

The Meet Application

01

Overview

The Meet App is a serverless, progressive web application built in React using test-driven development.

02

Purpose

The goal of this project was to build a strong understanding of serverless architecture and progressive web application development. Throughout each phase, we applied test-driven development (TDD) principles to ensure functionality, maintainability, and reliability.

Technologies

* **React**

* **AWS Lambda**

* **Recharts built on D3**

* **Jest | jest-cucumber
Puppeteer**

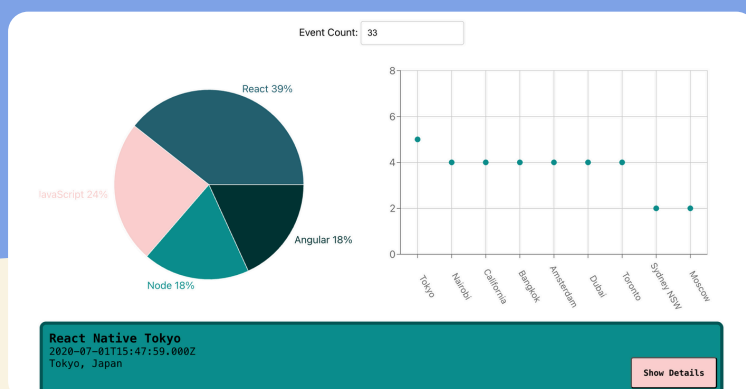
03

Objective

The application enables users to access and interact with it both online and offline. Users can select a city to view a list of upcoming events in that location, explore detailed information for each event, filter the number of events displayed, and visualize key event data within the selected city.

Project Developer

I served as the primary developer for this project, applying key concepts provided through CareerFoundry's curriculum.



The Meet Application

Continued

04

Serverless Deployment

This application was built using a Function-as-a-Service (FaaS) architecture to support serverless deployment. I found this approach especially beneficial for smaller applications, as it allows for fast and efficient deployment, automatic scalability, and cost-effective resource management—charging only for what is used. Implementing AWS Lambda provided valuable hands-on experience with cloud infrastructure, as it handled the underlying processes required to scale the application dynamically.

Setting up the Lambda function to trigger specific events and activate cloud services deepened my understanding of how serverless systems operate. One challenge I encountered was the “cold start” delay that occurs after periods of inactivity, which highlighted the importance of optimizing function performance. Overall, this experience helped me appreciate the advantages of serverless deployment and inspired me to continue learning more about scalable cloud solutions.

05

Data Visualization

In this project, I had the opportunity to work with Recharts, a library built on D3.js, to visualize event data retrieved from the Google Calendar API. I’ve been eager to explore data visualization ever since hearing a podcast that highlighted its potential to communicate complex information in intuitive and meaningful ways. The primary goal of this feature was to present both quantitative and qualitative insights through clear, engaging visuals.

Guided by my curriculum, I used Recharts to streamline the process—leveraging its templated components to efficiently build out charts without starting entirely from scratch in D3. Incorporating these visualizations added depth and interactivity to the application. I found that creating dynamic data charts brings a sense of playfulness to the user experience, much like exploring an interactive art exhibit powered by data.

The Meet Application

Continued

06

Test-Driven Development

A key focus of this application's curriculum was test-driven development (TDD). We began by writing user stories using the Gherkin "Given, When, Then" syntax, which provided clear direction for designing and structuring tests. Unit testing with Jest came first, allowing specific parts of the code to be isolated and simulated to confirm functionality and prevent potential issues early on. Next, integration testing ensured that the application returned expected results when multiple elements interacted together.

We also implemented user acceptance testing (UAT) with jest-cucumber, enabling stakeholders to view test outcomes in an easy-to-read, Gherkin-style format and identify which user stories failed. Finally, end-to-end testing with Puppeteer simulated real user interactions, verifying that all graphical elements and workflows performed as intended. Watching these automated tests replicate user behavior was both insightful and rewarding, as it showcased how the application performed in real-world scenarios.

Although testing can be time-intensive upfront, it plays a vital role in maintaining software quality. This experience taught me how structured testing not only prevents bugs but also supports scalability and long-term stability as applications grow.

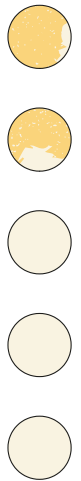

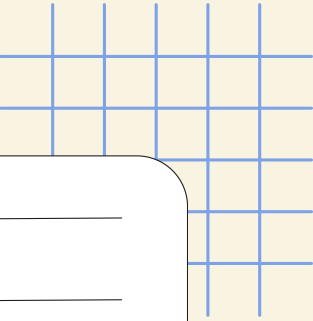
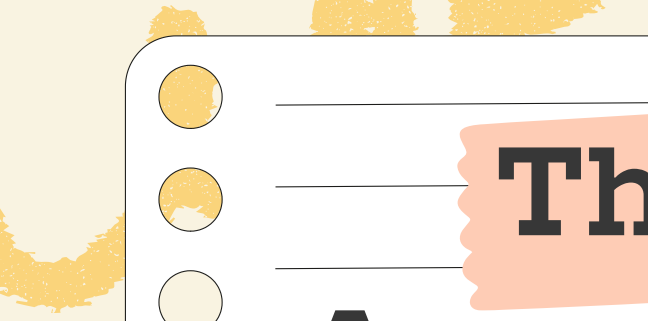

07

OAuth

This Meet application implements Google OAuth to authenticate users and provide access to event data from the Google Calendar API. OAuth works by verifying a user's identity and assigning a unique key and secret associated with their credentials. This process allows secure authentication without requiring clients to share their login information directly with the server. Once verified, OAuth issues a token that enables users to make authorized requests to the Google API.

While OAuth requires some initial configuration, it greatly simplifies the authentication process by eliminating the need to build custom login logic within the React application.

Throughout this course, I've had the opportunity to experiment with various authentication methods, and I found OAuth particularly effective due to its seamless integration and reliance on credentials many users already possess through their Google accounts.



The Meet Application



Final Thoughts



08

Overall



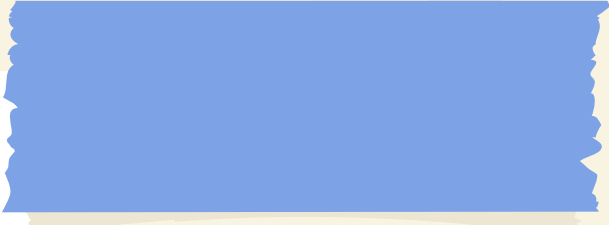
This project was an enjoyable and rewarding experience. I particularly appreciated integrating the Google Calendar API to dynamically render event data, as well as working with D3.js to create engaging data visualizations. I found the process of transforming data into visually appealing and user-friendly displays especially fulfilling. Overall, this project offered a diverse range of learning opportunities across multiple aspects of development.



09

Learning


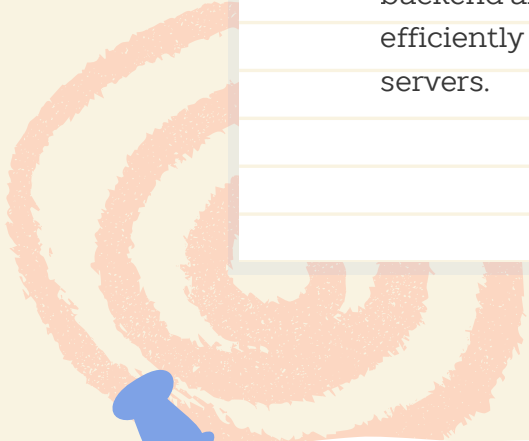

Writing tests is a critical aspect of application development, though it requires careful planning and strategy. I've learned that integrating testing throughout the development process is the most effective approach. As I continue to grow as a developer, I'm eager to strengthen my skills and explore different testing methodologies to ensure more reliable and maintainable code.



10

Challenges

Implementing AWS Lambda presented a valuable challenge, as I had no prior experience with the service. Through this process, I gained a deeper understanding of backend architecture—specifically how AWS Lambda can efficiently handle front-end requests and route them to servers.



The Future

Moving forward, I would like to enhance this application's mobile responsiveness. Currently, it is not fully optimized for mobile users, and improving responsive across devices would significantly enhance the user experience. Given its foundation as a Progressive Web Application (PWA), adapting it for mobile use would add substantial value and accessibility.

