**Increase you page load component with best practices**

Step1: Remove unwanted code and commented code

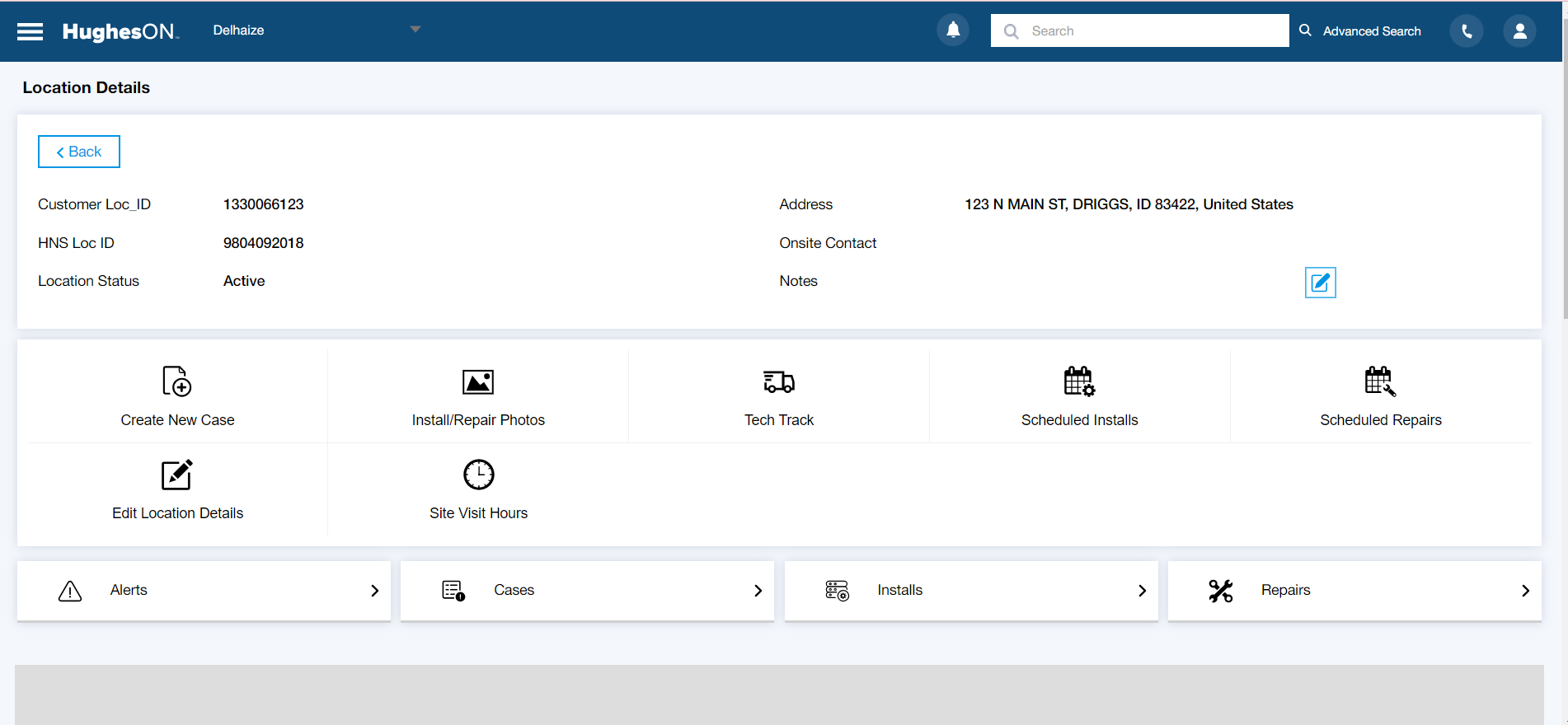
Step2: In Js file remove console.log statements once your code runs successfully.

Step3: Remove unused variables if you’re not using anywhere in your code.

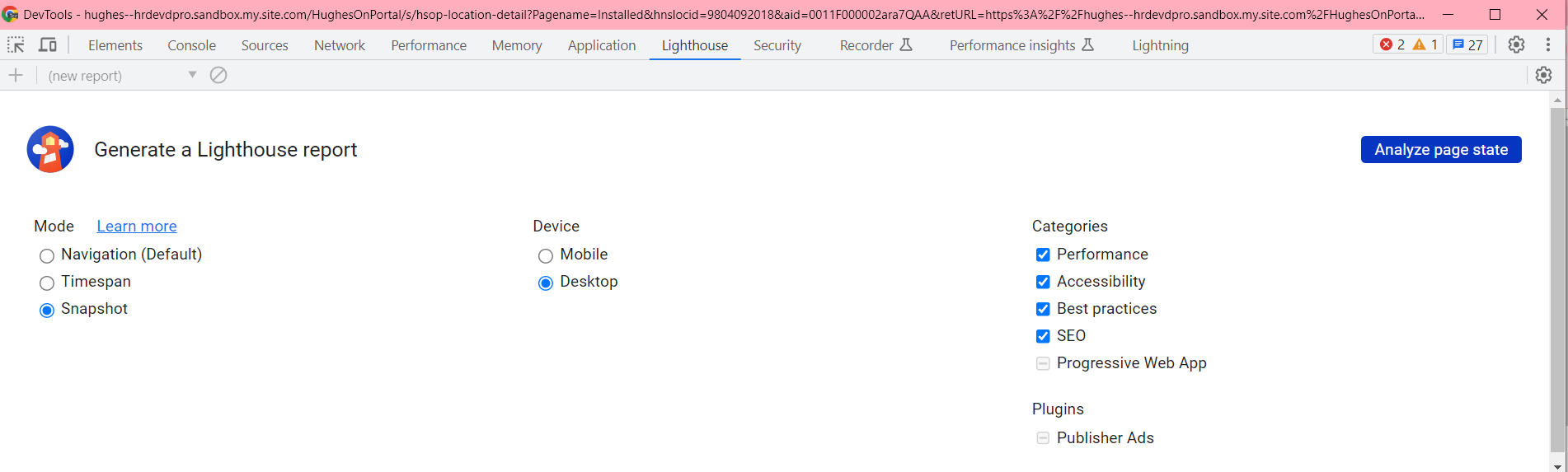
Step4: If your method has huge functionality break it into small pieces and bind it into main method and if the same code, you’re using it in different places make it as a common method and use it in wherever it is required.

Step5: Now in HTML and CSS wise if you want to follow the best practices follow below process

* Now take any one page from our application lets take as location details page



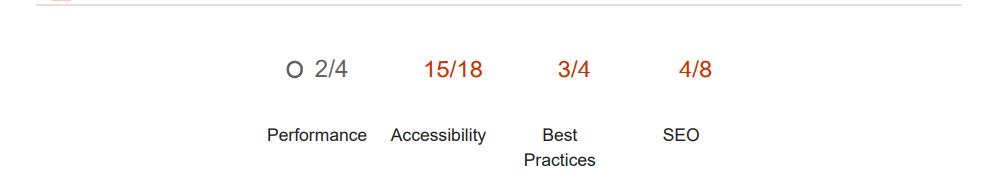
* Now install the lightening Inspect Element extension to your chrome browser. Now in inspect element click on **lighthouse**



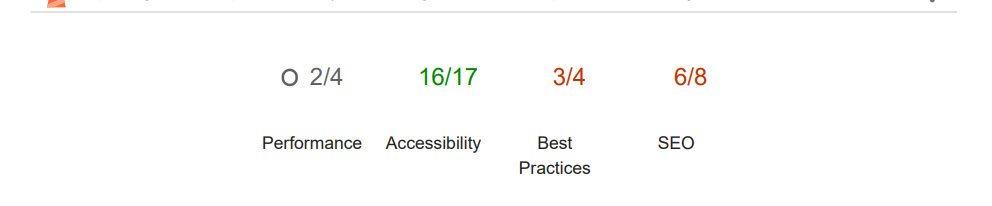
* Now select snapshot mode and check all the checkbox and click Analyze page state and check the report

Step6: My previous and current report for the location details page load changes

Previous



After Changes

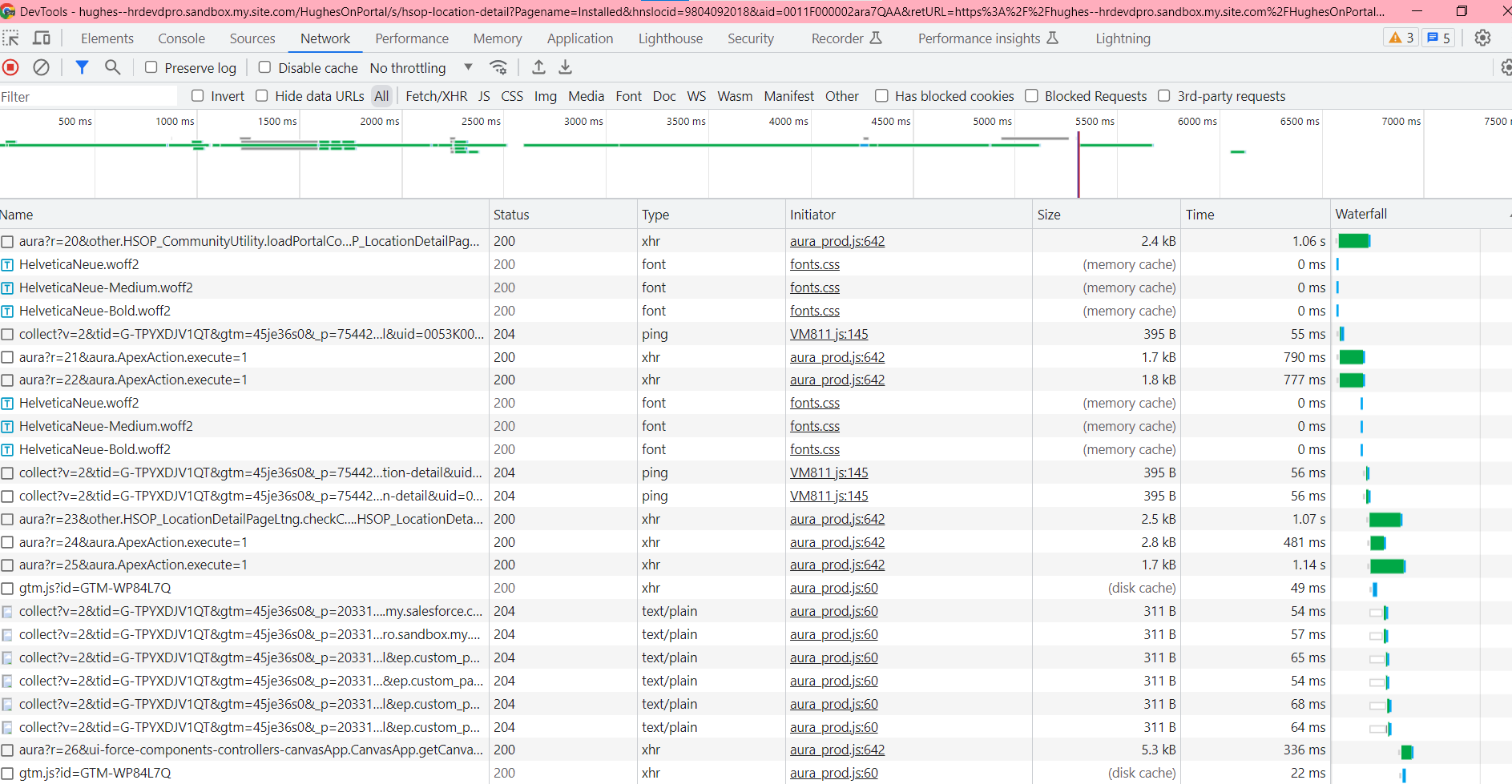


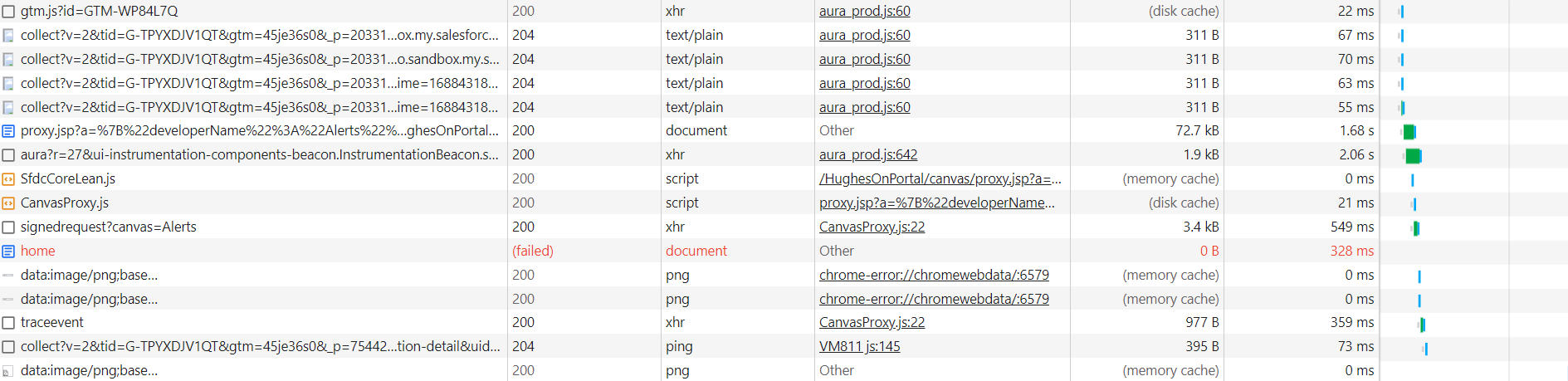
Made changes for Accessibility and SEO rest it is coming from the common components like header, left menu and footer component so warnings we can ignore only we can fix like errors and critical same as clayton issues.

Also, you can download that result as a PDF format based on the instructions, we can do modifications in your component.

This is how we can increase the page load related issues in each page of our application

Step 7: From the same inspect element dev tools check for the network tab for API calls not to use duplicate call





If any duplicate API calls comes remove those duplication and make it a single call so that it will reduce the time load complexity of your page.

And remove unused methods if its no longer not needed you can remove those methods

Example 1:

import { loadScript, loadStyle } from "lightning/platformResourceLoader";

we have imported this but no longer we are not using this in js file so these kinds of things we can remove from the component level it will increase your page load

Example 2:   
  
connectedCallBack(){  
 // here your calling API directly from LWC lifecycle hook

getResult({}).then({})  
}

But still you’re calling API from the below method and not using it any where

getDetails(){

getResult({}).then({})

}  
So, avoid these kinds of issues from the component level