Know more about the Technologies 🖥️ 📱 🌐

React v/s Angular

|		
History	React is not a framework, but a JavaScript library developed and maintained by Facebook, released in 2013, and being used at Facebook since then.	Angular is a TypeScript based JavaScript framework, developed and maintained by Google. The latest version of Angular is Angular 8 now.
Most Prominent Features	- React uses JSX, an XML-like language built on top of JavaScript instead of classic templates. - XSS protection. - No dependency injection. - Fetch for Ajax requests. - Utilities for unit-testing components. React also provides some popular libraries to add functionalities: - React-router for routing. - Redux or MobX for state management. - Enzyme for additional testing utilities.	- Provides templates, based on an extended version of HTML. - Provides XSS protection. - Provides Dependency injection. - Provides Ajax requests by @angular/HTTP. @angular/router for Routing. - Component CSS encapsulation. - Utilities for unit-testing components. @angular/forms for building forms.

React v/s Angular

TypeScript v/s JavaScript	React uses JavaScript which is a dynamically-typed language, so you don't have to define the variable's type. It makes it easy to use.	Angular is written in TypeScript, so you must be comfortable with TypeScript before using Angular.
Used Templates	React uses UI templates and inline JavaScript logic together which was not done by any company before. This is called JSX. React uses component which contains both the markup AND logic in the same file. React also uses an XML-like language which facilitates developers to write markup directly in their JavaScript code. JSX is a big advantage for development, because you have everything in one place, and code completion and compile-time checks work better.	Angular uses enhanced HTML templates with Angular directives i.e. "ng-if" or "ng-for" etc. It is quite difficult because you have to learn its specific syntax.
Scalability	React is more scalable than Angular.	Angular is easy to scale.

React v/s Angular

	React	Angular
Architecture	React is a simple JavaScript library giving much more freedom to choose your own libraries.	Angular is a full MVC (Model, View, and Controller) framework.
Speed	React is faster than Angular.	Angular is fast as compared to old technologies but React is faster than Angular.
Size	The size of React is smaller than Angular, so it is a little bit faster.	The size of Angular is large, so it takes longer load time and performance on mobile.
Companies Using	- Facebook - Airbnb - Uber - Netflix - Instagram - WhatsApp - Dropbox, etc.	- Google - Nike - Forbes - Upwork - General Motors - HBO - Sony, etc.

Progressive Web Application
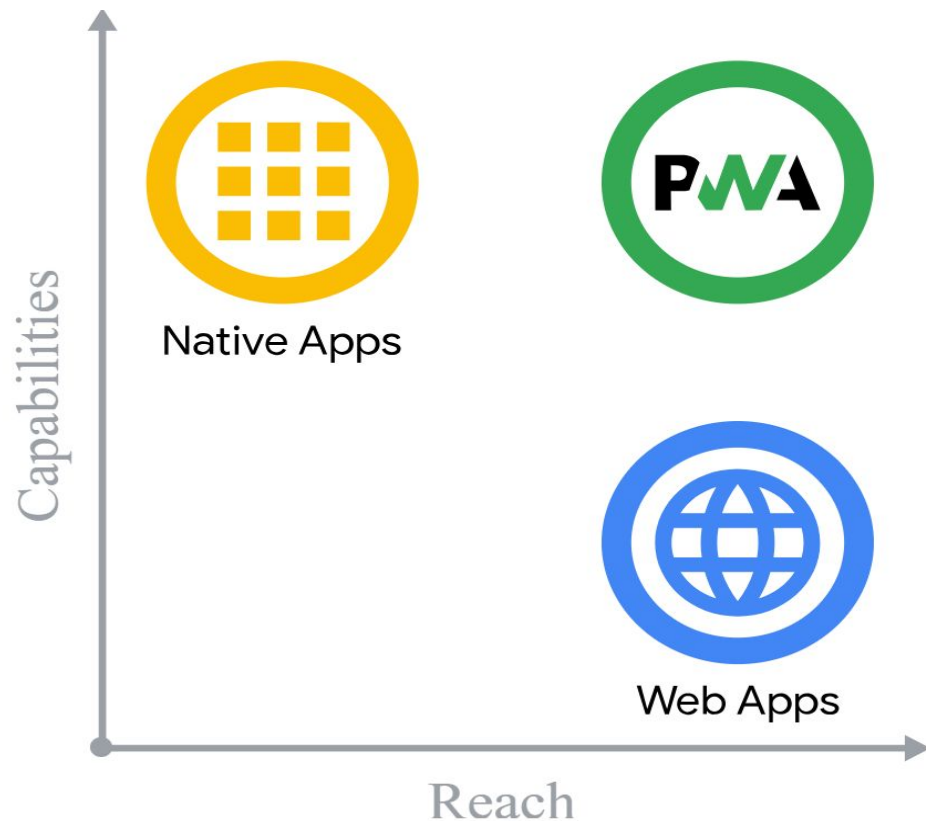
If you think about platform-specific apps and web apps in terms of capabilities and reach, platform-specific apps represent the best of capabilities whereas web apps represent the best of reach. So where do Progressive Web Apps fit in?

Progressive Web Apps (PWA) are built and enhanced with modern APIs to deliver enhanced capabilities, reliability, and installability while reaching anyone, anywhere, on any device with a single codebase.

Find out more about PWAs on,

https://web.dev/articles/what-are-pwas
https://youtu.be/QLfmzSR_xSY?si=SxPYcu567z56gGch

Capabilities vs. reach of platform-specific apps, web apps, and progressive web apps.

Progressive Web Application

Definition: Progressive Web Apps are web applications that have been designed to be capable, reliable, and installable. These three pillars transform them into an experience that feels like a platform-specific application.

Progressive Web Apps will offers your users a lovely experience. Using the latest web features to bring enhanced capabilities and reliability, Progressive Web Apps allow what you build to be installed by anyone, anywhere, on any device with a single codebase.

Capable	The web is quite capable in its own right today. For example, you can build a hyper-local video chat app using WebRTC, geolocation, and push notifications. You can make that app installable and take its conversations virtual with WebGL and WebVR. With the introduction of WebAssembly, developers can tap into other ecosystems, such as C, C++, and Rust, and bring decades of work and capabilities to the web. Squoosh.app, for instance, uses this for its advanced image compression. Until recently, only platform-specific apps could really lay claim to these capabilities. While some capabilities are still out of the web's reach, new and upcoming APIs are looking to change that, expanding what the web can do with features like file system access, media controls, app badging, and full clipboard support. All of these capabilities are built with the web's secure, user-centric permission model, ensuring that going to a website is never scary for users. Between modern APIs, WebAssembly, and new and upcoming APIs, web applications are more capable than ever, and those capabilities are only growing.

Progressive Web Application

Reliable	A reliable Progressive Web App feels fast and dependable regardless of the network.
	Speed is critical for getting users to use your experience. In fact, as page load times go from 1 second to ten seconds, the probability of a user bouncing increases by 123%. Performance doesn't stop after the onload event. Users should never wonder whether their interaction—for example, clicking a button—was registered or not. Scrolling and animation should feel smooth. Performance affects the entire user experience, from how they perceive your application to how it actually performs.
	Finally, reliable applications need to be usable regardless of network connection. Users expect apps to start up on slow or flaky network connections or even when offline. They expect the most recent content they've interacted with, such as media tracks or tickets and itineraries, to be available and usable even if getting a request to your server is hard. When a request isn't possible, they expect to be told there's trouble instead of silently failing or crashing.
	Users deserve apps that respond to interaction in the blink of an eye, and an experience they can depend on.

Progressive Web Application

| Installable | Installed Progressive Web Apps run in a standalone window instead of a browser tab. They're launchable from on the user's home screen, dock, taskbar, or shelf. It's possible to search for them on a device and jump between them with the app switcher, making them feel like part of the device they're installed on.

New capabilities open up after a web app is installed. Keyboard shortcuts, usually reserved when running in the browser, become available. Progressive Web Apps can register to accept content from other applications, or to be the default application to handle different types of files.

When a Progressive Web App moves out of a tab and into a standalone app window, it transforms how users think about it and interact with it. |
|---|---|

Progressive Web Application

Conclusion:

Progressive Web Apps are just web applications with progressive enhancement, enabling new capabilities in modern browsers. Using service workers and a web app manifest, your web application becomes reliable and installable. If the new capabilities aren't available, users still get the core experience.

Companies that have launched Progressive Web Apps have seen impressive results. For example,

- Twitter saw a 65% increase in pages per session, 75% more Tweets, and a 20% decrease in bounce rate, all while reducing the size of their app by over 97%.

- After switching to a PWA, Nikkei saw 2.3 times more organic traffic, 58% more subscriptions, and 49% more daily active users.

- Hulu replaced their platform-specific desktop experience with a Progressive Web App and saw a 27% increase in return visits.

Design References

- Pharmaceutical App - Only for Demo - With animations
  - Figma Link

- Fitness App - Only for Demo - With animations
  - Figma Link
  - https://www.behance.net/gallery/176911417/Step2earn-Fitness-feet-tracking-mobile-app

- Pet Care Service App - Only for Demo - With animations
  - Figma Link
  - https://www.behance.net/gallery/181414885/Pet-Care-Mobile-App

- Financial Loan App - Only for Demo
  - Figma Link
  - https://www.behance.net/gallery/159555259/Urban-Finance-Fintech-app

- Payment Transfer App - Only for Demo
  - Figma Link
  - https://www.behance.net/gallery/156157731/Fastpay-App

- Dating/Swinger App - UI/UX available - No animations
  - Figma Link
  - https://www.behance.net/gallery/147960483/Dating-App

Contact Us,

Keep in Touch

HQ Address
517-520, Center Square,
Nadiad City, India

Phone Contact
+91 903 3578 484
+91 940 9123 477

AU Address
Level 5, 4 Columbia CT,
Norwest NSW, 2153,
Australia

Submit a Contact Form

Email Contact
kartik.bhaviyash@groovyweb.co

Website
www.groovyweb.co

Connect with us:

Thank You and All The Best!

Let's join 🤝 to design an app that adds immense value to the life of your users!