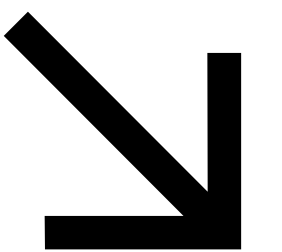


BM

Meet

Case Study



Overview

Meet is a react serverless, progressive web application (PWA) using a test-driven development (TDD) technique. The application uses the Google Calendar API to fetch upcoming events.

Statistics

Events by Genre

21%



28%

Meet App

Filter

Statistics

Events by Genre

Events in Each City

Events

AngularJS Workshop

Mar 29, 2023, 4:00 PM
Cape Town, South Africa

[show details](#)

Fun with Angular

Mar 29, 2023, 4:00 PM
Dubai - United Arab Emirates

[show details](#)

React Workshop

Mar 29, 2023, 8:00 PM
Toronto, ON, Canada

Intro to AngularJS-Remote

Mar 29, 2023, 10:00 PM
New York, NY, USA

[show details](#)

React Native Tokyo

Mar 30, 2023, 9:00 AM
Tokyo, Japan

[show details](#)

Use jQuery, bring in

Mar 30, 2023, 2:00 PM
Mumbai, Maharashtra

Learn JavaScript

Mar 30, 2023, 4:00 PM
London, UK

[show details](#)

Fun with Node.js

Mar 30, 2023, 6:00 PM
Nairobi, Kenya

[show details](#)

jQuery and More

Mar 31, 2023, 1:00 PM
Santiago, Chile

Events

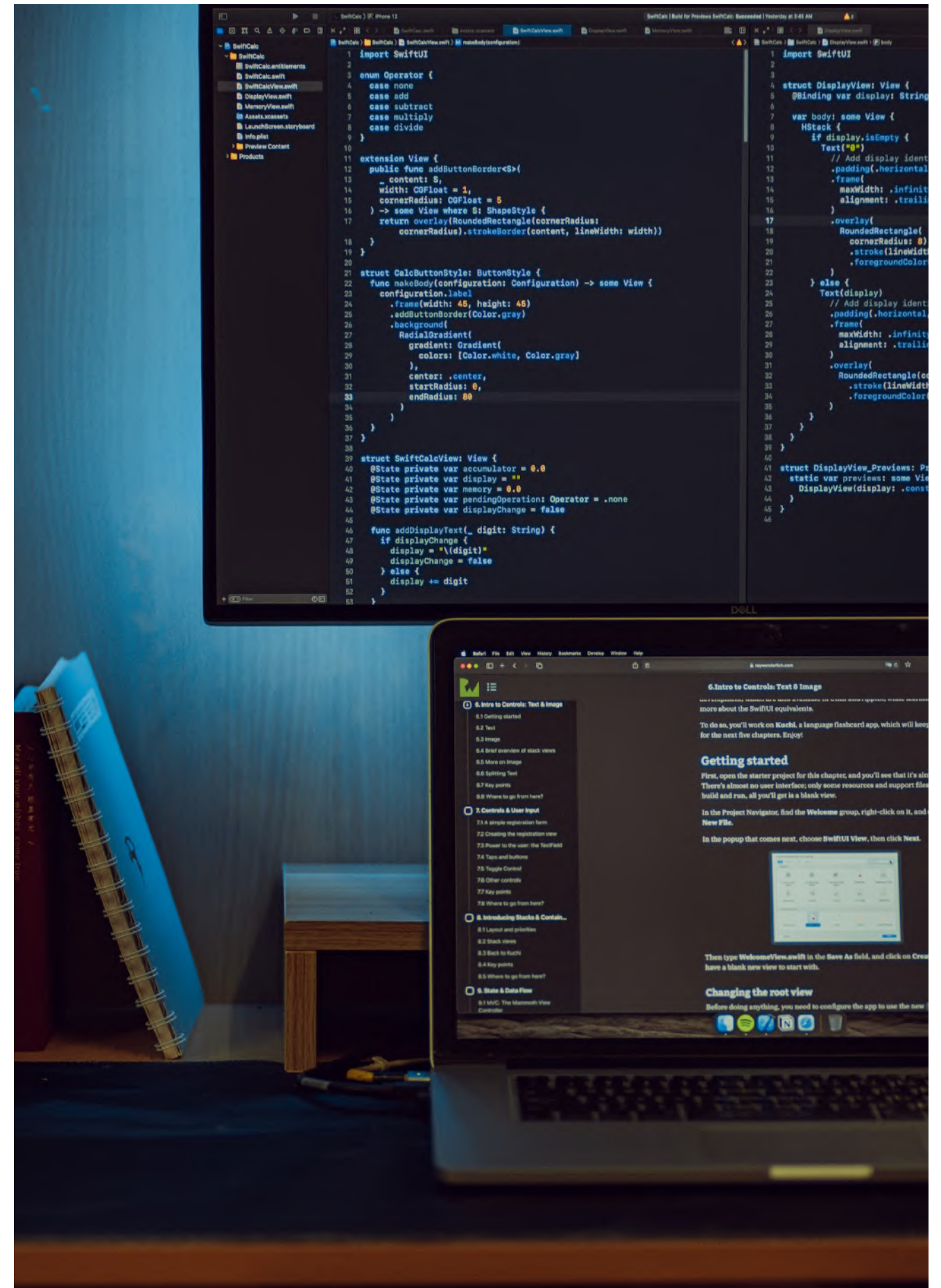
AngularJS Workshop

Purpose & Context

Meet was a project I built as part of my web development course at CareerFoundry to demonstrate my mastery of test driven development and PWA's.

Objective

The aim of the project was to build a progressive web app with the ability to work offline and a serverless backend developed using a TDD technique.





Duration

16 days

Credits

Tutor: Terver Aosu

Mentor: Stephen Barungi

Built with

- React
- Amazon(AWS)
- Google Cloud
- Jest
- Bootstrap

Approach

01 Testing

First I: Applied a test-driven development, defining user scenarios for Jest (unit and integration), Cucumber (acceptance), and Puppeteer (end-to-end) tests based on user stories.

02 Serverless Functions

Then I: Implemented serverless functions such as display events from Google Calendar using AWS Lambda

03 Continuous Delivery

Next I: Created a project in Atatus to automate tracking the performance of the app to resolve any issues

04 Object-Oriented Programming

Then I: Used object-oriented programming to set up notification warning, info and error alerts

05 PWA

Next I: Implemented PWA using Lighthouse, a free tool offered by Google that's used to analyze apps. allowed my app to be installed on users' devices by editing the manifest file from create-react-app and also implemented the offline logic by using localStorage.

06 Data Visualization & Deployment

Finally I: Used Recharts library to add two charts one that shows how many events will take place in that city on upcoming days in the form of a scatterplot chart, and another that visualizes the popularity of event genres in the form of a pie chart and then deployed the updated app on GitHub pages.

Retrospective



What went well?

I like the concept of creating user scenarios and testing before coding. I could see this being an error free approach that saves a lot of time and helps in keeping code consistent across a team. I also enjoyed using Kanban to organize my task and achieve an orderly productive work environment.



What didn't go well?

Upon completing this achievement I ran into some issues with testing in enzyme. My tutor and I could not find a solution as to why some of my test were not passing. I tried debugging and troubleshooting but to no success. My guess is that enzyme is no longer supported with the latest react and could be causing some dependency issues. However the app does still work as intended.



Future Steps

I would like to continue to build on the app by creating my own API or using another external API. I would like the data to be more diverse.



Final Thoughts

Overall, this was an insightful introduction to test driven development. I've learned quite a bit about serverless functions, defining user scenarios, unit and integration, acceptance and end to end testing.



roderickfmoise@gmail.com

+49 163 7126279

End

