StrayerFitness Website

Creator: Kaylee Strayer (Me)
Academic Advisor: David Eve
Project Advisor: Michael DeSanty

What is the project?

1. A website to advertise my personal training business but also a place for people to get basic knowledge about the health and fitness world for free.

2. People can also register to become clients and schedule sessions once they are signed up. Trainers that work for me can see their schedules (upcoming client sessions).

3. I was motivated to choose this project because I am very passionate about the fitness space and was thinking about one day having a personal training business.

Audience/ Differences

- People new to the fitness space
- The original scope didn't include:
  - Calendar UI
  - Google Auth
  - Individualized scheduling (client login)

Knowledge and Technology Used

Academic Courses

- Full Stack Development
- Advanced Web Development
- JavaScript, React-Bootstrap based UI
- PostgreSql based API
- GitHub and Git Bash/Command Prompt

Research

- Google Authentication
  - Chosen because it is one of the most popular with this type of project and is accessible
- Calendar UI
  - React-Bootstrap was my Calendar of choice as it was already in use in my project
- Different types of login levels
  - Needed for both clients and trainers

API Setup

- PostgreSQL Database connected to the rest of the API by the pgClient config file.
- Controllers: Clients, Trainers, Sessions, Reasons, Auth.
- Middleware: Separate auths (google and regular sign-in) as well as record validation.
- Routes: Clients, Trainers, Sessions, Reasons, Auth. As well as the standard Server.

UI Setup

Auth and State Management: Slice, Action, Modal, Store.

Components: Sessions (Create, Edit, Delete, Form), Clients (Create, Edit, Form), Knowledge-Sharing Components (Ex: Nutrition).

Miscellaneous: Main, App, and CSS.

Google Authentication

Challenge 1: A completely new technology to learn and research

Solution 1: I looked at videos and explanations about how to sign up for a Client account using Google Auth and what JavaScript code as well as HTML should accompany it.

Challenge 2: Google Auth was failing to take even though the website was showing that a client was logged in (the button changed).

Solution 2: I looked into all the possible reasons this could be and tinkered until one eventually went through. It ended up having to do with the allowed origins page in the Client section of the Google auth website and how my application was handling cross-origin navigation.

Calendar UI

Challenge 1: I had never worked with any calendar ui interface before.

Solution 1: Look at the react-bootstrap calendar documentation and see if the look fits what I need. Also research how to implement the interface. The second mini-challenge was figuring out the CSS styling rules for the calendar as there were so many small nuances to it because the interface is made of so many different parts (ex: certain box type colors can be controlled separately).

Multiple Logins

Challenge 1: Only clients should be able to schedule sessions for themselves. This requires a client to have a login as well as the trainers.

Solution 1: I first had to figure out how to set this up. I wanted to keep the two tables separate but also have login logic stay simple. Because of this I decided to just move login logic from both into a user_accounts table and include a role row to distinguish whether the user had trainer permissions or was just a client. This required a lot of work to implement because I had to change the whole login logic not only in the backend, but also how this logic was received and treated in the front-end (ex: queries).

Session Form

Challenge 1: The session form appeared to not be submitting once I changed the population of the client selection box (it now was supposed to only show the client that was logged in).

Solution 1: I stuck console outputs into the code in two different places. The first was placed to make sure that the front end was receiving the information from the form. The second one was to display the information that the form was receiving and sending to the back end. I found out that the client id that the front-end was sending was not matching up with what it should have been.

Continued...

Solution 1: Then I realized why (the id for the user_accounts table was being accessed not the client_id in that table). From there it was just a matter of switching things around to make sure that I passed the client_id in the token so it was more easily accessible by the front-end.

Conclusion

In the end, I was successful in creating a working full-stack application to support the further development of a personal training business. In the future the feature that I would like to add would be the ability for clients to add a description when they register and figure out how to structure it in a way to prevent abuse. I would also like to add a footer that has copyright information as well as more links like other business websites have. Through this project I learned how to have patience and persevere through challenges that seem impossible. I also deepened my understanding of how each element of a full-stack application works together and learned new skills such as multi-type login, calendar ui setup with React-Bootstrap, and the Google Authentication framework.