Abstract—Many platforms are emerging recently and businesses need to work with all of them, therefore, creating applications that work on multiplatform like Android, IOS, Windows ...etc. is necessary these days. Building a separate native application for each platform as well as working separately with both Frontend and Backend is costly in several respects, including time, effort and cost. Using Cross-platform tools and frameworks is the best solution to the problem of multiple platforms and the need to deal with all of them. This paper discusses the most famous, modern, and widely used Cross-platform tools to introduce the best way to build an E-learning application that works natively in multiplatform using only one programming language, and even it doesn’t require work with the Backend side. Finally, we have created a sample E-learning application using Flutter with Firebase as a Backend. Due to the Corona pandemic, e-learning has become a mandatory requirement as a primary or secondary method of education, this application work on all platforms (Mobile and Desktop), and we tested it on Android and Web Browsers.

Index Terms—Cross-Platform, E-learning, Firebase, Flutter.

I. INTRODUCTION
Software development process becomes tremendous complex these days, not because of the complicated of programming languages and frameworks, on the contrary, it has become easier, but because nowadays the business needs software that works perfectly on all platforms like Mobile (both Android and IOS), all types of Browsers, Desktop (Windows, Linux, and MAC) and sometimes even on the smartwatches and sensors for IOT systems (Jasim et al. 2021).

The evolution and prevalence of all these platforms made the process of developing applications that work on all of them a necessity, not a luxury (Solmaz and van Gerven 2022).

The fact that we need a different programming language and different tools to create the same application which works on each platform is scary because we don’t need only a different language or/and tool for each application, but a different language for Backend and Frontend as well, therefore, for each application in each platform we might need two different languages, and many times it requires a developer or team for each platform and one for Backend and other for Frontend (Kaoudi and Quiane-Ruiz 2018).

Using Cross-Platform Software Development is a big deal, and a great solution, because it enables software developers to build applications that work on multiple platforms with the same codebase (Nawrocki et al. 2021).

From the Backend we can use cloud services like Firebase which enable developers to work with the Backend without working with Backend details or creating an API, just using the same programming language for the Frontend to interact with the Backend.

E-learning and Mobile Learning applications become necessary and widely used because of COVID-19 and even in normal situations, therefore, in this research, a simple application for eLearning has been created with Cross-platform tools, this application can work on Android, IOS, and Web Browsers, therefore, it can work on all platforms (Mobiles, Desktops) with any operating system (Windows, macOS, Linux, Android, and IOS).

II. BACKGROUND
A LOT OF TOOLS AND FRAMEWORKS EXIST FOR BUILDING CROSS-PLATFORM APPLICATIONS, WE WILL DISCUSS THE SOFTWARE DEVELOPMENT APPROACHES INCLUDING THE MOST IMPORTANT AND MOST PREVALENT CROSS-PLATFORM APPROACHES.

A. NATIVE SOFTWARE

Like a native language for humans, which is the language that has been learned in infancy from parents or family (“What Is Native Language | IGI Global” n.d.), a native programming language is a language provided by a specific platform vendor with some libraries and tools under what is called a software
development kit (SDK), therefore by using this language and SDK, developers can target a specific platform or operating system, for instance (Android SDK) for build an Android application (Biørn-Hansen et al., 2020). The native app is an application written specifically for one platform, using a native programming language provided by this platform, like java or Kotlin for Android, Objective-C, or Swift for IOS (Jobe 2013).

As demonstrated in figure 1, there are two sides, application side, and platform (Operating System) side, in native development, native code can directly access all platform-specific services and hardware for a platform as well as rendering processes made on the platform side, by using the native widgets (UI components) that hold information about their position in the screen and how they should look, then pass this information to the canvas which responsible for displaying it in the platform screen. And events are also read by the platform and passed to the widgets to process it by the native code to define how should react to these events like the click or touch (Mainkar and Giordano, n.d.).

A. CROSS-PLATFORM SOFTWARE DEVELOPMENT

There are many definitions for Cross-Platform (Biørn-Hansen et al. 2019) that define it as a substitutional way for developing software by reusing the same code over multiple platforms. It is the method of one application that works on all types or multiple types of operating systems (Nitze and Schmietendorf 2013) This will help developers to create and deploy a solution on a variety of operating systems like Android, IOS and Windows with minimal time and cost (Zohud and Zein 2019).

With Cross-Platform frameworks, we need only a general code for multiple platforms with no Platform-Specific code (Biørn-Hansen et al., 2020).

several approaches exist to implement Cross-Platform:

1. **cross-compilation** technique by using a cross-compiler which is compiling code for one operating system like Android on another operating system like windows, for instance, enables decoupling of the build platform from the target platform. This technique works as follows:
   a. a cross-platform framework will provide an independent API (Application Programming Interface) using any programming language like Kotlin, Java, Dart, which means it is not related to any platform or operating system.
   b. developers can use this API to create an application.
   c. after that the Cross-Compiler within the framework will compile it into a native code for each targeted operating system (“Cross-Platform-Publication,” n.d.). Flutter is the example of using this approach (Biørn-Hansen et al., 2020).

2. **the runtime-based approach** is a type of interpreted approach (El-Kassas et al. 2017) using it is own runtime environment to run the application, therefore, it builds a runtime layer for each targeted operating system, in this case, the cross-platform frameworks provide the General API for programming along with the runtime environment to run the application. This runtime environment will execute in the device (like a mobile device) and allows the application to run on any device independently of the operating system (“Cross-Platform-Publication,” n.d.) & (El-Kassas et al. 2017), these types of frameworks typically use general programming languages like JavaScript to allows developers to write only one code-base application (Biørn-Hansen et al., 2020).

3. **A virtual machine (VM)** is a software representation of a machine, like a computer or mobile, which runs a program exactly like a real machine (Heistermann et al. 2015). VM normally simulate exactly all machine hardware like CPU and registers, in the cross-platform software development context, using VM categorize under the interpreted approach, VM allows an application to run on multiple platforms with the same source code but on a VM depending on the platform type (“Cross-Platform-Publication,” n.d.). The example of this type is Java Virtual Machine (JVM), in this approach, the Virtual machine, and programming language are separated (El-Kassas et al. 2017).

4. **The hybrid approach** is another approach, it is an application written with Web technology like (Html, CSS, and JavaScript) but the code will run within a native container, therefore this application will have access to all native device APIs and features like camera, Bluetooth, GPS … etc. (Jobe 2013).

The native container here is holding the same feature as the browser, and the ability to access the specific-platform services is done indirectly by using Bridge (Faust, n.d.), Ionic framework is an example of adopting this approach (Biørn-Hansen et al., 2020).

**Mobile Web applications**, which use the same technology of Web applications like (HTML, CSS, and JAVASCRIPT) to

DOI: http://doi.org/10.24086/cocos2022/paper.511
simulate the native application for target operating system (host) like Android, but at the end, this application will run on the host Web browser (“Cross-Platform-Publication,” n.d.).

B. TOOLS AND FRAMEWORKS FOR CROSS-PLATFORM SOLUTION

1) React Native
It is a JavaScript library that uses a native UI component to create a native application for android and IOS (“React Native · Learn Once, Write Anywhere” n.d.). In react native, JavaScript is used to create the applications, therefore, the interpreted technique has been used here. In this approach, JavaScript code defines generalized widgets not specific-platform widgets, but under the hood, each widget has an associated widget in each platform supported by the framework (Mascetti et al. 2021). Interaction between the application side and platform side has been done by the bridge, which converts the generalized widgets to the platform-specific widgets depending on the platform on which the application running, therefore, the rendering process happens on the platform side, not an application.

Unlike native, the communications between the application and the platform services and features like (Bluetooth, Wi-Fi, GPS) are done through the Bridge not directly (Faust, n.d.).

2) Flutter
Flutter is a toolkit (collection of tools) for building applications compiling natively for Mobile (Android & IOS), Web, Desktop (Windows, Linux, and MAC), and even embedded devices (“Beautiful Native Apps in Record Time | Flutter” n.d.). Flutter’s main goal is to make the development process easier and faster and more productive, therefore, it comes with significant features like hot-reload which helps developers test the changes on application faster, and using pre-defined widgets as well as allows the community to create and deploy hundreds of plugins to use it by developer everywhere (Mainkar and Giordano, n.d.).

Flutter follows a different approach to render widgets and connect with the platform-specific services, all rendering process is done on the application side (within the framework), flutter approach the rendering engine which is responsible to draws the widgets, therefore, it uses its widgets, not platform-specific widgets. Flutter frameworks enable communication directly with the platform canvas to display those widgets on the platform screen. And to communicate with the platform-specific services, flutter also uses what is called a platform channel, which is a new way to enable developers to create a plugin to interact with platform-specific features like GPS, WIFI … etc. (Faust, n.d.).

a) JIT AND AOT IN FLUTTER
Flutter support two types of compilation, just in time (JIT), and Ahead of Time (AOT) (Singh and Lodhe 2021).

JIT means to compile the necessary parts of source code at the runtime, which means when the application running (Viglas 2014), this type of compilation is used by flutter in the development phase (stage) to speed up the development and testing process, by using Hot Reload, therefore, programmers don’t need to recompile the whole source code every time to test the changes in source code, instead just inject the updated code at run time (Singh and Lodhe 2021).

AOT means to compile all source code into native binary code for each platform (Gunningberg, Association for Computing Machinery., and Association for Computing Machinery. ).
Special Interest Group on Embedded Systems., n.d.), flutter used this type in the deployment stage when the application is ready to deploy to each platform, therefore with AOT, the application will work faster and natively (Singh and Lodhe 2021).

b) **Flutter Compilation for Android, IOS, WEB, and Desktop**

Flutter application consists of two main parts, the flutter skin engine which is created by C and C++ languages, and the Dart code and dart SDK, in an android platform, C and C++ engine compiled by android NDK (Native Development Kit) which is come with Android SDK, and the Dart code will compile ahead of time (AOT) to the native code.

In the IOS platform C and C++ engine will compile with LLVM (Low-Level Virtual Machine) (“FAQ | Flutter” n.d.), which is a cross-platform compiler that comes with Xcode as a part of the IOS development platform (“What Is LLVM? Definition and Related FAQs | OmniSci” n.d.), and the Dart code compiling using ahead of time (AOT) compilation to the native code, and macOS platform with Xcode is required to build the IOS application (“FAQ | Flutter” n.d.).

In the Web, the flutter framework uses the dart2js library to compile dart code to the JavaScript code which works in browsers (“Dart2js | Dart” n.d.).

For Desktop applications, the flutter app should build in the targeted platform, which means, build windows applications in the windows platform, macOS on the macOS, and so on, because to create a desktop application, flutter framework uses the platform-tools, like Visual studio for windows, Xcode for macOS and Clang, CMake, GTK …etc. for Linux beside the flutter SDK (“Desktop Support for Flutter | Flutter” n.d.).

![Figure 5: Compilation in Flutter to multiplatform](image)

**Figure 5: Compilation in Flutter to multiplatform**

C. **Backend Vs Frontend**

The Frontend is the part of an application which user can see it and interact with, therefore, everything running on the client device is a Frontend, including User Interface and other logic, for instance, Flutter, React Native and HTML, CSS and JavaScript, all those technologies and languages using to build a Frontend application either for Mobile devices or Web Browsers (Gerasimov et al. 2020).

On the other hand, Backend is everything working with the server-side, including Interaction with Database, authentication, Storage, for example, PHP, Laravel, node.js, Firebase, all those technologies, and languages are used to working with the Backend side (Amza, Cox, and Zwaenepoel 2003).

D. **APPLICATION PROGRAMMING INTERFACE (API)**

It is like an interface between a software application and other software applications, enabling other applications and developers to access and use the data and services provided by this application without understanding or working with its underline procedures.

API enables applications to exchange data and functionality by using a set of methods, objects, and URLs (Meng, Steinhardt, and Schubert 2018). In most modern mobile applications, it is required to use a strong architecture by connecting it with a Backend server and central database, to do that, it is required to use an API to connect between Backend and Frontend like the diagram below (DOSPINESCU and PERCA 2013).

![Figure 6: API](image)

**Figure 6: API**

E. **FIREBASE**

Firebase is a Backend as a Service, which means it is a cloud platform that offers Backend solutions for developers by the cloud, therefore, by using Firebase, developers don’t need to take care of the Backend works anymore.

Firebase gives Backend services to mobile applications and Web applications as well as embedded devices (Sharma and Dand 2019a). Firebase uses Firestore service to provide NoSQL database which using collections and documents like MongoDB and other types of NoSQL rather than Relational Database (Khan et al. 2022).

The traditional way to create a mobile application connected with a central database is to build a database and make relationships, after that create the API with authentications; while with using Firebase, Firebase will provide hosting, APIs,
authentication system, and everything for Backend, developers only use these services with the same tools and programming languages used in the Frontend side to create the application (Khawas and Shah 2018). Firebase enables developers to build applications in less time and cost, by escaping the server-side programming and database management systems issues (Rahmi, Piarsa, and Wira Buana 2017). The most important feature with Firebase is presenting a lot of services for the Backend, like NoSQL database, storage, Authentication, Messaging, Analyzing, Test Lab … etc. (Chandakant, n.d.).

Three mobile Web applications developed (Qadir and Cooper 2020) used google flutter framework to create a cross-platform application connected with Firebase as Backend, the application is a cargo tracking system using Global Positioning System (GPS) to track cargo, this application is working on both android and IOS platforms, and also on the Web, and using Google map API to provide a map for users, this application uses many other languages and frameworks beside flutter and Firebase, like HTML, CSS, JavaScript and Node.js framework as well as a Web server as a host.

(El-Kassas et al. 2017) survey the existing cross-platform approaches for mobile developments to help developers choose the best and most recent approach. As well as find the strength and weaknesses in each approach. This work focuses on making a merge among multiple approaches to reduce the best one. The result of this paper was proposing a new approach like (component-based approach), (cloud-based approach) and (merged approach). And introduce a detailed description of each approach and the pros and cons of them as well as introducing a sample solution for each approach.

This paper found that using native cross-platform with Titanium and Xamarin and hybrid approach with PhoneGap is most widely compared with Web approach because users cannot download the Web app from the app stores.

(Jobe 2013) explore if it is possible to replace native mobile applications with Web applications in developing countries. This paper develops two different mobile Web applications to test this ability.

The first one was (tracked run), and the second one is (booking system for scheduling “slum runs”). Those applications are tested by semi-professional Kenyan runners.

The result after 6 months of testing found that the mobile Web app for (tracking runs) had poor performance compared with native, because of the poor GPS performance, while the second Web app for booking system was working well with good performance.

This paper concludes that using mobile Web applications with apps requiring intensive interactions with native features like camera, GPS, WIFI …etc. is not a good alternative for native applications, while it is suitable as a replacement if the application only needs like native interface and content-focused.

(Bjorn-Hansen et al. 2020b), focused on comparing the native android applications with cross-platform mobile applications in the matter of performance, this paper examines the performance from many features or variables like time of completion, CPU usage, memory consumption, Accelerometer, contacts, file system, and geolocation.

The result indicates that using any of cross-platform frameworks may decrease the performance of application compared with using native SDKs, but at the same time, some cross-platform frameworks can produce the same or better performance than native on certain features like using Native Script led to reducing the usage of CPU and the time of compilation as well as using Flutter framework yields

III. RELATED WORKS

(Lachgar 2017) Created a framework to examine the best tools and methods to create mobile applications with given requirements, like available on android and IOS, access the network, notifications, camera, GPS ..etc., this framework is divided into two stages, first one for selecting the best approach (native, hybrid or Web), and second stage for select the best tool to implement the work, his framework helps developers to decide which approach is better in specific situations like with time and cost constraints for example.

This research found that using the native approach is the best way to fulfill 62% of requirements, after that using a hybrid approach, can implement about 56% of requirements, and using the Web approach can fulfill about 18% of requirements.

And for the best tools, it was found that the best tool is using the platform-specific development kits for each platform, then using Titanium framework for cross-platform, after that using IONIC for PhoneGap.

(Nawrocki et al. 2021) has created a applications in some modern cross-platform frameworks like React Native, Flutter, Xamarin, beside the Native android and IOS applications to compare it depend on some factors like start-up time, size, CPU, Memory Usage and development experience, the result found that using Native solution is the best choice for all factors except user experience when the flutter become the first, and directly after native, flutter is the best solution as Cross-platform solution.

DOI: http://doi.org/10.24086/cocos2022/paper.511

Figure 7: using Firebase as Backend with Flutter
application with lowest computed RAM usage, but there is no framework are best in all features. (Khawas and Shah 2018) introduce an android application connected with Firebase as a Backend, this paper focuses on the importance of Firebase as a substitution of using different programming languages and database management systems to work with Backend for android application, and highlighting the features of a Firebase by building a real-world android application with using Firebase as Backend service. (Sharma and Dand 2019b) claim that Firebase is the best Backend as a Service for mobile applications, and in his paper, he introduces a mobile application connected with Firebase to fulfill this claim. The result was creating a college application, and this application provided many features to both students and faculty and it is covered all requirements that resulted from requirements analysis for application.

### Table 1
Comparison between the widely used Mobile Development Technique

<table>
<thead>
<tr>
<th>Factors</th>
<th>Native</th>
<th>Flutter</th>
<th>React Native</th>
<th>WebView</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance</strong></td>
<td>The Best</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Low performance</td>
</tr>
<tr>
<td><strong>Platforms Support</strong></td>
<td>Only one Operating system like Android</td>
<td>Android, IOS, Windows, Mac, Linux, Web and Embedded System</td>
<td>Android and IOS</td>
<td>Android, IOS, and Web</td>
</tr>
<tr>
<td><strong>Widget Rendering</strong></td>
<td>Using the platform widgets directly</td>
<td>Use flutter own widgets and draw them directly on the platform.</td>
<td>Using Bridge to convert generalized widgets to the Platform-specific widgets</td>
<td>Using the WebView (like an internal browser) which is a native widget to render the Web app.</td>
</tr>
<tr>
<td><strong>Connect with Platform-specific services</strong></td>
<td>Connect directly</td>
<td>Using the Platform channel to connect.</td>
<td>Using Bridge to connect.</td>
<td>Using like a Bridge to connect.</td>
</tr>
</tbody>
</table>

(Mainkar and Giordano, n.d.; Faust, n.d.)

In Table 1, a comparison has been made between some of the most important and latest technologies and frameworks for developing mobile applications in terms of performance, support, and some technical details, this information was obtained from the results of some relevant studies. The results of the comparison showed that there is a discrepancy in preference according to different factors, and no one of these techniques can be considered as absolutely the best, but it depends on the use, time, cost, and required platforms.

### IV. THE RESULT

For this research, (Our Class) application has been created, a simple application to allow teachers to add the classes, and subjects as PDF files, or YouTube Videos, in this application only one programming language has used which is dart with flutter framework, and for Backend, Firebase Cloud Platform has used to work with the database and authentication system, this application works on android, IOS, and Web Browsers.

#### A. THE Backend

Firebase has been used as a Backend for this application, Firestore and Authentication system has used for NoSQL database and mobile authentication.

#### B. THE CROSS-PLATFORM APPLICATION

To build a cross-platform application, the Flutter framework has been used, it is only required one programming language which is Dart, this application works on both Mobile and Web, the application support both Arabic and English Languages, and Mobile phone Authentication has been used. After the user inserted his phone number, the application will send a message to this number including the code to ensure that is his mobile number and open another screen to insert this code (OTP code).

After the user inserted his phone number, the application will send a message to this number including the code to ensure that is his mobile number and open another screen to insert this code (OTP code). Then students can see all sections or classes screen.

If the student wasn’t subscribed to in any class, he should insert the class code provided by his lecturer to join the class. After that, students can access all the subjects within the class which the students are.

In this application, there are only two types of lessons that can be uploaded, either a link to a YouTube video or a link to a PDF file, to reduce the storage space in Firebase. Thus, reducing the cost of subscriptions, regardless of the number of lessons uploaded.

The simple dashboard has been created for inserting the classes and subjects by the teachers, only teachers who registered by the admin of the application could access these screens.

### C. RESULT ANALYSING

Although Flutter supports all different platforms, our application doesn’t work on Desktop (Windows, Linux), because Firebase does not support desktop applications except macOS, therefore, our project works on all platforms except the desktop, but in practice, and since it works on the Web browsers, it can be said that the application can be used on all platforms.

Sometimes and in rare cases, we need to use two different codes to work with different platforms in a flutter, for instance in our project, and to work with mobile authentication, we have used two different codes to check if the platform is Web.
we used (signInWithPhoneNumber()) otherwise use (verifyPhoneNumber).
And we check if the platform is the Web as well to add an
tab instead of a thumbnail because the function to create a
thumbnail from a YouTube video doesn’t work with the Web.

CONCLUSION
There are many ways to build applications that work on multi-
platforms, but using a flutter framework with Firebase as a
Backend enables the developer to create an application that
works on most types of platforms natively with one codebase,
and it doesn’t require working with Backend stuff.
The project (Our Class) is a sample application that works on
Mobile (Android, IOS) and Web Browsers, this application can
be used to facilitate access to lectures for students via any
device, and by building this application and testing it we could
practically understand how does cross-platform tools works, as
well as the cons and pros and the limitations, the research found
that with flutter and in rare cases it is required to use a different
codebase for some platform, but it is easy to handle it by just
check in which platform that code will run and change the code
accordingly. On the other hand, Firebase can easily integrate
with flutter by using the flutter SDK with the Dart programming
language, therefore it doesn’t require working with the Backend
side, but Firebase doesn’t support all Desktop operating
systems like Windows and Linux.

ACKNOWLEDGMENT
I would like to thank Cihan University-Erbil since it was the
first place for me to start my academic road and writing this
research. I thank also Mr. Abdullah A. Nahi -Computer Science
department-cihan university for providing supports and advice
to accomplish this research.

REFERENCES
“Distributed Versioning: Consistent Replication for Scaling
Back-End Databases of Dynamic ContentWeb Sites.” In ,
“Beautiful Native Apps in Record Time | Flutter.” n.d. Accessed
November 12, 2021.
https://flutter.dev/?gclid=CjwKCAiAvriMBhAuEiwA8Cs5len
qlpHEJF5am44bnr4RmhTAOphVS-TznpGAGXahlAtisSlqCSjM7xoCpPEQAvD_BwE&gclsrc=a
w.ds.

Biørn-Hansen, Andreas, Tor-Morten Grønli, Gheorghita Ghinea,
Platform Mobile Development in Industry.” Wireless
Communications and Mobile Computing 2019 (January): 1–

Biørn-Hansen, Andreas, Christoph Rieger, Tor Morten Grønli, Tim
Investigation of Performance Overhead in Cross-Platform
Mobile Development Frameworks.” Empirical Software

Overhead in Cross-Platform Mobile Development

Chandakant, Mayuri. n.d. “International Research Traditional

“Cross-Platform-Publication.” n.d.

https://dart.dev/tools/dart2js.

“Desktop Support for Flutter | Flutter.” n.d. Accessed December 1,

DOSPINESCU, Octavian, and Marian PERCA. 2013. “Web
Services in Mobile Applications.” Informatica Economica 17
https://doi.org/10.12948/issn14531305/17.2.2013.02.

El-Kassas, Wafaa S., Bassem A. Abdullah, Ahmed H. Yousef, and
Mobile Applications Development Approaches.” Ain Shams
https://doi.org/10.1016/J.ASEJ.2015.08.004.

Gunningberg, Per, Association for Computing Machinery.,
https://doi.org/10.1016/J.ASEJ.2015.08.004.

Gunningberg, Per, Association for Computing Machinery.,
https://doi.org/10.1016/J.ASEJ.2015.08.004.

Heistermann, M., S. Collis, M. J. Dixon, J. J. Helmus, A. Henja, D.
of the American Meteorological Society 96 (10).
https://doi.org/10.1175/BAMS-D-14-00220.1.

Jasim, Khalid F, Reem J Ismail, Abdullah A Nahi Al-Rabeeah, and
Symmetric Cipher Algorithms Suitable for the Security of
https://doi.org/10.24086/cuesj.v5n2y2021.

DOI: http://doi.org/10.24086/cocos2022/paper.511