

# CASE STUDY: STARBUCKS PROGRESSIVE WEB APP

UNIVERSALLY ACCESSIBLE ORDERING FOR ESTABLISHED AND EMERGING MARKETS

---

**BACKGROUND:** Starbucks released their mobile ordering app nationwide in 2015. When it came time to add ordering functionality to the Starbucks website, they wanted to create a system that would be universally accessible to both established and emerging markets, so they called in Formidable to help them create a Progressive Web Application (PWA). Because PWAs are built around the concept of "offline-first", they are great for emerging markets, like rural communities, where internet connections are less reliable. They're also great for people on the go, who may go in and out of internet connectivity throughout their day. The goal was to build a Starbucks PWA that would allow customers to browse the menu, customize their orders, and add items to their carts, even if they weren't online.

---

## CHALLENGE:

- + Create a performant Progressive Web App with dynamic functionality that feels native, despite being on the web platform
- + Build a user-friendly ordering system that accounts for complex business logic and works smoothly throughout the entire ordering process

## SOLUTION:

- + Create performant, content-specific animations that will give users an experience that rivals that of a native app
- + Work with the existing APIs to accommodate complex business logic and customer interactions within the ordering system. Utilize GraphQL to organize data and streamline business logic
- + Implement client-side caching to enable offline functionality
- + Scale the supporting Node server by improving performance and adding layered caching

## RESULTS:

- + A fast, performant, intuitive PWA that looks and feels like a native app
- + Emerging markets will be able to utilize the new web app even without consistent internet access
- + Customers can customize their food and drink orders seamlessly
- + The new GraphQL layer has made it possible for Starbucks to use consistent business logic and data across all platforms, so it no longer has to be recreated each time it needs to be used.

## BENEFITS:

- + The PWA ended up being 233KB, 99.84% smaller than the 148MB size of the iOS mobile app
- + The PWA project introduced the Starbucks team to GraphQL, which they now plan to use for future projects that need to organize complex logic and data
- + With GraphQL, less customization is needed for specific versions since it can be used any client (iOS or Android)
- + The PWA shows that Starbucks is committed to elevating the modern web

#### CHALLENGE :

- + Despite the existence of the mobile app, the PWA ordering system would need to be built from the ground up. Because customers are used to using a dynamic, native mobile app for ordering, it was important to create a similar experience within the PWA. The app would need to incorporate images, animations, and interactions, while still fulfilling the requirements for a PWA. Additionally, the Formidable team would need to do extensive work to ensure that the ordering system was user-friendly and functional, so both the customer and the business would have a positive experience through the entirety of the transaction. Because the app allows for myriad customizations, complex business logic would need to be implemented so that every possible deviation from the default menu items would be accounted for.

#### SOLUTION :

- + In order to make the ordering system work efficiently and effectively, complex business logic needed to be accounted for. Each menu item includes many different customization options, each of which has its own unique business logic, so the team had to consider what options would be displayed to the customer, how the customer could modify the product, and what information would get sent to the POS. Each order would need to specify the default recipe and any ways in which the order deviated from that standard so the barista could make the order without any hiccups.

Because no existing APIs could account for all of this logic right out the box, it all needed to be written in. Formidable utilized GraphQL to organize the complex data set and business logic for use across all clients, and to wrangle a legacy API into an efficient, performant one.

In order to create the offline functionality for the PWA, complex caching had to be set up to download the information for each menu item, store information on customers' previous orders, and store static pages from the website. Formidable's team set up the server so it could handle in-memory caching and efficient server-side rendering.

Formidable utilized React to create performant, content-specific animations, that would provide feedback to the user and help the content feel alive. These kinds of animations are common in native apps, but a luxury in web apps.

#### RESULTS :

- + Starbucks now has a fast, performant, easy-to-use PWA that looks and feels like a native app. Smooth animations make the app feel native, and the tiny app size makes it fast and ultra responsive. The implementation of caching has given customers the option to use the app to look at menus and nutrition information, accurately customize their food and drink orders, and more, all without an internet connection. Once online, customers can use the PWA to view pricing specific to the location they're ordering from and place their order. Because much of the app is available without an internet connection, the app is viable for emerging markets that may not have consistent access to the internet, and to customers who may lose access periodically throughout their day.

#### BENEFITS :

- + The PWA had a lot of positive results beyond the main scope of the project. By virtue of the specific requirements of a PWA, the final version of the PWA clocked in at 233KB, 99.84% smaller than the 148MB size of the mobile iOS app. The PWA is also faster than Starbucks' existing iOS app, making the web app a favorite amongst its users. The project introduced the Starbucks team to GraphQL, which will now be used for future projects that need to organize complex logic and data. Additionally, with GraphQL, any client (iOS or Android) can use the logic so less customization is needed for specific versions. The PWA shows that Starbucks is committed to elevating the modern web, making it just as performant (if not more so) than a native app.

#### CONCLUSION :

- + The new ordering PWA shows Starbucks' commitment to providing accessible, user-friendly online ordering to all of their customers, regardless of whether they are in an established or emerging markets.