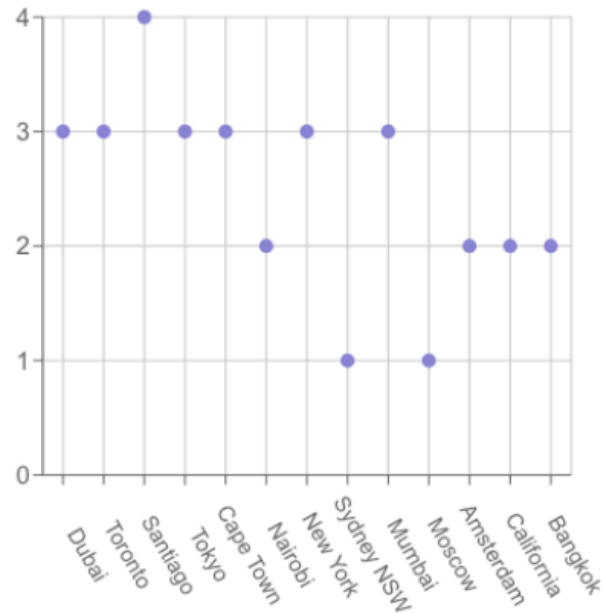
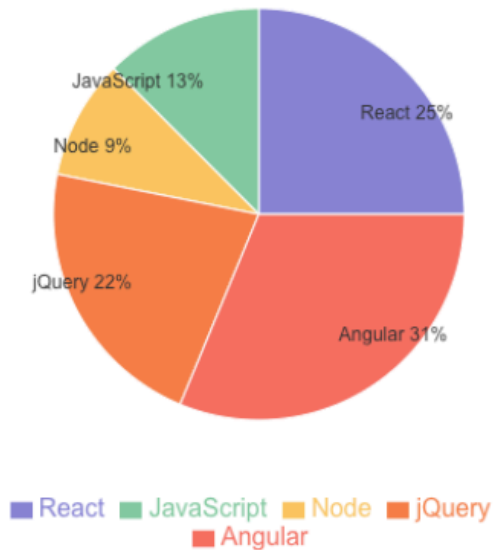


# Meet App Case Study



## Executive Summary

*A responsive, serverless web app that helps users discover, explore, and visualize upcoming events across cities.*

### Introduction

The Meet App is a progressive web application (PWA) that helps users find upcoming events in different cities. It works online and offline, can be added to a phone's home screen, and shows visual charts that break down events by genre and location. I created this app to strengthen my skills in React, data visualization, and test-driven development while focusing on building a smooth and reliable experience for users.

## **The Problem**

Event apps often feel overwhelming or require creating accounts to use basic features. Some don't work well offline or lack visual tools to help users understand the content. I wanted to create an app that worked smoothly from the first click, felt simple, and offered clear visuals to explore events in a meaningful way.

## **My Role**

I designed and built the full frontend of this app using React. I integrated the Google Calendar API, handled city and event filtering, and added visual charts using Recharts. I also implemented offline functionality using Workbox and followed a test-driven development approach using Jest.

## **Timeline**

This project took around 3 weeks from concept to completion.

## **Tools and Skills Used**

Tools and Libraries:

- React
- Recharts (for data visualization)
- Workbox (for offline features)

Backend Integration:

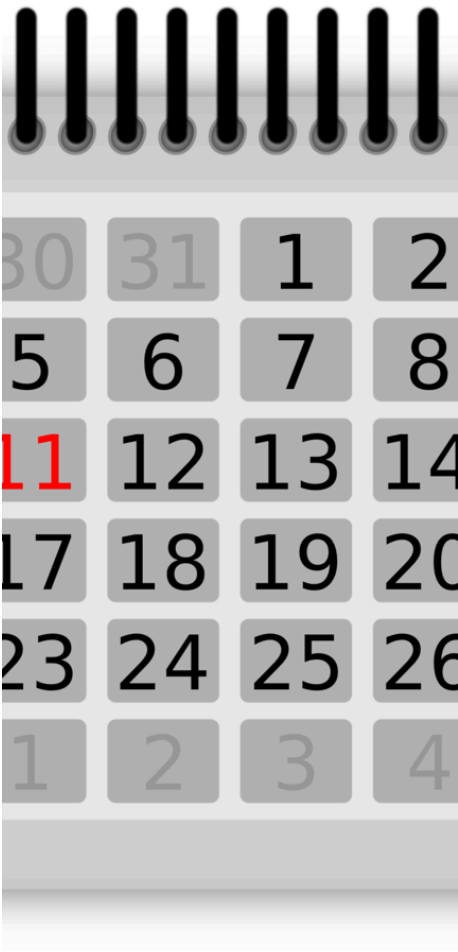
- Google Calendar API

Other Technologies:

- Jest (for unit and integration testing)
- GitHub Pages (for deployment)

Skills I practiced:

Test-driven development, responsive design, data filtering, building PWAs, working with APIs, data visualization, and offline-first development.



# Key findings

## Key Finding 1:

Users liked being able to search for events by city and set the number of events shown.

## Key Finding 2:

Charts that showed events by genre and city location helped users better understand the types and distribution of events.

## Key Finding 3:

Offline access gave users peace of mind and let them explore previously loaded data without needing a signal.

# Solutions

## Solution 1

Used the Google Calendar API to fetch real event data and allow filtering by city

## Solution 2

Added user controls to set the number of events shown and toggle event details

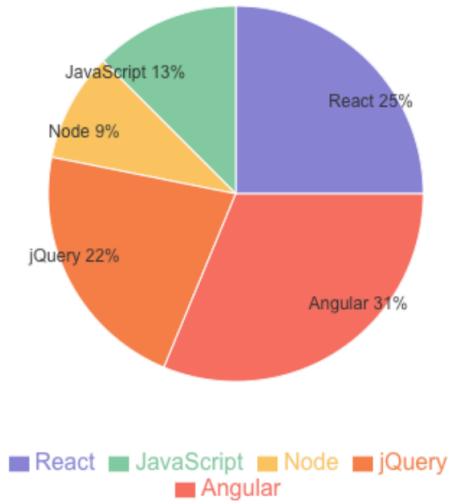
## Solution 3



Used Recharts to create pie charts and scatterplots that visualized the event data in a more meaningful way

#### **Solution 4**

Built in offline support so users could access the app without internet after initial load



## Impact

### **Impact 1**

The app gave users a quick and flexible way to find events without needing an account or login

### **Impact 2**

Data visualizations helped users see patterns in event genres and city locations

### **Impact 3**

I gained hands-on experience with TDD, data visualization, and offline-first web app design

# Case Study Introduction

## Background

This project was part of my full-stack development program. I created the Meet App to build something real using test-driven development (TDD), data visualization, and responsive design.

I wanted users to be able to discover events in different cities and understand those events through visual summaries. I built the app using React and connected it to the Google Calendar API to fetch real event data.

## Objectives

### **Objective 1**

Create a web app that helps users find and filter events by city

### **Objective 2**

Allow users to control how many events they see at once

### **Objective 3**

Include a visual breakdown of events by location and genre

### **Objective 4**

Make the app installable, responsive, and usable offline

## Problem

Some event apps don't work well offline or lack visual tools to help users understand the content. That can be really frustrating when

you're at a conference and are not sure where your session is located.

I wanted to create an app that worked smoothly from the first click, felt simple, and offered clear visuals to find events and to use the data in easily.

# Recommendations

**Recommendation 1**

Build visual features that help users quickly understand content, like charts and graphs

**Recommendation 2**

Let users control what they see by adding filters and options like number of events or expanded details

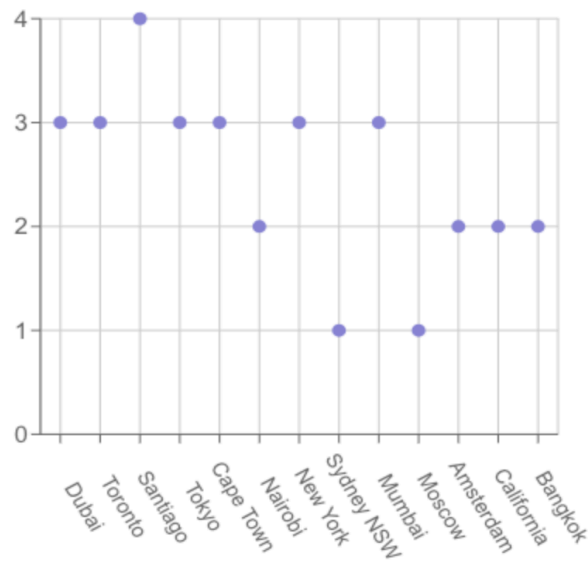
**Recommendation 3**

Offer offline access so users aren't limited by their internet connection

## Impact of recommendation

SOLUTION	EFFECTIVENESS	IMPACT	NOTES
City filter and event limit	Highly effective ▾	Helped users focus only on the events they wanted	Simple and intuitive filtering experience
Chart visualizations	Highly effective ▾	Made patterns in the data more obvious	Supported with scatterplots and pie charts
Offline access	Highly effective ▾	Made the app more reliable and accessible	Powered by Workbox and PWA features

# Analysis



## Research methods

- Reviewed other event apps to find common UX pain points
- Tested the app's behavior across browsers and mobile devices
- Used feedback from peers and mentors to improve layout, functionality, and visual clarity

## Approaches used

- Test-driven development with Jest
- Responsive layout using React and CSS
- Data integration and charting using Google Calendar API and Recharts
- Offline setup using Workbox

# Relevant facts and information

- Duration: 3 weeks
- Tools: React, Recharts, Google Calendar API, Workbox
- Features: Search by city, event filters, event detail toggles, offline access, charts and graphs

## Challenges

One challenge was handling data formatting from the Google Calendar API to match what was needed for the charts.

Another challenge was balancing offline performance with live updates. Getting everything to work smoothly with Workbox and ensuring a reliable experience without the internet required careful setup and testing.

## Conclusion



## Summary of findings

The Meet App gave users a clear and visual way to find and explore upcoming events. With filtering options, offline access, and interactive charts, the app helped users discover events in a way that felt simple and insightful.



# Implications of the study

This project taught me how to build more interactive and user-friendly apps. I got to practice working with real APIs, charting tools, and offline technologies.

It also gave me the chance to focus on writing clean, testable code through test-driven development.

**“Meet App showed me how powerful it can be to combine data, design, and interactivity in one tool that helps people connect with events in their area.”**