

Part 1: Stories

1. Brainstorm

- a. Working on demo, learning how to screen share to show how a project works
- b. Javascript beginners, learning how to learn different perspectives
- c. Flexbox to lay out the page evenly and easier than manually trying to position things
- d. Back end servers, how the back end works with the front end
- e. Servers, how to call different things to get them to work how you want
- f. How to use semantic tags in html to make css work better together
- g. How to write a bug report when doing qa to make things more efficient
- h. Why i learned back end server for an app
- i. Learning how to deploy
- j. Flexbox for shapes lab

2. Steps 2 and 3: Write

a. Story 1

i. Learning flexbox to make css easier to lay out the page (CSS shapes lab)

1. I made a shapes page that needed to be positioned a certain way. I looked up and learned how to use CSS flexbox in order to position the shapes in a better way that is more even, even when the page is responsive. Positioning the shapes directly on the shape manually one at a time is very difficult because you often can't get it exact and it will change as the size of page changes. I chose flexbox because I wanted to learn how to use it better even though I am still a beginner. It also makes your life easier when using CSS and is very helpful to learn. I would love to continue my learning of flexbox so that I can get better at CSS. I learned that learning something new in a language that can ultimately save you time and headaches is definitely worth it and can help over time as you practice and continue your learning.
2. Some of my favorite learning points with flexbox is the way that you can position anything really in any direction. They can be reversed or rearranged. The idea of flexing is making everything more visually flexible. This is so handy when the page is responsive and is different sizes. The layout of pages can be so touchy so taking the extra time learning flexbox has been very important. I will continue to take the time to learn these certain shortcuts because it saves so much more time and can be used for any page.

b. Story 2

i. Learning how to deploy (deployment lab)

1. It was very exciting learning how to deploy a page that I had written out for the deployment lab. We used heroku to deploy where it is hosted somewhere else outside of my computer

making it more accessible. I definitely learned a lot more when using heroku and going through the actual process of looking at the differences between my localhost page and then what is actually being deployed. There were so many bugs to work out and things that needed many touch ups since one of the main problems was getting the CSS to come through from my code to the actual live site. The CSS and the images were having a hard time coming through when being deployed. One of the things was setting up the server the right way using get requests in order to send the files through the right path in order to get it to actually deploy. Everytime I would deploy and the styles and images did not come through, I knew that it had to be that the method was not set up correctly. I had to learn step by step on how to set up the method correctly with the back end in order to deploy correctly. I had to go over the syntax and make sure everything was coded correctly.

2. Learning how to deploy with heroku has definitely been challenging and I need a lot more practice. It has helped me in my perspective of how I can code in the future and how I want to code neater and tidier and well organized. It makes me think of how I can set up my code to work better together as I set up the images and CSS specifically since that is what I had trouble with the most when trying to get the styles and images to come through. Setting up the server and double checking the syntax and continually going through the quality assurance to make sure that things make sense and run smoothly for a smooth deployment is important. Knowing what to look for when things do not deploy correctly is very handy when things are not working out. All of these little mistakes are helpful in the long run when we can learn from these mistakes and make better decisions that make coding more efficient in the future.

c. Story 3:

i. experience building movies list app in demo

1. Building out the movies list app seemed very complicated since there is so many different files and trying to keep everything straight can be confusing. I learned how the back end works with the front end and how to call different functions to each other which is requesting and getting responses back. I learned how important it is to keep your work organized and in different folders so you can keep everything straight. It's also important to reference the files at the top that are getting information from another file. It's important to keep your code organized in each file to more easily read. I use all of my future projects this way to get it more organized and it just makes better sense to make a server

folder that does everything for the server and so on. One of my main problems is getting overwhelmed with files and not quite sure in how to organize everything. It also helps to know what the front end does and what the back end does in order to make it less complicated when trying to think what file is doing what.