Criticisms and Action Taken

Front-End

Criticism	Action
“You need an about page to teach the user how to use the site.”	Instead of adding an about page, info popups were added to the New Job form to explain what each field means. I think this is enough, since the users of the site are going to have a vague idea of what all the options mean.
“I would like to see more comments...”	Many comments were added to the code.
“I did not see unit tests.”	We determined that unit testing was not worth our time, although this is something we may explore between the code freeze and the expo, just for our clients.
“The presenters said that they are waiting for a new ruleset from their employer, so it seems like once that is handled the requirements will be met.”	New ruleset was given to us by our client and fully implemented.
“It’s possible that refactoring could be done, because there are some very long functions.”	Some code refactoring was done to make functions more efficient.
“I am sure there are possible ways to improve the code, but I think the current react code is efficient enough that the marginal benefit of rewriting it is not worth the extra time.”	Some code refactoring was done to make functions more efficient. Some of the more complicated functions were not rewritten because of time constraints.
“I did not see any unit tests for the react code. However, I am not a strong believer in unit tests for react code as I don’t really see the benefit in it, so I wouldn’t recommend the team spend time on it anyways.”	Unit tests were not added for the UI. This is something we considered doing for our clients after the code freeze, but I did not think it was worth the time to do it before then.
“In JobResults.js, in the fetchJob function, it uses both Async/Await and .then(). I know that’s really picky, but it really stuck out to me.”	Attempted to fix this by only using Async/Await or .then(), but it only works with both. Without Async/Await, the react code moves along before the function can finish. Without .then(), the API call moves along before it finishes. I decided to leave the code as-is, since it doesn’t really hurt anything to use both Async/Await and .then().

Back-End

Criticism	Action
The names of variables and classes seemed very explicit in what they meant. I would like to see more comments, especially in the more complicated parts of the backend code.	More comments added for both back-end servers.
I saw some sections of the code labeled “old”. I’m assuming the comment label means these bits of code are no longer useful, so they should be taken out.	Old code removed after the code clean up.
I did not see unit tests. During the presentation Postman was used and I saw a lot of tests, but there did not seem to be any structure among them. I would recommend exporting a collection of Postman tests and storing it on the repo for other people to use to test the program.	Sample Postman test case added in the ReadMe file.
The presenters said that they are waiting for a new ruleset from their employer, so it seems like once that is handled the requirements will be met.	New ruleset file updated
I could not find any obvious abstractions, or a way to write functionally equivalent code. Its possible that refactoring could be done, because there are some very long functions.	Helper functions added to reduce the duplicated code.
Setting up tomcat was a challenge. I know I need a tomcat file servelet. The ReadMe for the RulesEngine states I must “move all the Json and YML files to that folder to serve it to the server”. What files must I copy into the /static folder for tomcat? That is the part I am stuck on.	Update the instruction to make it more clear.