**Technology Requirements**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Database** | **Programming languages** | **Framework** | **Backend** | **Server** |
| MongoDB | JavaScript | React JS | Node.js | Express |

|  |  |  |  |
| --- | --- | --- | --- |
| **Technology** | **PROS** | **CONS** | **Why it fits our project** |
| **MongoDB** | -High speed and easy processing  -Simplicity of use  -Flexibility | -Hard to join relational databases  -Limited data size  -High memory usage on system | Mongo DB is our choice because of its flexibility and ease of use. We do not require a complicated database system so this will be fine for our use cases. |
| **React JS** | -Dynamic web applications  -Simple set up and ease of use  -Great performance  -Good library support | -Implementing views  -Many updates and changes | React JS was chosen because of its good performance and cross platform compatibility. It is a good framework that we are all familiar using. |
| **Node.JS** | -High efficiency and speed  -Many free tools and mature ecosystem  -Can be scaled easily | -Can have performance bottlenecks  -weak library  -bad API | Node JS was chosen because of its compatibility with the other systems of our program. |
| **Express** | -Very scalable  -Another JS environment  -Easy integration and good compatibility | -Issues with certain callback functionality | Express JS was chosen because it is a good server side framework for Node JS. Also good for communicating with third party resources. |
| **JavaScript** | -Very versatile language  -Dynamic application design  -Very cross compatible  -Familiarity for the team | -Malware and security concerns | Javascript is the primary language used for this program. All the parts are compatible with javascript so that development can be simplified. |