

## Foundations Capstone demo plan

Start at main page

Project intro:

Introduce myself and discuss what the project is and the purpose of the site aka. To give people access to muscle specific random exercises, allowing them to ensure that they don't plateau on their exercise journey. (10 sec)

1. Colin
2. What is the site
3. Purpose of the site
4. IT IS A WORK IN PROGRESS

Explaining what and how I built it:

I developed the web pages using css and html for the majority of the layout and used node javascript for the functionality portion of it. Navigate through the pages with the menu button explaining its complexity(navigate through the 4 html pages while talking

5. What was used in building of the site?
6. Heroku database
7. Sqltabs for testing
8. Sequelize for running in seed.js file
9. Post man for get requests
10. Heroku was also used for deployment
11. Navigate through the 4 different html pages while talking

Explain the randomizer and delete button difficulties:

12. Originally built out the sql tables with each individual workout in its own column. This made the data extremely difficult to manipulate once it was pulled. The goal was to get only the amount of workouts that would be performed in one session. The problem I was having was that the way the data was entered it was either all of the entries or none of them. I was successful in showing only one of the columns but the way I put it, all of the information was in a single column of the table. This made it very difficult to try and pull and show just part of the column. I attempted to create an object and put the data in there to then try and send the part of that object to a new object that would store only the data that I wanted to show. To correct this I then remade the tables so that each muscle group table had a workout name and workout description that would later in the sql be inserted into the tables. This allowed for the data to be handled much much easier.
13. Once the data was corrected I created an array with each index being each individual workout. Then using a simple randomizer function pulled one workout and then ran it again for as many workouts were necessary. Then using an axios call I got and sent the data to the web page.
14. Then I realized if I was a user and didn't care for or had some kind of limitation with a particular workout that I may want to remove and then get another one so I added a remove button. This allows the user to remove one exercise at a time.
15. To do this I created a button after the random workout is created, by creating an element with the context of remove. Then adding an event listener on click to run the reset workout function. That function is targeted to the parentNode of the parentnode and then I used the remove child method.
16. Touch on the fact that I used postgresSQL and javascript

