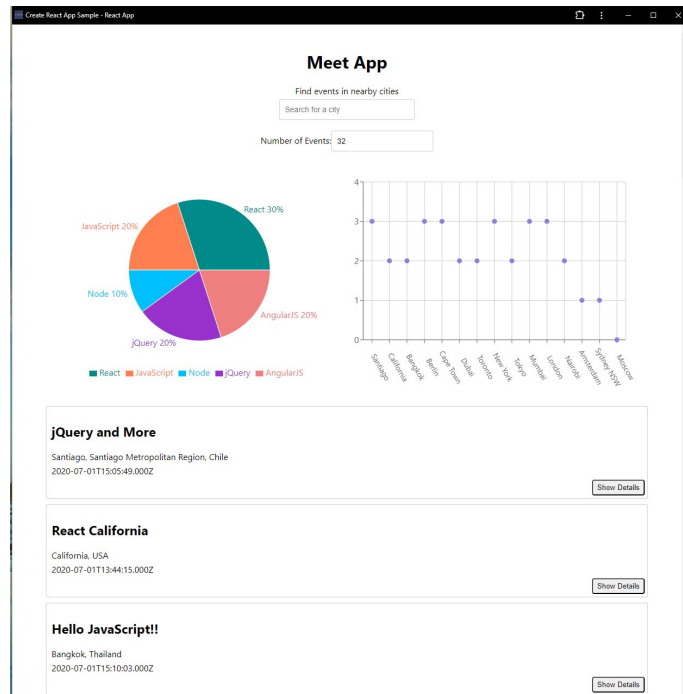

Meet App

A PWA Case Study
Stephanie Leon

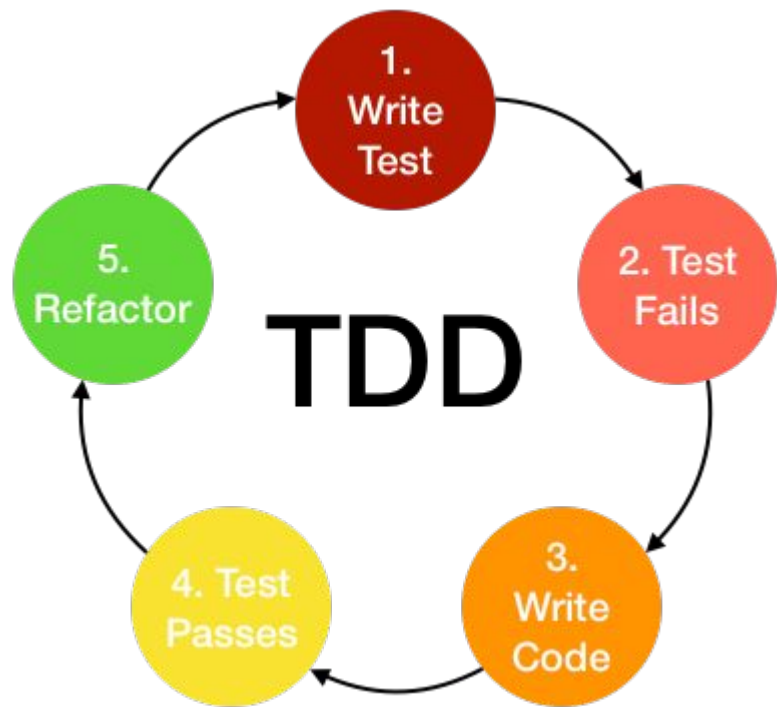
Overview

Meet is a serverless, progressive web application designed with React using a test-driven development technique. The app uses the Google Calendar API to fetch upcoming events.

Users can filter events by city, show/hide event details, specify the number of events shown, view charts visualizing event details, use the app when offline, and add an app shortcut to their home screen.



Purpose & Context



Meet was created as part of a full stack web development course I completed through CareerFoundry. This was my first opportunity to create an application specifically using a TDD approach.

Objective

The aim for this project was to combine two concepts that are growing in popularity for web development: serverless architecture and PWA.

I also practiced writing user stories based on the app's key features and writing unit tests, integration tests, acceptance tests, and doing end-to-end testing.

Feature 1: Filter events by city

User Story-

As a user,
I should be able to filter events by city
So that I can see a list of events taking place in that city.

SCENARIO 1: When user hasn't searched for a specific city, show upcoming events from all cities.

Given: user hasn't searched for any city;
When: the user opens the app;
Then: the user should see a list of upcoming events.

SCENARIO 2: User should see a list of suggestions when they search for a city.

Given: the main page is open;
When: user starts typing in the city textbox;
Then: the user should receive a list of cities (suggestions) that match what they've typed.

SCENARIO 3: User can select a city from the suggested list.

Given: user was typing "Berlin" in the city textbox AND the list of suggested cities is showing;
When: the user selects a city (e.g., "Berlin, Germany") from the list;
Then: their city should be changed to that city (i.e., "Berlin, Germany") AND the user should receive a list of events taking place in that city.

Development Process

Technologies and Tools

- React for building an interactive and dynamic user interface
- Bootstrap for responsive design and UI components
- PWA Support to allow the app to be installed as a shortcut on mobile devices
- Jest and Puppeteer for automated testing and end-to-end test coverage
- CI/CD Pipeline was implemented using GitHub Actions to automate testing, deployment, and ensure code quality. Every pull request triggered the pipeline to run unit tests, linting, and build processes before merging.

Development Process

Challenges:

- API Integration: Managing third-party API rate limits and data consistency
- Offline Access: Ensuring seamless functionality without an active internet connection
- Scalability: Designing for future growth and increased user traffic



Solutions:

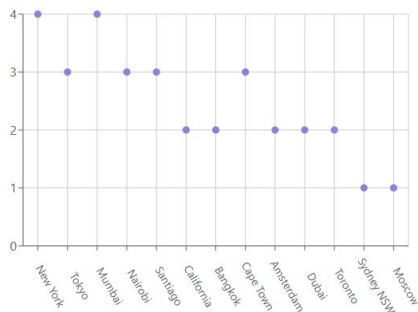
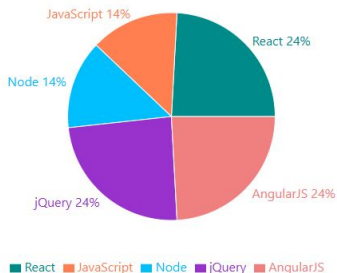
- Implemented caching and optimized API calls to reduce redundancy
- Used service workers to cache key data and improve accessibility
- Adopted modular code architecture and used robust libraries

Features

Meet App

Find events in nearby cities

Number of Events:



Intro to AngularJS-Remote

New York, NY, USA

2020-07-01T13:23:24.000Z

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.

[See details](#)

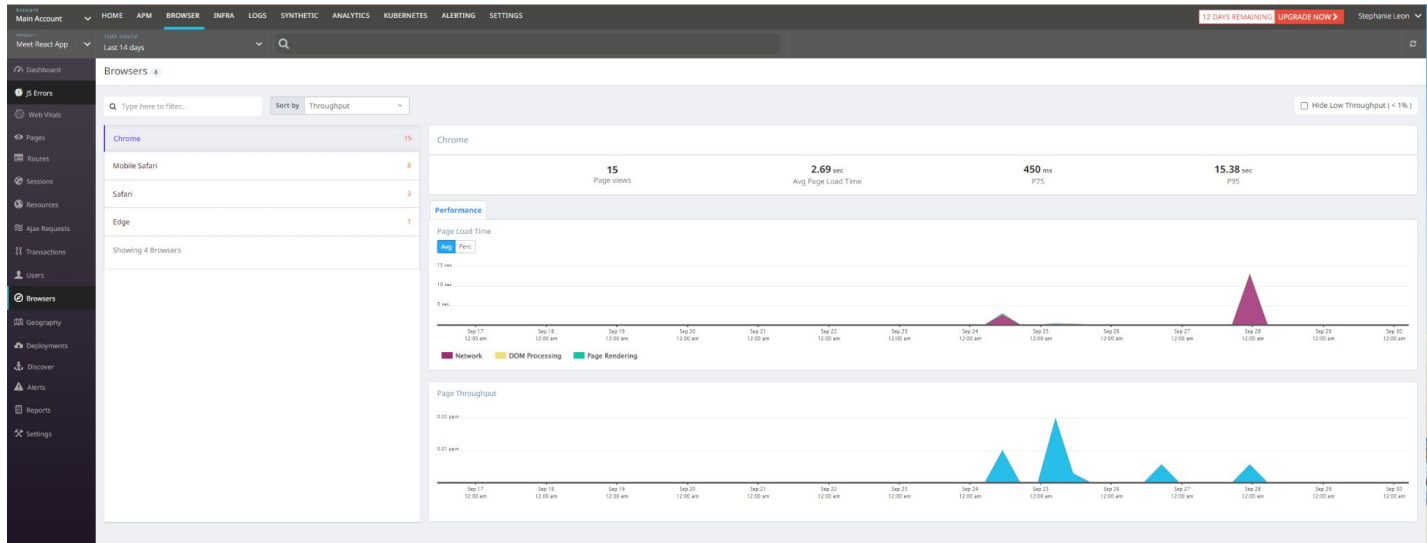
Hide Details

Event Search: Users can search for events by city or keyword

Responsive Design: Optimized for various screen sizes to ensure usability across devices

Event Details: Comprehensive event details, including data, time, location, and description

Features



Multiple Platforms: App must work on the latest version of Chrome, Safari, Firefox, Edge, etc

Offline Support: Users can view previously accessed events without an internet connection

CI/CD Automation: Ensures seamless delivery of updates and bug fixes

Testing Framework: A strong focus on testing guarantees reliability

Test-Driven Development

Unit Testing used to ensure individual components function as expected using Jest.

Integration Testing used to verify that multiple components work together seamlessly.

End-to-End Testing completed to simulate user interactions and to validate workflows.

Automated Testing done through the CI/CD pipeline to ensure each code commit to detect and address issues early.

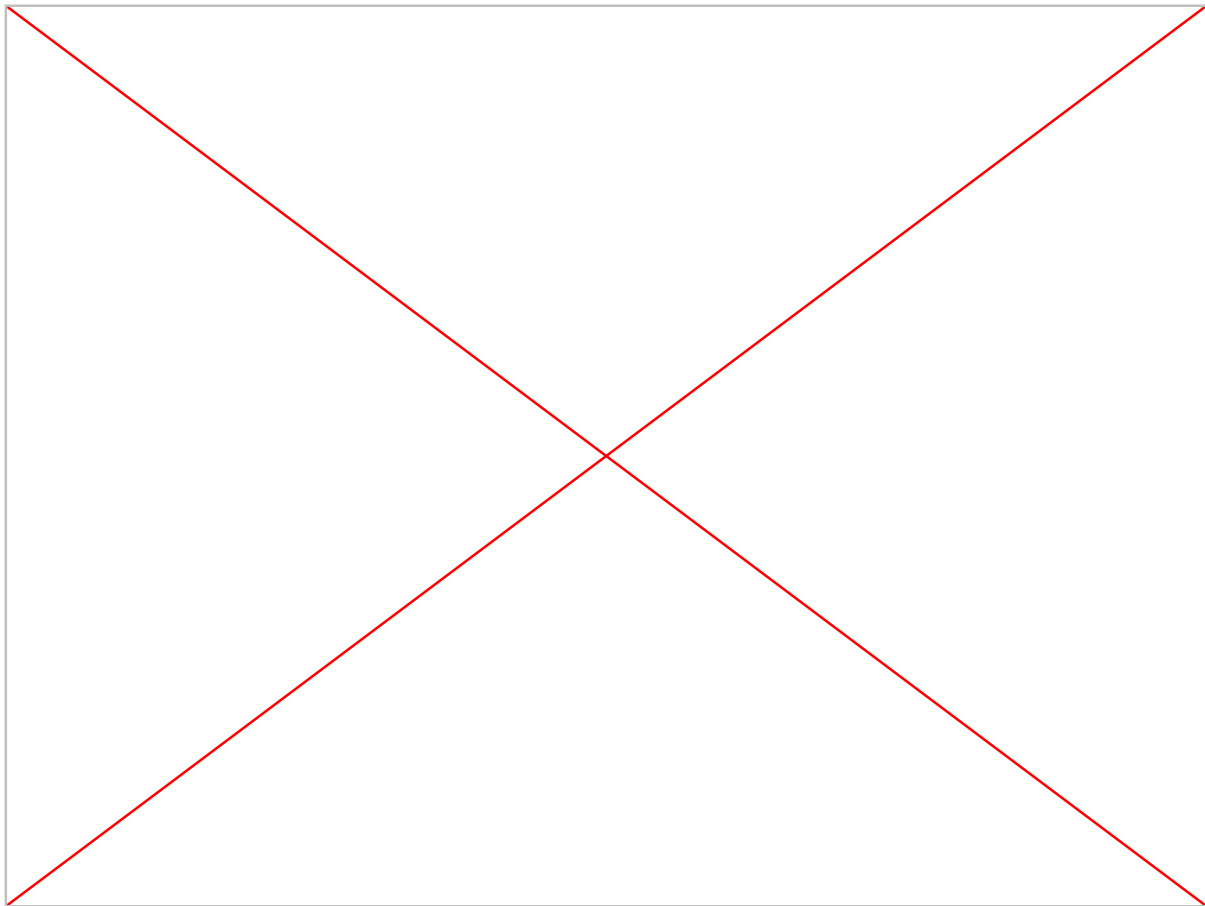
Test Coverage was over 90% and the app was monitored using Atatus.

```
src> _tests_ # EndToEnd.test.js
1 import puppeteer from 'puppeteer';
2
3 describe('show/hide an event details', () => {
4   let browser;
5   let page;
6   beforeAll(async () => {
7     browser = await puppeteer.launch();
8     page = await browser.newPage();
9     await page.goto('http://localhost:3000/');
10    await page.waitForSelector('.event');
11  });
12
13  afterAll(() => {
14    browser.close();
15  });
16
17  test('An event element is collapsed by default', async () => {
18    const eventDetails = await page.$('.event .details');
19    expect(eventDetails).toBeNull();
20  });
21
22 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
23
24 PASS src/features/filterEventsByCity.test.js (34.418 s)
25 PASS src/_tests_/CitySearch.test.js (14.504 s)
26 PASS src/_tests_/NumberOfEvents.test.js
27 PASS src/_tests_/Event.test.js
28 PASS src/_tests_/EventList.test.js
29 (node:6500) [DEP0040] DeprecationWarning: The "punycode" module is deprecated. Please use a userland alternative instead.
30 (Use 'node --trace-deprecation ...' to show where the warning was created)
31
32 One of your dependencies, babel-preset-react-app, is importing the
33 "@babel/plugin-proposal-private-property-in-object" package without
34 declaring it in its dependencies. This is currently working because
35 "@babel/plugin-proposal-private-property-in-object" is already in your
36 node_modules folder for unrelated reasons, but it may break at any time.
37
38 babel-preset-react-app is part of the create-react-app project, which
39 is not maintained anymore. It is thus unlikely that this bug will
40 ever be fixed. Add "@babel/plugin-proposal-private-property-in-object" to
41 your dependencies to work around this error. This will make this message
42 go away.
43
44 PASS src/_tests_/EndToEnd.test.js (21.675 s)
45 node:events:497
46   throw er; // Unhandled 'error' event
47   ^
48
49 Error: kill EPERM
50   at ChildProcess.kill (node:internal/child_process:512:26)
51   at C:\Users\stefi\OneDrive\Documents\vaet\node_modules\puppeteer\src\node\BrowserRunner.ts:211:20
52   at ChildProcess.emit (node:events:430:5)
53   at ChildProcess.emit (node:events:519:24)
54   at maybeClose (node:internal/child_process:1185:16)
55   at Process.ChildProcess._handle.onexit (node:internal/child_process:385:5)
56
57 Emitted 'error' event on ChildProcess instance at:
58   at ChildProcess.kill (node:internal/child_process:512:12)
59   at C:\Users\stefi\OneDrive\Documents\vaet\node_modules\puppeteer\src\node\BrowserRunner.ts:211:20
60   [... lines matching original stack trace ...]
61   at Process.ChildProcess._handle.onexit (node:internal/child_process:385:5) {
62     errno: -4048,
63     code: 'EPERM',
64     syscall: 'kill'
65   }
66
67 Node.js v12.7.2
68
69 Test Suites: 7 passed, 7 total
70 Tests:       30 passed, 30 total
71 Snapshots:   0 total
72 Time:        36.246 s
73 Ran all test suites.
74
75 Watch Usage: Press w to show more. |
```

Final Solution

A user-focused platform that successfully addresses the core challenges of event discovery and management.

Click to see a video of the app and it's features! →



Future Iterations

- Social features like users being able to connect with others attending the same events
- A notification system to send reminders for upcoming events users have selected

Duration:

This project took 3 months to complete. More time than expected was needed as my computer was unavailable for multiple weeks while being repaired and I moved states.

Credits:

Lead Developer: Stephanie Leon

Tutor: Jesus Diaz

Mentor: Shreyansh Kumar