MEET APP

A FULL STACK CASE STUDY

Jamie Tracy

OVERVIEW

This serverless, progressive web application (PWA) was built with React using a test-driven development (TDD) technique. The application uses the Google Calendar API to fetch upcoming events. Users can login using their Google information and search for relevant events in their area.

CHECK IT OUT

OBJECTIVE

This project was part of the the full-stack immersion course from the web development program at CareerFoundry to demonstrate my knowledge and skills of PWAs. The goal was to create a serverless progressive web application with authentication and authorization of users to showcase in my portfolio.

DETAILS

Roles

Role: Lead Developer
Tutor: Ebere Iweala
Mentor: Mahesh Rodrigo

Duration

This project took me nearly 8 weeks to complete from start to finish.

Methodologies

- React
- React Bootstrap
- React Testing Library
- Jest
- Puppeteer
- AWS

THE PROCESS

Getting Started

I started a Create React App and wrote user scenarios using the Gherkin “given, when, then” syntax.

Setting things up

I connected my app with a Google Calendar API for events and set up OAuth authorization/authentication for users.

Going serverless

I set up the serverless toolkit through AWS and and created and deployed AWS Lambda serverless functions in my code.

Testing

I conducted unit tests and integration tests for React components and app features using the React testing library and Jest.

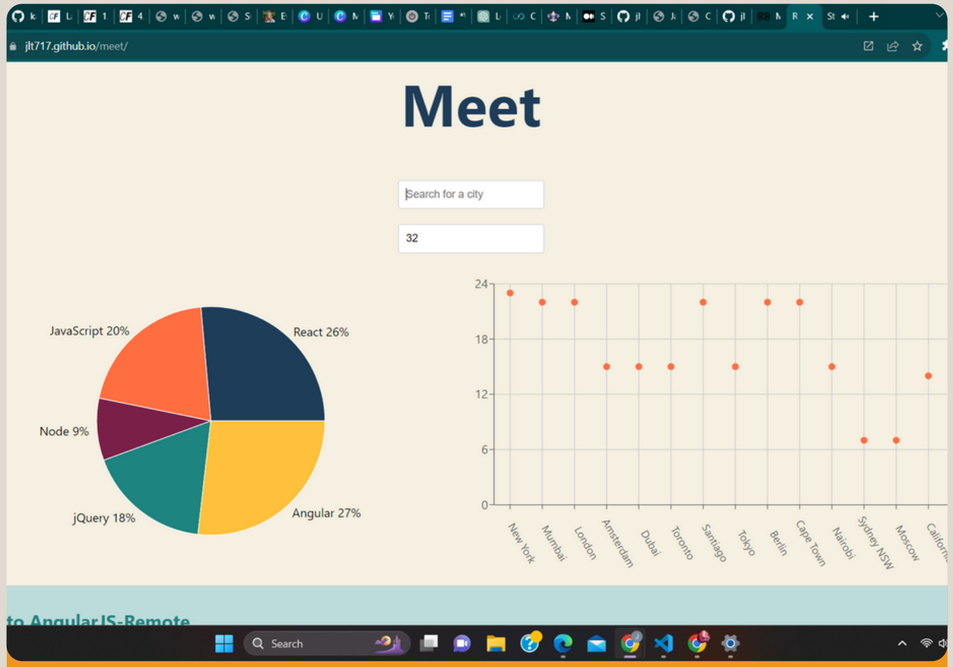
Continued Testing

I then conducted end-to-end testing using Puppeteer.

App Performance

I followed up by using Atatus to track overall app performance and Lighthouse to convert my app to a PWA.

SCREENSHOTS



Home Page

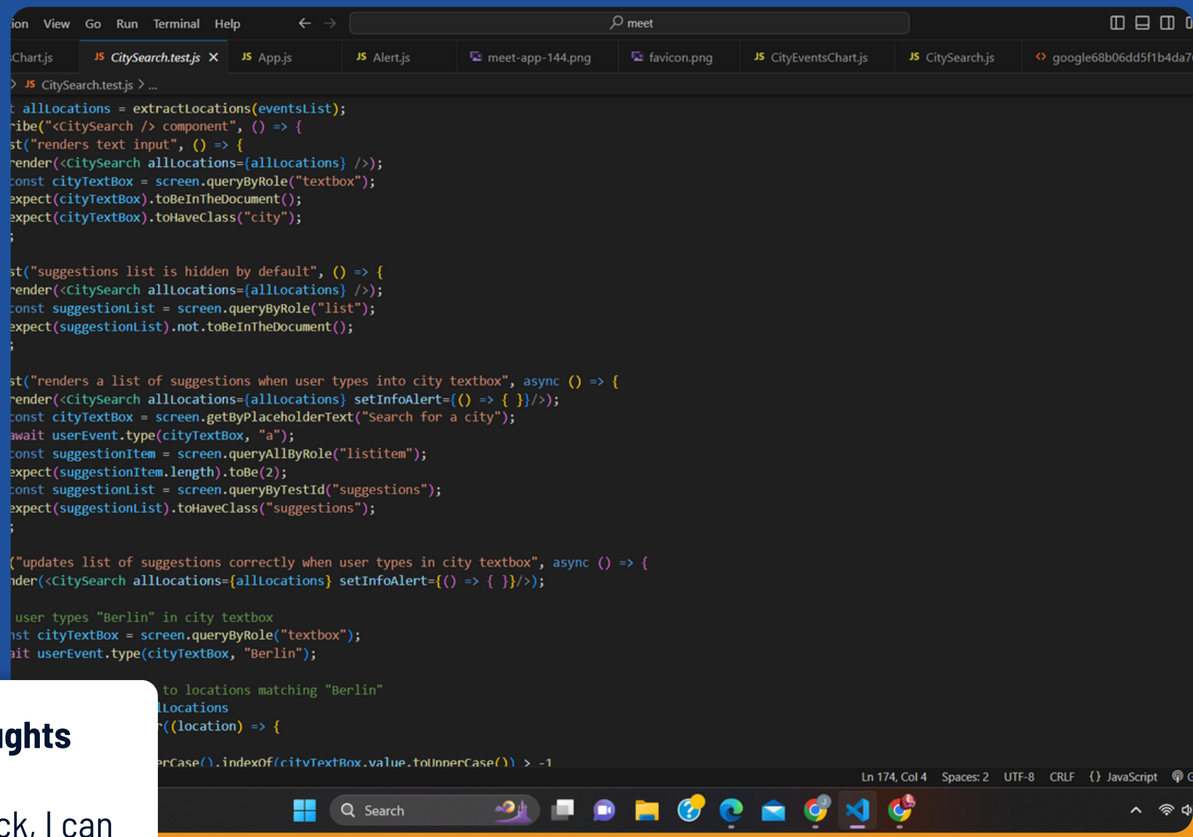
Home screen with pie chart and scatter plot showing events by subject and by location as well as search feature.

Event Details

Each event can be clicked to reveal details about location, time, and event summary.

RETROSPECTIVE

This project challenged me and pushed me to persevere and try to break down and understand new concepts.



What Went Well

I enjoyed focusing on user scenarios and functionality in this project. It was very informative and user-focused.

Challenges

I struggled with reinterpreting code into different scenarios for various testing. This project was among the most frustrating I've worked on.

Final Thoughts

Looking back, I can see that I should've spent more time researching and practicing each type of testing. I did, however, after the fact.