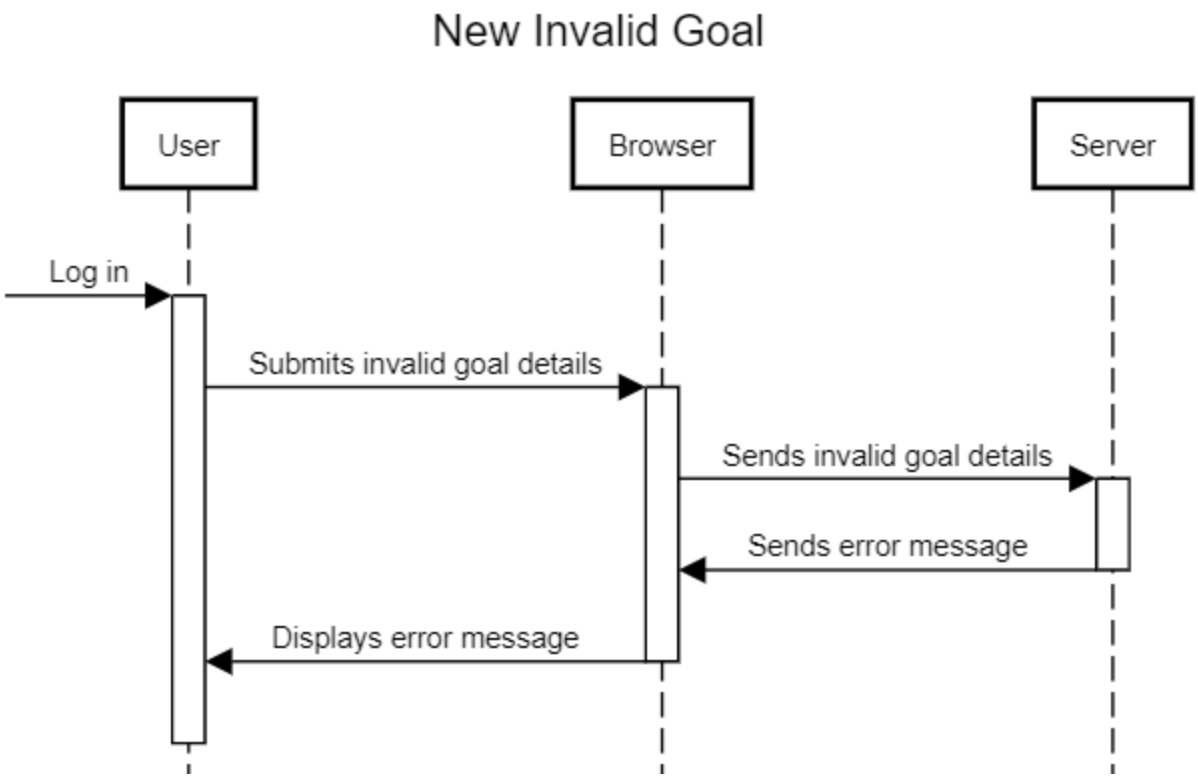

Previous Workouts
User
Browser
Server
Database
Login
Clicks 'Show All'
Request Previous Workout Data
Retrieve All The User's Previous Workouts
Previous Workouts
Send Previous Workout Data
Display Previous Workout Data
User Filters Previous Workouts
Requests Matching Workout Data
Retrieve Matching Workout Data
Matching Workout
Send Matching Workout Data
Display Matching Workout Data
Clicks On A Specific Workout
Requests Specific Workout Data
Retrieve the Specific Workout
Specific Workout Data
Send Specific Workout Data
Display Specific Workout Data
Clicks Reuse Previous Workout
Requests Workout To Be Reused
Retrieve The Specific Workout
Specific Workout Data
Create A New Workout Database Entry
Confirms Action Was Successful
Workout Created Confirmation
Clicks Back Button
Requests Dashboard Page
Send Dashboard Page
Display Dashboard Page

Invalids for Previous Workouts

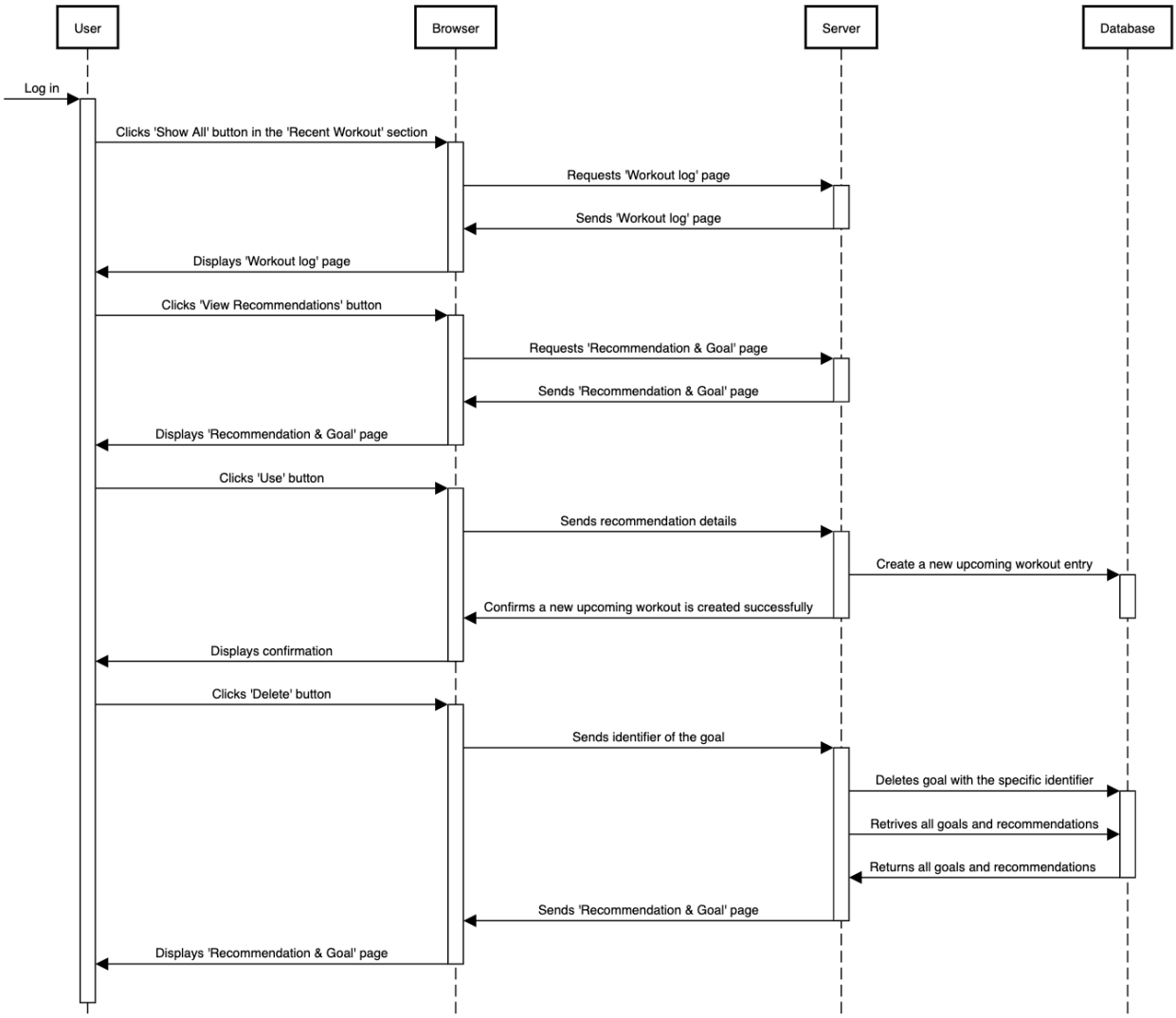


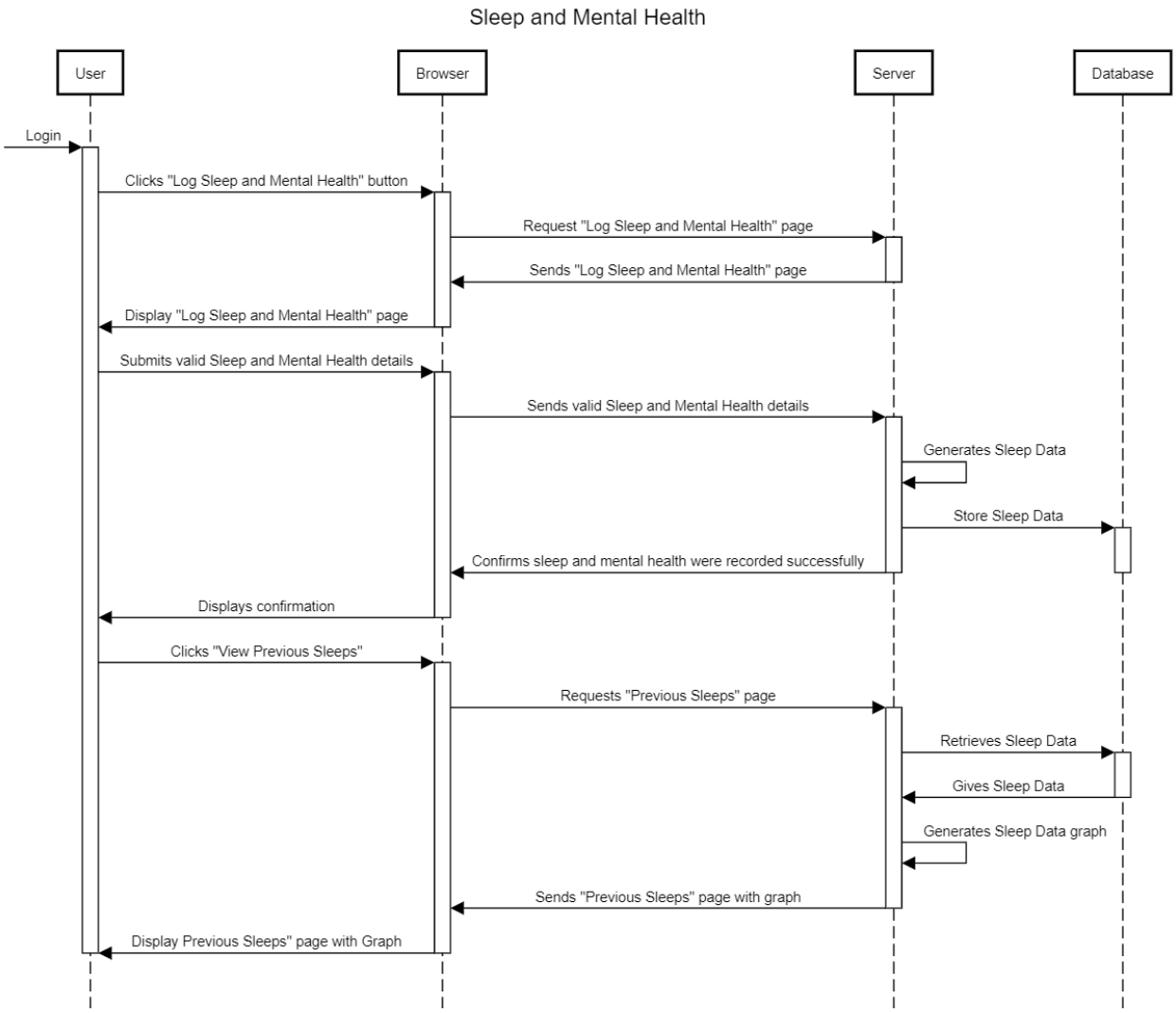
New Goal

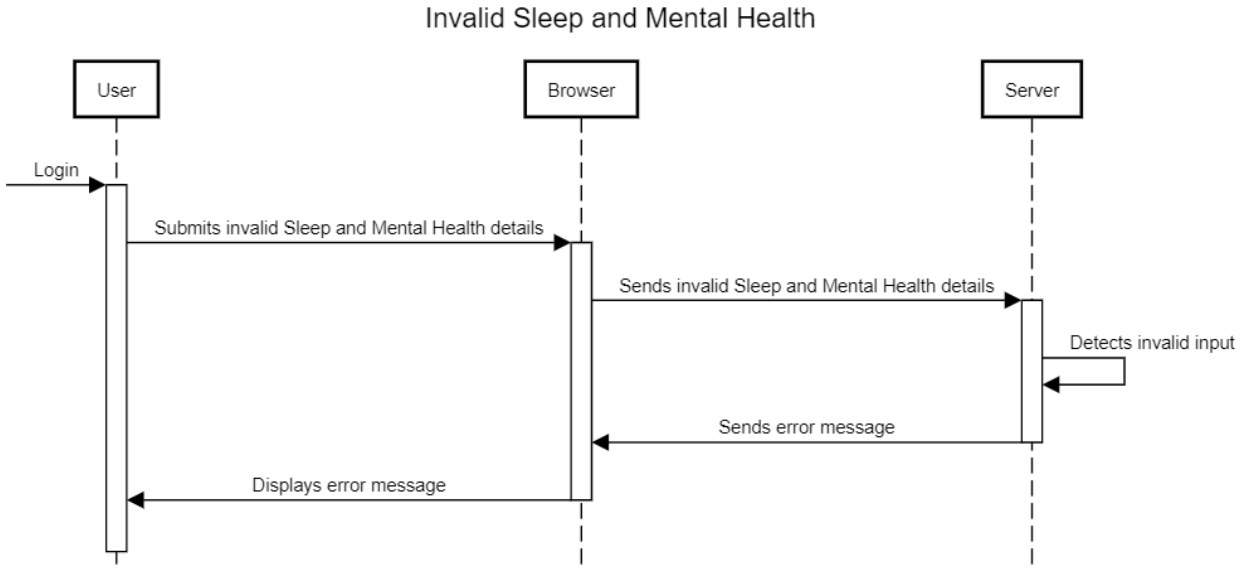



New Invalid Goal
User
Browser
Server
Log in
Submits invalid goal details
Sends invalid goal details
Sends error message
Displays error message

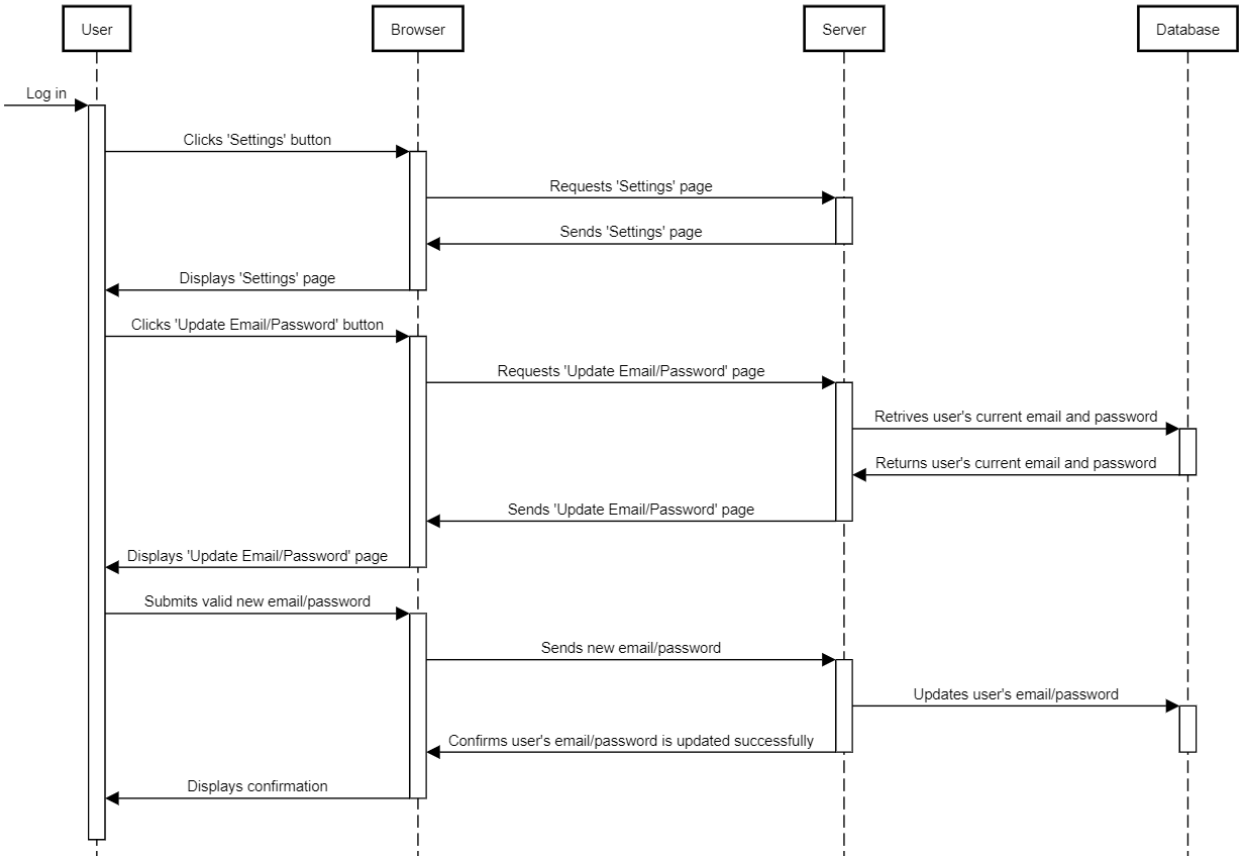
Goal & Recommendation



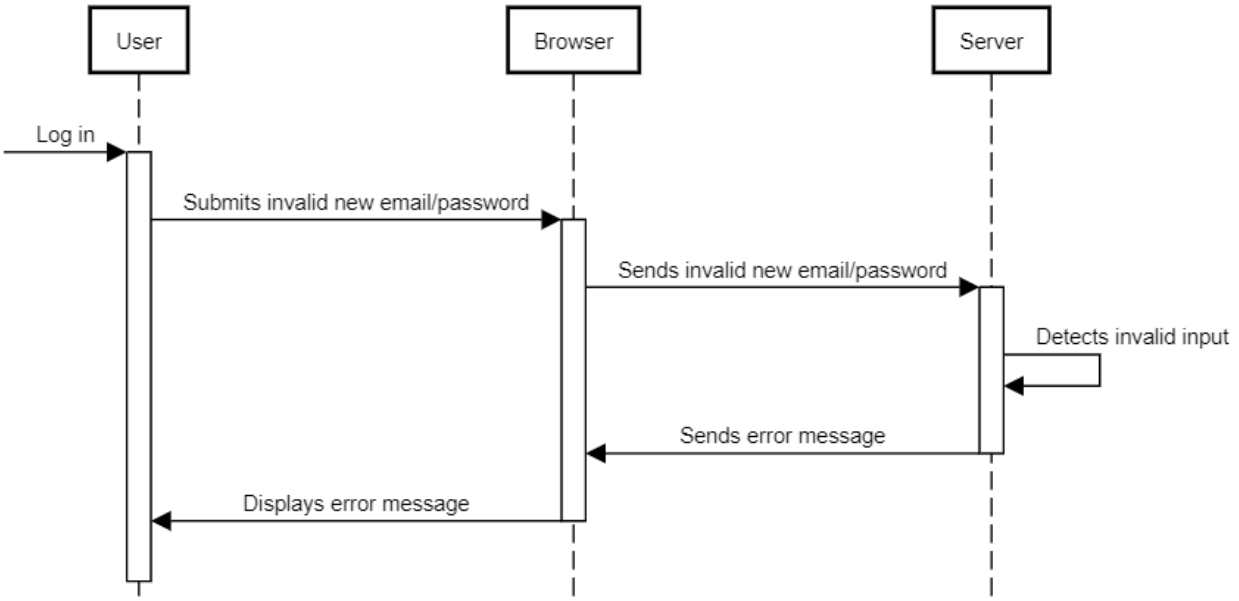

Sleep and Mental Health
User
Browser
Server
Database
Login
Clicks "Log Sleep and Mental Health" button
Request "Log Sleep and Mental Health" page
Sends "Log Sleep and Mental Health" page
Display "Log Sleep and Mental Health" page
Submits valid Sleep and Mental Health details
Sends valid Sleep and Mental Health details
Generates Sleep Data
Store Sleep Data
Confirms sleep and mental health were recorded successfully
Displays confirmation
Clicks "View Previous Sleeps"
Requests "Previous Sleeps" page
Retrieves Sleep Data
Gives Sleep Data
Generates Sleep Data graph
Sends "Previous Sleeps" page with graph
Display Previous Sleeps" page with Graph


Invalid Sleep and Mental Health
User
Browser
Server
Login
Submits invalid Sleep and Mental Health details
Sends invalid Sleep and Mental Health details
Detects invalid input
Sends error message
Displays error message

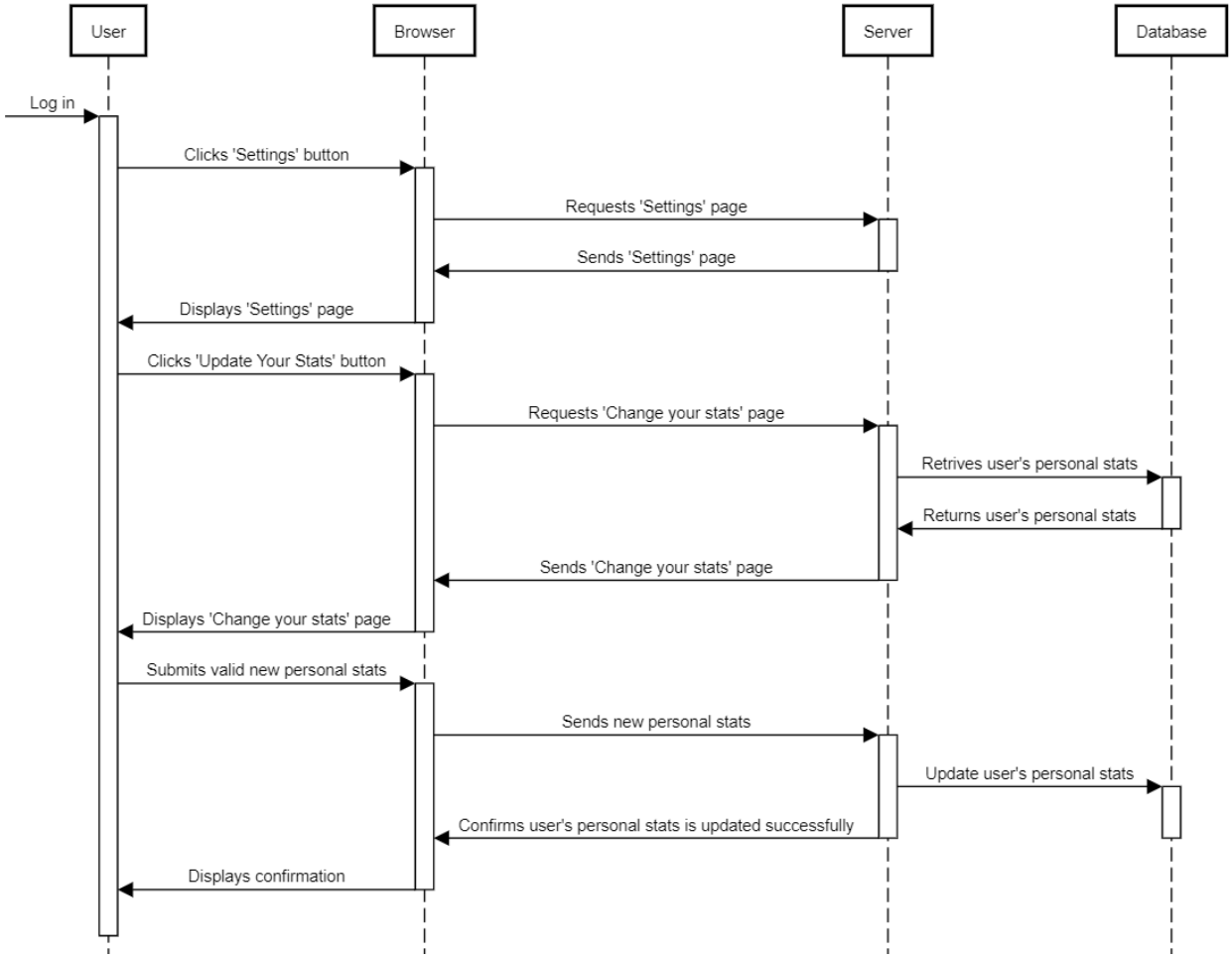
Update User Info



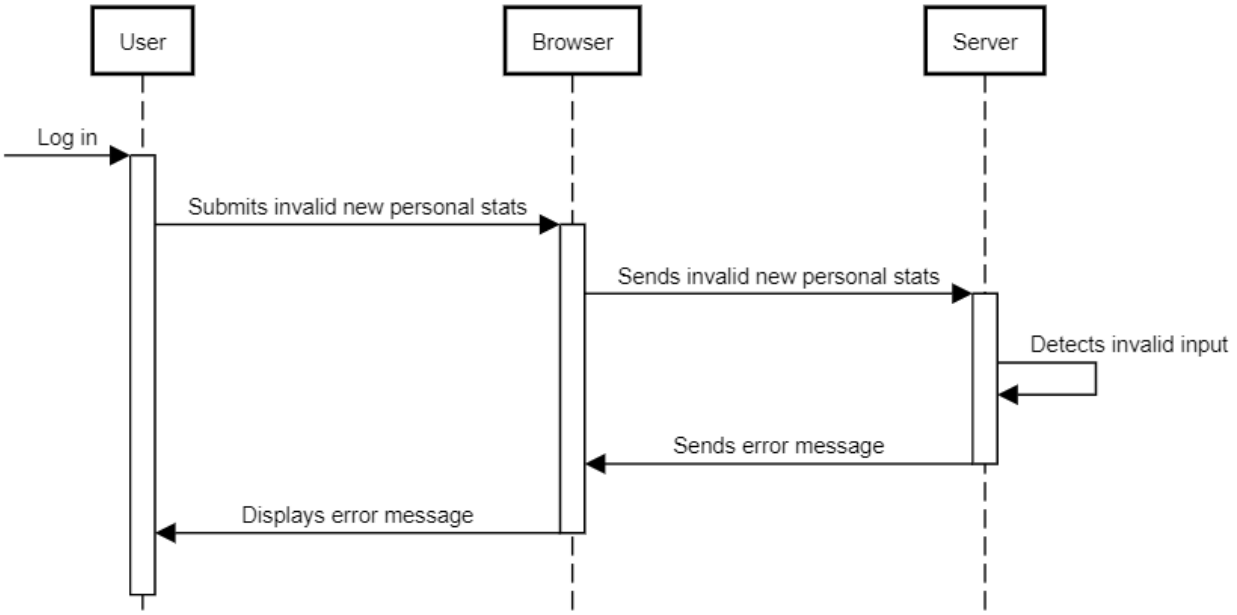
Invalid Update of Email/Password



Update User Info



Invalid Update of User Info



Key Technologies

Choice of a Platform

The final system will run its backend & frontend servers (Flask & React, respectively) on Debian GNU/Linux. Specifically, we plan to implement this code on a CSE machine or emulator for consistent system requirements and capability. The application itself is a web app and will be tested to ensure it runs successfully on both Firefox and Google Chrome. We chose to use our CSE web servers running Debian as it made for the perfect development/testing environment. Paid hosting solutions, such as AWS require a full set up of operating systems, programming languages, frameworks and modules, while our CSE machines had all of these ready-to-go, and was an environment consistent across group members. Given that we were making use of a React framework, minimal testing across browsers was required for the majority of the frontend, however we paid careful attention to ensuring our OpenLayer API Maps worked in these different browsers, as it was not as thoroughly pre-tested as the React Framework. In our testing we found that the entire application worked as expected in Chrome, Safari, Firefox and Microsoft Edge, tested on both Debian and Windows 10.

Summary of the Key Benefits and Achievements of Architectural Choices

We had a great deal of success in developing our prototype, our chosen technologies were well suited, aside from the frontend which underwent some changes. We previously made sure that our implementation would have high cohesion and low coupling by using technologies that we know integrate well together. We also chose software tools that the team was previously familiar with in order to streamline development and a platform that would ensure uniformity between group members as well as a stable development environment.

Frontend
For our final prototype, after careful consideration of the feedback we were given, we made the choice to switch our frontend to make use of the React framework, rather than no framework, as the benefits simply outweighed any negatives we had considered. In the end we had significantly expedited development by switching, and produced a more polished product.

Backend
For our Backend, we used the Flask web server framework, in Python, which processed and delivered content from the database to the frontend via a HTTP request API (GET, POST, etc). Flask ended up suiting the scale and nature of the project perfectly, and we didn't encounter any issues with it. It was easy to add endpoints, and easy for all group members to contribute to, as everyone had used it before. Python also made it simple to interact with our database by using the JSON library, making most functions called simple to write and understand.

Database
For our prototype, due to the very small scale/demonstration nature we made use of a series of JSON files. We would obviously change this to the aforementioned SQLite in a production environment.

API
While choosing which API's we could implement, we decided that it was important that they did not over-complicate our design. For example, there were a plethora of options for the map API, including some from large organisations such as Google and Apple. However, we felt that the majority of these would distract from the goal of the application which is to provide a simple way for users to input their workout data onto a map. In the end, we decided to use Openlayers 6 API, which is an open-source mapping tool. The main advantage of this decision is that OpenLayers allows the user to click on the map to draw their path. This means that it will effectively calculate the distance travelled without staying on predetermined paths. Since many people workout in places like parks. As such, OpenLayers 6 has novel features which will greatly benefit our application.
We were also planning to use a fitness API to calculate statistics such as BMI and calories burned. However, we quickly realized that it would be much simpler to just implement these calculations ourselves since they only relied on a few variables which were available to us.

Summary of the Key Benefits and Achievements of the Design and Implementation

Our design aims to target anyone who is looking to improve their physical and mental health. It allows users to plan exercises in advance or record previously completed workouts with the added benefit of including approximately how many calories were burned. The more workouts that are logged, the better the predictions are of suggested future workouts. Users can create personalised and achievable goals to aid in their progress. Additionally, logging sleep and satisfaction levels with the functional user interface enables users to gain a greater holistic view on their sleep and see where they might need more sleep. All of this is able to be done through a simple to use UI.

The implementation of this project was focused on creating a simple and easy to use frontend and backend that all members had experience using. The frontend uses a React JS framework as it uses JS and provides enough features for what we need. Furthermore, our backend was created using Python. This was chosen over Node JS as all members in our group had experience coding in Python, with Node JS also being more suited to real time applications, which our program would make no use of. The Flask framework was used over Django as it is a lighter framework that is better suited for applications like ours. Django boasts a greater deal of web development tools consequently making the code harder to understand for novice web developers like us.

Team Organisation and Appraisal

Responsibilities and Organisation of the Team

Our team for this project was randomly allocated which meant that, for the most part, we had not worked with each other previously. Despite this, we were able to quickly build relationships with each other and collaborate well throughout the term. Our first task was coming up with problems ideas to implement for our project. It was challenging to come up with something new and unique without taking on too much work that would not be feasible to complete in the 10 week term. After a few days of individual and team brainstorming, we settled on a health tracking web application. An idea which continued to grow and develop throughout each stage of the project.

For each deliverable, we would meet at the start of the week to discuss what needed to be done. We would break up the work into manageable chunks and discuss what each section meant. Once in agreement, we would then divide this work evenly between ourselves and decide upon a completion date that would be well before the final due date so that we would have time to confer with each other and help each other out if somebody had difficulty completing their section.

This strategy proved to be quite effective in completing all tasks on time and to a high standard that all team members were pleased with. Below is a description of what each team member contributed to each deliverable.

Deliverable 1

For the first deliverable, after deciding on our theme of developing a health tracker web application, the project was divided into six epics for each of us to write problem statements and user stories individually. We then collaborated on a design for the low and high fidelity prototypes before designing the individual pages separately. Michael worked on UI and usability, Guy was in charge of planning workouts, Timothy did map utilisation, Ethan worked on tracking workouts, Ellan did goal setting and recommandations while Lachlan was in charge of sleep and mental health tracking. At the end, Ethan made sure that all of our prototype pages were collated together into the one diagram.

Deliverable 2

For the software architecture in part 1 of deliverable 2, we discussed what each question was asking as a team. Once a group knowledge had been attained, we then went and all answered these individually before coming back and sharing our answers to create a combined submission that was a product of the best elements of everybody's personal ideas. For part 2, Ellen was largely in charge of updating the user stories based on feedback from deliverable 1 and asking our tutor for advice. Together we decided on a uniform way of drawing up our sequence diagrams then went and did those individually with everybody working on diagrams for their epic as described above during deliverable 1.

Deliverable 3

Deliverable 3 involved a pitch demonstration to our tutor. This involved a slide presentation as well as a walkthrough of the front end of our project. We split up the work so that Ellen and Lachlan were working on coding and collating the front end while Michael, Guy, Timothy and Ethan collaborated on the presentation. Specifically, Guy spoke about the overall product and its key features, Tim presented our APIs and architecture description while Ethan discussed our planned implementation. The frontend was then demonstrated by Lachlan.

Deliverable 4

The major task involved in deliverable 4 was completing our frontend design as well as implementing the backend including API integration and database management. Ellen was designated the role of putting the finishing touches onto our frontend which included adding appropriate warning messages to the user and providing a more user-focussed experience in general. Michael worked on creating a subscription system to improve our business value, Timothy implemented the recommendation software to help users meet their goals, Ethan was in charge of calculating stats such as pace and calories burnt based on user input and Lachlan designed and formatting the backend database to provide a persistent application that would store all necessary data for each user. During this phase, Guy worked on an improved pitch including the new business proposition of adding a subscription to our application as well as spending time so that we could all get ahead on deliverable 5.

Deliverable 5

For the fifth and final deliverable, we had to include a summary of everything from previous deliverables as well as adding some new reflective content as well. This proved to be quite a large task but once again, we were able to break it up into smaller sections and complete each one well. Guy was able to work on transferring all relevant prior content from previous submissions and apply the necessary changes to make them even better based on feedback from our tutor. This included problem statements, features, user stories and sequence diagrams. After everybody else had finished with deliverable 4, we all divided up the remaining sections evenly between ourselves. Michael focussed on the achievements of our application including screenshots and descriptions, Tim wrote a description of our key technologies including our API usage, Ethan discussed the choice of platform including browser and deployment as well as described some of the problems we faced. Ellen then completed the section on the updated design and implementation and Lachlan wrote most of the team reflection section.

Overall, each team member was pleased with the organisation of the team and the way we were able to share responsibilities evenly. We would be more than happy to work together on a similar project in the future.

How Did the Project Go?

In general, we all felt that the project was a great success! Looking ahead from the start of the term, the scope of the project certainly did feel very daunting and we were uncertain how we would go about completing all aspects. However, as the project was broken down into deliverables and we then divided each deliverable between us, we found that we were meeting our goals in a timely manner and were often pleasantly surprised about how straightforward everything became when we all worked together. We all agree that many hands do indeed make light work.

Another major contributor to the success of our project was the fact that we spent time developing a well thought out application from the beginning. The health tracker web application theme gave us a strong foundation with similar products receiving success on the market already. We were also able to add new elements to ensure our project would be unique. The agile method of iterative development employed throughout this term also meant that we would be able to easily expand the scope of our application as time and ability allowed. For example, we were able to include sleep and mental health tracking, an idea we initially developed during deliverable 1 but decided against later on before being confident to bring it back for the final demonstration.

We have all learned a lot from this experience as a whole including how to build a strong team from complete strangers. The lack of face-to-face learning over the past 12 months due to COVID-19 has meant that we have not had as many chances to collaborate in our university courses. This project has been instrumental in allowing us to once again experience the benefits of teamwork and enjoy working together, something that will continue to be an important element of our future careers as software engineers.

We have also been able to considerably increase our technical skills in both front and backend development as well as requirements and design engineering. We have been encouraged to keep user stories and requirements at the forefront of our design and implementation process and this has helped influence the way we code and write software.

Issues or Problems Encountered

As with any task, the completion of this project was not without its inevitable hiccups and issues along the way. There were times when the criteria was not fully understood by the team and we were penalised when it came to marking. We also had some moments of miscommunication where some team members were unclear of their role for a particular section. However, these challenges have become learning experiences as we have been able to develop our knowledge and be more careful in understanding requirements and ensuring that all team members are on the same page.

Specifically, we had some issues with choosing a framework for our frontend and backend integration. We had originally decided to work in Vanilla JavaScript and we used this strategy for our deliverable 3 demonstration. However, the tutor was questioning this choice during the demonstration which was unexpected and caused us to delve deeper into the possibility of using more of a defined framework. After careful consideration to the given feedback, we decided to switch our frontend from Vanilla Javascript to the React Framework. Ultimately, there were a great deal of benefits to using a framework over none, that truly expedited our development more than we had initially expected, despite the fact that we had to rework some of the frontend code that we had already done. This resulted in us being able to produce a more polished final product for the deliverable 4 presentation. However, this certainly proved to be the biggest change throughout development.

Another significant issue we encountered was during our final demonstration, where we had made an untested change just prior to the presentation. Unfortunately, this broke previously completed software elements and caused performance issues, meaning we needed to switch presenters mid-way. However, the second presenter did not have the final version of the code stored locally as they had not run ‘git pull’. Thus, the most recently added feature of recommendation generation was not able to be demonstrated. The side effect of this meant that we were presenting a less-than-optimal product demonstration that may result in the prototype appearing to be less finished than it really was. This will certainly teach us all to not try to change the code so close to a submission deadline as any advantage gained may not be worth the consequences of not being able to thoroughly test our application.

Would You Do it Any Differently Now?

When first embarking on the first deliverable, we were not too sure about the overall scope of the project. We knew that we would have to implement a prototype for our application, but did not really know how or what that would look like. With the benefit of hindsight and knowing exactly what the final requirements are and what our prototype became, there are a couple of changes that the team could have made along the way to improve the overall process and end product.

In particular, we might have taken a different approach to the way we first started the project and developed our ideas. We began by thinking about a product that would be simple enough for us to implement while still being useful to a wide audience. However, this meant that we were struggling to find appropriate APIs and framework technologies to help us with this. In future, a better technique might be to start with a chosen technology stack and find some helpful APIs, then figure out how we can integrate all of that into one seamless application.

Another thing that might help us out in future would be to have more sprints so that we would have less work to complete during each iteration. This could have been achieved by starting the prototype implementation earlier. At times we felt overwhelmed by the task at hand as we were implementing a major feature, such as the database or API integration, in one sprint. This holistic approach meant that we were unable to test all features as thoroughly as we would have liked and so some bugs and errors were only discovered at the last minute. Therefore, being able to break these implementation stages into smaller chunks might have been beneficial to the project as a whole.

Timeline

Below is a Gantt Chart of our teams' progress throughout the term. Please see https://lucid.app/lucidchart/2cbc8740-5c48-4c48-a5ae-26752e1862f1/view?page=NUDq6GBwIiKa# for a full resolution image.