SEMINAR PRESENTATION
ON
" CROSS - PLATFORM APPLICATION DEVELOPMENT USING FLUTTER "

PRESENTED BY
MEGHSHAM VINAYAK KAPURE

GUIDED BY
PROF. A. S. SONDAKAR MAM

Agenda

- Types of Mobile Apps by Technology
- Native vs Cross Platform Vs Web
- Cross – Platform Mobile Development
  - Challenges of Mobile App Development
- Introduction to Flutter
- DART Programming
  - Interface elements in Dart
  - Libraries of DART
  - Packages of DART
  - Widgets in Flutter
- Application build with Flutter
- Advantages & Disadvantages of Flutter

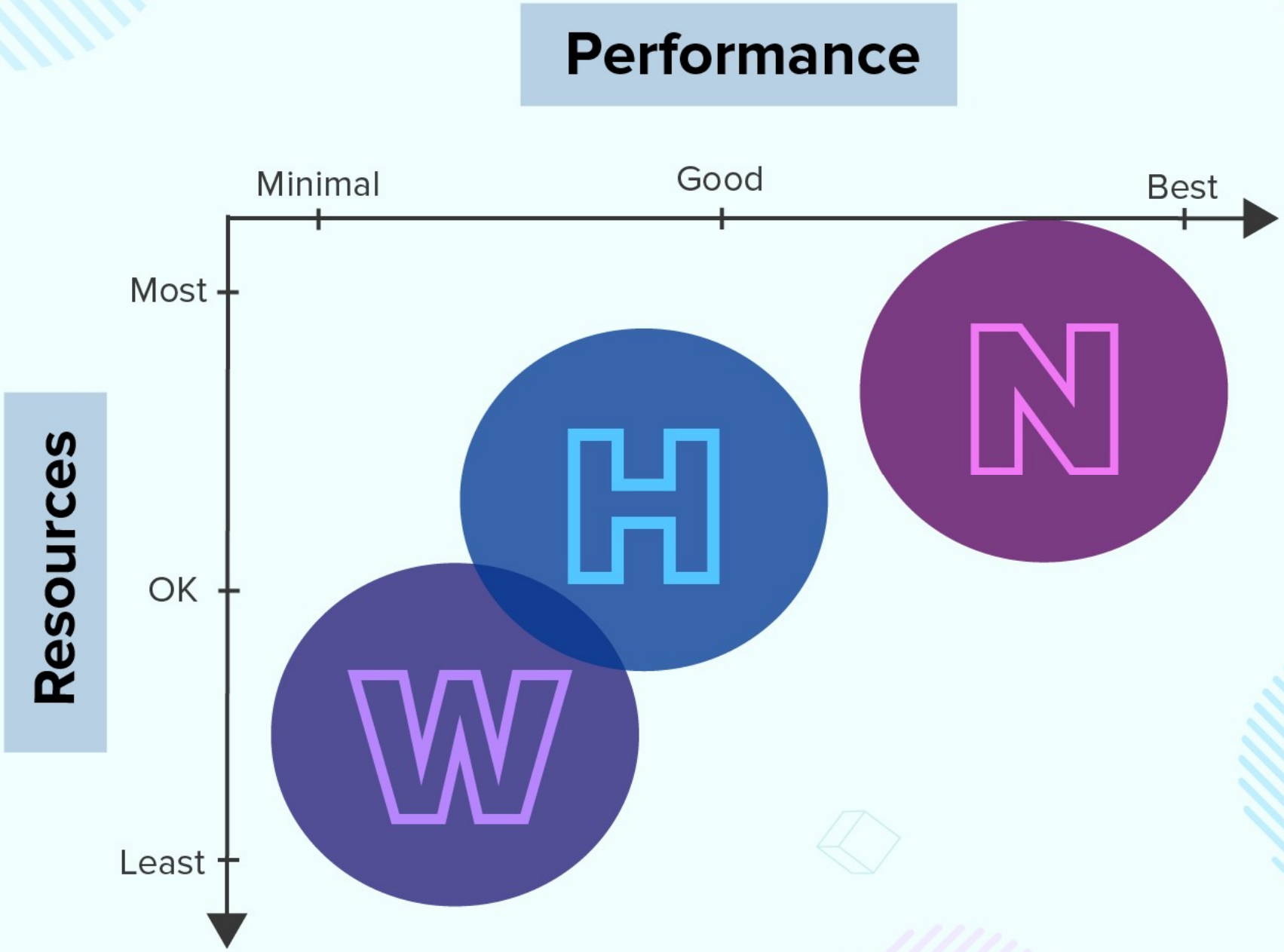
Types of Mobile Apps by Technology

There are three basic types of mobile apps if we categorize them by the technology used to code them:

- **Native apps** are created for one specific platform or operating system.
- **Web apps** are responsive versions of websites that can work on any mobile device or OS because they're delivered using a mobile browser.
- **Hybrid apps** are combinations of both native and web apps, but wrapped within a native app, giving it the ability to have its own icon or be downloaded from an app store.

Comparison Native vs Web vs Cross - Platform 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


Performance
Minimal
Good
Best
Most
OK
Least
Resources
N
H
W

Cross – Platform Mobile Development

- In computing, cross-platform software is computer software that is implemented on multiple computing platforms.
- A cross platform app is an application written or compiled in a way that allows it to run on multiple platform
- Cross-platform software may be divided into two types :-
    - Requires individual building or compilation for each platform that it supports
    - One can be directly run on any platform without special preparation

Cross Platform Frameworks

Introduction to Flutter

Flutter is Google's portable UI toolkit for crafting beautiful, natively compiled applications for mobile, web, and desktop from a single codebase.

Flutter works with existing code, is **used** by developers and organizations around the world, and is free and open source.

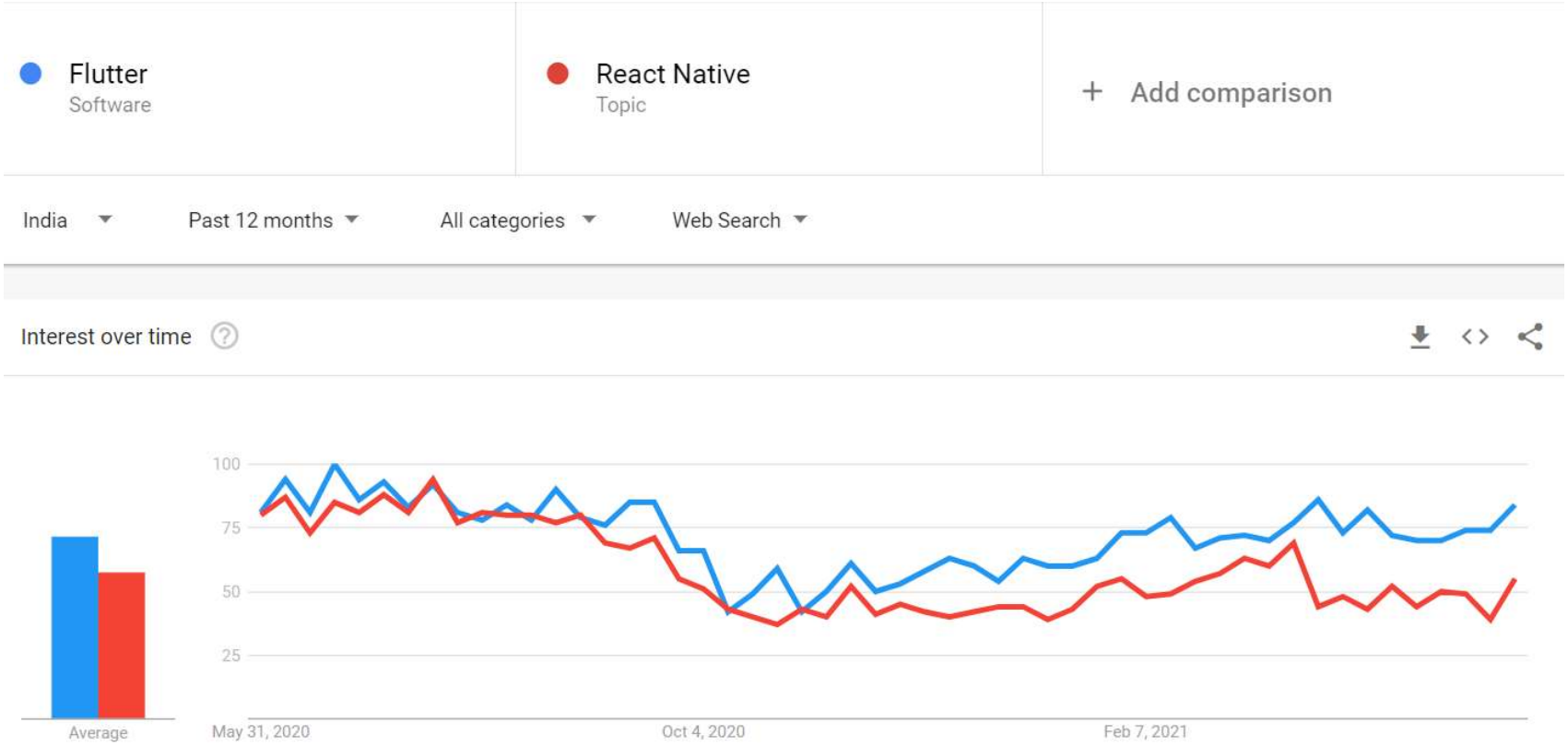
It is written using C, C++ and Dart.

The goal is to enable developers to deliver high-performance apps that feel natural on different platforms, embracing differences where they exist while sharing as much code as possible.

Future Scope of Flutter

1. The framework assists in developing an android application with comparatively low-end machinery in comparison to other technologies. Also, it constitutes the Hot Reload feature that brings an instant result on the changing of code.
2. It helps in defining the look and feel of the mobile app.
3. The open-source technology framework has the capability to deliver smooth and responsive animations to the user interface elements.
4. There are endless opportunities available when it comes to backend development and there is no exception to this.
5. Flutter renders every pixel of the screen. This means all the widgets will look alike on the mobile device.
6. Testing can be done with Flutter SDK and you can get the feedback accordingly.

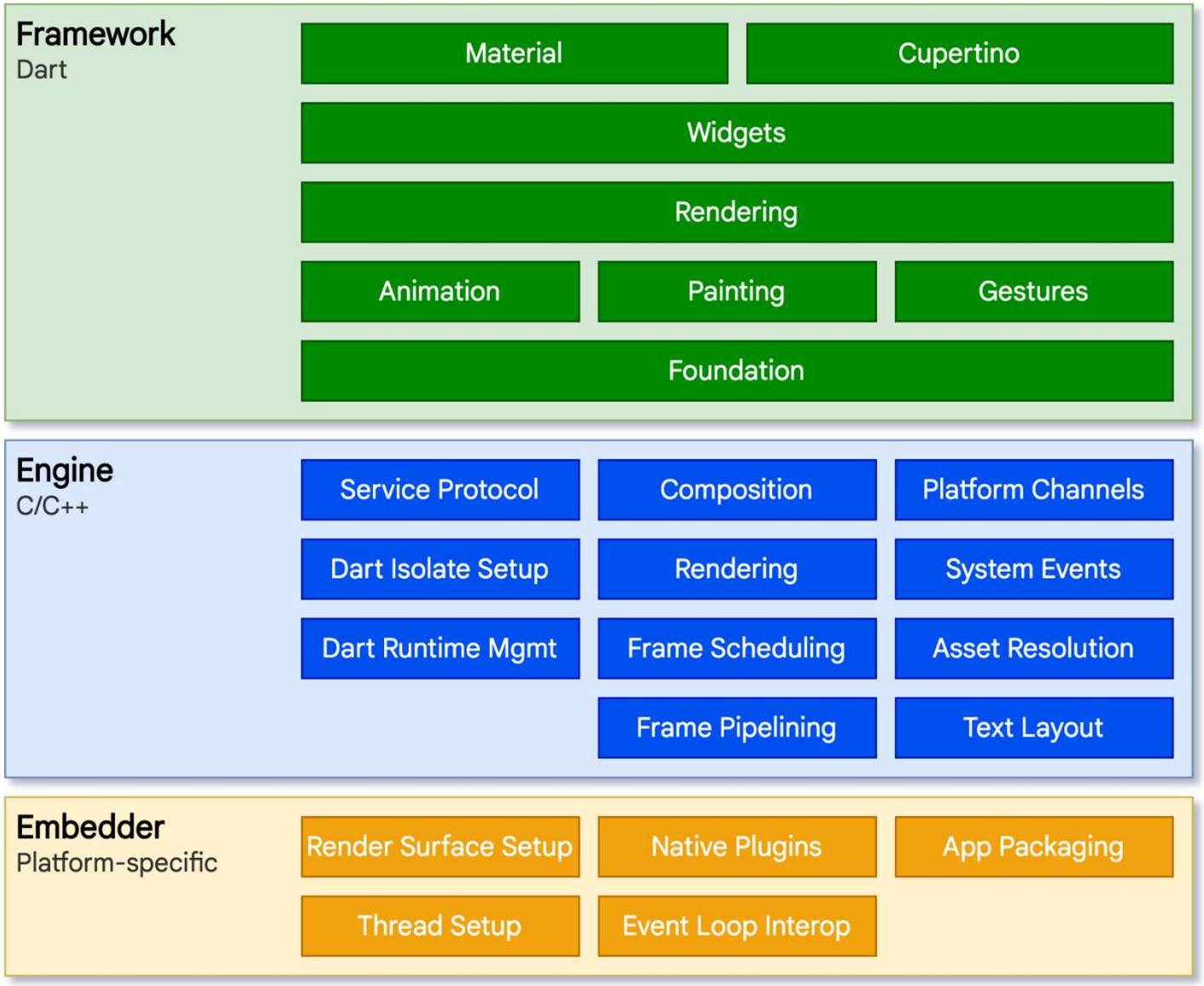
Flutter Trends



Features of Flutter Framework

- Open source Platform
- Amazing Widgets
- Custom Design & Navigation
- Faster Development Cycle
- Ease of code Sharing
- Object Operation
- Super Productive Dev Tools
- Reduce time for Quality Check and Assurance
- Enhance Performance with Dart programming

Flutter Architecture



Block Diagram of Technology

DART Programming

Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

Like most ALGOL languages (like C# or Java):

1. The entry point of a Dart class is the main() method. This method acts as a starting point for Flutter apps as well.
2. The default value of most data types is null.
3. Dart classes only support single inheritance. There can be only one superclass for a particular class but it can have many implementations of Interfaces.
4. The flow control of certain statements, like if conditions, loops (for, while and do-while), switch-case, break and continue statements are the same.
5. Abstraction works in a similar manner, allowing abstract classes and interfaces.

Interface Elements in Dart

Flutter has more app-specific libraries, more often on user interface elements like:

1. Widget:
   common app elements, like the Text or ListView.
2. Material:
   containing elements following Material design, like FloatingActionButton.
3. Cupertino:

   containing elements following current iOS designs, like CupertinoButton.

Libraries of DART

Dart has a rich set of core libraries that provide essentials for many everyday programming tasks such as working on collections of objects (dart:collection), making calculations (dart:math), and encoding / decoding data (dart:convert). Additional APIs are available in community contributed packages.

- Multi-platform libraries :-

  async, collection, convert, core, developer, math, typed_data

- Native platform libraries :-

  io, isolate, mirrors

- Web platform libraries :-

  html, indexed_db, web_audio, web_gl, js, us_util

Packages of DART

- The Dart ecosystem uses packages to manage shared software such as libraries and tools. To get Dart packages, you use the **pub package manager**.

- Importing libraries from packages

```
import 'package:js/js.dart' as js;
import 'package:intl/intl.dart';
```

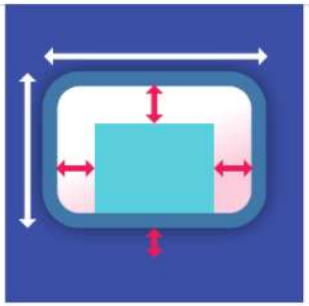
- This page lists some of the most popular and useful packages that Dart developers have published. Commonly used packages fall into three groups:
  - General-purpose packages
  - Packages that expand on Dart core libraries
  - Specialized packages

Widgets in Flutter

Flutter widgets are built using a modern framework that takes inspiration from React. The central idea is that you build your UI out of widgets. Widgets describe what their view should look like given their current configuration and state.

Basic widgets : Flutter comes with a suite of powerful basic widgets, of which the following are commonly used:

- Text : The Text widget lets you create a run of styled text within your application.
- Row, Column : These flex widgets let you create flexible layouts in both the horizontal (Row) and vertical (Colum n) directions.
- Stack : Instead of being linearly oriented (either horizontally or vertically), a Stack widget lets you place widgets on top of each other in paint order.
- Container : A container can be decorated with a BoxDecoration, such as a background, a border, or a shadow. A Container can also have margins, padding, and constraints applied to its size.

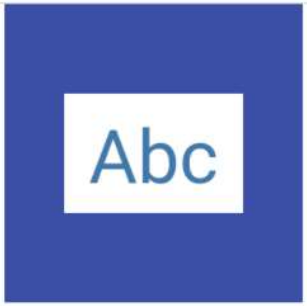


Container

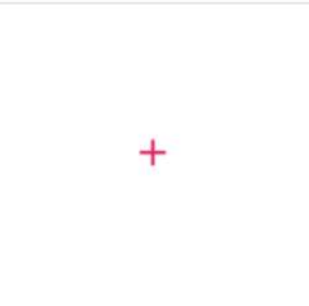
Row

Column

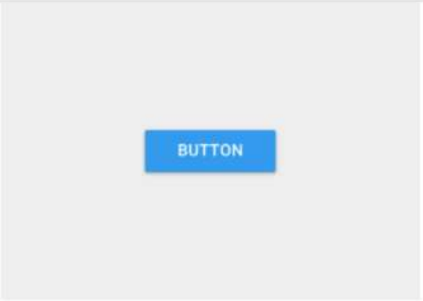
Image



Text



Icon



RaisedButton

Scaffold

Appbar

Application build with Flutter

- Google Ads (Utility)
- Alibaba (eCommerce)
- Reflectly (Lifestyle)
- Birch Finance (Finance)
- Hamilton Musical (Entertainment)
- Hookle (Social)
- Watermaniac (Health & Fitness)
- SG BusTracker (Maps & Navigation)

Advantages of Flutter

- **Faster code writing** : For developers, Flutter means faster & more dynamic mobile app development because of hot reload feature
- **One code for 2 platforms** : Developers write just one codebase for your 2 apps – covering both Android and iOS platforms
- **Less Testing** : If you have the same app for 2 platforms, it means less testing! Because of one codebase, the developers write automatic tests only once.
- **Faster apps** : Flutter apps work in a smooth and fast way, without hanging and cutting while scrolling.
- **Designs which your users will love** : Flutter is designed to make it easy to create your own widgets or customize the existing widgets.
- **The same app UI on older devices** : Your new app will look the same, even on old versions of Android and iOS systems.

Disadvantages of Flutter

- **Large File Sizes** : This object-oriented programming language is not as great as other languages like C#, Java, Objective C, and JavaScript.
- **Lack of Third-party Library :**These third-party libraries are normally free, open-source, pre-tested, and easily available. You may not find every single feature you need for the development, for now.
- **Issues with iOS :** Since Google is directly interested in fixing bugs in the shortest amount of time, building Android apps on Flutter is fast and enjoyable.
- **Dart :** This object-oriented programming language is not as great as other languages like C#, Java, Objective C, and JavaScript. Not many freshers will be able to develop an app using this language.

Conclusion

- Flutter is a software development kit from the house of Google that helps in the development of speedy and attractive mobile apps for iOS and Android.

- Flutter can be more efficient and affordable option for building application for small size businesses, start-up, or individuals product application.

References

- **Referred Books :**
  - Beginning App Development with Flutter by Rap Payne
  - Flutter Complete Reference by Alberto Miola
- **Websites :**
  - https://flutter.dev/
- **Other Resource :**
  - https://github.com/flutter/flutter
  - https://stackoverflow.com/tags/flutter

THANKS