|  |
| --- |
| Implementation Issues |
| MyNewMedia |
| Zachary Merrill, Stephanie Koesters, and John Mark De Los Reyes |

We used GitHub to download and upload files of code and documentations to our local computer and to the website. GitHub was able to upload files nicely to the website, but when it came to downloading files from GitHub, it resulted in a lot of issues because sometimes it didn't fully download everything from the file. With that, it resulted in a lot of frustration because it took a while to know which file(s) in the folder were missing. There were two cloning issues that resulted in a lot of frustration to the MyNewMedia team. The first problem was one of our group members didn't get a couple of files after he cloned it to his local PC. Another problem was that cloning the folder of files to our local PC didn’t always work, it then forced the group members to download the zip file. Even if you downloaded the zip file, there is a chance that a couple of files will be missing.

We had to download multiple third party plugins and there were several issues with coding when we installed them to make them work with our website. Since we had to use our command prompt to install them in our local computers, sometimes it didn’t install them properly, and sometimes it wouldn’t even install them at all. We were not able to get the full functionality of the website unless we downloaded all the needed Django applications. It took more time for team members to get started coding because we had to communicate with the person who added the Django application to see how he/she did it. There were a lot of tutorials for these applications, but that didn’t help much, and sometimes it actually made it worse because we installed something that we shouldn’t have. For example, one of the team members was trying to install a Django application called Django-taggit, and the problem was that the tutorial instructed to install a folder that he didn't need. This resulted in multiple errors to the code, but once he deleted the folders that he didn't need, the errors went away. Even though our group had trouble with installing Django applications, there were some packages that were outdated which resulted in us not being able to use them.

Setting up Django was very difficult because it was free and there was a lot to install, such as Python, an Apache server, and so on. Since there was a lot to do, it took the team a long time before they could start writing code to make the website work. Similar to installing Django applications, some applications didn't install in our local PC's nicely, others couldn't install to our local PC's at all. This resulted in a lot of time wasted because all the work we did before the other installations couldn't be put into use.

Lastly, reading other people's code took a lot of time because each person in our group thinks differently. Since we only meet twice a week, we had to email each other using our Gmail or by Facebook to understand specific lines of code or to get an idea on how to code something up in our website. With that being said, since most of the group members knew HTML and CSS, those languages were not a big issue. It was the backend part of the code that gave most of the group members trouble because programming in Python was new to everyone.