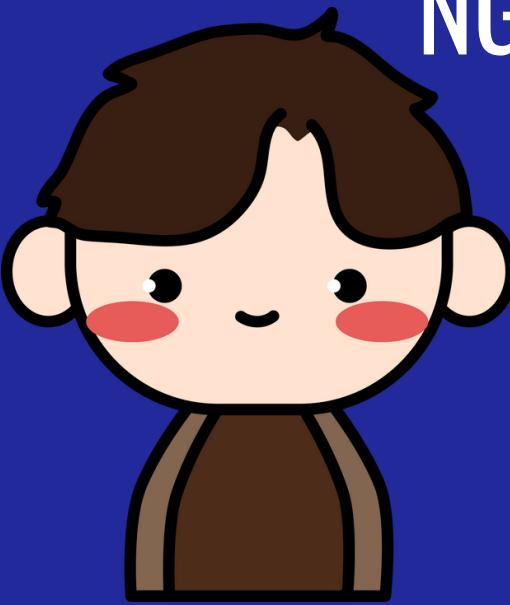
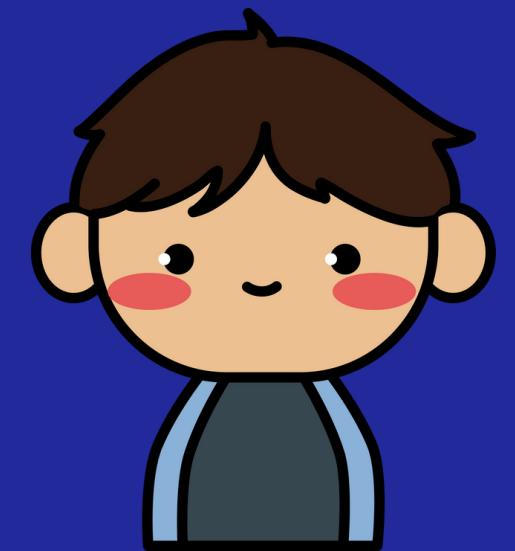


MOBILE APP DEVELOPMENT

Group 09



NGUYỄN NGỌC THIỀN ÂN



NGUYỄN MINH NHẬT



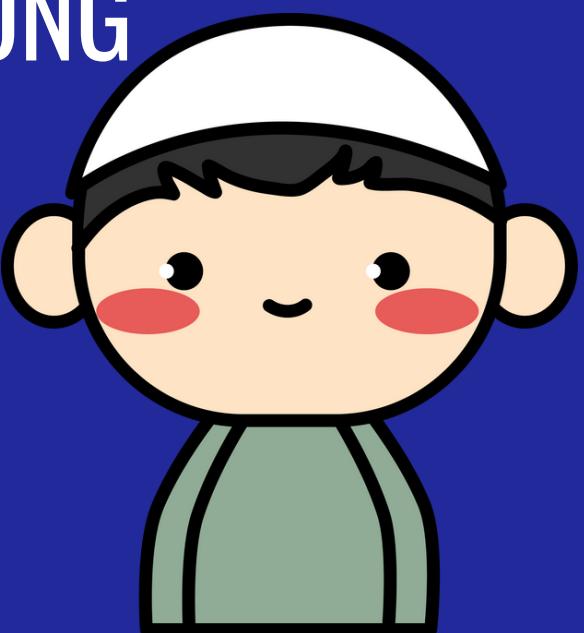
welcome to
our team



TÔ TẤN HIỆP



NGUYỄN TUẤN PHONG



NGUYỄN DUY KHANG

NỘI DUNG



01.
TỔNG QUAN
Mobile App Development là gì?

02.
QUÁ TRÌNH
Quá trình xây dựng Mobile App

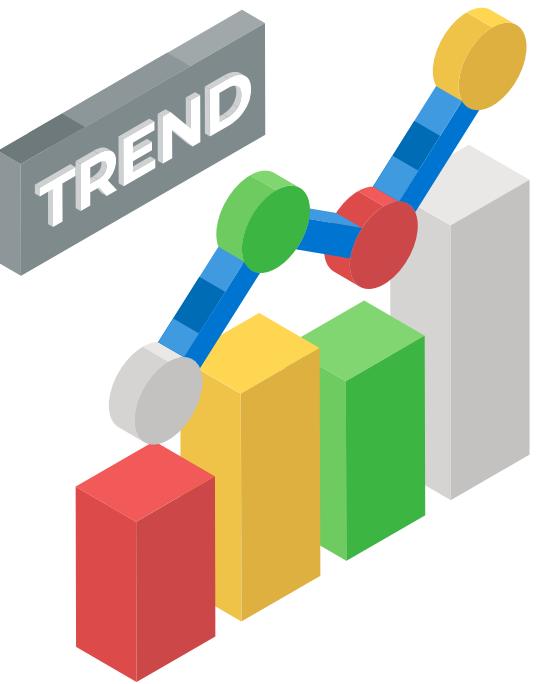
03.
PHÂN LOẠI
Native, Web, Hybrid

04.
**CROSS-PLATFORM
FRAMEWORKS**

TỔNG QUAN



WHY MOBILE?

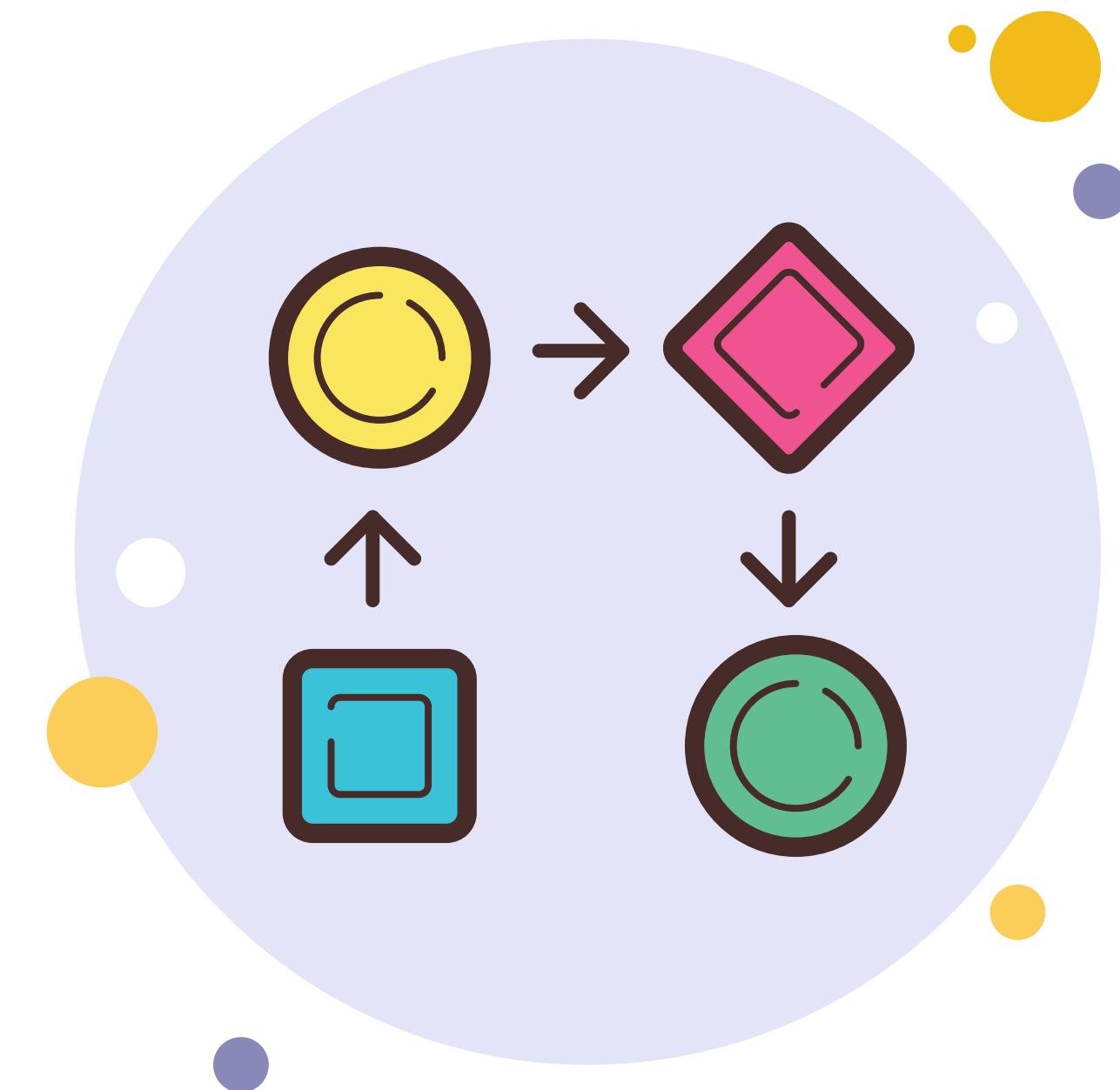


WHAT IS Mobile App Development ?

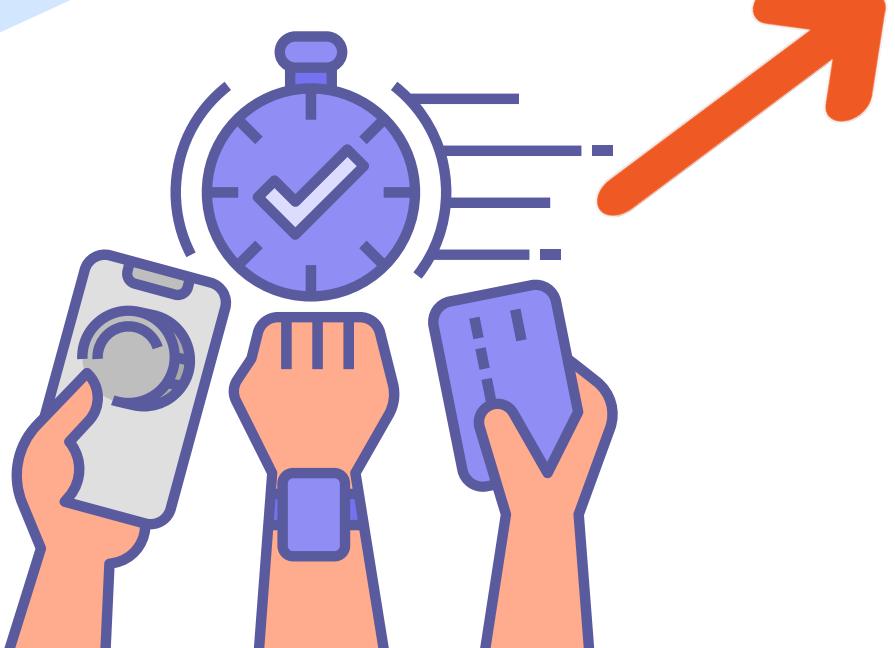
WHAT IS Mobile App Development ?

Là một quy trình tạo ra phần mềm cho thiết bị di động.

Các ngôn ngữ lập trình và đánh dấu được sử dụng cho loại hình phát triển phần mềm này bao gồm Java, Swift, C# và HTML5.



Tâm quan trọng



Nhu cầu





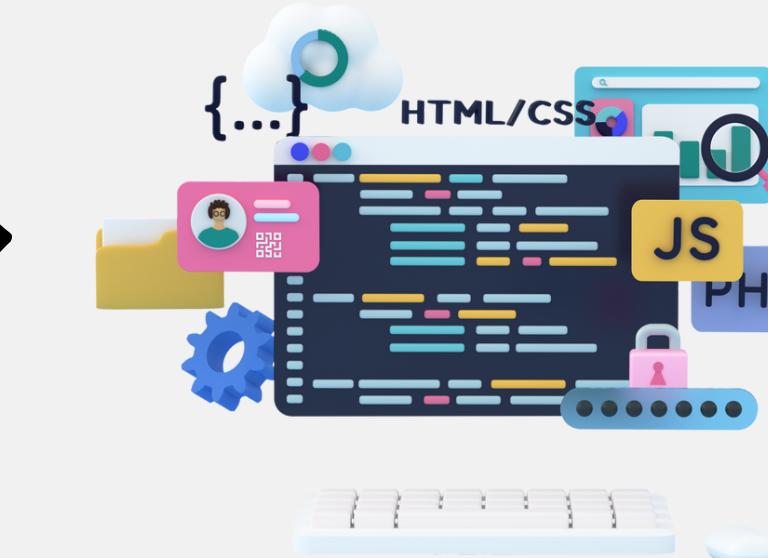
Thông kê sử dụng mobile app

Đến năm 2025, dự kiến sẽ có 7,49 tỷ người dùng di động trên toàn thế giới

QUÁ TRÌNH PHÁT TRIỂN



Quá trình phát triển một Mobile Application



1

Xác định chiến lược

Một chiến lược để xác định làm thế nào doanh nghiệp của bạn có thể phát triển thông qua một ứng dụng di động. Khách hàng của bạn sẽ được lợi ích gì? Nhân viên hoặc đối tác kinh doanh của bạn sẽ nhận được gì?



Xác định mục đích ứng dụng

Xác định đối tượng người dùng

Các yếu tố cạnh tranh

2

Phân tích và lên kế hoạch

Ý tưởng ứng dụng của bạn bắt đầu hình
thành và trở thành một dự án thực sự

- 01 Ý tưởng ứng dụng → Dự án: Chuyển đổi ý tưởng thành một dự án thực sự.
- 02 Phân tích & Kế hoạch: Xác định trường hợp sử dụng và yêu cầu chức năng chi tiết.
- 03 Lộ trình & MVP: Xác định hướng đi của sản phẩm và phiên bản sản phẩm tối thiểu có khả năng thực hiện.
- 04 Tên ứng dụng & Cửa hàng: Chọn một tên duy nhất cho ứng dụng và đảm bảo nó không bị trùng lặp trong cửa hàng ứng dụng.

3

Thiết kế giao diện người dùng (UI/UX)

"Mục đích của thiết kế ứng dụng di động là mang đến trải nghiệm người dùng mượt mà và dễ dàng với một giao diện thanh lịch."



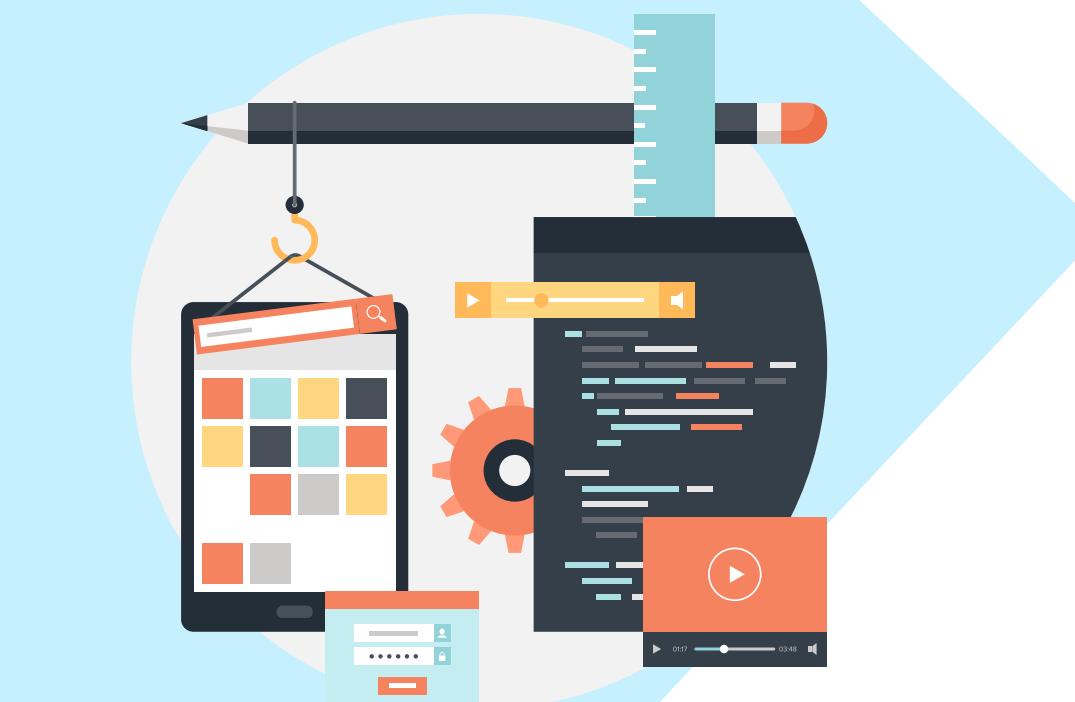
4

Phát triển ứng dụng

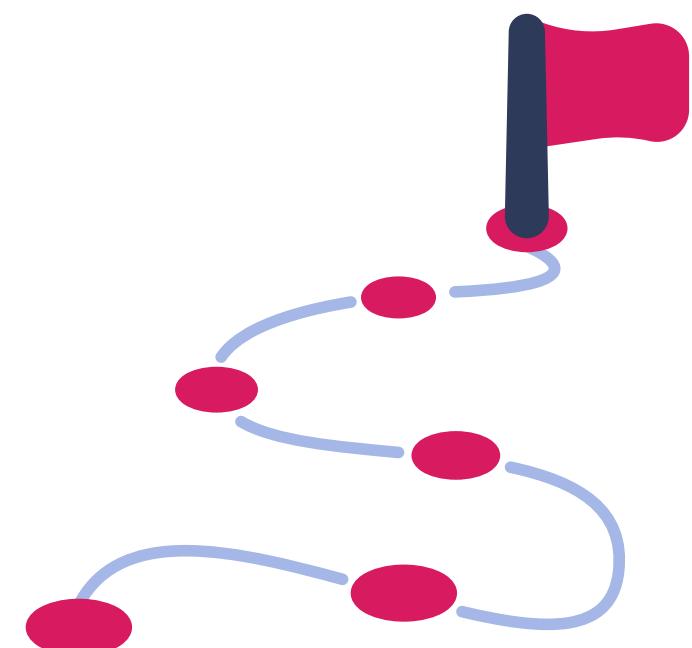
Trước khi bắt đầu phát triển ứng dụng, chúng ta cần phải:



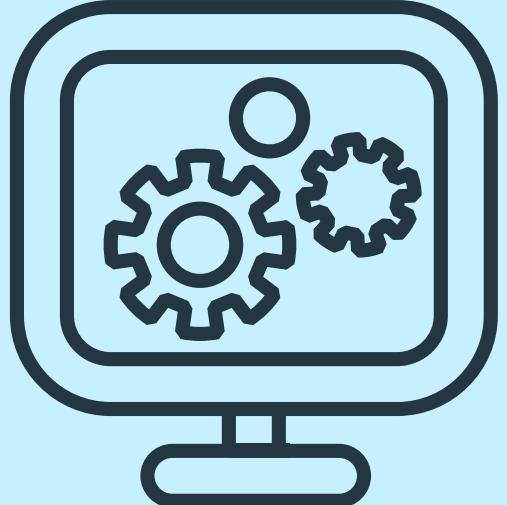
Xác định kiến trúc kỹ thuật



Lựa chọn bộ công nghệ

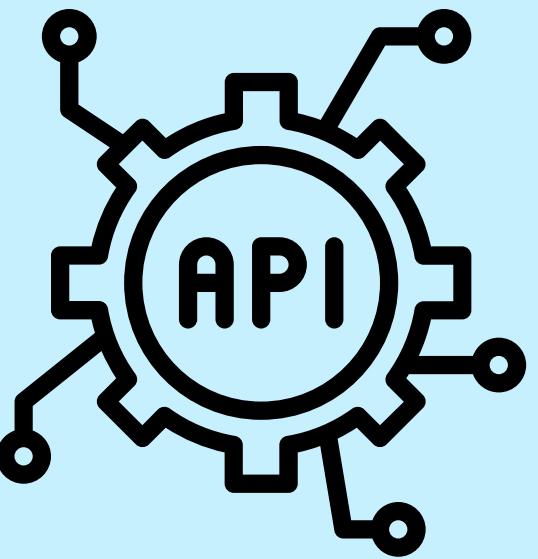


Xác định các cột mốc phát triển



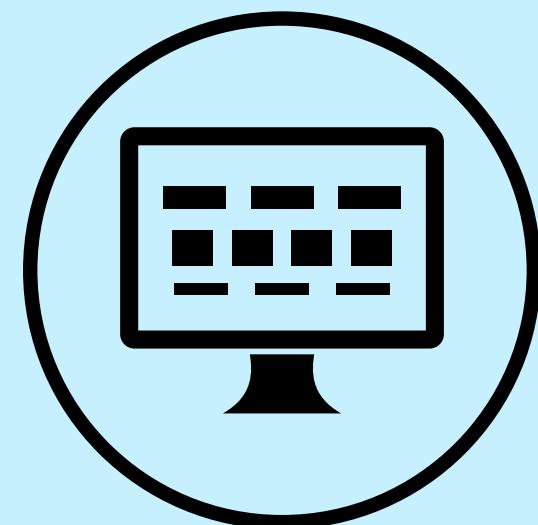
Back-end/Server technology

Đây là trái tim của ứng dụng, nơi quản lý dữ liệu, xử lý nghiệp vụ và cung cấp dịch vụ cho ứng dụng di động.



API(s)

Giao diện lập trình ứng dụng (API) giúp front-end và back-end giao tiếp một cách hiệu quả.



Mobile App Front-End

Phần này là giao diện người dùng, nơi người dùng tương tác với ứng dụng.

Cấu trúc cơ bản của dự án ứng dụng di động

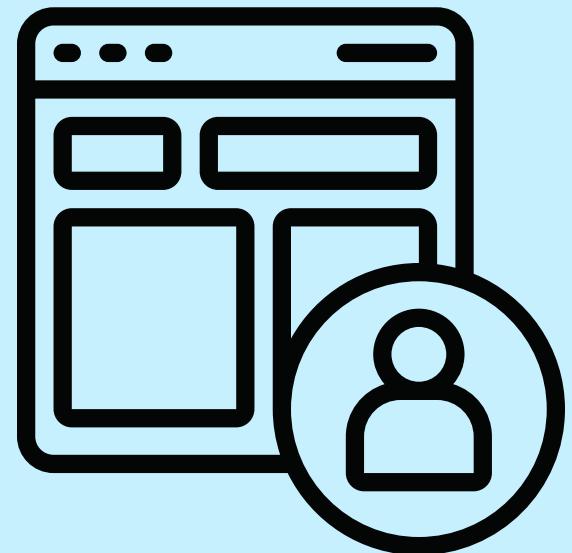
5

Kiểm thử ứng dụng

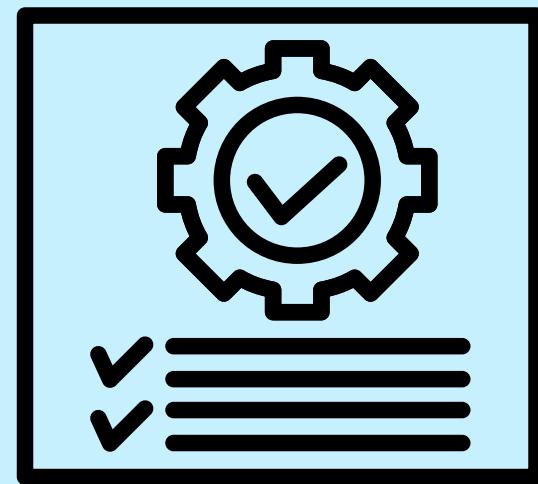
Đảm bảo chất lượng ứng dụng thông qua kiểm thử là bước quan trọng để ứng dụng hoạt động ổn định, dễ sử dụng và an toàn.



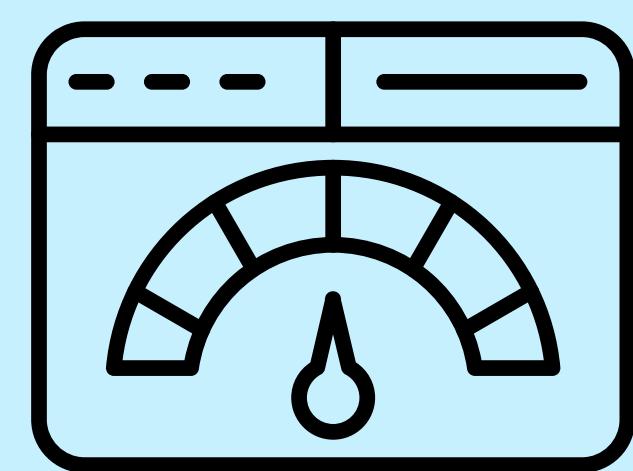
Các loại kiểm thử cần thiết



User experience
testing



Functional
Testing



Performance
Testing



Security
Testing



Device & Platform
Testing

TRIỂN KHAI VÀ HỖ TRỢ



Chuẩn bị trước khi phát hành

Triển khai ứng dụng

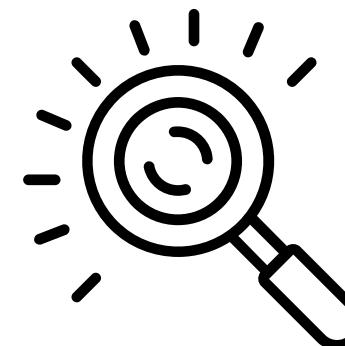
Việc phát hành một ứng dụng di động yêu cầu bạn đăng ứng dụng của mình lên các cửa hàng ứng dụng (như Google Play Store và Apple App Store).

Để có thể phát hành ứng dụng, bạn cần một tài khoản phát triển với các cửa hàng ứng dụng.

Thể loại



Mô tả



Biểu tượng khởi chạy



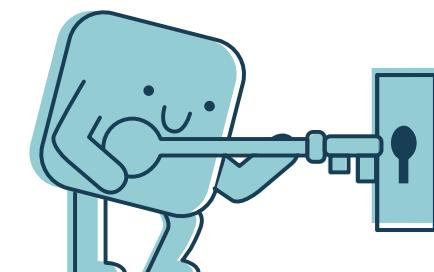
Tên ứng dụng



Ảnh minh họa
ứng dụng

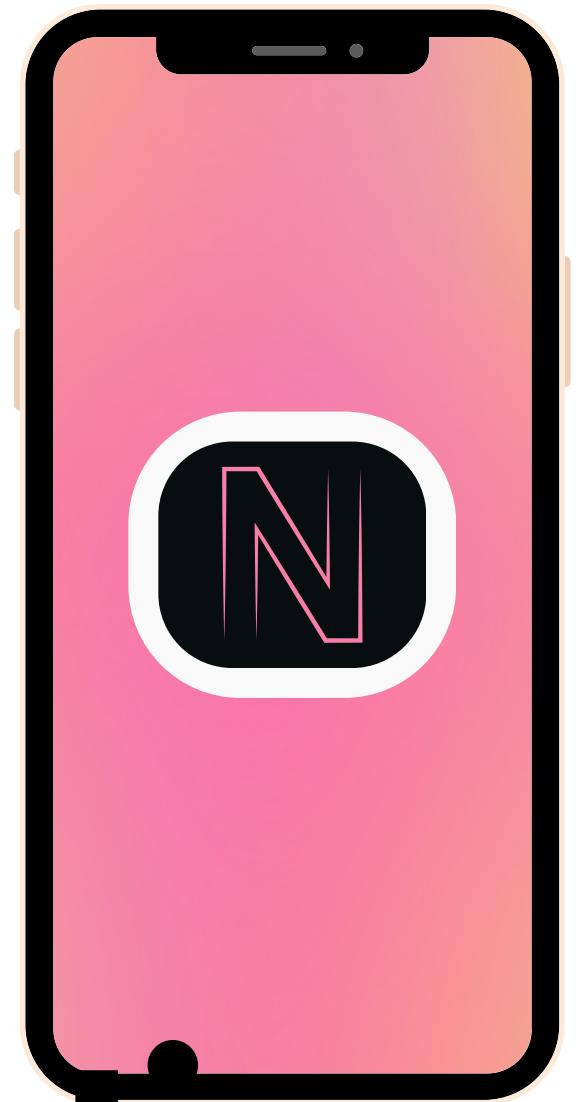


Từ khóa



PHÂN LOẠI





**native
app**

ỨNG DỤNG GỐC



**web
app**

ỨNG DỤNG WEB



**hybrid
app**

ỨNG DỤNG LAI

Native Apps

Công nghệ được sử dụng

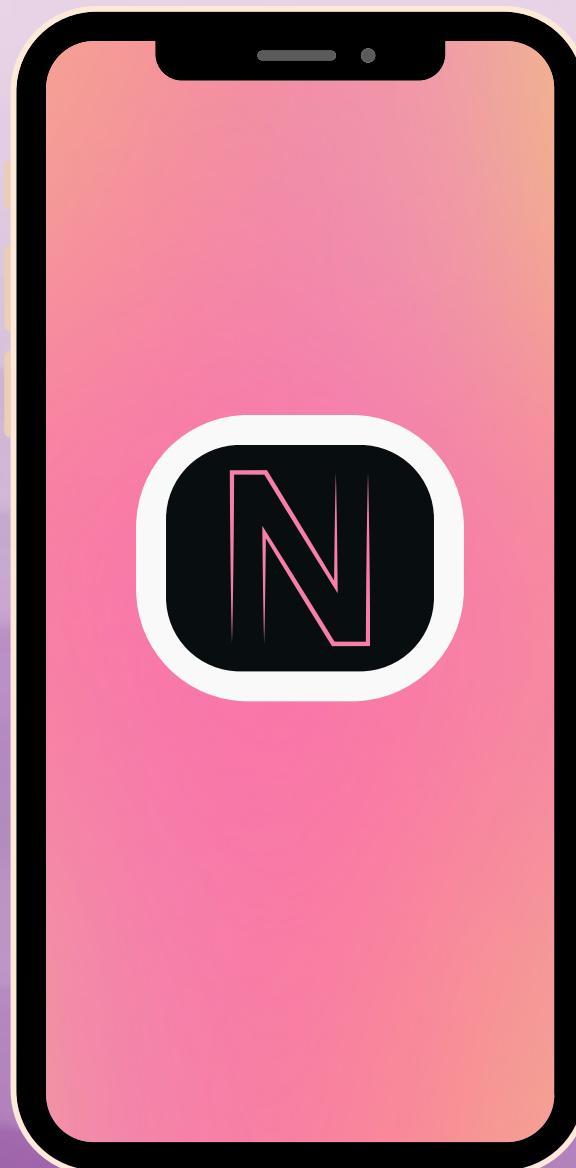
Java, Kotlin, Python, Swift, Objective C, etc.

“ Ưu điểm ”

1 Faster, Better Performance

2 Native UI

3 Can access device features



“ Hạn chế ”

1 Higher cost to maintain

2 Takes up space in the device

3 Updates must be downloaded

Web Apps

Công nghệ được sử dụng

HTML5, CSS, JavaScript, Ruby, etc.

“ Ưu điểm ”

1 Web-based so performs
on all devices

2 Easier to maintain

3 Users don't run out
of storage



“ Hạn chế ”

1 Dependent on a
browser

2 Needs an Internet
connection

3 May not always integrate
with device hardware

Hybrid Apps

Công nghệ được sử dụng

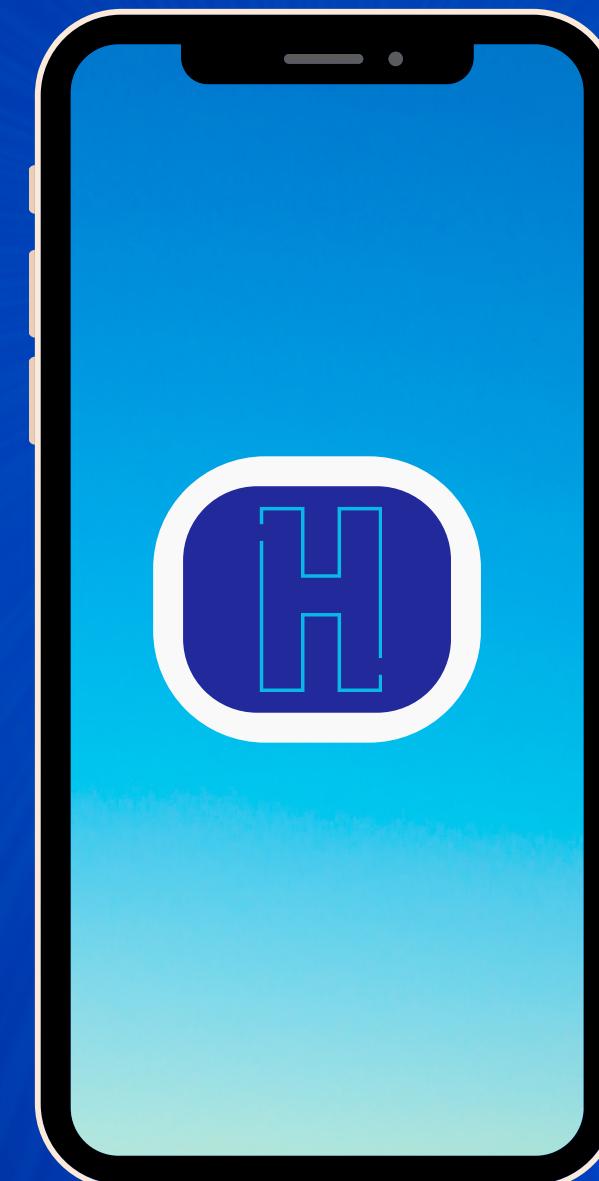
Ionic, Objective C, Swift, HTML5, etc.

“ Ưu điểm ”

1 Quicker/cheaper
to build

2 Load quickly

3 Less code to maintain



“ Hạn chế ”

1 Lacks power of
Native app

2 Slower since it has to
download each element

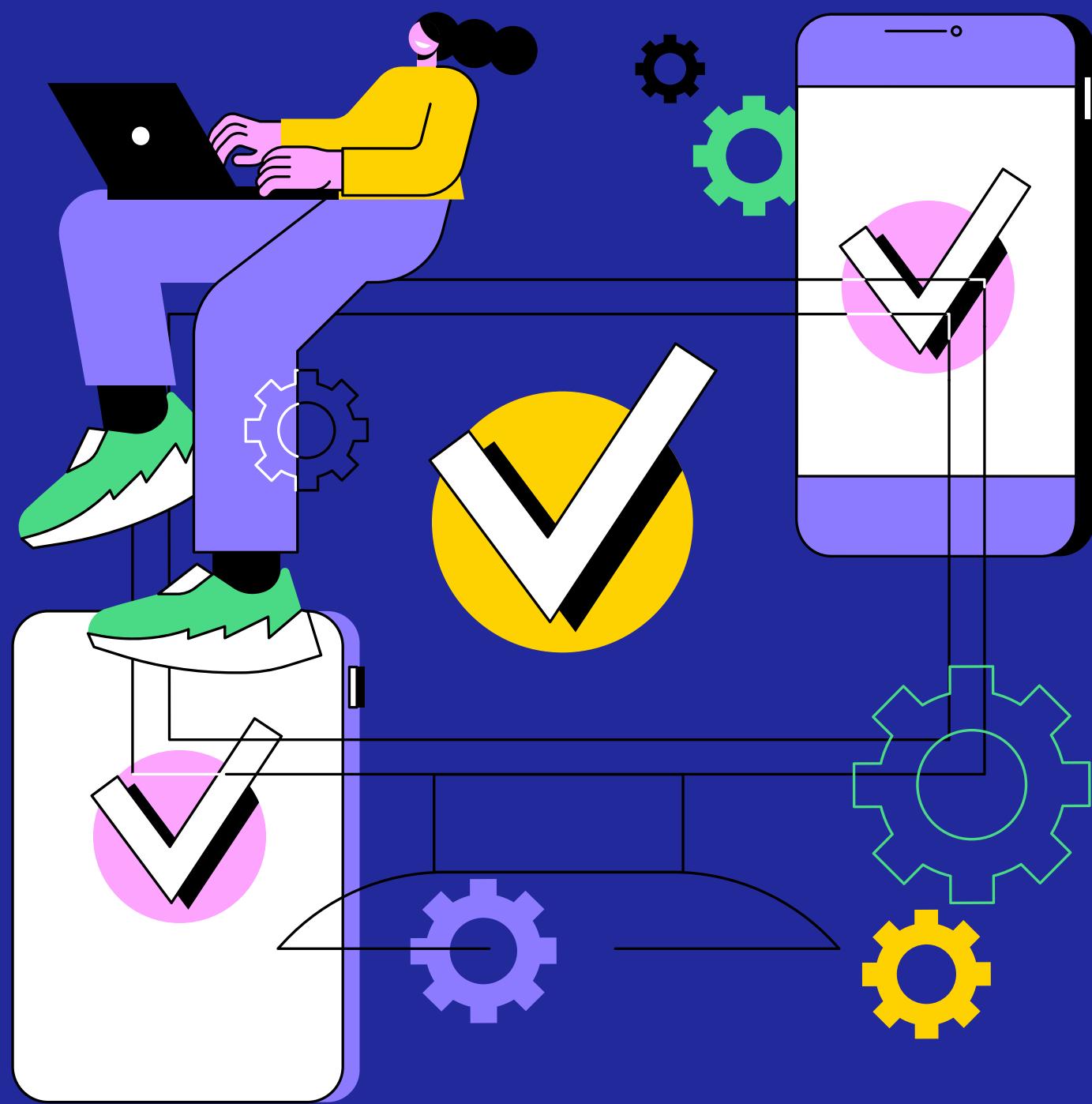
3 Certain features might not
be usable on devices

Comparison: Native vs. Hybrid vs. Web App Development

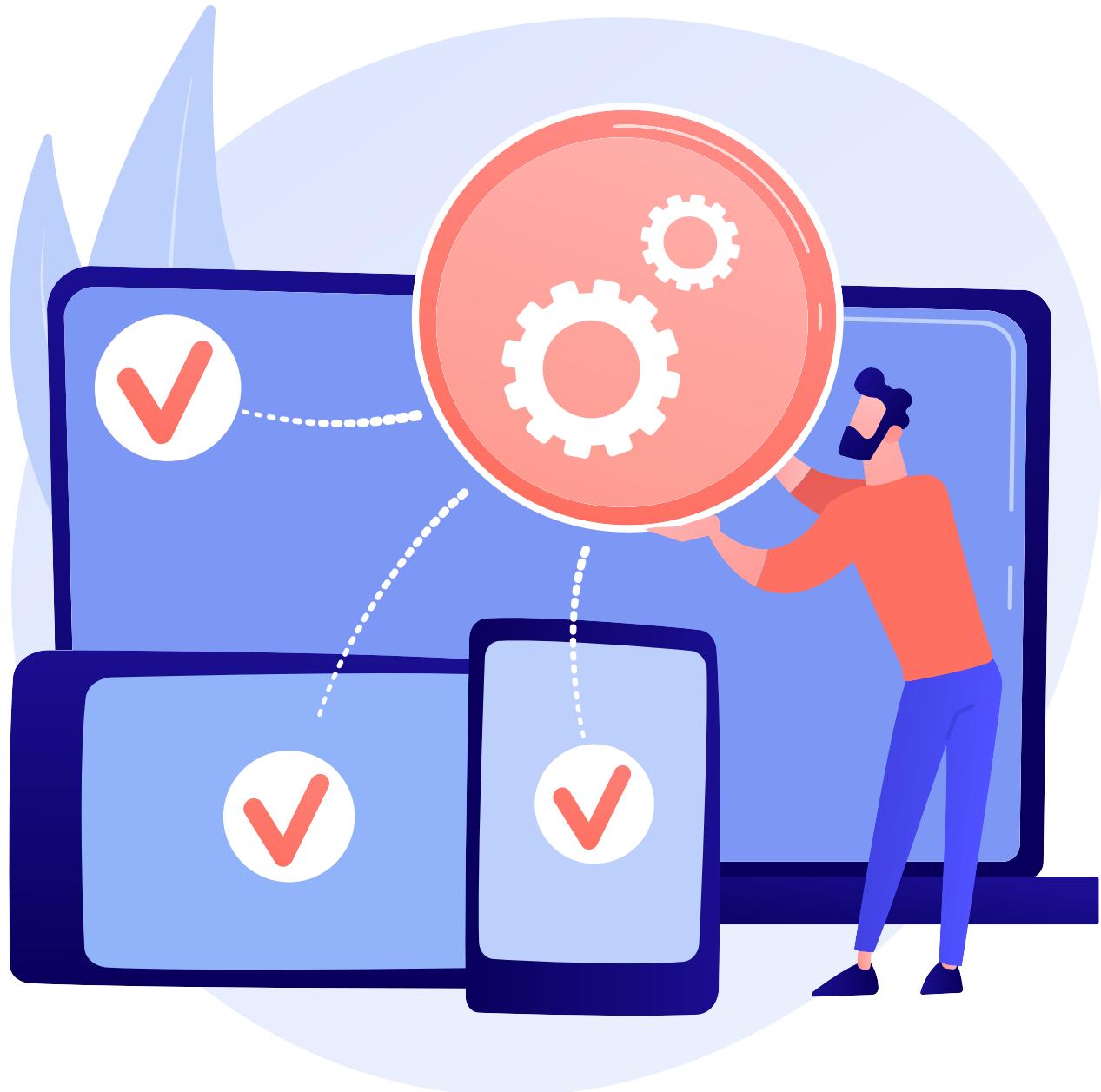
App Type	Native	Web	Cross-Platform
Performance	A +	B -	A
UI/UX	A +	B	A +
Code Reuse	—	—	+
Development	The most difficult	The simplest	Medium difficulty
Development cost	High	Lower	Cheaper than Native because of the reusable code
Offline Functionality	+	—	+
Upgrades	Automatic upgrades	Easily updatable with the Internet	Difficult to upgrade
Tools	Xcode, AppCode, Android Studio, Atom, Android IDE - AIDE, IntelliJ IDEA	Django	React Native, Flutter
Examples	Google Maps, Twitter, LinkedIn, Telegram, Facebook, WhatsApp, Artsy, Pinterest	OLX, Twitter Lite, AliExpress, Forbes, Pinterest	Instagram, Facebook Ads Manager, Storyo, Skype, Wix, The New York Times

Nguồn: [Listlink](#)

CROSS PLATFORM FRAMEWORK



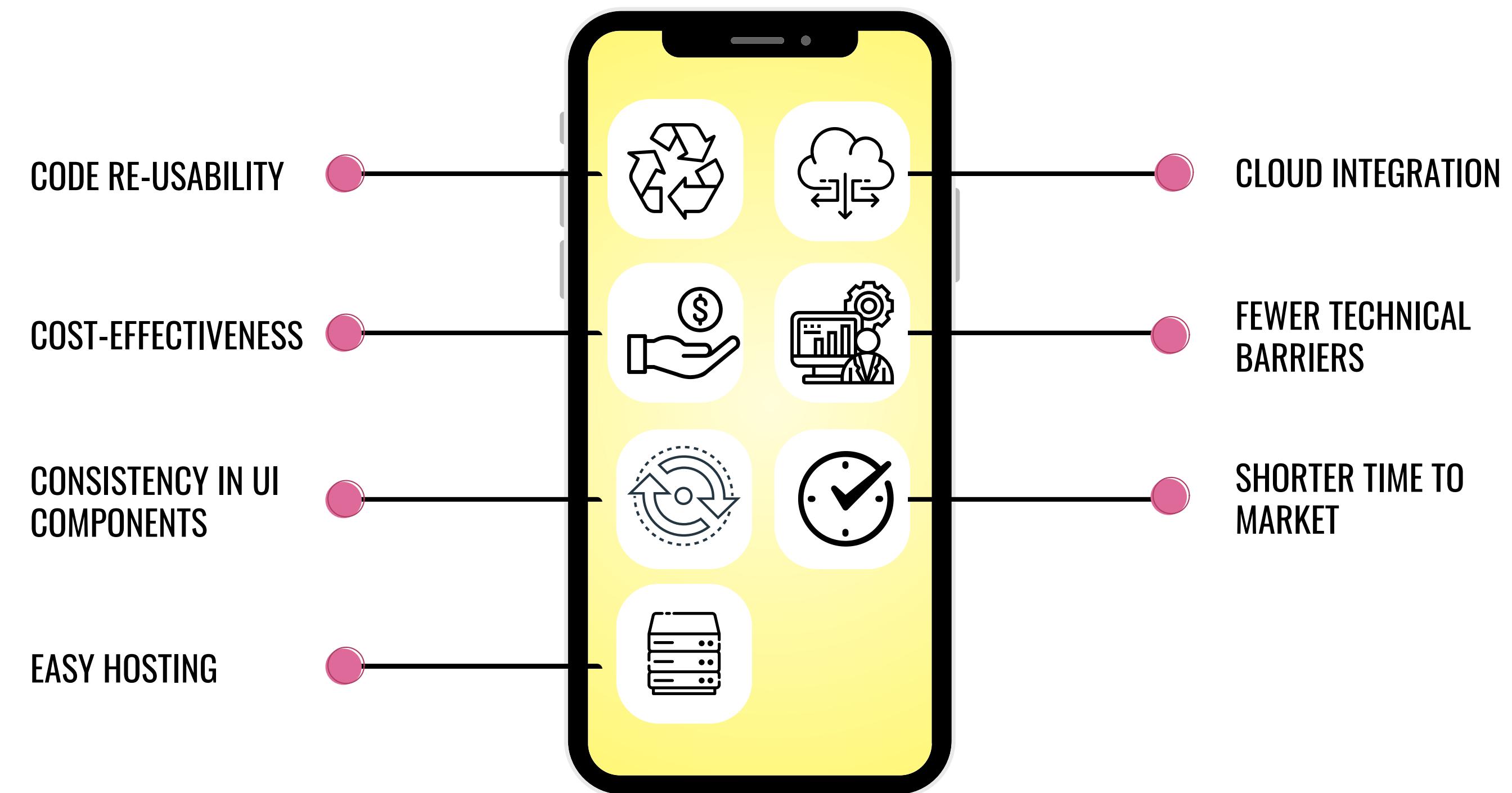
Cross platform



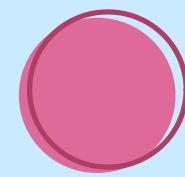
Là một thuật ngữ để chỉ việc xây dựng một ứng dụng duy nhất có thể chạy trên nhiều nền tảng hệ điều hành khác nhau, thay vì phát triển các phiên bản ứng dụng riêng cho từng nền tảng.

Hiểu đơn giản hơn là chỉ với một bản code, bạn có thể phiên dịch và sử dụng nó trên đa nền tảng ứng dụng, giúp bạn tiết kiệm được thời gian và công sức đáng kể.

Lợi ích của Cross Platform Framework



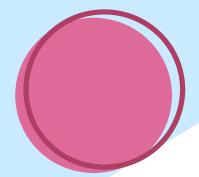
Lợi ích của Cross Platform Framework



CODE-REUSABILITY

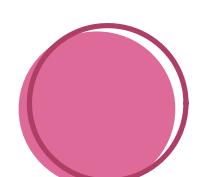
Lập trình một lần và có thể tái sử dụng cùng một mã nguồn để chạy ứng dụng trên các nền tảng khác.

-> Điều này giảm bớt các công việc lặp đi lặp lại, và do đó, tiết kiệm chi phí hoạt động.



QUICKER DEVELOPMENT TIME

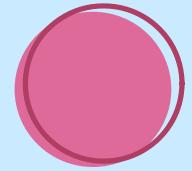
Việc tạo một mã nguồn duy nhất cho nhiều nền tảng giúp giảm thiểu công sức phát triển từ 60 đến 80%



REDUCED COSTS

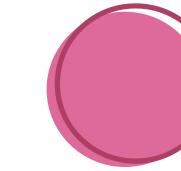
Bớt thời gian, nguồn lực và công sức bỏ ra cho việc phát triển ứng dụng di động có mối quan hệ trực tiếp với việc giảm chi phí.

Lợi ích của Cross Platform Framework



EASY CLOUD INTEGRATION

Lợi ích thêm từ việc sử dụng Cross Platform Framework là việc dễ dàng tích hợp với dịch vụ lưu trữ trên đám mây.

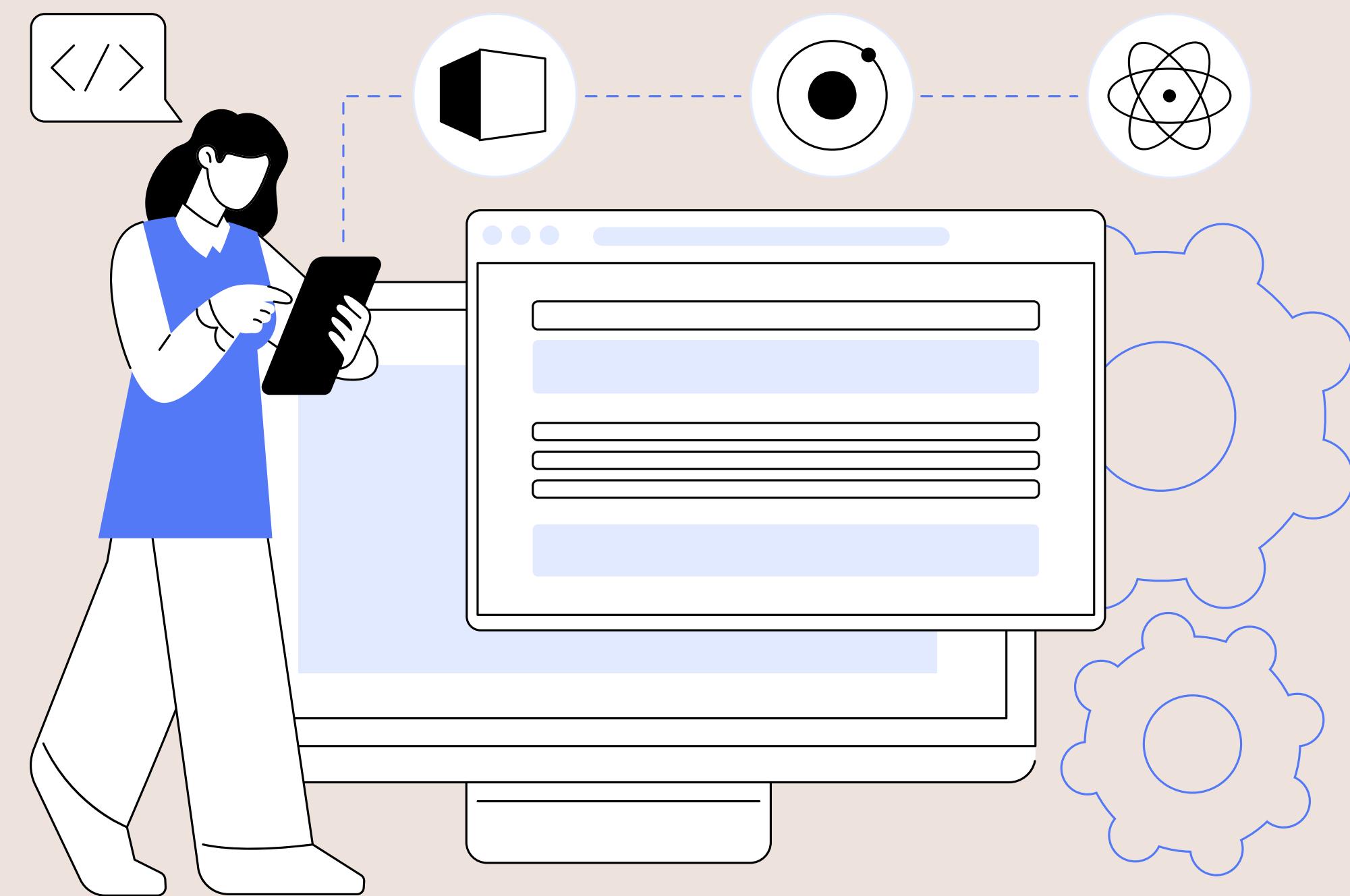


EASY MAINTENANCE

Với một mã nguồn duy nhất, việc bảo trì, cập nhật và thay đổi trở nên dễ dàng hơn.

Các cập nhật và thay đổi, một khi đã thực hiện, có thể được đồng bộ nhanh chóng trên tất cả các nền tảng và thiết bị, giúp tiết kiệm thời gian, tiền bạc và công sức.

CÁC FRAMEWORK PHỔ BIẾN CHO CROSS PLATFORM





Flutter



Do Google giới thiệu năm 2017.



Sử dụng như ngôn ngữ hướng đối tượng, thực hiện bằng Dart.

Đặc điểm:

- GPU di động cho giao diện mạnh mẽ.
- Khung phản ứng tự động cập nhật giao diện.
- Lựa chọn lý tưởng cho MVP: phát triển nhanh và tiết kiệm chi phí.
- Cơ chế tái tạo widget và điều chỉnh mã hiệu quả.
- Động cơ đồ họa tích hợp (GMD), không cần tạo giao diện riêng cho Android và iOS.

Famous Flutter Apps



Alibaba



Reflectly

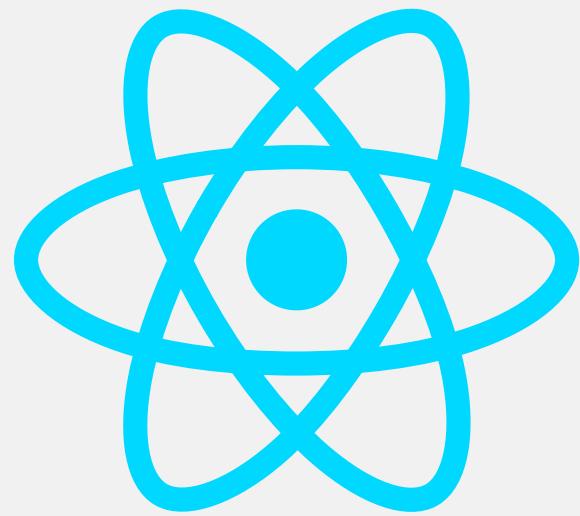


Google Ads



Birch Finance

Nguồn: [appinventiv](#)



React



Cross Platform Framework dựa trên JavaScript và React.



Hỗ trợ viết mô-đun bằng Objective-C, Swift hoặc Java.

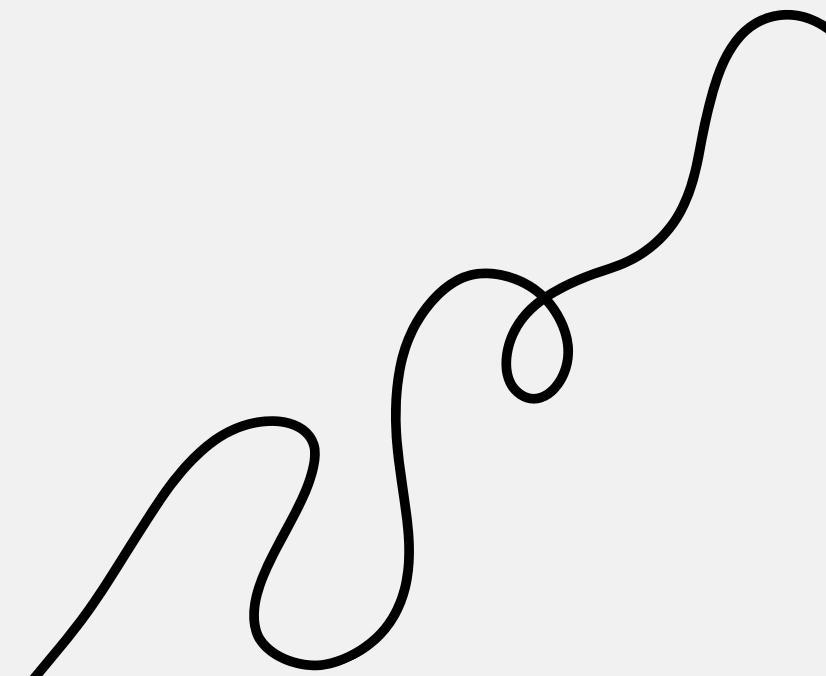


Khả năng thực hiện các hoạt động nặng như chỉnh sửa hình ảnh và xử lý video.

Đặc điểm:

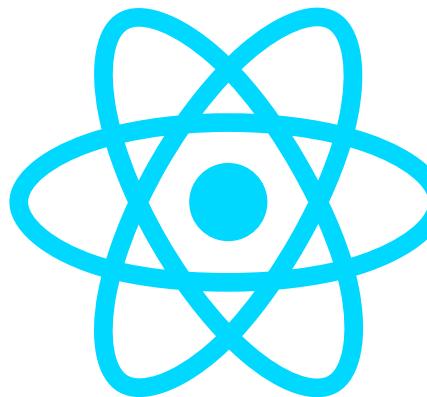
- Mã nguồn mở, có cộng đồng hỗ trợ lớn.
- Tốc độ hiển thị giao diện nhanh.
- Yêu cầu mã hóa một lần (WORA) cho Android và iOS, giảm thời gian phát triển và chi phí.
- Tương thích cao với các plugin bên thứ ba, ví dụ: Google Maps.
- Tập trung vào UI, mang lại giao diện phản hồi nhanh, giảm thời gian tải và mang lại trải nghiệm mượt mà cho ứng dụng.

React Native là lựa chọn ưa thích không chỉ của các nhà phát triển mà cả các doanh nghiệp.



Statistics on React Native

React Native



StackOverflow Voices

57.9%

Github Contributors

9.1k

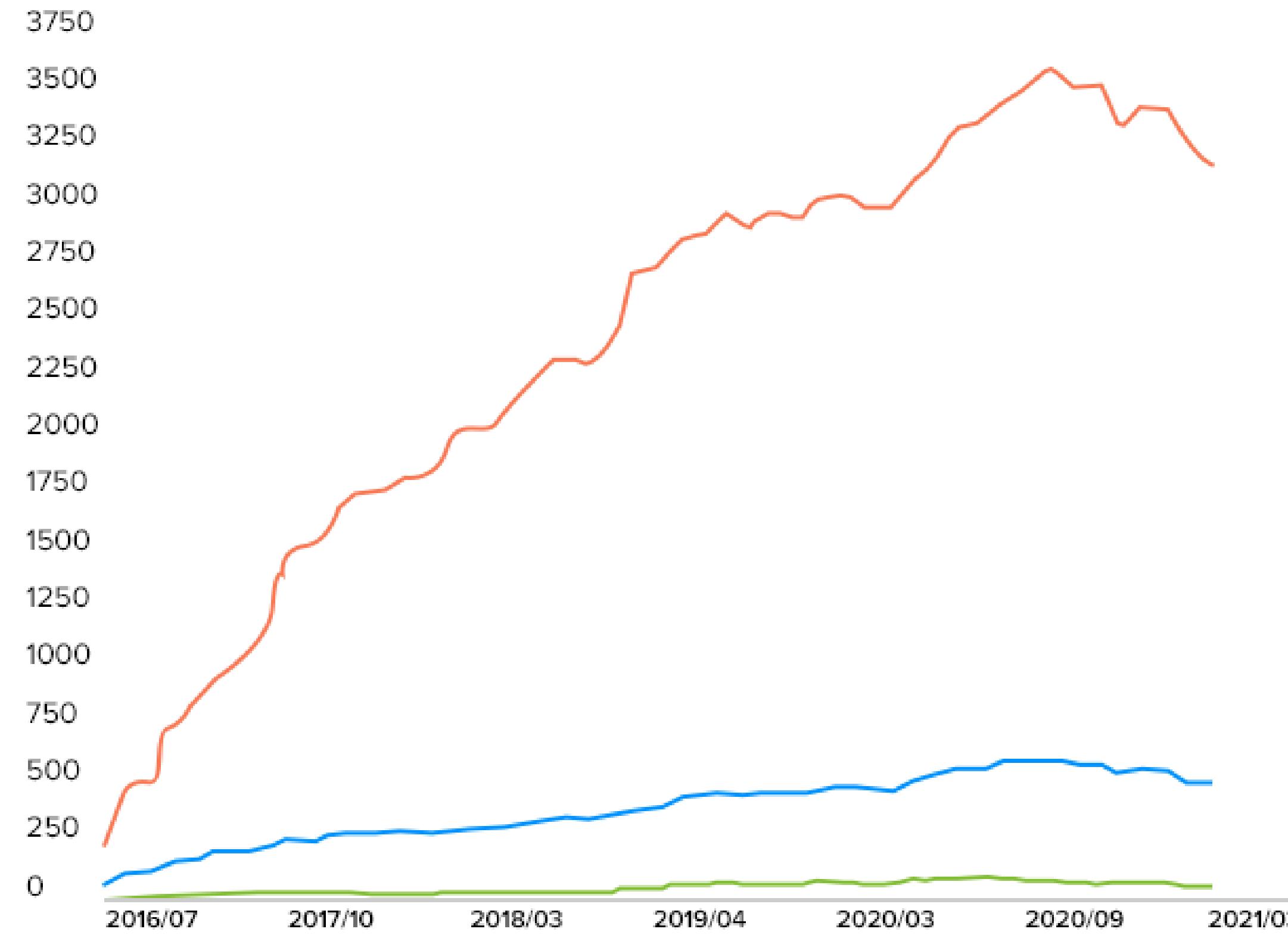
Google Trends Queries per day

58



-  **Khung ứng dụng đa nền tảng phổ biến, dựa trên AngularJS.**
-  **Cho phép phát triển ứng dụng bằng sự kết hợp của HTML5, JavaScript, CSS và Cordova để truy cập native platform controllers.**
-  **Tạo giao diện người dùng sáng tạo và tính năng thân thiện, với trải nghiệm giống như ứng dụng bản địa.**
-  **Dựa trên khung UI SAAS dành riêng cho hệ điều hành di động, cung cấp nhiều thành phần UI.**

Ionic Usage Statistics



Nguồn: [appinventiv](#)



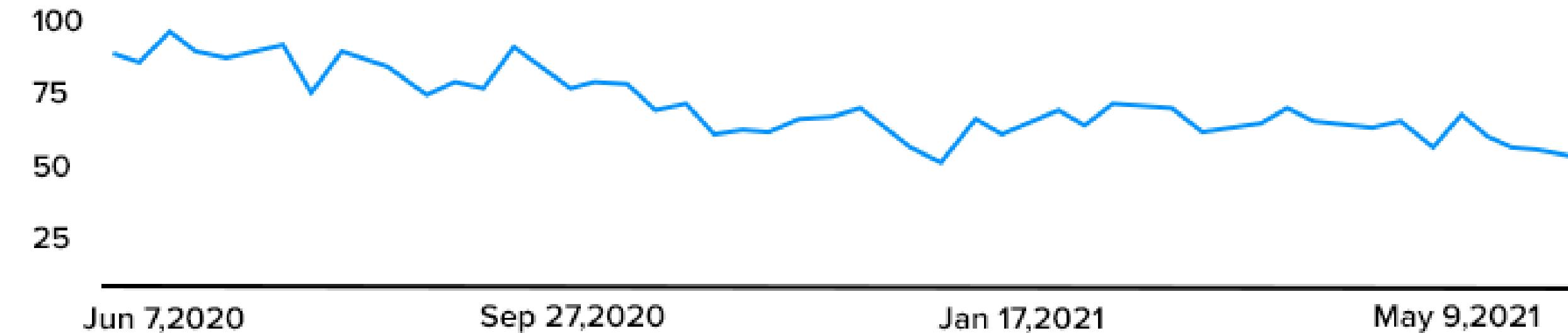
Đặc điểm:

- Mã nguồn mở và linh hoạt, cho phép các nhà phát triển chỉnh sửa mã dễ dàng.
- Dựa trên khung UI SAAS dành riêng cho hệ điều hành di động, cung cấp nhiều thành phần UI.
- Dựa trên AngularJS, giúp mở rộng cú pháp HTML và tích hợp tính năng vào ứng dụng.
- Sử dụng plugin Cordova để truy cập các tính năng bên trong của thiết bị như Camera, GPS.
- Cung cấp trải nghiệm giống ứng dụng bản địa, giúp ứng dụng hoạt động hoàn hảo trên nhiều nền tảng.



Xamarin

Xamarin Interest Over Time





Xamarin



Cross Platform Framework sử dụng C# và .Net để phát triển ứng dụng cho Android, Windows và iOS



Cho phép tái sử dụng đến 90% mã khi xây dựng ứng dụng cho ba nền tảng khác nhau.



Mang lại trải nghiệm giống native app nhờ API tuyệt vời của mình.

Đặc điểm:

- Sử dụng C# - một ngôn ngữ phát triển ứng dụng đa nền tảng hiện đại.
- Cung cấp chức năng ứng dụng mức , giảm thiểu vấn đề tương thích phần cứng nhờ plugin và API.
- Hỗ trợ tích hợp trực tiếp của các thư viện Objective-C, Java và C++, cho phép tái sử dụng nhiều mã nguồn từ bên thứ ba.
- Giảm thời gian và chi phí phát triển ứng dụng di động, hỗ trợ WORA và có một bộ thư viện lớn.
- Cung cấp kiểm tra tại thời gian biên dịch mạnh mẽ, giúp giảm lỗi thời gian chạy.
- Có giao diện và điều khiển người dùng bản địa tuyệt vời, giúp phát triển ứng dụng giống native app.



APACHE CORDOVA



Cung cấp giải pháp dựa trên đám mây, cho phép chia sẻ ứng dụng trong quá trình phát triển để nhận phản hồi từ các nhà phát triển khác.



Cross Platform Framework sử dụng CSS, JavaScript và HTML5 để phát triển ứng dụng di động.



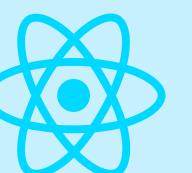
Phát triển ứng dụng ấn tượng dựa trên các công nghệ web hiện có và hỗ trợ đầy đủ các tính năng bản địa như GPS, Camera, Danh bạ, Lưu trữ, v.v.



APACHE CORDOVA

Đặc điểm:

- Cho phép tạo ứng dụng đa nền tảng sử dụng các công nghệ web hiện có như HTML5, CSS3 và JavaScript.
- Hỗ trợ sử dụng một mã nguồn duy nhất để tạo ứng dụng cho nhiều nền tảng như iOS, Android, Windows Phone, BlackBerry, v.v.
- Theo dõi kiến trúc có tính chất plugin, giúp mở rộng quyền truy cập vào **native device APIs** một cách linh hoạt.

	 React Native	 Xamarin	 Ionic	 Flutter	 Cordova
Programming language	JavaScript + Swift, Objective C or Java	C# with .net environment	HTML, CSS and JavaScript + Typescript	Dart	HTML, HTML5, CSS and JavaScript
Performance	Close-to-native	iOS/Android Close-to-native	Moderate	Amazing	Moderate
GUI	Use native UI controllers	Use Native UI Controllers	HTML, CSS	Use Proprietary Widgets and deliver amazing UI	HTML, CSS
Market and Community	Very strong	Strong	Strong	Not very popular	Average
Market	All apps	Simple apps	Simple apps	All apps	Simple apps
Use cases	90% of code is reusable	96% of code is reusable	98% of code is reusable	50-90% (approx.) of code is reusable	50-80% (approx.) of code is reusable
Popular App	Facebook, Instagram, Airbnb, UberEats	Olo, the World Bank, Storyo.	JustWatch, Pacifica, and Nationwide	Hamilton	FanReact, Untapped, Hockey Community
Pricing	Open-source	Open-source + Paid as well	Open-source + Paid as well	Open-source	Open-source
Creators	Facebook	Xamarin Inc.	Drifty Co.	Google	Adobe

**Cảm ơn mọi người
đã theo dõi !**

