Case Study: Meet App

Overview:
Meet App is a serverless, progressive web application (PWA) built with React using a test-driven development (TDD). The app uses the Google Calendar API to fetch upcoming events. It provides users access to different tech events and workshops around the world.

Purpose & Context:
Meet App was a project I built as part of my full stack web development program at CareerFoundry to demonstrate my mastery of test-driven development (TDD) and behavior-driven development (BDD).

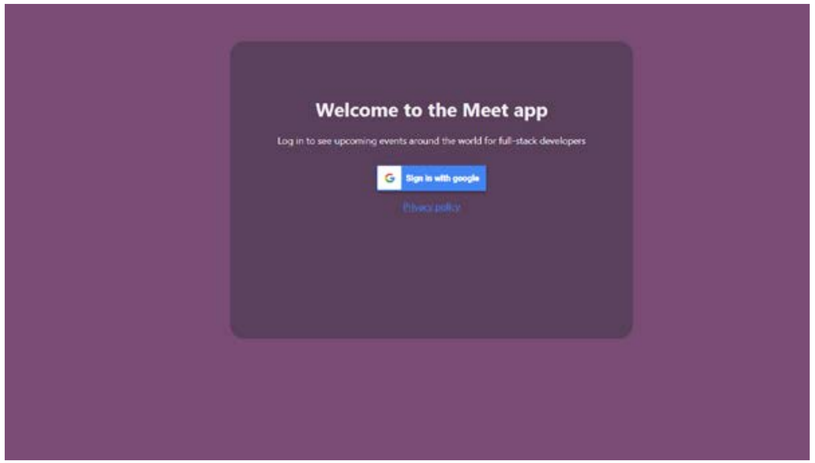
Objective:
The goal of the project was to develop a progressive web app that can work offline and has a serverless backend using a TDD approach.

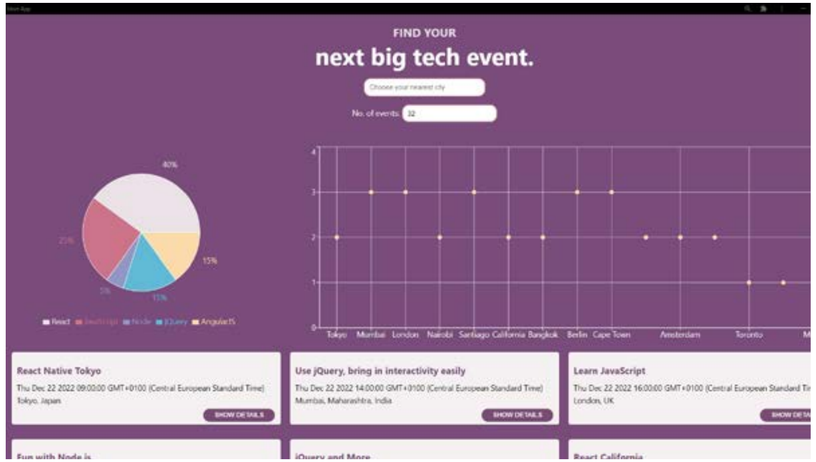
Approach:
I used AWS Lambda and the Serverless Toolkit to set up an authorization server that allows users of the app to connect to the Google Calendar API via an access token.

Based on the defined scenarios, I created unit and integration tests for the features and tested how the components of the app worked individually and as a whole. I then ran end-to-end tests using the Pupeeteer tool and reviewed the user flow throughout the app.

I implemented Web App Manifest and Service Worker to turn the app into a PWA, and used Lighthouse to evaluate its performance and qualifications. Finally, I used Recharts, a React-based visualization library, to add a pie chart and scatter plot to the app.

Challenges:
One of the biggest challenges I faced on this project was writing tests before implementing the code, as TDD was an approach I wasn't familiar with. So I had to spend a lot of time learning new tools, from Jest and Enzyme for unit and integration testing, to Puppeteer for end-to-end testing, to Cucumber for user acceptance testing. There is one thing I would do differently if I could, and that is to use the React testing library instead of Enzyme. This is because Enzyme is no longer actively supported in React-based applications.





Credits
Lead Developer: Minhaj Islam
Tutor: Blaise Bakundukize
Mentor: Stephen Barungi