Test Driven Development with Progressive Web Applications

A Case Study

Project Objectives:

- To design and build a serverless Progressive Web Application using React and Test Driven Development techniques.

- This application will use the Google Calendar API to retrieve events related to JavaScript coding and display the events along with data visualization tools to users.

- This single page application will be called MyMeetups!

How Did I Build MyMeetups?

- My Role: I completed this project as part of my Full Stack Web Development Course with CareerFoundry. As the Web Developer in charge of this project, my roles were to create wireframes, implement and test code, and host the application online. I built the entire application myself!

- Tools: HTML, CSS, JavaScript, and the React.js library, coded in Visual Studio Code.

- Timeframe for development: 3 weeks.

Introduction to the Project: Key Features

The MyMeetups application must allow users to:

- Filter events by city
- Show/hide event details
- Specify the number of events
- Use the app when offline
- View charts showing the number of upcoming events by city

This app must be developed using Test Driven Development (TDD) techniques!

Introduction to the Project: User Stories

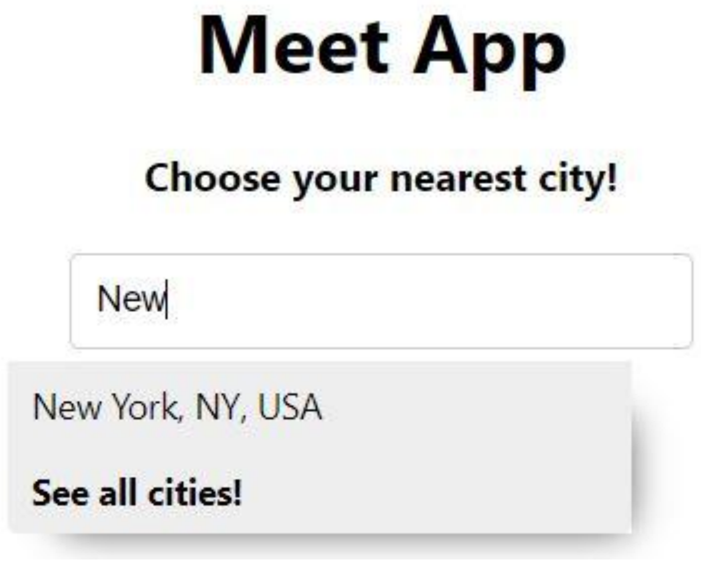
- As a user, I would like to be able to filter events by city so that I can see the list of events that take place nearest to me.
- As a user, I would like to be able to specify the number of events that show up on my application so that I can see more or fewer events at once.
- As a user, I would like to be able to use the application while offline so that I can see the events I accessed while I still had a connection to the internet.
- As a user, I would like to be able to show/hide event details so that I can see more information about events that interest me, but hide them so that the application isn’t cluttered.

What is Test Driven Development?

- Test Driven Development, or TDD, is a process for developing software and applications where features are made into test cases before being fully developed and implemented on the application.
- When a developer uses TDD, they write tests for features before they write the code for the features themselves. Developers using TDD especially use automated tests to speed up the process and remove human error. This makes for less buggy code, more scalable applications, and focused software development!
- So, when I made the MyMeetups App, I wrote tests for features in the app before I created and implemented the features. Pretty cool!

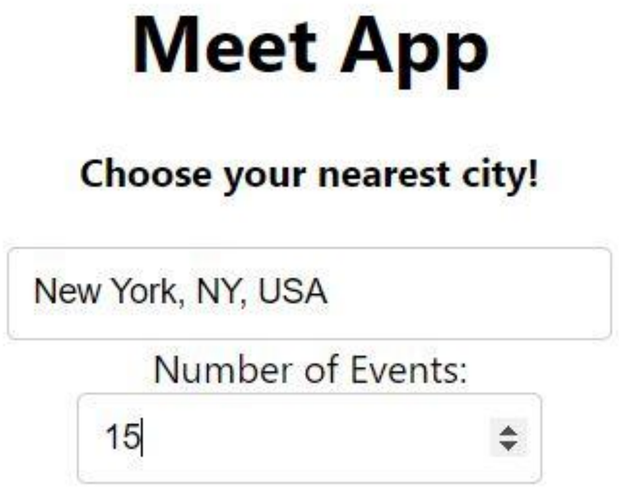
Implementing Key Features: Specify City

One of the key features I implemented for this application was a search bar for users to look up their city and a drop down list of all available cities so that users could narrow down their search. Users can also press the **See all cities!** button to receive search results from events around the world!



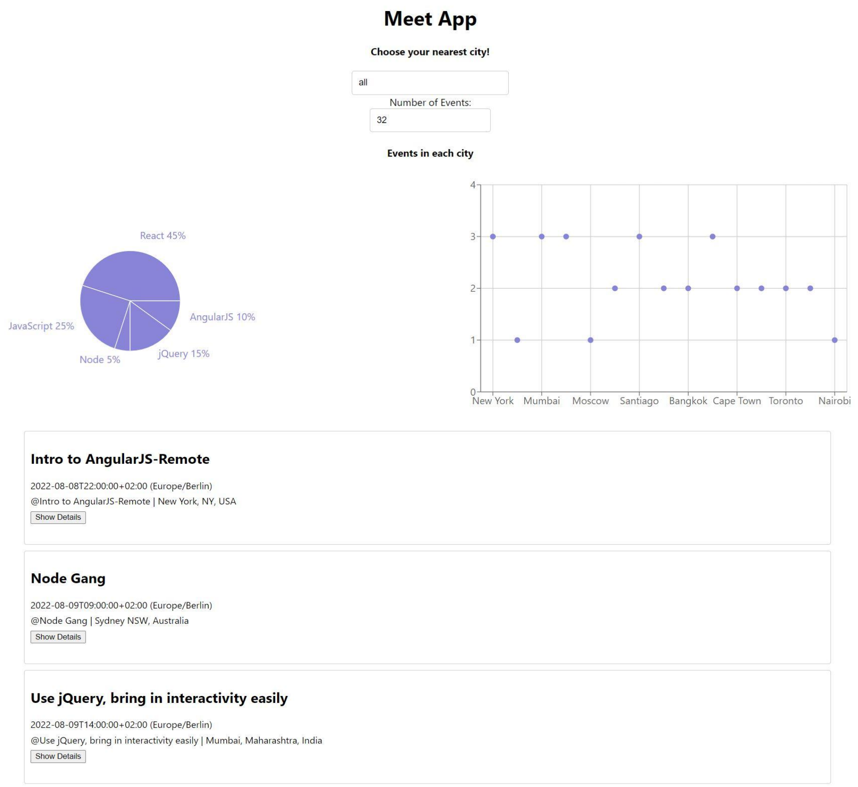
Implementing Key Features: Specify Number of Events

I also implemented a feature so that users can specify the number of events they want to see on the application, either by inputting the number themselves or increasing or decreasing the value. I added this feature because the number of events that was returned to the user could be overwhelming on the application!



Implementing Key Features: View the Events!

Here's the most important part: allowing the users to see the events they're interested in! The above the events themselves, the application allows users to visualize the results with a pie chart and a graph. Users can choose to display more information on events that interest them to prevent the application from becoming cluttered.



Meet App

Choose your nearest city!

all

Number of Events:

32

Events in each city

Intro to AngularJS-Remote

2022-08-08T22:00:00+02:00 (Europe/Berlin)

@Intro to AngularJS-Remote | New York, NY, USA

Show Details

Node Gang

2022-08-09T09:00:00+02:00 (Europe/Berlin)

@Node Gang | Sydney NSW, Australia

Show Details

Use jQuery, bring in interactivity easily

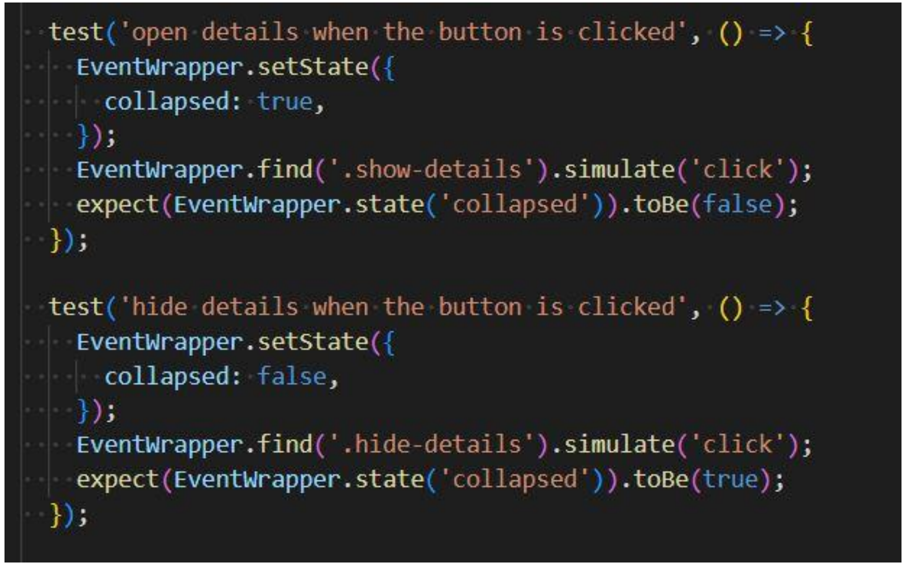
2022-08-09T14:00:00+02:00 (Europe/Berlin)

@Use jQuery, bring in interactivity easily | Mumbai, Maharashtra, India

Show Details

How Does Test Driven Development Fit In?

Before I implemented these neat features for my application, I wrote out unit, integration, and end-to-end tests! An example of my actual tests is on the right. These tests specified the desired feature in the application and checked when the application rendered to see whether or not the code is working, all automatically, saving time and removing human errors!



```
test('open details when the button is clicked', () => {
  EventWrapper.setState({
    collapsed: true,
  });
  EventWrapper.find('.show-details').simulate('click');
  expect(EventWrapper.state('collapsed')).toBe(false);
});

test('hide details when the button is clicked', () => {
  EventWrapper.setState({
    collapsed: false,
  });
  EventWrapper.find('.hide-details').simulate('click');
  expect(EventWrapper.state('collapsed')).toBe(true);
});
```

Challenges and Solutions During Development

- Challenge: Tests are failing, but features are working
    - Solution: selectors on the tests aren't coded right! I reworked all of the tests and they all passed.
- Challenge: Users can specify too many events for the application
    - Solution: code alerts to let users know the range of events they can specify for the application to display.
- Challenge: Application isn't receiving events from the API
    - Solution: code error with the way the application is linking up with the events on the Google Calendar API; reworked code until the application retrieved the events.

Concluding Thoughts

What a fun application to code and develop! I learned so much using the Test Driven Development techniques and I believe that developing an application in this way would be especially beneficial when working with a team and if I had to hand off this application to another developer. This way, by looking at the tests, they would know all about the features for the application, what they're for, and would be alerted if anything in the application broke while they were adding new features!