Group Members:
Esther Martens (esm108), Ashvi Shah (as2767), Vivian Trieu (vtt11)

Group Project: Learn You a Language

We worked on a web app project that uses flask, html, css, racket, haskell, and python. Our project is a web application tool that helps people learn more about paradigms and different types of languages. We included descriptions and explanations for each of the paradigms and gave examples of some of the languages that correspond to those paradigms. In another section of the web app, we also introduce all of the languages we are working with and explain their history and relevance. This further helps people understand the languages and paradigms, as well as gives them context. For people who have not coded, or are not as exposed to different types of languages, the "Meet the Languages" page is a great way to familiarize yourself with the language and also learn where these languages are being used and for what purpose.

After introducing the paradigms and languages, we delved into applying the languages by writing data structures. We were able to write fully functioning programs for stacks, queues, linked lists, and binary search trees. We chose not to include arrays because it is not exactly a data structure, and short of creating a class for it with different types of data types inside, we can only show how to traverse it, which is fairly basic. Instead of spending time on that, we decided to focus on making the code for the other structures functioning and error free. Our goal however, was to write all of those as well as, hash tables and heaps. We were unable to finish writing these two additional structures because making flask work took longer than we expected. In addition to this, finding a good host also took time because a lot of websites only support static web app hosting, and would not work with our webapp. We ended up finding heroku last minute and had to learn how to utilize it according to our needs. The difficulty that game with heroku was that additional files was needed to be put into our project in order for it to work on heroku. This took time to figure out because of the lack of documentation there was out there on how to format the files. The files needed to be formatted in a very specific manner for heroku to work. Furthermore, none of the group members knew any UI, so we also had to learn how to write HTML and CSS. We made the decision that UI was just as important as the actual content of the web apps, so we cut our time that would have otherwise been dedicated to writing programs, to make the website look more visually appealing. One of the reasons we wanted to make a webapp was so we could learn more about UI/UX. So far, we have only learned back end programming at Rutgers, so this was a great opportunity to learn more about the front end.

On the pages that illustrate the use of each data structure in python, racket, and haskell respectively, we analyzed the Big O and the implementation of each data structure. This was an essential part of our project because by being able to analyze and determine the major differences of how to implement different data structures in each language helped deepen our understanding of each paradigm. By analyzing the Big O and algorithms, it emphasized how

each paradigm has its own pros and cons. As we had learned in class, programming languages do not all fall under just one paradigm. They can be under more than one, with each having its own uniqueness and attribute. Additionally, by coding the four different data structures in the three languages that we had learned in class exemplifies our understanding of each language because there is a very distinct difference in implementation. For example, in python we were able to utilize for loops as well as recursion to build the data structures. On the other hand, the racket and haskell required recursion because of its functional nature.

Despite our accomplishments, we had hoped to implemented embedded coding. This would have made the website much more interactive in that the user may have compiled the code on the web app rather than just reading it. Additionally, we would have liked to added in C and Java coding as well because it represents imperative and static typed languages. However, due to time constraint (such as other final projects, final exams, Vivian's car breaking down, etc.), we were unable to implement these features. Although we were not able to incorporate C or Java, we specifically focused on including python, racket, and haskell because these programming languages were taught in class. Overall, despite not being able to fully achieve all our set goals, what we accomplished is something we are proud of with the zero experience we have in UI/UX development. We were able to overcome the varying challenges that came with learning new languages (HTML and CSS) and a new framework (Flask), but yet, enhancing out existing knowledge by practicing python, racket, and haskell.