SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER APPLICATIONS

23CAT607- CROSS-PLATFORM APP DEVELOPMENT

I YEAR II SEM

UNIT 1 – INTRODUCTION WEB DESIGN

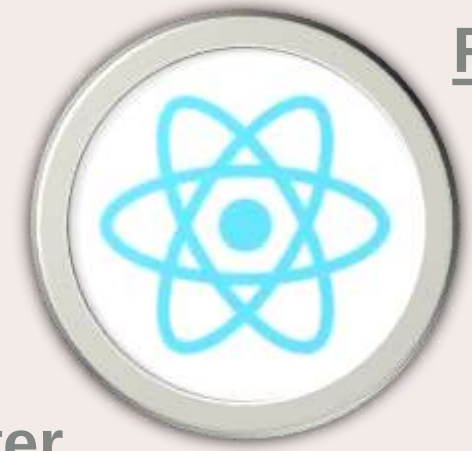
TOPIC 4 – Tools of Cross-platform applications

Agenda

1. Tools Cross-platform App
2. Comparison
3. Tools with language
4. Why flutter is preferred
5. Summary

Cross-platform App

Flutter



React Native

Appcelerator

Xamarin

Qt

Adobe PhoneGap

Safari web inspector tools

Cross-platform App

Sencha

Apache Cordova

Unity

Ionic

Solar2D

Cross-platform App

☺ Pros	☹ Cons
Fast app development	Lack of seamless hardware support, such as camera or GPS
Reusable code	Sluggish code
Reduced overheads, development, and maintenance costs	Slow feature updates
Greater market reach	Lower performance & flexibility compared to native apps
Easier to build a user-friendly UI & UX	Integration difficulties

Cross-platform App

Framework	Currently owned by	First introduced in	Performance	Development language	Learning curve	Hardware APIs support	Community
React Native	Meta	2015	Average due to the use of bridge	JavaScript	Easy	Limited (direct API's and third-party plugins)	Huge and growing
Flutter	Google	2017	Native-like	Dart	Easy	Extensive (direct APIs and third-party plugins)	Fair-sized but rapidly growing
Xamarin	Microsoft	2011	Close to native-like	.NET	Difficult	Extensive (direct APIs and third-party plugins)	Fair-sized but shrinking

Cross-platform App

Ionic Framework	OutSystems	2013	Average due to use of web technologies	Web languages (HTML, CSS, JS)	Easy	Limited (third-party plugins only)	Huge and growing
Apache Cordova	Apache Software Foundation	2011	Average due to use of web technologies	Web languages (HTML, CSS, JS)	Easy	Limited (third-party plugins only)	Huge (possibly larger than Ionic's) but shrinking
NativeScript	OpenJS Foundation	2014	Great (but not as good as Flutter's)	JavaScript or its variants	Fairly easy	Limited (direct access and third-party plugins) but can be expanded easily	Small and fairly stagnant
Kotlin Multiplatform	JetBrains	2017	Native-like (and often better than Flutter)	Kotlin	Moderate	Extensive (expect/actual mechanism)	Small but rapidly growing

Cross-platform App

Tool	Programming Language(s)
React Native	JavaScript, React
Flutter	Dart
Xamarin	C#
Ionic	HTML, CSS, JavaScript, Angular
NativeScript	JavaScript, Angular, Vue.js, React
Apache Cordova (PhoneGap)	HTML, CSS, JavaScript
Qt	C++
Unity	C#
Electron	HTML, CSS, JavaScript
Progressive Web Apps (PWAs)	HTML, CSS, JavaScript

Cross-platform App

1.React Native: Developed by Facebook, React Native uses JavaScript and the React library to build native mobile apps for iOS and Android.

2.Flutter: Created by Google, Flutter uses the Dart programming language to build natively compiled apps for mobile, web, and desktop.

3.Xamarin: Developed by Microsoft, Xamarin uses C# and .NET for building cross-platform apps for iOS, Android, and Windows.

4.Ionic: Ionic is an open-source UI toolkit that uses HTML, CSS, JavaScript, and the Angular framework for building mobile and desktop apps.

5.NativeScript: NativeScript allows developers to use JavaScript, Angular, Vue.js, or React to create truly native mobile apps for iOS and Android.

Cross-platform App

1.Apache Cordova (PhoneGap): Apache Cordova (formerly PhoneGap) uses HTML, CSS, and JavaScript to build cross-platform mobile apps with access to native device features.

2.Qt: Qt is a cross-platform application development framework that primarily uses C++ for building applications that run on various software and hardware platforms.

3.Unity: Unity is a cross-platform game engine that primarily uses C# for creating 2D and 3D games, simulations, and other interactive content.

4.Electron: Developed by GitHub, Electron uses HTML, CSS, and JavaScript to build cross-platform desktop applications.

5.Progressive Web Apps (PWAs): PWAs are web applications built using HTML, CSS, and JavaScript that can provide an app-like experience across platforms and devices.

Cross-platform App

1.Native Performance

- ✓ Flutter compiles directly to native code, providing performance comparable to natively written apps.

2.Single Codebase

- ✓ Write one codebase in Dart and compile it to build native apps for multiple platforms (iOS, Android, web, desktop, and embedded devices).

3.Hot Reload

- ✓ Flutter's hot reload feature allows developers to see code changes instantly reflected in the running app, streamlining the development process.

4.Extensive Widget Library

- ✓ Flutter comes with a rich set of customizable widgets that follow Material Design and Cupertino (iOS-style) design principles, enabling beautiful and consistent user interfaces across platforms.

5.Native Access

- ✓ Flutter provides easy access to native platform features and APIs, such as cameras, GPS, sensors, and more, through a set of packages and plugins.

6.Growing Community and Ecosystem

- ✓ Flutter has a rapidly growing community of developers, contributors, and third-party packages, offering better support, more resources, and a richer ecosystem.

Cross-platform App

1.Open Source and Backed by Google

- ✓ Flutter is an open-source project backed by Google, ensuring long-term support, regular updates, and a commitment to its development and improvement.

2.Reactive Programming with Dart

- ✓ Flutter uses the Dart programming language, which supports reactive programming, making it easier to build responsive and efficient user interfaces.

3.Ahead-of-Time (AOT) and Just-in-Time (JIT) Compilation

- ✓ Flutter supports both AOT and JIT compilation, allowing for optimized performance and faster development iterations.

4.Platform Views

- ✓ Flutter allows embedding platform-specific views (e.g., WebViews, maps) within the Flutter app, enabling integration with existing native code.

5.Stateful Hot Reload

- ✓ Flutter's stateful hot reload feature preserves the state of the app during code changes, making it easier to iterate and test different scenarios.

6.Consistent Look and Feel

- ✓ Flutter's widget library and design principles help ensure a consistent look and feel across platforms, making it easier to maintain branding and user experience.

Thank you

REFERENCE:

- ✓ A Complete Guide to Cross-Platform App Development 2023 (ongraph.com)
- ✓ Introduction to cross-platform app development - itCraft blog (itcraftapps.com)
- ✓ Introduction to Cross-Platform App Development | AppMaster