Michael Young

Project Intro

My site is a basic homepage that was built on Facebook's React.js library. The idea is that you can get relevant information to your life every time you open your browser. I am unaware of any bugs that affect specific browsers, one of the issues I am aware of is a size limit on local storage. I haven't run into any issues, I've tested images up to 6mb, but I'm unsure what will happen if its too large for local storage. I am also aware that of an issue with choosing the weather form a city, certain (abnormal) queries can take you out of the US search limit which will display weather "correctly," but be from a city name but no state name. This is because there is no state name in the other country. I'm also aware of an issue where if the user input has a word-character in front of the intended string (i.e. "A Key Largo" when "Key Largo" was intended), it returns either "Invalid Input" or results from another location ("A Key Largo" happens to return a weather station in Northern Norway by the name of "A"). To test the site, you can simply open the index.html file. As long as you have an internet connection there are 4 sections that can be tested.

Customizations

Pretty much the entire site is custom beyond class examples. Using React.js deprecated functions such as innerHTML (which is dangerouslySetInnerHTML in React.js). For this project I had to learn to create a webpack server, map out all my dependencies and dev-dependencies, and figure out what loaders I need to produce a plug and play system. Needless to say, I've spent a lot of time working on the behind the scenes of this site. It's been a seriously rewarding, and frustrating, process because things don't work like I expect them to. I actually spent hours trying to figure out how to use webpack's file-loader so I could set the background image in my css file instead of through javascript.

Very little of what I ended up using was explicitly covered in class, but everything I used was based on concepts learned in class and labs. I had to use arrays, strings, loops, if statements and functions that relate to each to produce my site. One of the more interesting things I did was to use a forEach loop to produce an array of custom html elements that are then output to create the list of news articles.

Design, UI/UX, Site Content

Basically all of my content comes from API's. The date is produced through javascript's date function and then I use some substring functions to return the pieces of the date string I want and then output them through React states. The tough part is outputting the API data into a user friendly and interesting site.

There is user input only for the city the user wants to pull weather from, it is validated by weather underground's API, although I'm also validating it with RegEx and the trim function to make sure only a-z and spaces are sent to weather underground as part of the request.

Dynamic HTML/Interactivity

My site allows users to go to the settings and change various aspects of their experience such as the metric, city, and state used for the weather. The user can also "upload" their own background image which is stored and pulled from local storage. The user must interact with the site in order to see the information they want by changing settings or clicking on a news article to show that article's snippet. To change settings the user must click the settings "button" in the upper right corner of the page which then shows options to customize the background and weather module.

Michael Young

JSON API

I'm connecting with the weather underground API and the NY Times Top Stories API. For the Weather Underground API I'm passing in user input city, state, and my API key. For the NY Times Top Stories API I pass in my API key and desired section which I have hard-coded. The weather underground API can be customized to any US city with a wunderground weather station which makes it possible for the user to see the weather in their city. The NY Times API brings the up-to-date stories from the NY Times and provides the title, thumbnail, snippet, and a link to the full article. This is relevant to the user because its important to keep up-to-date with news. These API's provide me with the basic information necessary to bring value to the user.

Extras

I used a timer to call the time function repeatedly because I wanted the time to be real-time and not be stuck on the time when the page was loaded. I used local storage to make the custom-background persist across reloads and visits. If I had known how easy local storage was when I built the weather module, I would have implemented a local storage aspect that saved the desired location to persist as well. I also used RegEx to validate the requested city for weather. I look for all non a-z and _ characters and take them out as well as trim leading and tailing whitespace. Finally, the big one, I used Facebook's React.js to build my entire site, which was quite the learning experience and allowed me to realize how much I have left to learn about web development.