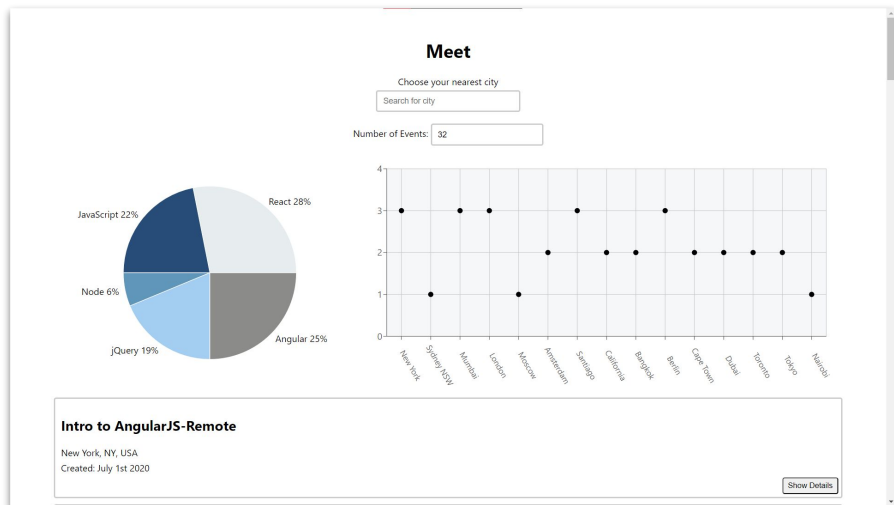


Case Study for Meet App



New York, NY, USA
Created: July 1st 2020
Developed by Google, AngularJS is a relatively new JavaScript, and it is designed to make front-end development as easy as possible for you. Join us to get introduced to this wonderful framework and dive deep into its features.

[Hide Details](#)

Node Gang
Sydney NSW, Australia
Created: July 1st 2020
We meet every Tuesday to talk about Node or JavaScript in general. Node Gang is an inclusive community that tries to cater to all levels of learners during meetups. Join us if you are curious to hear about what's new in Node, patterns, interesting npm packages, and practices.

[Hide Details](#)

Use jQuery, bring in interactivity easily
Mumbai, Maharashtra, India
Created: July 1st 2020
Do you know jQuery is used by around 70 percent of the 10 million most popular websites as of May 2019? Though many consider it dead after Angular and Express gained popularity, jQuery is still an important part of many websites. In our workshop, we teach basic to advanced jQuery where you will also be able to build a simple app using it. If you are familiar with JS, join us to learn probably its most popular library.

[Hide Details](#)

Angular Moscow
Moscow, Russia
Created: July 1st 2020
Developed by Google, AngularJS is a relatively new JavaScript, and it is designed to make front-end development as easy as possible for you. Join us to get introduced to this wonderful framework and dive deep into its features.

[Hide Details](#)

Overview

Meet app is a serverless PWA built with React using a test-driven development (TDD) approach. It provides movie details, lets users manage profiles and favorite films, and integrates the Google Calendar API for event tracking with interactive charts.

Purpose & Context

Developed for a CareerFoundry web development course, Meet app showcases full-stack JavaScript skills, API integration, and data visualization while following industry best practices.

Objective

The goal was to build a scalable, full-stack app with React, Recharts, and GitHub Pages, demonstrating expertise in modern web development for a professional portfolio.



Approach

Serverless

Using a Google Calendar API, I am able to pass that data into an easy to read list of events. This is a Behavior-driven development(BDD) test suite for a React app, verifying that event details toggle correctly when users interact with them. It uses Jest-Cucumber for structured testing and React Testing Library for rendering and interaction.

Jest-Cucumber

Key features include offline support, push notifications, responsive design, and cross-platform compatibility, ensuring a seamless and scalable user experience.

```
test("User should see a list of suggestions when they search for a city.", ({
  given,
  when,
  then,
}) => {
  let AppComponent;
  given("the main page is open", () => {
    AppComponent = render(<App />);
  });

  let CitySearchDOM;
  when("user starts typing in the city textbox", async () => {
    const user = userEvent.setup();
    const AppDOM = AppComponent.container.firstChild;
    CitySearchDOM = AppDOM.querySelector("#city-search");
    const citySearchInput = within(CitySearchDOM).queryByRole("textbox");
    await user.type(citySearchInput, "Berlin");
  });

  then(
    "the user should receive a list of cities (suggestions) that match what they've typed",
    async () => {
      const suggestionListItems =
        within(CitySearchDOM).queryAllByRole("listitem");
      expect(suggestionListItems).toHaveLength(2);
    }
  );
});
```

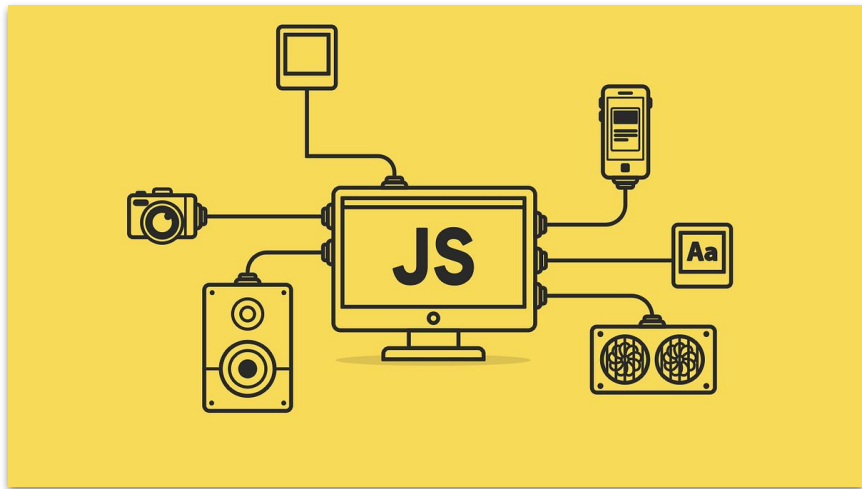
Serverless Functions



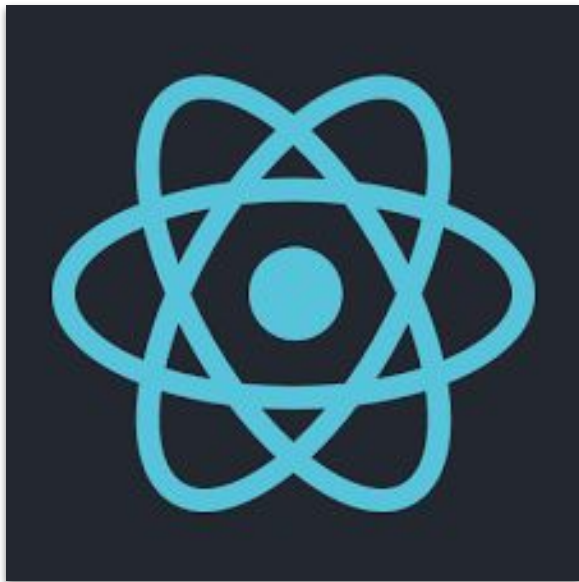
This application leverages serverless functions, specifically AWS Lambda, to handle backend operations. These functions authenticate users with the Google Calendar API and securely retrieve event data. By utilizing serverless technology, the app automatically scales to meet demand, reducing operational costs and ensuring high availability—without requiring manual server management.

Challenges

I found this project better to manage, as this was one of the later projects that I built, within Careerfoundry. I still learned a lot though. This project pushed me to master serverless functions with AWS Lambda, Google Calendar API authentication, and database structures. Building a PWA with React using TDD was challenging, especially implementing offline functionality and event filtering. Debugging frontend-backend interactions improved my API integration and serverless skills. Overcoming these challenges refined my React expertise and problem-solving abilities. I'm grateful for my mentor and tutor for their invaluable guidance.



Duration



This project spanned approximately 5 weeks to complete. I had a lot of fun with this project, especially with learning more about API's. I look forward to creating more serverless projects with the React framework.

Thank You For Viewing!

Would you like to connect with me?

[My Gmail: VincentAThorne2005@gmail.com](mailto:VincentAThorne2005@gmail.com)

