**Project Reflection:**

Throughout the completion of *Project Juice*, we used an agile approach to software development. This methodology allowed us to make the necessary changes to our goals and vision throughout the development process without sacrificing our ability to complete the final product. When we first envisioned our product, we had initially wanted to look to not only drinks, but food as well. However, we quickly realized that we would be better able to provide a cohesive experience for the user in the given timeframe if we focused solely on drink recipes. We had smaller pivots throughout the process as well. These mostly took place as a result of our time constraints and previous lack of experience. Since none of our four members had any experience with web development, we needed to spend much of our later sprints learning how to complete the very basics of our user and functional requirements. This required us to make certain changes to our overall vision. For example, we decided that implementing a way for users to input and suggest their own custom recipes would take too many of our resources away from our ability to complete the css necessary to make our website aesthetically pleasing. We determined that this was more significant to the project at large and decided to move away from crowd-sourced recipes. In this move, we also moved from crowd-sourcing ratings for the different recipes in our database. This pivot was made to free up development time for other areas as implementing a rating scheme would have necessitated a complete restructuring of our database scheme.

GitHub was used in two capacities. One was web hosting yet the major one was code management. We effectively used GitHub as best as we needed. We uploaded all of our code as it was completed and used it as our central repository for all things *Project Juice*. Yet, we did not successfully use all of its functionality. Both for the sake of simplicity and because there was never a clear need, we only ever worked in the single master branch. This was not significant as the group was small enough to avoid any merging or overwriting conflicts but had the project been any bigger, it would have been a very useful feature to use.

Amazon Web Services and MySQL workbench were used to work with our database. There is not a lot to say about these tools. We were very successful in using them. We hosted our database using AWS’ SQL database with php integration and used MySQL workbench to directly link to it.

During this project, we officially used Slack as our primary means of communication. However, this was not our actual primary means of communication. Since the group was four people and we never assigned specific, completely disconnected roles to any single member, we primarily used group SMS messaging for a large portion of the project. One of Slack’s main features for group communication is the ability to have multiple channels of discussion happening instantly and simultaneously. This is incredibly useful if certain team members do not need to see the technical discussion of another team. However, since everyone on our group was working on such similar project aspects at the same time, there was no advantage of using slack over any other form of communication. As a general-use communication platform, we did not use it to its fullest potential. Though we were unsuccessful in this regard, we did successfully use it to distribute significant technical information. This took the form of us posting Linux shell commands to properly set up the php localhost server that was used for the first part of our front-end development. It also took the form of posting links to various resources containing information about whatever the team needed to learn at that time.

One tool that we did use successfully was Trello. In the early stages of the project, we were somewhat effective in our use of this tool. We used it to lay out what needed to be done by the group. However, no group members used all the features properly. We failed to move the cards to the appropriate lists when they were either in progress or completed. We also failed to use the “Change Members” feature to clearly show which member was completing each card. As the project went on, we slowly got better at properly using these features though doing so earlier would have contributed to our overall efficiency. Beyond this, we eventually came to a usage scheme for Trello that greatly improved our team’s work efficiency and clarity. For our last two sprints, we implemented a leveling system for task management. This meant that each card would have a number associated with it in its title. This allowed us to clearly display dependencies that needed to be completed before moving on to future tasks. For example, in the first sprint we had four different levels. The “Level 0” task was downloading MySQL workbench and using it to communicate with our database hosted on AWS. The was a Level 0 task as it did not directly contribute to the completion of the project so much as it enabled us to work on it. A Level 1 task from this sprint was implementing a login system. This needed to be completed before we could complete the Level 2 task of the 21+ age gate for *After Dark*. All the Level 2 tasks needed to be completed before we could complete the Level 3 task of compiling the pages into a single website and so on. This was a very useful way of using this website as it allowed us to clearly see how well we were progressing through the requirements of our project.

We are very happy with how *Project Juice* turned out by the end of the semester. Due to the small size of our group and our previous inexperience in web-development, we are proud of what we accomplished. Through much hard work, we managed to create a functional product.

**Project Report:**

Project Juice is fully functioning and ready for use. Of the initial goals for the website, the majority were included in the final product. Project Juice is a drink recipe website targeted for college students and young adults. It allows users to create accounts, search recipes on the Project Juice Classic, or search for alcoholic beverage recipes on After Dark, provided the users are 21+. The implementation of Project Juice utilized various development technologies including SQL, MySQL Workbench, PHP, HTML, CSS, and Bootstrap. An SQL database was used to create the website’s back-end. The database included a table for user information, including login, password, and birth date, a drink recipe table for the Project Juice Classic, and a separate alcohol drink recipe table for After Dark.

The website’s front-end was developed using PHP and HTML along with MySQL queries to collect data from the website’s database. The main web pages included the register user functionality, login and logout functionality, and the ability to search for drink recipes on both the Classic and After Dark versions of the website.

While the initial functioning website was made up of PHP and HTML, it lacked an original design or consistent styling on the various web pages. To create an attractive, streamlined website, Bootstrap was applied to the existing PHP files. Bootstrap provided many templates for interactive website development which were highly customizable. Project Juice user relied on a generic template for the website layout, but significant changes were made to the template, particularly the graphics and color scheme. The resulting website has a contemporary design which is appealing to target users such as college students. While most pages included the same layout made for Project Juice Classic, separate graphics and color scheme were designed for the After Dark recipe page. The After Dark recipe page had a darker color scheme and graphics that emulated actual “after dark” signs.

Accomplishments:

Implemented Agile software development to deliver constant improvements to the project

Learned syntax for MySQL queries

Created unique graphics for the website

Connected SQL database back-end to front-end via PHP and MySQL

Created an extensive drink recipe database

Implemented user login and logout functionality

Implemented user registration functionality

Learned website development as none of our team members had done any prior web development

Integrated Bootstrap into website’s PHP and HTML files

Hosted website using Amazon Web Services and Github.io

Outstanding Issues:

Lack of sessioning throughout website – only some pages reflect when a user is logged in

Future goals:

Implement rating system where users can rate recipes and save their favorite recipes

Add additional recipes and include graphics for each recipe

Allow for sorting recipes by difficulty and number of ingredients