Our team was considering between Django and Express.js. After some experience with both, we decided to continue using our Django prototype. We considered Django and Express.js because of our group’s familiarity with Python and Node. Initially, using Express to set up the back end routes was simple, straightforward, and easy to understand. However, we had some trouble understanding how to add packages and deal with package-lock.Json and the node-modules. Using React for the front end required getting around the CORS policy (cross origin resource sharing), which had to be done through setting a particular header onto the Express server. Although Express has countless plugins available, figuring out how to use some of them was challenging, and not all the issues we ran into had solutions.

Compared to Express.js, Django was generally easier to learn. It has a “battery included” philosophy so we would not have to worry about functionality and infrastructure at the moment. This also allows us to take care of security, URL routing, templates, and database all within Django. Because of our group’s familiarity with Django, we were able to get a successful API call through the framework. Our next API call would be for a Boston Streetlight Locations database, and accessing that database through Django will be easier than through Express. In general, our progress using Django to develop our web application was greater than our progress using Express.js. As our team continues this project, we can continue to access the documentation we used to successfully complete our Django prototype to develop the rest of our project.