When first tasked with this assignment we had to make sure we had a clear idea of what we were trying to accomplish before we chose any stacks to use. The user stories played a huge part in helping us accomplish this task but what helped the most was discussing during the lab what exactly we wanted our application to do. Once we had a clear picture of what we wanted, we needed to decide what stacks would be best for integration and easiest for implementation. It ultimately came down to using Node.js which meant we would be writing javascript and Flask which was in python.

Because the python language has simple syntax and a multitude of libraries that are easy to follow and implement, python/flask became one of our two choices for the stack we would end up using. Flask is considered among many to be the best framework for simple web applications which is what we were doing in this case and with its extensive documentation it would be easier for us as beginners to use. We would have more control over how we want things to be done in the application since most of the framework is open to change. The only drawbacks to flask is the scaling of it. With the way flask handles requests it can only handle one at a time and because we are doing multiple requests at once for our application, it would end up resulting in a possibly slower response which would negatively affect the user experience, something we don’t want to do. While doing research we found out that there are workarounds for this however, from our understanding they increase the risk of breaches in security.

Javascript is the language we were all least familiar with but we knew for this project that at the very least one of us would have to learn it. Because we would have to learn javascript, that would mean that we would need to get familiar with the Node.js stack. Node.js is a great middleware that makes it extremely easy to handle API’s. With Node.js it is possible to share and reuse code with ease. Another big pro for Node.js is the number of free tools that are available for it, making the handling of data even easier for us. In addition to this Node.js can handle requests without delay unlike Flask, making the performance better. The only possible problem we face with Node.js is the issue of bottleneck. While this can easily be worked around by the use of worker threads, that relies on people having multiple cores which not everyone has. Thankfully though we should not have to worry about that since we are doing a pretty simple application.

In conclusion, both of these frameworks are very good but we feel that using Javascript and Node.js would work the best for our project. There are two main reasons why we chose Javascript/Node.js over Python/Flask. The first reason is because as stated above we have to learn javascript for the front end so we thought it would make sense if we implemented javascript on the back end as well, making sure that they integrate properly together. The second reason is because of everyone’s experience when it comes to web development. Everyone in our group either has limited experience in web development or none at all. Because of this, we thought Node.js would be a better starter since this past tuesday the professor demonstrated how this would work in class on the back end, giving us some sort of starting place with this project.