**Choosing a Technology Stack**

Why MEAN?

Everyday MEAN is increasingly adopted by many companies. MEAN uses JavaScript throughout, and JSON Objects simplify communication between backend and frontend. Only having to use one language will simplify the programming we need to do. It will also make debugging easier because the objects stored in the database are identical to the objects the client-side JavaScript deal with. Also Node.js has a large module library with thousands of packages available for download. AngularJS simplifies client side logic. Express makes it simple to write an API.

Having experience with MEAN will be the most likely framework to help us in our future careers. If we decide to work more on this project and expand it, the scalability will be helpful. Node.js supports multithreading. Using Node.js can improve the performance of the application, as Node.js is entirely non-blocking and event-based, allowing for true concurrency among requests.

Using the MEAN stack provides us with the opportunity to make Mean Girl jokes.

Why not LAMP?

LAMP is not as well integrated as MEAN. It will be harder to debug when the language used changes from frontend to backend. We also don’t have time to learn several brand new formats/languages/frameworks. People in our group have experience with JavaScript, JSON files, and MongoDB, but not with PHP or Apache. LAMP is also limited to running on Linux, while MEAN can also run in OSX and Windows. MySQL, the db in the LAMP stack, is a relational database. These databases are more complex to set up, and we will not require the structure that a relational database provides. In our case, it would also be easier to perform server-side logic with a NoSQL database.