For the programming languages, we chose to write the entire project in Javascript. In the back-end, we used the relational database, postgresql, to store information to the project. In the frontend, we used react.js to build the pages, and react-bootstrap for the overall styling of the project.

For the first iteration, the tasks we completed were mainly the highest priority tasks designated in the Prioritization List. The tasks were mostly for posting, displaying, and removing listings. For this iteration, the goal was mainly to get the system on its feet with the basics working properly so that we could expand upon it in the next two iterations.

For the second iteration, we added the rest of the functional requirements that were high priority. These functionalities included the ability for an admin to log in and checking the login credentials in the back end. It also added a few features regarding reviews including the admin’s ability to edit and remove reviews. We also added pages for employers which dynamically rendered information about each employer.

For iteration three, we added the ability for the admin and employers to edit and remove employer profiles. Employers can only do this to their own profile, while the admin can do it to any. We also expanded on the reviews system by allowing users to flag them for the admin to take a look at. We implemented a page which contained all flagged reviews, which the admin can then unflag or delete them. The admin can also delete reviews by looking at the individual employer profiles as well. Additionally, we created an account creation system which requires an email address for the username and a password that contains at least eight characters, one capital letter, one number, and one special character. Contact information was also added as a required field for posting jobs.

While we were able to get a great deal of functionality into our final product, we found that email verification for new accounts and email -based notification for administrators was ultimately something that wasn’t a priority for us in the first 3 iterations. While many of our group members had had experience in using React and Postgresql to create forms and populate databases, creating this email system would require expanding the applications we were presently using and devoting time to a function with limited usage. We did, however, meet 19 of 21 requirements we set out to accomplish for our final product, and as a result we were still satisfied with what we ended up with in our final product and furthermore believe that this functionality could definitely be added to the site in the future.

Security is an issue. We used cookies to set whether a user was an admin or employer for the login process, which is far from ideal. We also didn't include a way to verify email addresses when people sign up. Using a login platform like Google Firebase would have been a much better solution, but we were reluctant to try it in what would be the first time using React for nearly half of the group.If the project were expanded, it would be interesting to add the ability for students to log in, upload a resume and set what they are looking for in a job.