Jüri Kiin
Arctic Fox
Rich Media Web Application Development II

Remember when you were young, and calling 5's on a seat was not enough? Remember when you waited for exactly 5 minutes to take your friend's seat only for them to come back and say they called Arctic Fox? This site is exactly that: Save your spot forever. With it's simple UI, ArcticFox lets you remember places in the world that you need for later use. In 3 clicks, you can save a location forever, and even come back and edit it later. You can even see places that other people saved!

It's API handles loading, adding, and updating places. The API also handles the filtering of the places, letting you sort them by date and name. This will let you find recent places, or even help you locate places based on a certain name.

Very little in this project went not as planned. A lot of the driving content in my project is the Google Maps API I'm working with. I have a lot of prior experience working with the API, so setting up and manipulating it for my needs was very easy to set up. I also often spend a lot of time getting CSS to look good, however I've developed a CSS framework that I used for this project. This let my style the site rapidly. This sped up the amount of time I needed to develop on the front end, leaving me more time to focus on functionality.

The thing that took longer than expected was setting up my PUT requests. I was able to get the server-side functionality built fine, however the FE tasks were more difficult than imagined. It took a bit longer to understand how to handle the PUT requests in my .then() functions as they had no content in the response, but I managed to figure that out. I also had trouble deciding how I would update the FE to account for changes in a PUT request. I was thinking about giving each item a reference to the whole place object, but felt that it was too much. I ended up deciding to gray the list out to indicate that it needed refreshing, and game an update button that let the user fetch places again.

If I were to continue with this app, I would definitely add some more things. Firstly, I would transition the data storage to a database so it would persist. I would also allow for user registration, so you can store places you've made on that account. In that vein, I would let the user make public and private places. They might want to save places that they don't want people to know about (Where they put buried treasure!). This would also let the user be able to quickly find places that they saved without having to search through everyone's places. I would also add some sort of notification to the page that let it know to update the list (versus forcing the user to do it themselves). I could have done that on a setInterval, but felt that if data isn't changing, don't keep making server requests. I would also add functionality to delete certain places. I understand the app is to save things forever, but if someone doesn't need to save it anymore, they should be able to delete it. Another FE feature I would add is the ability to add a custom map marker image, that could more personally separate your places from ones around it. It would also be cool to see places and be able to see who added them.

There are several ways I went above any beyond for this project. On the FE, I used my own personal CSS framework to style the site. I used the google maps API in order to help collect data, and set up markers and info windows within that API. On the server side, I included 2 more ways to filter the main get request. I added sorting by name & date. This seemed like a good idea anyway, but it was more than the 1 required parameter. These parameters also work together so you can limit the size of the request, as well as pricing a date or name filter.

I did use some external resources:

Generating a Global User ID: http://stackoverflow.com/questions/105034/how-to-create-a-guid-uuid-in-javascript

Sorting an array by date: https://stackoverflow.com/questions/10123953/how-to-sort-an-array-by-a-date-property

Toast CSS taken from: https://www.w3schools.com/howto/howto_js_snackbar.asp