**Frontend Technology:**

***Angular***

**Angular** was developed as a frontend JavaScript framework in 2010 by Google designed to make single page applications.

**Angular Pros:**

* One of the most popular web frameworks currently
* Feature rich with numerous libraries, templates, and testing utilities at a developer’s disposal Two-way Data binding to transfer changes in the Model to the View
* Ideal for large scale applications
* Used by many renowned companies and has frequent updates every six months
* Supports native, hybrid and web app development

**Angular Cons:**

* Steep Learning Curve as a familiarity in Typescript is needed as well as time to integrate with the large volumes of libraries
* Slow performance for heavy applications due to the large amount of features
* Mixed reviews from developers who have used this framework

***React***

**React** was launched in 2013 by Facebook, designed to divide the UI into components that would build into a single page application.

**React Pros:**

* The most popular web framework currently and for the near future
* Virtual Dom allows for more efficiency for updating the view of a webapp
* We already have some experience in React
* Frequent updates by the Facebook team
* Supports cross-platform development

**React Cons:**

* Due to the dynamic nature of react, large scale applications may not be ideal for react
* Required to learn JSX and other Quirks of the framework

***Vue***

**Vue** was released by a former google engineer and is one of the newer frameworks. Since its release, it has steadily gained popularity.

**Vue Pros:**

* Virtual Dom allows for more efficiency for changing UI states
* Straightforward Learning Curve
* Quick deployment of application with high performance, easy development
* Newer framework that is gaining in popularity
* Most highly rated framework on Github (Stars)

**Vue Cons:**

* No prior experience
* Not ideal for more heavy weight applications due to lack of libraries
* Small community of developers

**The frontend technology we decided to use is React.**

Due to the limited time for this assignment, one of the most important criteria for us was ease of development. Since we are still new web developers, we also wanted a popular language so there would be more resources to learn from. We chose React because of the experience we have with the framework, its large community, and the fact it is ideal for a lightweight application. A feature-rich framework like Angular would not be needed for a smaller project like this.

Vue was also heavily considered due to its high ratings, it being light weight and relatively easy to pick up. Angular though, was one we were not high on due to the tougher learning curve, its lower reviews by developers and it being more suitable for larger scale projects. Since all four choices are frontend oriented, there is not much difference in the domains covered by each framework.

**Backend Technology:**

***Node.js***

**Node.js** is a runtime environment based on Chrome’s V8 JavaScript engine introduced in 2009.

**Node.js Pros:**

* Has recently become a very popular web backend
* JavaScript Full stack development allows:
  + Easy transition for frontend developers to work on backend
  + Code sharing and reuse
  + Developer productivity, efficiency
  + Superior speed and performance compared to other backend technologies
* Prior experience in Node.js backend
* Npm, the default Node.js package manager provides close to a million available libraries

**Node.js Cons:**

* Slow processing when it comes to heavy computation due to single-threaded nature of JS
* Open source nature causes some tools in npm to lack quality control

***Python***

**Python** is one of the world’s most popular languages many top companies include the use of python in their technology stacks.

**Python Pros:**

* Numerous frameworks and libraries to help with web development
* Easy learning curve: syntax is pseudo code like
* Versatile as it can be used for webapps, desktop apps, ML, DevOps and more
* Django, a python framework works well with many well-known databases
* Extremely popular, used in almost every major company

**Python Cons:**

* Lack of microprocessor support resulting in code limitations
* Not good for memory intensive tasks due to its dynamically typed nature
* Less experienced developers in its community than others

***PHP***

**PHP** is an open-source, server-side scripting language developer in 1994 for the web. Around 80% of all websites are built around PHP.

**PHP Pros:**

* Massive community with lots of frameworks, documentation, and other resources
* Works efficiently with HTML, servers, and databases
* Faster webpages compared to other technologies
* Low barrier to entry means a relatively easy learning curve

**PHP Cons:**

* Declining popularity
* Lack of libraries that support modern needs such as ML
* Security flaws when used by novice programmers
* No prior experience

**The backend technology we decided to use is Node.js**

In choosing this backend technology, we as with the frontend wanted something that we were familiar with and that reduced the choices to python or node.js. We also wanted a technology that would still be widely used in the future and due to PHP’s older libraries and declining popularity, we did not think it was the right choice. Finally, what made us choose Node.js over Python was because the research we did showed that integrating a Node backend with a React frontend was a more seamless and efficient option compared to a Python backend.

**CI/CD:**

**Database:**