



Case Study: *Meet App*

Caleb Dalrymple



Challenge

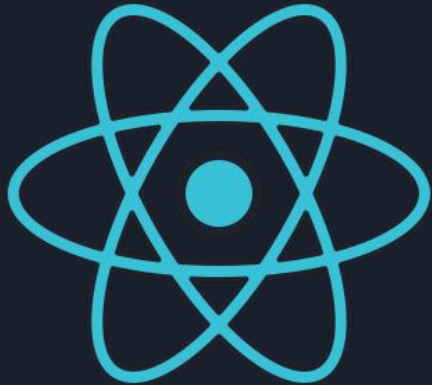
To build a serverless, progressive web application (PWA) with React using a test-driven development (TDD) technique. The application uses the Google Calendar API to fetch upcoming events.

[Click here to see the live site!](#)

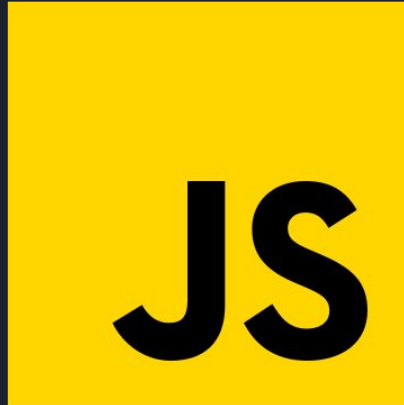


The Tech

React



JavaScript

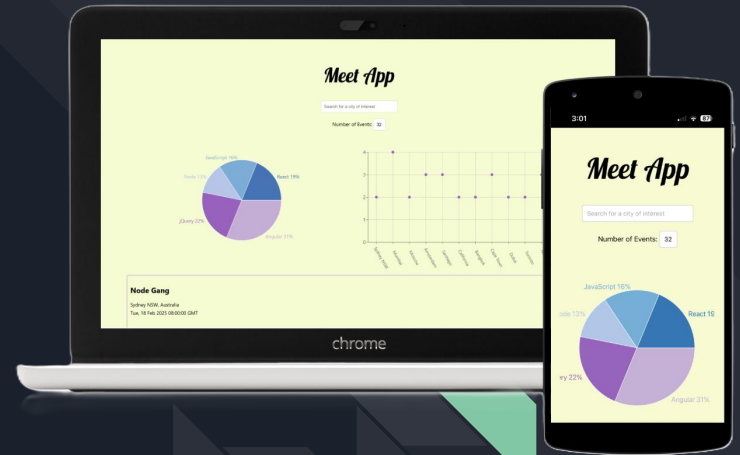


AWS Lambda



Meetup Like App

Meet App is essentially a way to find events happening in different cities. Users can choose a city of interest and the number of events to display. Using Recharts, I implemented a pie chart to show the percentage of events in each category and a scatter plot to show how many events are in each city.





PWA

I created a Progressive Web App (PWA) because I wanted to build a fast, reliable, and engaging experience that works seamlessly across devices. I was focused on providing users with a smooth experience regardless of their network conditions. Additionally, I appreciated the simplicity of maintaining a single codebase while still delivering an app-like experience without requiring users to install anything from an app store. Constructing a PWA aligned perfectly with my goal of creating a modern, responsive, and user-friendly application.



AWS Lambda

I used AWS Lambda functions to process the event data efficiently. When a user selects a city and the number of events to display, a Lambda function queries the google calendar API and retrieves the relevant information. These functions also handle the generation of data for the graphs, ensuring that the most up-to-date and accurate event data is used for visualization. AWS Lambda's serverless model allowed me to scale the backend easily without worrying about server management, while ensuring fast response times when generating the event data and graphs for the user.



TDD

The first time I used a Test-Driven Development (TDD) approach to building an app was both challenging and eye-opening. I started by writing a simple failing test, which felt counterintuitive at first, but it quickly helped me clarify the functionality I needed to implement. At times, I was tempted to jump ahead and write the implementation first, but sticking to the process forced me to think more critically about my design choices. By the end, I had a well-structured, fully tested feature.