Engineering Journal

Louis La

1. Capture short term goals or motivations

Every action you take should be driven by some challenge you are facing. Many times this will be simply the requirement of the project. However, you should be able to frame each challenge in your own terms. Own the motivations in the context of your own project.

2. Capture the actions taken or changes made

This portion should be the most detailed. Make sure to be technically precise and remove any ambiguity related to the action you are taking.

3. Capture the results observed

Results can be recorded in many ways. The best type are objective numerical results, such as the output of a test run. However, you can also make visual observations or subjective observations reflecting on your work.
Reflect on the impact of these results. In many cases, the results that you observe will lead you to identifying your next challenge. Whether it is something to be worked on immediately, a ticket created for work later, or the ability to move on to the next phase, you should evaluate what your results mean for the project.

4. Capture the impact of these results

5. Maintain and organize the above in a clear, digestible, and accessible format.

Tuesday, July 13, 2021

Overview

- First day of FEC
- Met up with the group for the first time

Goals - Challenges - Motivations

First goals are to create a group Trello account, GitHub organization, and make sure everyone has access to them.

Actions Taken

We met as a group during class time to create the above components needed. Everyone has access to all the needed accounts and are added as admins.

Results

Today was a successful day, we accomplished what was needed for the daily tasks.

Thursday, July 15, 2021

Overview

- Phase 0 is completed and slowly moving into Phase 1

Goals - Challenges - Motivations

The main goal for today is to set up our GitHub repo, Atelier, with a working environment. React, Webpack, and Express need to be all running.

Actions Taken

I cloned down our master branch and our group worked together to install the needed dependencies and boilerplate code to have the repo run properly.

Results

Everything went smoothly, we have the basis of our project incorporated. Babel is transpiling properly and the server is running as well.

Saturday, July 17, 2021

Overview

- Fiddling with the Atelier API
- ESLint

Goals - Challenges - Motivations

The main goal for today is to incorporate an axios request and the Atelier API into the repo and to install a linter.

Actions Taken

We had some issues getting the Atelier API to work with our request. After playing around with it, we were able to get the data from the API. We also installed ESLint.

Results

The API is set up with our git token and ESLint is installed. We still need to play around with ESLint settings or learn more about it because a lot of our code was not properly formatted.

Tuesday, July 20, 2021

Overview

- Setup test suite
- Shared States

Goals - Challenges - Motivations

The goal for today is to set up the test suite and make sure it is working properly.

Actions Taken

The group worked together to set up Jest in the main repo. Also, we discussed what states need to be shared in our app.jsx file.

Results

Our Jest tests are working properly. We tested it with a simple math function and it passed. In terms of our state, we worked out a few shared states that each of our widgets will be using.

Thursday, July 22, 2021

Overview

- Tickets
- First merge from a branch
- Testing with enyzme

Goals - Challenges - Motivations

Actions Taken
Our group met up to try to merge the first branch for a test. In addition, we tried to incorporate enzyme.

Results
Enzyme is not currently working, we will continue troubleshooting it. The first merge came through with a few bumps, but we were able to figure it out.

Saturday, July 17, 2021

Overview

- Enzyme
- React Components

Goals - Challenges - Motivations
Check off remaining weekly deliverables for Phase 1.

Actions Taken
We met up to take a look at the enzyme/jest test suites that we put together. Also, we made sure everyone is still on track with the current schedule deadlines.

Results
Enzyme was set up with a separate test suite for each person. In addition, we went over any packages we needed to install for our components (Moments, Multer).

Tuesday, July 27, 2021

Overview

- Skeleton

Goals - Challenges - Motivations
Go over weekly goals.

Actions Taken
Our meeting today involved going over each person's progress on their widget.

Results
Everyone has their basic skeleton done and are on track to finish up a good amount of functionalities for Thursday's meetup. We also shared issues we each had to see if we are able to help each other solve.

Thursday, July 29, 2021

Overview

- Merge

Goals - Challenges - Motivations
Merge everyone's skeletons.

Actions Taken
Our meeting today involved attempting to merge our three widgets' skeletons together.

Results
We had some issues with the merge. There were a few conflicts that caused other components to not render properly. After trial and error we were able to get it to function with all three widgets together. Our webpage properly renders what each of us expected.

Saturday, July 31, 2021

Overview

- Code Coverage Tests
- Progress Demo Video

Goals - Challenges - Motivations

Have working tests that will show our code coverage percentage as well as record our progress demo video to this point.

Actions Taken
We spent most of the class time working on our tests and recording our demo video.

Results
Our tests with jest were having some issues so we opened up a help desk ticket. Even though all components were rendering properly, the tests seemed to think otherwise. So we had to find workarounds in how we wrote our tests. Also, we were successful in getting our jest/enzyme tests to show a code coverage percentage. In addition, we rehearsed and recorded our progress demo video. Each team member introduced their widget and spoke about challenges and successes up until this point for each widget.

Tuesday, August 3, 2021

Overview

- Lighthouse Audit

Goals - Challenges - Motivations
- Get lighthouse audit report and merges

Actions Taken
We spent most of the class time figuring out our merge conflicts. Then we proceeded to getting our lighthouse audit reports.

Results
Our group is still learning about pull requests and merging so we've been going through the steps together during each class before we merge. Any conflicts we are resolving together so that we can learn the process.

In addition, we tested the lighthouse audit reports.
Link to report:
https://drive.google.com/file/d/16qMTFva6ob_5LpPaV7JJKoZtYIKGjUpn/view?usp=sharing

---

Thursday, August 5, 2021

Overview

- Merges

Goals - Challenges - Motivations
- Merges and ESlint

Actions Taken
Our group used all of class time to do our merges together and also change our linter.

Results
We had merges to work on together and added a flex box for our entire app so that we can see where everything is at.
We also change our linter from the Airbnb style to the Hack Reactor style. Our previous linter was giving us issues so we decided to change it.

Saturday, August 7, 2021

Overview

- Reflection Form
- Confirming CSS

Goals - Challenges - Motivations
- Merges and reflection forms

Actions Taken
Our group met up for about two and half hours to work on reflection forms and cover any issues we were having.

Results
We did code reviews over zoom so that we can all learn from each other as well as doing the pull requests. There was an issue with one pull request so we worked together to try and resolve that conflict. The merge was not successful so we ended up with compiling issues and the CSS not showing up. We resolved this by going through each error one by one.

Tuesday, August 10, 2021

Overview

Thursday, August 12, 2021

Overview
Lighthouse Audit results:
Performance: 31
First Contentful Paint
13.0 s
Speed Index
13.0 s
Largest Contentful Paint
19.0 s
Time to Interactive
15.0 s
Total Blocking Time
0 ms
Cumulative Layout Shift
0.109

Saturday, August 14, 2021

Overview

- Reflection Form
- Confirming CSS

Goals - Challenges - Motivations

- Merges and reflection forms

Actions Taken
Our group met up for about two and half hours to work on reflection forms and cover any issues we were having.

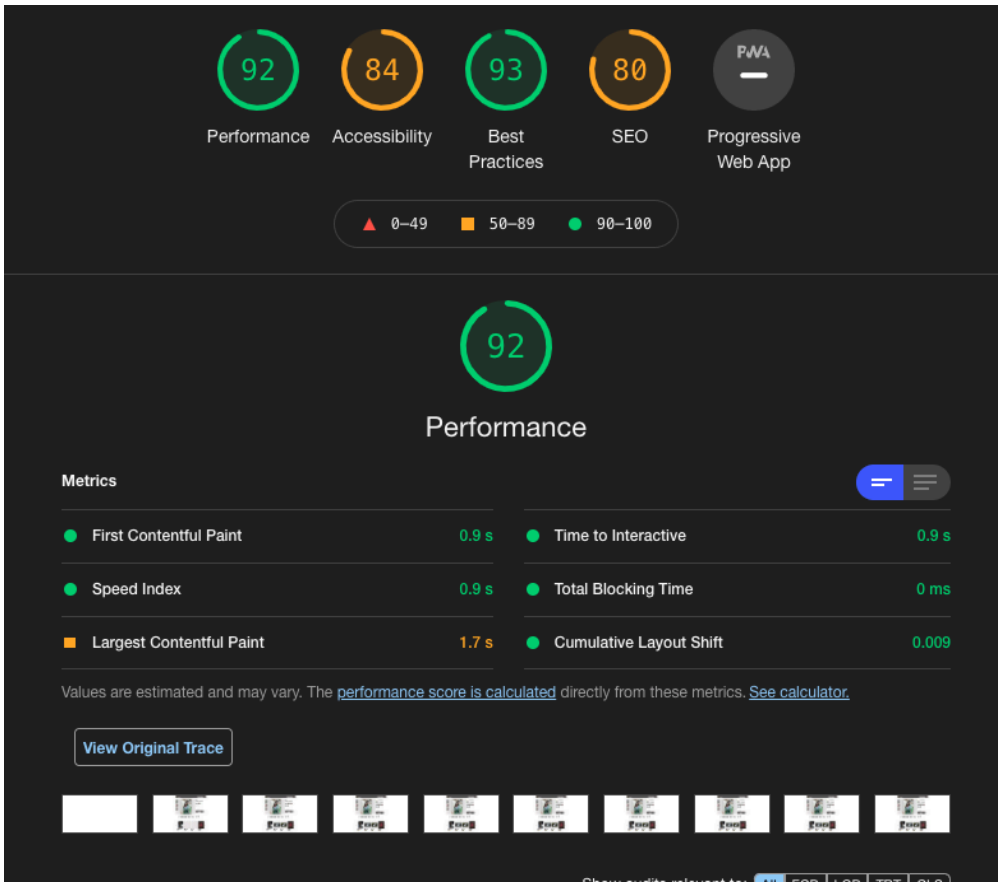
Results
We did code reviews over zoom so that we can all learn from each other as well as doing the pull requests. There was an issue with one pull request so we worked together to try and resolve that conflict. The merge was not successful so we ended up with compiling issues and the CSS not showing up. We resolved this by going through each error one by one.

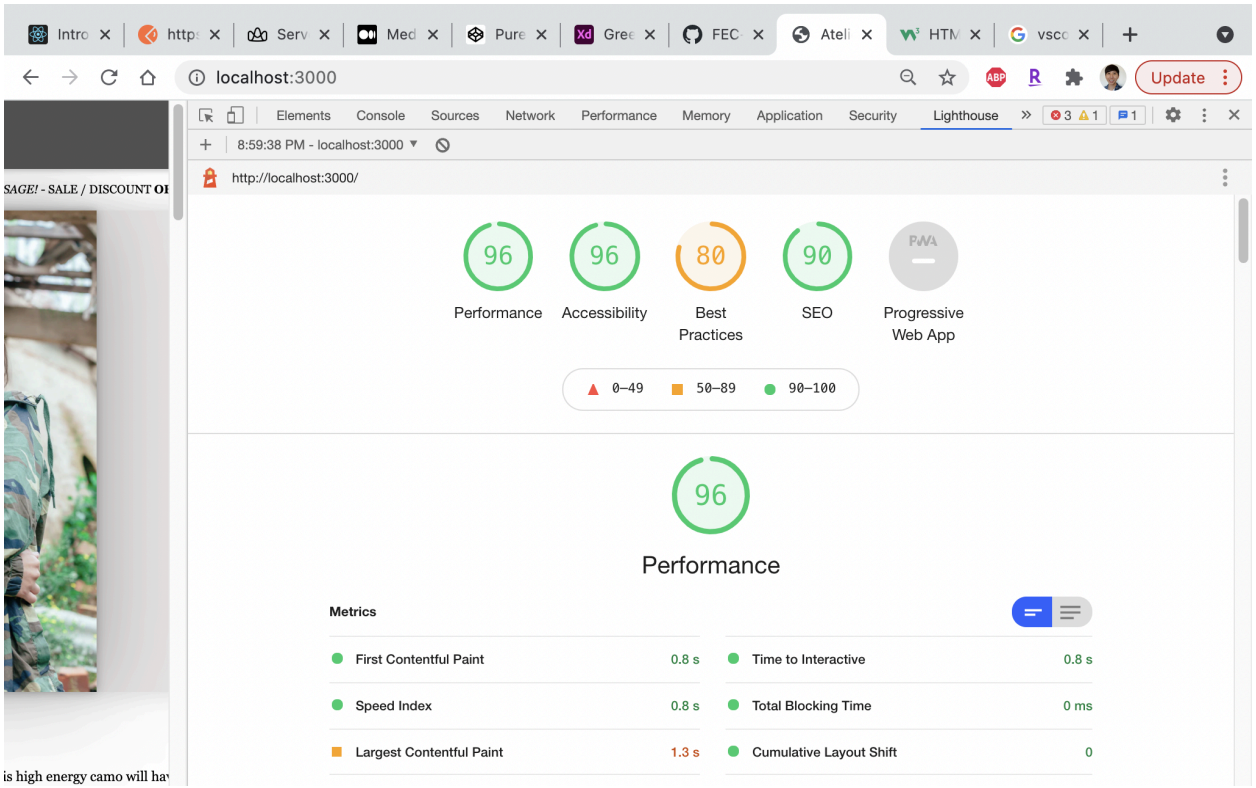
Friday, August 20, 2021

Day before FEC Demo

My tests that I have written are not passing but I've chosen to optimize the application so that it hits the benchmark goals in the business document.

Before optimization, we were in the 60s overall. After putting preload images, adding alt images, fixing react errors (key props, classes),
Compression npm package for bundle.js. - The numbers jumped to 96. This is for the browser





Requirements

Quality Control and SLAs

Page performance of the new web-portal should be equivalent to or greater than the legacy site.

- Time to First Paint: *0.8 seconds*
- Time to First Meaningful Paint: *2.0 seconds*
- Time to Interactive: *2.5 seconds*

Our numbers are greater than the legacy site finally!

Monday, August 23, 2021

Recap and Final Thoughts

Concepts Applied and Used

- ★ Making Axios requests to a REST API
- ★ Making sure all functions are asynchronous when needed
- ★ Understanding React's lifecycle methods
- ★ Properly handling state
- ★ Using third-party react components
- ★ Installing a proper linter
- ★ Writing and setting up tests for react components (jest)
- ★ Code coverage
- ★ Clean CSS
- ★ Git Branching
- ★ LightHouse Audit
- ★ Deploying with AWS and EC2
- ★ Optimization
- ★ Code Review

Challenges

1) **Where to put 'state'**
    - The first challenge was figuring out where all the state should live. Initially, I was having state in each component of the module (product information, gallery, add to cart) and I didn't fully understand the lifecycle methods of react. So I was running into issues with components not being able to render and kept getting "undefined." In addition, components were mainly not rendering because the lifecycle methods finished before the API calls were able to complete their requests. Basically, I was rendering nothing because props had not yet received the data from the API.
    - After reading articles and understanding how react works when rendering, I refactored all of my code. Now all API calls are made in the main class component, ProductOverview. All the requests are made here and sets the state with all the needed data to later be used for each component.
    - I then pass down the state (this includes product styles, skus, quantities, sizes, etc) as props to each of our components.

- This helped solve the main issue of components not being able to render because of an “undefined error.”
- Lastly, I hardcoded placeholder data in a separate file so that the main component uses that data in state to render until the API calls finish and re-renders the new data.

2) **Zoom in 2.5x on main image**
    - One of the features for the photo gallery section is when the user clicks the image (during full screen mode), the image zooms in 2.5x and allows the user to pan around the image on hover.
    - Initially, I was searching up ways to do this with raw CSS and JS. This included adding event listeners. I tested this without react and webpack and it worked fine. But once it goes through webpack compilation, it does not work. I didn’t dive too deep into the reasoning behind this as at this point I had already spent half the day trying this feature.
    - I decided to go back to square one and looked into a react component package. Once I’ve selected one made by other developers, I npm installed it and followed the simple 2 step instructions. This immediately worked.
3) **Open size select menu on click**
    - When a user attempts to click the button, “Add to Bag,” without selecting a size, there should be a message that tells the user to select a size and the select menu for the size should automatically open up.
    - After going through many stackoverflow posts and other articles, it seems that it is not possible to have a select tag open as a result of an onClick action by using plain CSS and JS.
    - But what I can do with CSS and JS is to manipulate the select option and have it change to a scrollable menu opened. This was an alternative that I decided to do instead. It definitely does not look as good as the proposed select option menu, but it works.
4) **Re-renders when given a different product id**
    - One of the main features overall in the project is when a user clicks a different product (related product), it will re-render all the components to reflect the information of the clicked product. This feature is supposed to be held by the team member who is creating the “Related Products” module. But since we do not have a fourth team member in our group, that module is omitted.
    - I hardcoded a basic version of the related products module. It’s main purpose is to change the product id on the main application file.
    - The issue we were having at first is that nothing was re-rendering even though the product id has been changed. So we looked into the issue and

implemented a “componentDidUpdate” when the product id being passed down to each module has been changed.
    - This somewhat worked for my module but not fully. Not all features on my module were re-rendered.
    - So I went into each component that wasn’t re-rendering and implemented “componentDidUpdate” and had to have it also reset certain states. Some of the issues prior to that the previous state saved and so if the user selected a size and quantity, those numbers and size would still be the same when a new product has re-rendered.

Future Plans

- Redo CSS for the other components
    - One thing that I would really like to do for the project is to redo the CSS for the other two components (Ratings & Reviews and Questions & Answers) so that the style matches better with mine.
- Use state manager
    - As per the challenge I had with controlling “state,” I would like to learn to use Redux or another state manager so that I can control it better.

Decisions

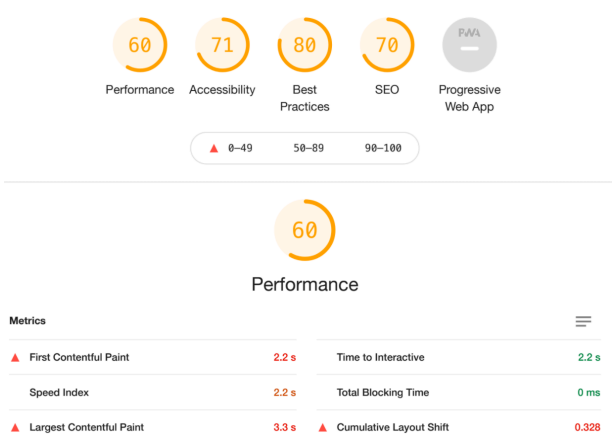
- **Using raw CSS instead of Bootstrap or CSS libraries**
    - One mistake a I made was that I didn’t look ahead on the recommended lists on things to use (on Hack Reactor gLearn page). One of it was bootstrap and other CSS libraries.
    - If I had looked at that earlier, I would have tried to learn it and implemented it. By the time I saw it, I would have had enough time to learn and implement it.
    - I made the decision to learn “FlexBox” so that I would be able to control where each section of my component would go. This definitely helped align each div much better and cleaner.
    - Also I decided to use all raw CSS to build out my module because it was a skill that I significantly lacked since the beginning of the program. Once I saw the wireframe for this project, I immediately already knew that it was out of the scope of my CSS knowledge. So I knew I had to take a step back and re-learn basic CSS.
    - One of the ways I learned better CSS techniques was investigating other shopping websites that had similar design as our given wireframe. Some of the sites I visited were etsy, Uniqlo, Gap, H&M, eBay, and American Eagle. I clicked around the websites and inspected their pages to see how

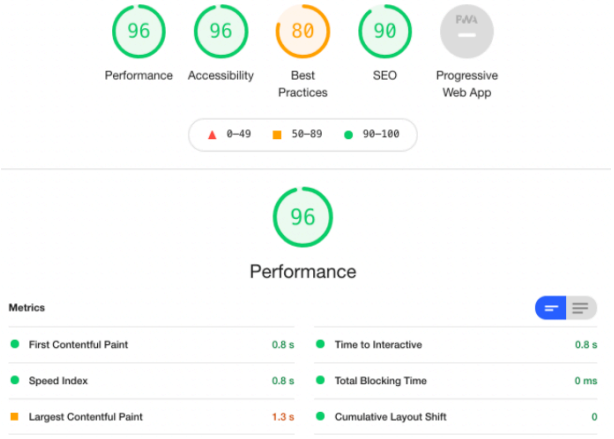
they structured their divs for each component. I also inspected their CSS properties for each div.
    - This taught me the basics of what makes a website look clean and appealing in the eyes of the customers. Some of this included not using harsh solid colors like plain black, blue, red, etc. Always use a shade of a color and never the solid colors themselves.
    - Include box-shadows and border radius for boxes. A lot of this things sound minimal but make a huge difference to the viewers.

My Role

- I would say my main role during this project was playing as the team leader. It's always good to have someone who keeps track of what needs to be done weekly and that everyone is communicating well. It's important to make sure everybody is on track for the weekly deliverables and for the whole group to be on their way to completing the project together.
- My definition of a team leader is someone who takes initiatives on starting the day and setting up the meeting times. And also trying to be one step ahead in the game on what needs to be done. To be honest, I wasn't always able to be ahead of the game because I was also struggling on working my part of the project. But I tried my best to always read ahead so that we know what's in the upcoming weeks.
- Overall, I think I did a decent job at playing this role. Our project was completed and met all bare minimum requirements ahead of the deadline. There were bumps during the weeks but we overcame them by helping each other with the problems. For example, if one member has issues solving a JS problem, we would group together and try to solve it together and learn from it. Ultimately, our teamwork was fantastic.

Optimization (before & after)





- Adding a preload photo to the main image and compressing bundle.js increased the performance at least 30 points
- Removed unnecessary console logs
- Fixed most of the react errors such as each props (div, span) needing a key
- Specifying a language in index.html
- Specifying name and content in meta for index.html
- Having a preload script
- Adding alt text to all image tags