WILLIAM YAU FULL STACK WEB DEVELOPER CASE STUDY

PROJECT OVERVIEW

“Meet” is an application created with React that allows users to find events that they can filter by location and technology. This project uses AWS Lambda, Google OAuth, Recharts, Jest, Cucumber and Puppeteer. It is a serverless, progressive web application that is built using a test – driven development technique. The application itself uses the Google Calendar API to fetch upcoming events.

GOAL AND DURATION

The goal of the project was to build a serverless PWA using a TDD technique. The application had to have some key features in mind which was to:

- Filter the events by city
- Show/Hide event details when viewing the list
- Specify the number of events when viewing the list
- Be able to use the application offline
- Display a chart visualizing event details

June 8th:
Started project with the React template and started to plan out the project using user stories

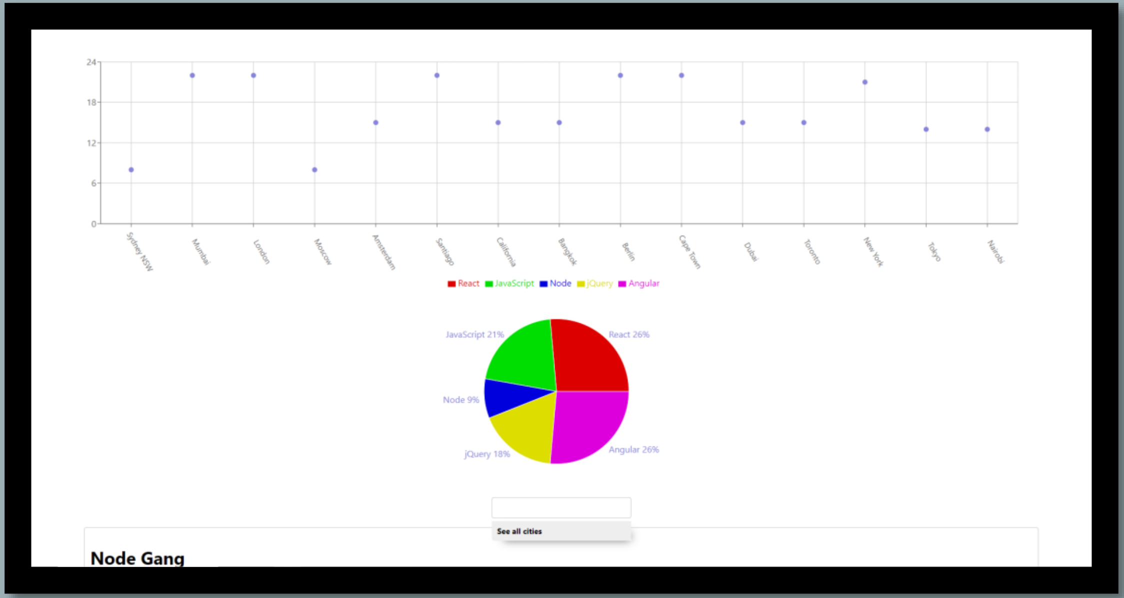
June 14th:
Integrated OAuth and AWS Lambda to allow serverless functionality

June 19th - July 9th :
Started Unit testing, Integration testing, Behaviour – driven development testing and End to end testing with Jest, Cucumber and Puppeteer

July 10th:
Completed application by creating chart visualisations with Rechart

PURPOSE

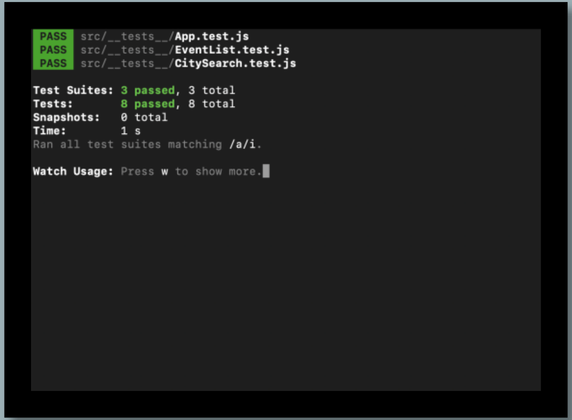
This endeavour afforded me invaluable insights into the imperative practice of meticulously testing my code and features, encompassing a comprehensive array of methodologies such as Integration Testing, Unit Testing, Behaviour - Driven Development (BDD), and End-to-End Testing. Moreover, it served as an initiation into the realm of serverless functions, enhancing my proficiency in crafting sophisticated graphs and charts for comprehensive data visualization.



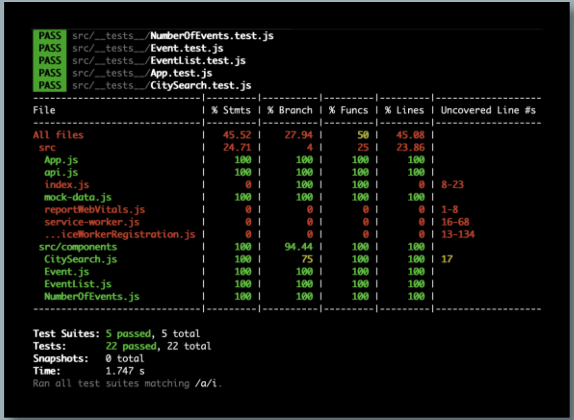
Node Gang

TESTING PHASE

The testing phase constituted a substantial component of the project, during which I leveraged industry-standard tools such as Jest, Cucumber, and Puppeteer to meticulously assess the functionality of my application features. Commencing with unit testing, I ensured the precision and correctness of individual features. Upon confirming the seamless operation of each feature, I transitioned to integration testing to ascertain the robustness of their interactions within the system. In alignment with the user stories that guided the project's development, I employed Behaviour-Driven Development (BDD) tests to enhance collaboration skills and facilitate a more natural and comprehensive approach. Subsequently, End-to-End (E2E) testing was employed to scrutinize the application's entire workflow, encompassing the initiation to the conclusion of its operational processes. This comprehensive testing strategy aimed to validate the integrity and seamless functionality of the application across all levels of its architecture.

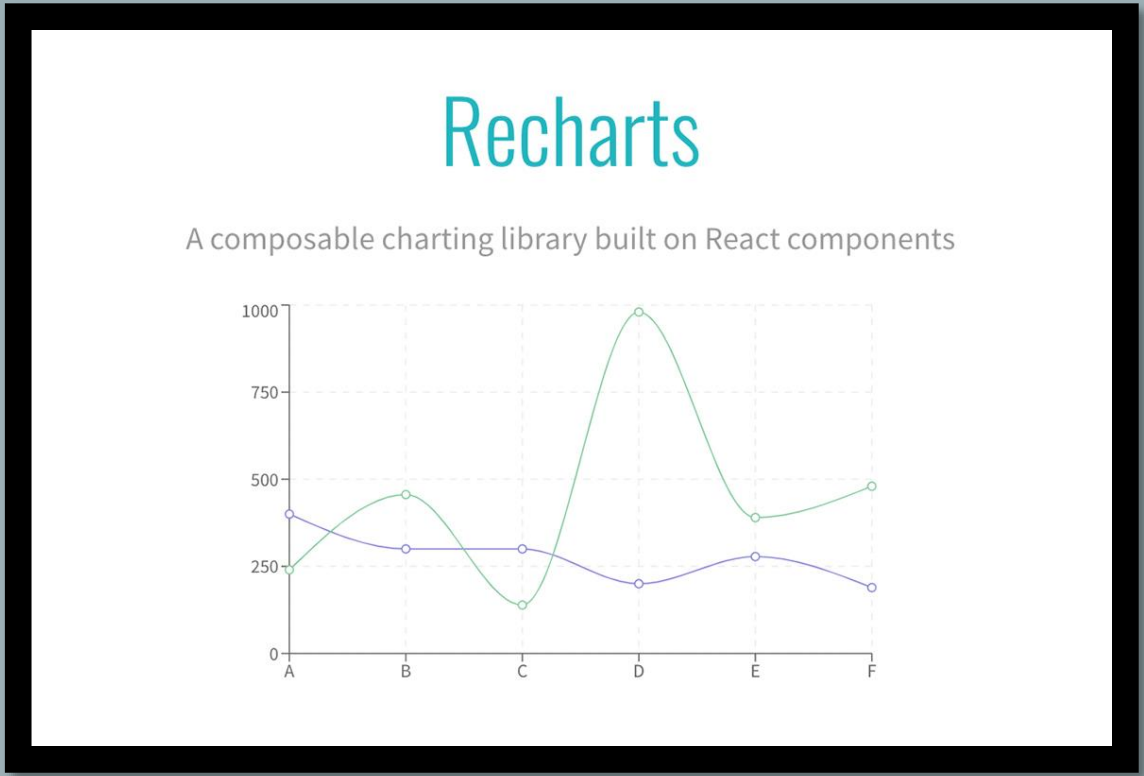


```
 PASS  src/__tests__/App.test.js
 PASS  src/__tests__/EventList.test.js
 PASS  src/__tests__/CitySearch.test.js

Test Suites: 3 passed, 3 total
Tests:       8 passed, 8 total
Snapshots:   0 total
Time:        1 s
Ran all test suites matching /a/i.

Watch Usage: Press w to show more.
```



```
 PASS  src/__tests__/NumberOfEvents.test.js
 PASS  src/__tests__/Event.test.js
 PASS  src/__tests__/EventList.test.js
 PASS  src/__tests__/App.test.js
 PASS  src/__tests__/CitySearch.test.js
-----------------------------|---------|----------|---------|---------|-------------------
File                         | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------------------|---------|----------|---------|---------|-------------------
All files                    |   45.52 |    27.94 |      50 |   45.08 |
 src                         |   24.71 |        4 |      25 |   23.86 |
  App.js                     |     100 |      100 |     100 |     100 |
  api.js                     |     100 |      100 |     100 |     100 |
  index.js                   |       0 |      100 |     100 |       0 | 8-23
  mock-data.js               |     100 |      100 |     100 |     100 |
  reportWebVitals.js         |       0 |        0 |       0 |       0 | 1-8
  service-worker.js          |       0 |        0 |       0 |       0 | 16-68
  ...iceWorkerRegistration.js |      0 |        0 |       0 |       0 | 13-134
 src/components              |     100 |    94.44 |     100 |     100 |
  CitySearch.js              |     100 |       75 |     100 |     100 | 17
  Event.js                   |     100 |      100 |     100 |     100 |
  EventList.js               |     100 |      100 |     100 |     100 |
  NumberOfEvents.js          |     100 |      100 |     100 |     100 |
-----------------------------|---------|----------|---------|---------|-------------------

Test Suites: 5 passed, 5 total
Tests:       22 passed, 22 total
Snapshots:   0 total
Time:        1.747 s
Ran all test suites matching /a/i.
```

DATA VISUALIZATION



Recharts

A composable charting library built on React components

A significant aspect of the project centred around data visualization, with a primary focus on presenting information in a manner conducive to user analysis. To achieve this objective, I employed the Recharts library to craft visually compelling graphs based on the available data. The deliberate selection of a pie chart and scatter chart effectively communicated the distribution of events across specific technologies and cities. This approach not only proved gratifying but also afforded me a broad spectrum of options for data visualization, contributing to the overall user experience by providing clear and insightful representations of the project's underlying data.

FINAL CONCLUSIONS

The positives:
I take great pride in acquiring the skill of effectively testing my code, a crucial proficiency that empowers me to ensure the reliability and integrity of my software solutions. Additionally, the exploration and mastery of a versatile data visualization tool represent a significant accomplishment, as it equips me with a transferable capability applicable to diverse projects.

The negatives:
While the styling aspect of the application was not my primary focus, I acknowledge that there is room for enhancement in this domain. Moving forward, I am open to dedicating additional attention to improve the application's visual presentation, ensuring a harmonious blend of functionality and design excellence.

What could be done for next time?
To further this application, I could create a feature where users could upload their own event and add it onto the system.

Final thoughts:
While I recognize my accomplishments in acquiring proficiency in testing within the development process. This self-assessment underscores my commitment to continuous improvement, as I aspire to create software solutions that not only meet the testing standards but also exhibit a robust feature set and a well-defined architecture for an optimal user experience.