

## Mobile Application Development

### Android SDK

The Android SDK (Software Development Kit) is a set of tools and resources that developers use to create and publish Android apps. It includes an Integrated Development Environment (IDE) called Android Studio, which is used for writing, testing, and debugging code. Additionally, it includes libraries, sample code, and documentation to help developers get started with app development.

### Android Emulator

The Android emulator is an Android Virtual Device (AVD), which represents a specific Android device. We can use the Android emulator as a target device to execute and test our Android application on our PC.

The Android emulator provides almost all the functionality of a real device. We can get the incoming phone calls and text messages. It also gives the location of the device and simulates different network speeds. Android emulator simulates rotation and other hardware sensors. It accesses the Google Play store, and much more.

What is an IDE?

An integrated development environment (IDE) is software for building applications that combines common developer tools into a single graphical user interface (GUI). An IDE typically consists of:

- **Source code editor:** A text editor that can assist in writing software code with features such as syntax highlighting with visual cues, providing language specific auto-completion, and checking for bugs as code is being written.
- **Local build automation:** Utilities that automate simple, repeatable tasks as part of creating a local build of the software for use by the developer, like compiling computer source code into binary code, packaging binary code, and running automated tests.
- **Debugger:** A program for testing other programs that can graphically display the location of a bug in the original code.

**Android Studio** is the official Integrated Development Environment (IDE) for android application development. Android Studio provides more features that enhance our productivity while building Android apps.

## **Gradle**

Gradle is a build automation tool known for its flexibility to build software. A build automation tool is used to automate the creation of applications.

The building process includes compiling, linking, and packaging the code. The process becomes more consistent with the help of build automation tools.

## **Mobile application concepts**

### **What is Mobile Application**

A mobile app (or mobile application) is a software application developed specifically for use on small, wireless computing devices, such as smartphones and tablets, rather than desktop or laptop computers.

### **Types of mobile applications**

#### **1. Native Apps**

Native apps are built for specific operating systems on mobile devices. That means the app can run on Android devices or Apple iOS devices but not both.

#### **Advantages**

- Native Apps Offer Speed

Because native apps are native to the platform, they work faster. Many elements come preloaded. The user data is fetched from the web rather than the entire application, and since they work with the device's built-in features, they are speedy.

- Native Apps Work Offline

Native apps work even if there is no internet connectivity. So, in situations where you are stuck somewhere with limited or no service, such as an airplane, underground tunnel, or subway, native apps are accessible.

- Native Apps Provide a Recognizable Look and Feel

Native applications are a nuanced version of their device's default apps. When a user performs some functions, he quickly understands the natural flow of the application because it is similar to apps already on the device.

- Native Apps Maintain Aspect Ratios

Aspect ratio is the ratio of width to height of different screens. It is an important factor that determines the quality of an image. Many apps collapse and start functioning improperly upon changing the device's size and shape. Native apps have better control over the orientation, size, and resolution of the app.

### **Disadvantages**

- Lengthy Downloading Process

Native apps are ready to use only when users download them from the app store (Google Play or Apple App Store). This involves a lot of steps – going to the app store, finding the application, accepting its terms and conditions, and then finally downloading it.

- No Flexibility

Developers have no flexibility regarding the platform for developing native apps. Developers have to code for one platform at a time, with separate coding for Android and iOS. Much of the time, if you are hiring app developers for your native app idea, you'll have to hire two teams of developers – one for Android and one for iOS.

- Expensive Development

The programming used in native apps is quite tricky, and developers who code in this language are scarce. Thus, native app development requires more Labor, which adds to the time and cost of development.

- Time-Consuming Development

Every platform, such as iOS and Android, needs a separate set of codes, which means more time is required, as it is equivalent to coding for two different apps.

- Native Apps Require Frequent Upgrades

If a bug is fixed in native apps or a new update is about to be launched, developers have to first submit the updates to the app store and then hope that users will update their apps to the

new versions. If users don't update their app - maybe they didn't notice the update, or they don't have enough storage space - those users might abandon the app due to unfixed glitches.

## 2. Hybrid Apps

Combines elements of both native and web applications. They are built using web technologies but are wrapped in a native app shell, allowing them to be distributed through app stores.

Instagram and Gmail are 2 extremely popular examples of hybrid apps. These are technically web apps, but they behave and act like native apps. They run within an app-embedded web browser. Unlike native apps, however, hybrid apps can function on multiple platforms and operating systems.

### Advantages

- **Instant access:** Unlike native apps, hybrid apps can be loaded through a link, which you can distribute through social media and email marketing. By immediately getting users to interact with your app, you're bypassing common friction points (app stores).
- **Cross-platform:** The most obvious reason why businesses opt for hybrid apps is because they're built for multiple platforms; so long as your users keep their web browser updated, they'll be able to access your software on all of their devices.
- **One-size-fits-all:** Hybrid apps can cut development time, as most of the functionality can be built with existing languages like HTML, which are easier to develop and test on multiple platforms. An iOS app that takes three months to develop might be ready in a couple of weeks if you opt for a hybrid app instead, helping you launch faster.
- **Offline access:** On some hybrid apps, data can be stored locally for offline access – ideal if you're offering functionality like mapping, video tutorials, or user manuals.
- **Easy maintenance:** Thanks to HTML and other web languages, hybrid apps are relatively easy to maintain; you don't have to worry about new operating system releases that could change the infrastructure of your application or limit functionality.
- **Affordable:** Hybrid apps can be developed quickly and require less maintenance than iOS or Android apps, and are therefore much cheaper for small businesses.

- **No need for approval:** If you've ever developed an app, you'll know how difficult it can be to overcome Apple and Google's strict guidelines and operating system protocols – often requiring lots of back and forth. Web apps don't require approval, making them the ideal choice for businesses looking to launch their software quickly.
- **Automatic updates:** Rather than having to submit changes to your app to the App Store and Play Store, and then wait for approval, your developer can instantly make amendments to your hybrid app and they'll appear immediately. Every time a user logs onto your hybrid app, they'll access the latest version, enhancing security.

### Disadvantages

- **Limited functionality:** Although hybrid apps are considerably more flexible as mobile web browsers gain new functionality and performance improvements, your developer won't be able to access specific hardware features like augmented reality.
- **Browser needed:** Users will need to open their web browser to run your app, which requires more steps and a URL or bookmark. With a native iOS or Android app, your software will always appear on your users' home screens.
- **Discoverability:** As we've just touched upon, getting users to find and interact with your app can be tough without an App Store listing. Be creative with marketing and encourage users to bookmark your app or save it as an app shortcut on their phone.
- **User management:** Unless you ask users to create an account and log in whenever they load your app, you'll struggle to collect usage and performance metrics which could improve your app. On the other hand, adding a login screen every time users load your app could limit engagement and increase abandonment and bounce rates.

Native apps	Hybrid Apps
Developed for a specific platform (iOS or Android) using platform-specific programming languages (Swift or Objective-C for iOS, Java or Kotlin for Android)	Developed using web technologies like HTML, CSS, and JavaScript

Provides better performance and faster load times because it is optimized for the specific platform	Slower performance and longer load times because it is running in a web view
Can access all the features and capabilities of the device, such as camera, contacts, and GPS	Limited access to device features and capabilities
Requires separate development for each platform, which can be time-consuming and expensive	Can be developed once and deployed on multiple platforms, reducing development time and cost
Provides a better user experience because it is designed specifically for the platform	User experience may be compromised because the app is designed to work on multiple platforms
More expensive to develop but can deliver higher returns on investment due to better performance and user experience	Less expensive to develop but may not deliver the same level of performance and user experience as native apps

### 3. Web Apps

Web apps run in a web browser. They are accessed on a mobile device and don't require downloading on the part of the user. Commonly, developers use traditional web development languages like HTML5, JavaScript, CSS, and others to create web apps.

#### Advantages

- A Better User Experience – With responsive design, it's a lot easier and cheaper to make a web based system user friendly across multiple platforms and various screen sizes.
- Flexible Access – Employees can work from anywhere with internet access.
- Client Secure Login – Impress clients with a modern web portal and improve customer service with automated processes.
- Easy Setup – It takes a couple of minutes to setup a new user; provide a URL, username and password and they're away.
- Always Up To Date – As everyone is accessing the same version of the web app via a URL, they will always be accessing the most up-to-date version of the software.

- Storage Increase – With the availability of the cloud, storage space is virtually infinite.

### Disadvantages

- Internet reliance – Whilst 4G & Wi-Fi internet access is available in many locations, if you happen to lose connection you will not be able to access your web app.
- Security – Whilst many business people may believe that data is less secure in a cloud environment, we beg to differ... There are ways in which you can reduce risk of a data breach, such as SSL enforcement for a secure HTTPS access to your app.
- Reduced Speed – It's likely that a web app will operate at a slightly slower speed than one hosted on a server locally.
- Browser Support – Unfortunately, we don't all use the same browser. This means during development you'll need to ensure your app is supported across a variety of browsers.

### Mobile application development platforms

When it comes to mobile applications, the two most famous operating system that comes to our mind are Android by Google and iOS by Apple.

#### What is Android?

Android is an open-source mobile operating system introduced by Google. It is the most widely used operating system supporting a wide range of devices. Android is written in Linux kernel with the help of programming languages like **C++** and **Java**. It is built particularly for tablets and touchscreen mobile devices.

An Android application is a software developed to run on an Android device. It is an APK (Android Package) file, Zip archive which contains app source code, resources, and metadata.

#### What is an Android Mobile App Development Platform?

An **Android app development platform** provides a set of tools, software, debugging tools, libraries, etc. to develop an Android application. All these elements provided by the **mobile app development framework** along with source code are used to develop apps for various mobile devices. This strong foundation is useful in developing apps with the help of in-built modules that provide functionalities like networking, user interface, storage, etc.

## Advantages of Mobile App Development Platforms

- **Cost efficient:** With the help of app development platforms, developing apps become easy at less cost and can be distributed across multiple platforms. This saves the cost of hiring different teams for each platform.
- **Ease of integration:** These platforms provide various libraries and resources that can be used in apps instead of writing complex codes.
- **Enhanced performance:** Platforms like Flutter and React Native have enhanced performance than the others. They focus on improving the user experience as well as making the app fast and responsive.
- **Built-in components:** They provide built-in features that can be customized according to the app's requirements saving both time and effort for the developers.

## Android App Development Platforms

### 1. Flutter

**Flutter** is an open-source and cross-platform Android app development platform. It was created by Google that enables developers to develop apps for Android, Windows, iOS, and the web with the help of a single codebase. It is written in the Dart language. It contains widgets, debugging tools, a rendering engine, and tools to integrate APIs to create apps. Applications such as Google Ads, Google Pay, eBay Motors, etc. are developed using [Flutter](#).

#### Features:

- *Flutter has a **Hot Reload** feature because of which only the changes will get loaded instead of the entire application. This is very useful in saving time as output can be seen in **real-time** with **less compilation** time.*
- *Various backend features and services can be integrated into the app like user authentication (through **Google**, **Facebook**, **Email/Password**, **Phone**, **Microsoft**, [GitHub](#), etc), **cloud services**, **storage**, and push notifications.*
- *Provides various animations and **UI enhancement** features to make the app look more appealing.*



- *Provides performance closely related to that of native.*

## 2. Android Studio

[Android Studio](#) is Google's official Android **Integrated Development Environment (IDE)** that provides a lot of features and functionalities to develop mobile apps. Developers can easily write the source code, debug and test the quality of their application. It supports various programming languages like **Java**, **Kotlin**, **Swift**, **C++**, etc. Google continuously updates it to meet the latest trends and requirements of developers and allows them to customize it as per your needs. It has a large community of Android developers to help in case you are stuck.

### Features:

- *Has virtual emulators as well as physical connection with devices both wired and wireless.*
- *Has Gradle build support*
- *Provides Google Play services like Google Ads, Google Maps, in-app purchases, etc.*
- *Has GitHub support so you can perform any Git command just with a few clicks*
- *Integrated with tools and frameworks for testing*

## 3. React Native

**React Native** is an open-source, accessible, cross-platform app development platform developed by **Meta** that can help developers in creating native applications for Android, iOS, and the web. It uses **JavaScript** and [React](#) library to develop, test and deploy the app. It allows fast development and deployment of mobile apps.

React Native has a syntax same as [CSS](#) but it does not use [HTML](#) or CSS. To take advantage of native views, the JavaScript thread's messages are used. Applications such as **Skype**, **Airbnb**, **Amazon Prime**, etc. are developed using React Native.

### Features:

- *The app is built using native elements and **APIs** which provide a smooth user experience.*
- *Components can be reused and third-party extensions can be easily integrated.*

- *With React Native, a **component-based GUI** can be created for **front-end applications**.*
- *A single codebase can be used across all platforms to create **platform-specific versions** of numerous functionalities.*
- *Provides a variety of services to create the backend of the application.*

#### 4. Ionic

Ionic is an open-source, cross-platform app development platform developed with the help of **Apache Cordova** and [Angular](#). It enables developers to build robust and high-feature native apps both in Android and iOS with the help of HTML, CSS, and JavaScript. It can be used to develop hybrid and **Progressive Web Apps (PWAs)** as well.

##### Features:

- *It is simple to work with as it uses technologies that most developers are already familiar with.*
- *Various components such as inputs, easy navigation, filtrations, views, etc. can be integrated to improve the user interface.*
- *The user interface is consistent and provides flexibility as well.*
- *The **Ionic Command Line Interface** is a tool that provides commands to **start, build, run, and emulate** the apps.*

#### 5. Xamarin

**Xamarin** is an open-source, cross-platform Android app development platform owned by **Microsoft**. It can build apps with the help of **C#** and **.NET** that can run on mobile operating systems like Android, iOS, and **Windows**. It provides the developers with a variety of tools to create, test, debug and deploy the apps and can be integrated with cloud services like **Microsoft Azure**.

It enables the developers to share the scripts across Windows and **macOS**. Developers can create a development environment with the help of [APIs](#), components, **backend services**, etc. by using **tools, frameworks**, and that are supported by an active community.

##### Features:

- *Developers can use the same code across multiple platforms without compromising the quality.*
- *Xamarin's intermediate layer manages the relationship between the standard code and the core system code.*
- *Less time is required in coding as C# uses few lines of code.*
- *Its integration with **Microsoft Visual Studio** can help in managing and increasing the productivity of app development.*

## 6. Native Script

**Native Script** is an open-source, native **mobile app development** platform developed by Progress. It uses CSS, JavaScript, Angular, [Typescript](#), and [Vue.js](#) to develop mobile apps. **Native Script Sidekick** is a Graphical User Interface used to create tasks via a GUI tool.

[XML](#) files are used to define platform-independent user interfaces. It is possible to construct application logic independent of the target platform using Angular and TypeScript.

### Features:

- *Native Script has direct access to iOS and Android APIs.*
- *Provides strong backend support to apps.*
- *Has a **native UI** without **Web Views**.*
- *Integration of other frameworks and libraries is easy.*
- *Uses less code and time to load the app on the system.*

## 7. Apache Cordova (Formerly Phone Gap)

**Apache Cordova** is an open-source, cross-platform app development platform that uses **HTML5**, **CSS3**, and JavaScript to develop mobile apps. It supports operating systems like Android, Windows, and iOS. It provides various plugins to developers to integrate hardware features like **camera**, **GPS**, **accelerometer**, etc. to provide an interface like native.

### Features:

- *Cordova utilizes a single **codebase** to develop mobile apps and **third-party tools** for management.*
- *The development process is very smooth.*
- ***Cordova Command Line Interface** is a tool that is used to build and install **various plugins** used across different platforms.*
- *Various core components are provided by Cordova to create the base of the application on which developers can implement their own logic.*

## 8. Sencha Ext JS

**Sencha** combined with **Ext JS** is a cross-platform mobile application development platform. It is a **JavaScript framework** to create data-driven mobile and web apps. It has a wide range of built-in **UI components**, extensions such as grids, calendars, windows, trees, toolbars, etc., and has various themes to enhance the appearance of the elements. **Sencha Architect** is a tool provided by Sencha Ext JS to shorten the time it takes to get an app to market.

To change a theme, you need to change the CSS and **Script code** of the theme. Ext JS is available in two types: **Modern toolkit** and **Classic toolkit**.

### Features:

- *Provides strong **data analytics**.*
- *Has a flexible **layout** system.*
- *Has a high level of **Data-Driven Documents (D3)** package*
- *Has a graphical data representation.*
- *Handles and manages millions of data.*

### Mobile application development approaches

1. Native Development: Building separate versions of an app for each platform (iOS, Android, etc.) using platform-specific languages and tools. This approach provides the best performance and access to platform-specific features but requires more time and resources.
2. Cross-Platform Development: Using frameworks like React Native, Flutter, Xamarin, or Ionic to build a single codebase that can be deployed across multiple platforms.

This approach can reduce development time and cost but may not provide the same level of performance or access to all platform-specific features.

3. **Progressive Web Apps (PWAs):** Building web applications that use modern web technologies to provide a mobile app-like experience. PWAs can be accessed through a web browser and can be installed on a device's home screen, offering offline functionality and push notifications.
4. **Hybrid Development:** Using web technologies (HTML, CSS, JavaScript) to build an app that is wrapped in a native container, allowing it to be distributed through app stores. This approach combines elements of both native and web development.

### **Reasons for mobile application development**

- **Reach and Accessibility:** Mobile apps allow businesses to reach a large audience of mobile device users, increasing accessibility to their products or services.
- **Customer Engagement:** Apps provide a direct and personalized way to engage with customers, offering features like push notifications, in-app messaging, and loyalty programs.
- **Brand Presence:** Having a mobile app can help strengthen a brand's presence in the market and improve brand recognition among users.
- **Enhanced User Experience:** Mobile apps can offer a more seamless and intuitive user experience compared to mobile websites, leading to higher user satisfaction and retention.
- **Access to Device Features:** Apps can leverage device features like GPS, camera, and accelerometer to provide unique and interactive experiences.
- **Monetization Opportunities:** Mobile apps can be monetized through various means, such as in-app purchases, ads, subscriptions, or paid downloads.
- **Competitive Advantage:** In many industries, having a well-designed and functional mobile app can provide a competitive edge over competitors who do not have one.
- **Data Collection and Analysis:** Apps can collect valuable user data, which can be used to analyze user behavior, preferences, and trends, helping businesses make informed decisions.
- **Offline Access:** Some apps can provide limited functionality even when offline, which can be useful in situations where internet connectivity is limited or unavailable.

- **Innovation and Differentiation:** Developing a mobile app can showcase a company's innovation and commitment to technology, helping to differentiate it from competitors.