The Cleaning Ladies

CI/CD integration plan

Continuous Integration is the practice of consistently and frequently updating, testing, and developing software.

Our team plans to integrate this practice by pushing our code to Github. The first thing a team member will do is go to the tutorial linked in Github and complete said tutorial. This will ensure that each member of the team is familiar with the code structure and how to make successful changes and run the project. All team members must be familiar with the code, and able to run the code on their devices. Any code changes, however minor, should be pushed to Github as soon as they are done. We will do this in small fragments. Small code pushes will ensure that any errors are negligible and easy to fix. This will also ensure that the team members are holding themselves accountable and to a higher standard. When pushes are not frequent and are too large, they may harm the other team members. Code between members may not be compatible and some code may also be dependent on other code. All code will be pulled and run whenever entering a coding environment. Whenever a team member begins coding on a given day, that team member will pull all Github code and run the software on their device. Sometimes when code is not pulled, a developer may overwrite the existing code, this will ensure that the issue does not occur. Running the code when pulled and every time a member is developing will ensure that the member is educated about the state of the code. This will help the team and code stay homogenous. The team will check the hosted site at least once a week. The team will also make sure that all code is running and working online. All code pushed to GitHub will be commented on and clear, assuring that anyone who seeks to review the code will be able to understand it.

Continuous delivery automates the testing environment and consistently allows testers, developers, and users to be able to use and refer to changes made.

The team will launch the project on a hosting site like Azure, AWS, or Google. This will let any user, tester, or developer maintain a working website. This will also help show the developers what the code will look like and how it will perform in real-time on a physical platform. The team will update the code on a public GitHub repository. The code will be updated frequently and in small fragments. The code will have detailed comments. Frequently updated and commented code will help any testers and developers clearly understand what is going on. It will also help testers be able to use and run the code. The GitHub will have a tutorial on how to run the code on a local environment, this will allow any testers and also the developers to not only be able to use the site on a hosted platform but also on a local platform.

The architecture for the product includes Github, .NET, and a hosting environment. Developers will need to configure .NET and Github on their local machines. Using an IDE such as Visual Studio, developers will use the command line to install .NET. The IDE will also have Github support, where developers will push and pull code with frequency. This will allow developers to run and edit the code on their local machines.

Users will most often use the online hosted version of the website. However, users will also be able to pull code from GitHub and run that on a local machine.