MEET Up

Objective

Create a serverless web application, with React using a test driven development approach. While using Google Calendar API to fetch upcoming events.

Goal

Allow users to easily search for events in different cities around the world. Make the app available for use offline and allow the user to save it to their home screen.

Tools Used

- React
- CSS
- AWS Lambda
- Google Calendar API
- Jest
- Puppeteer

The Process

1. Serverless Functions & Authentication
Allowed me concentrate more on the code and less on the infrastructure because the cloud provider handled the deployment.

2. Google Calendar API
Access Google's database of events using the Calendar AAPI.

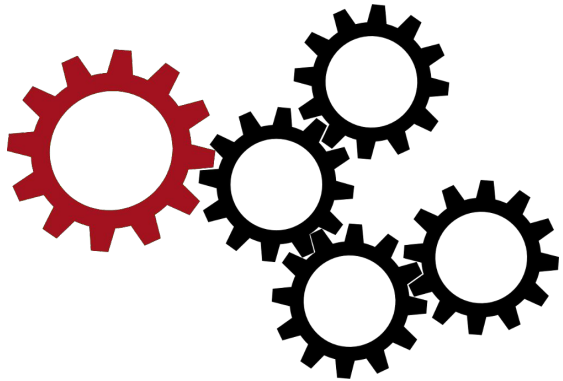
3. Unit & Integration Test
Ensures the code has the essential parts to function properly.

4. Convert App into a PWA
Allows the app to be installed on both mobile devices and computers. As well as making it available offline.

5. Design with CSS
Create dynamic design.

Serverless Functions & Authentication

Serverless functions hosted on AWS Lambda were used to obtain access tokens from the authorization server (Google OAuth Provider). Once the access code was granted and a request was made to the protected resource (Google Calendar API) for information on events the events were then returned to the user.



Testing using Jest

Test Driven Development

Unit Testing

Using Jest, test was made for each unit (function) of code.

Integration Testing

This was implemented to ensure that each individual part of the app works with every other part.

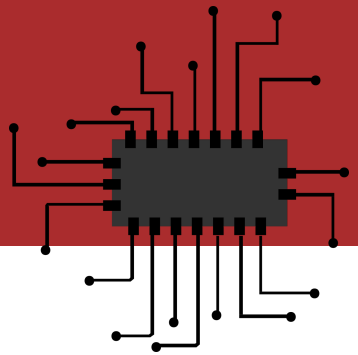
End to End Testing

Using Puppeteer end to end testing was created to test the entire application.

Converting the App into a PWA

(Progressive Web Application)

Service Worker & App Manifest



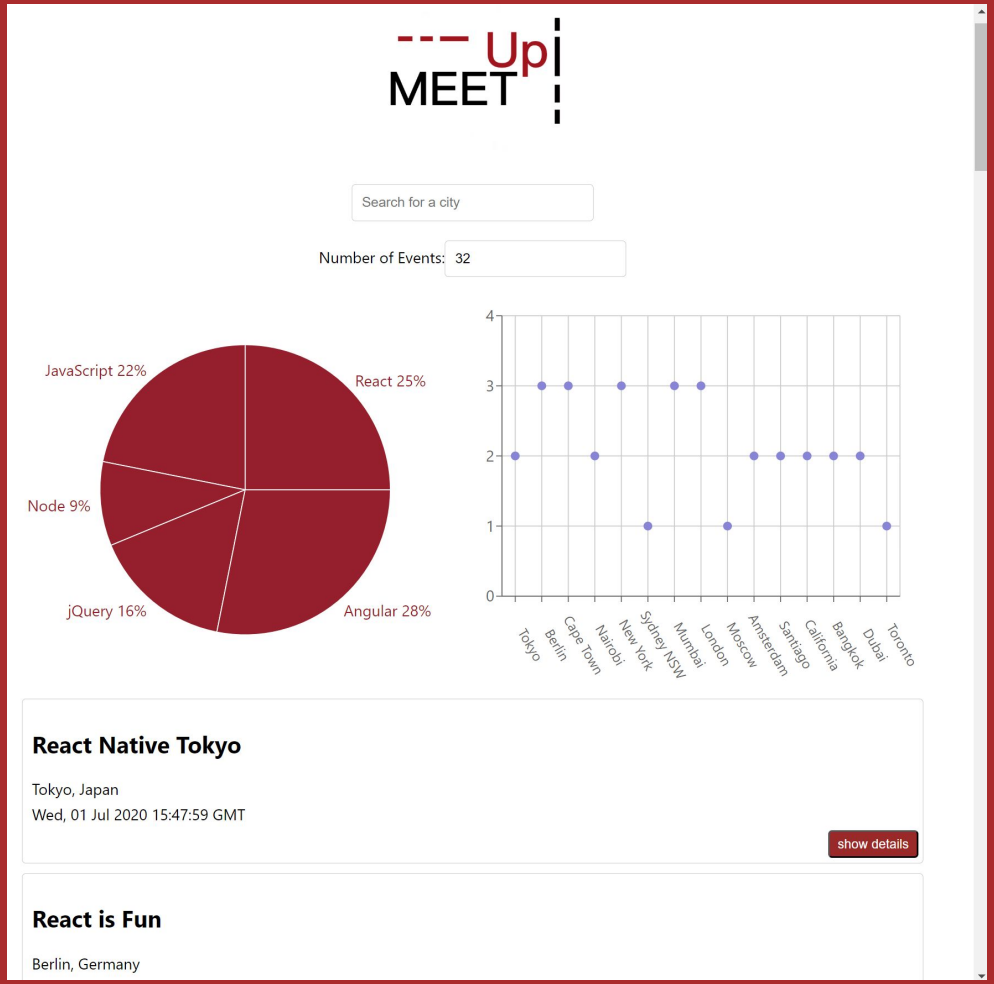
Step 1

Create a service worker. This allows the user to be able to use the app while offline because it loads data already stored in the cache instead of downloading it again from the browser.

Step 2

Create a web app manifest. This is a JSON file that defined how the app will be displayed to the user and is referenced by the PWA. It contains metadata that is used with the app is installed on the user's device. Such as the app's name, icon and the starting URL.

The Final Product



Retrospective

The goal was to create a Progressive Web Application which I was able to accomplish using AWS Lambda. Towards the end of the project data visualization features were also implemented into the applications UI, which I think added a nice touch. The most difficult part of this project was creating the unit test and ensuring they pass, but I was able to accomplish this with the help of my instructors. In the future I would like to add more color to my applications to create a more unique experience for the user.

Check out my App
https://darnobles.github.io/Meet-Up/