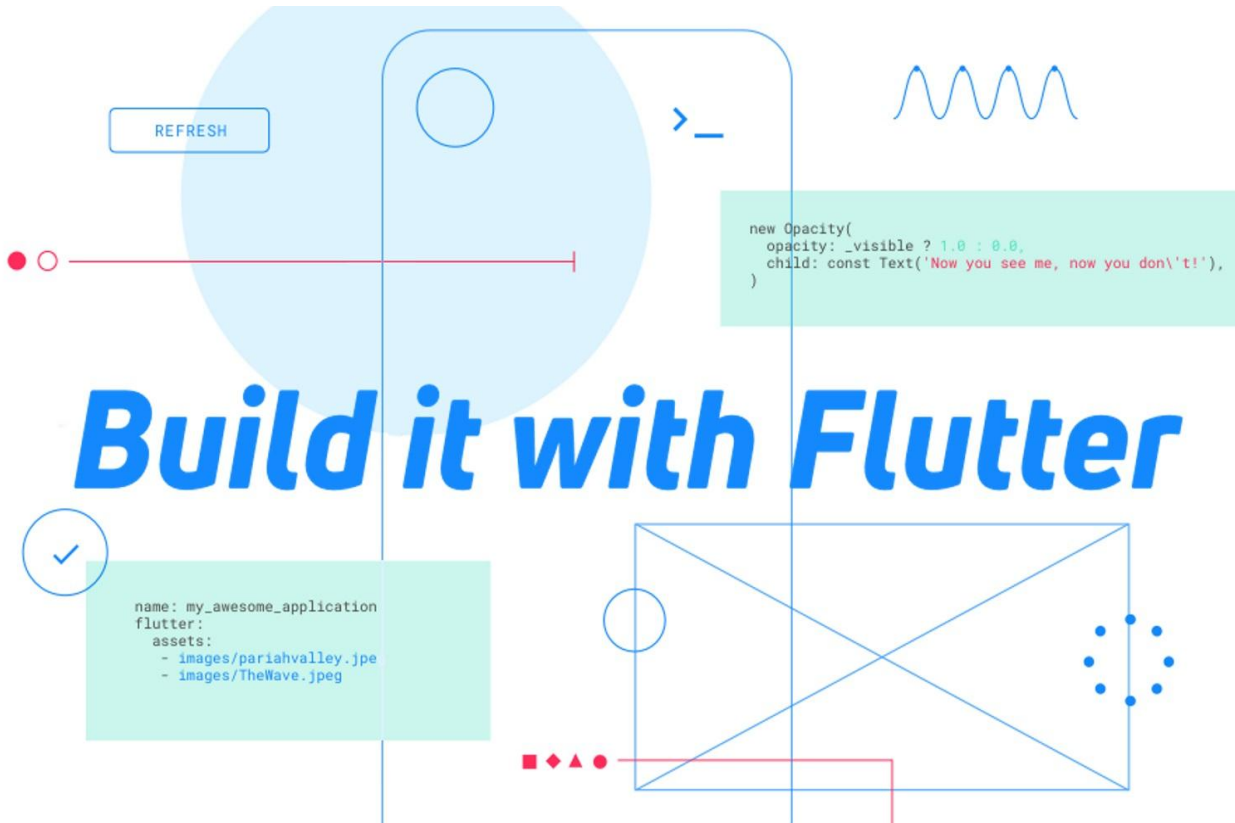




# Mobile App Development with Flutter

Francis Deh





REFRESH

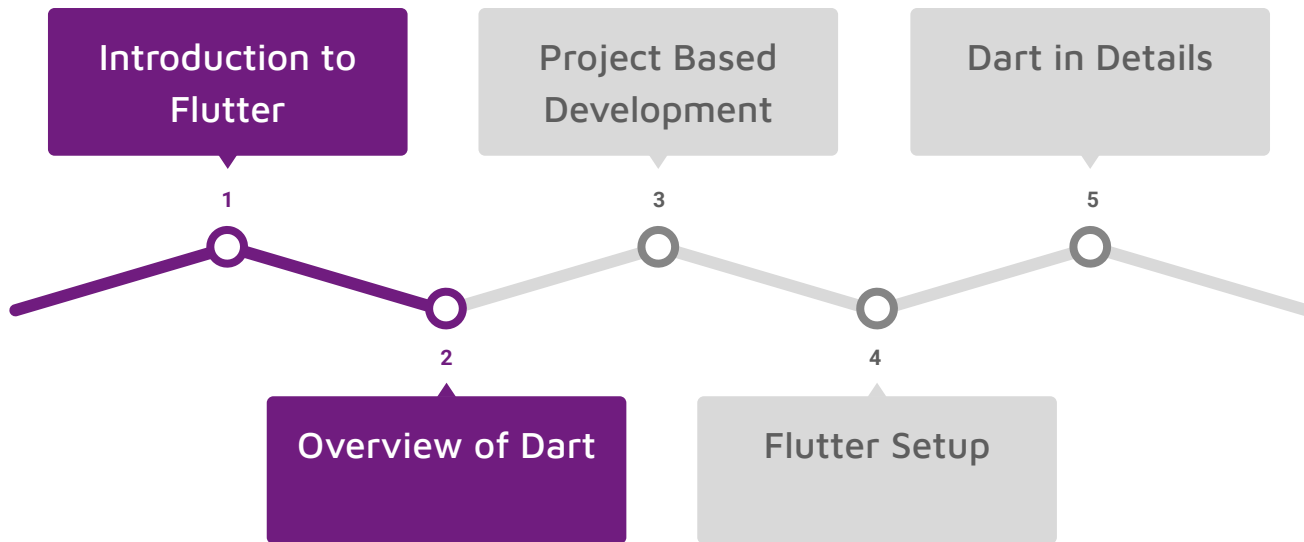
```
new Opacity(  
  opacity: _visible ? 1.0 : 0.0,  
  child: const Text('Now you see me, now you don\'t!'),  
)
```

# *Build it with Flutter*

```
name: my_awesome_application  
flutter:  
  assets:  
    - images/pariahvalley.jpe  
    - images/TheWave.jpeg
```

Mobile App Development with Flutter

# Objectives





# Introduction to Flutter

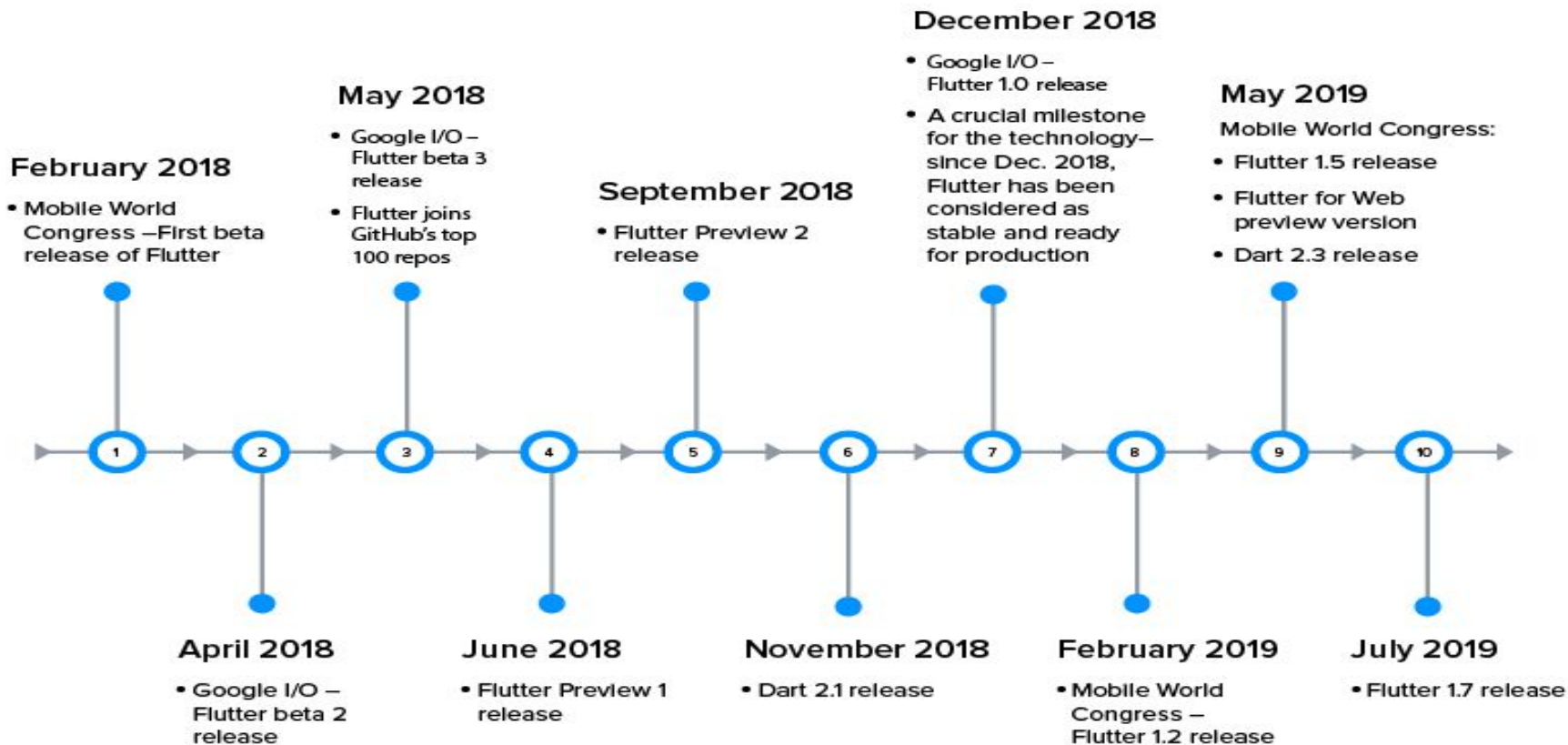


## What is flutter?

Flutter is a set of tooling that allows us to create beautiful apps that run on iOS, Android, the Web, and desktop.<sup>1</sup>



# A brief history of Flutter



# Advantages of Flutter



Hot Reload



Fast development



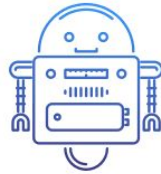
Screen reader



Quick rendering



Cross-platform



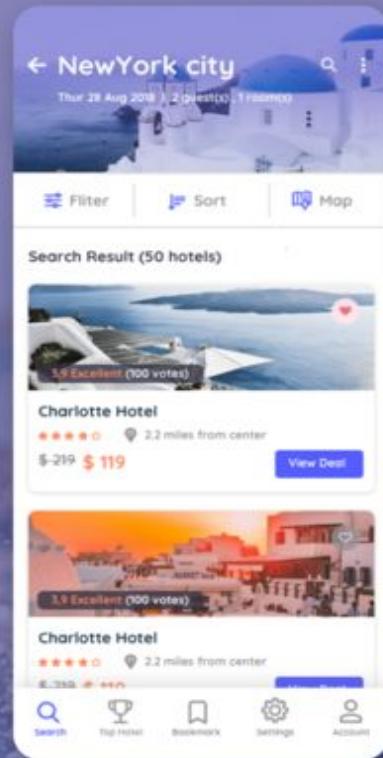
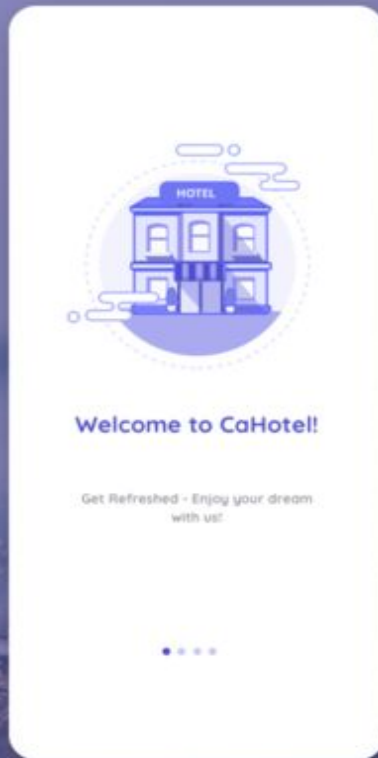
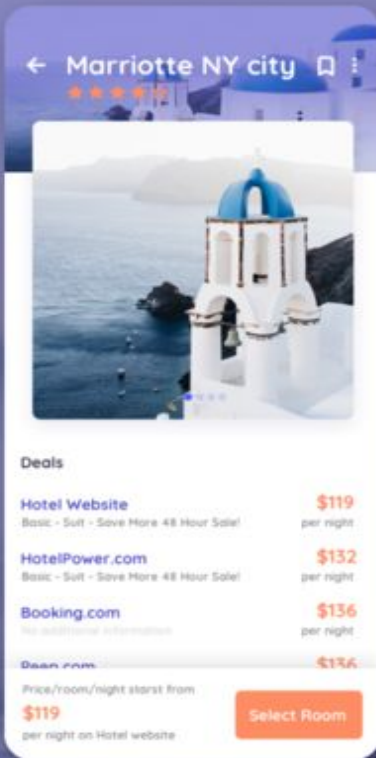
Flutter goes native



Open source & free



Themes for Android &  
iOS





## My profile



Kevin Allen



3285 Traveller points

My next trip

28 Nov

## My bookings



Paris

28 Nov, 2019



London

8 Sep, 2019



Prague

30 Aug, 2019



## Hôtel Le Littré

Paris - 28 Nov, 2019

320€

4 nights

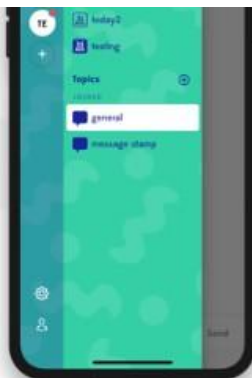


24 days

until trip



× Cancel trip



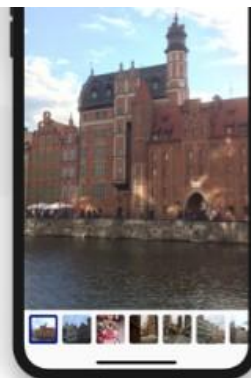
iPhone X — 12.0



iPhone X — 12.0



iPhone X — 12.0



iPhone X — 12.0





Mobile



Web



Desktop



Embedded



Flutter







**Table 1-1.** *Cross-platform development categories*

	<b>Some technologies</b>	<b>Cons</b>	<b>Pros</b>
<b>Progressive Web Apps (PWA)</b>	HTML/CSS, React, Angular, Vue	Not a real app. Runs in a web browser. Not available in app stores. Hard to create a desktop shortcut. Cannot access many of the device's resources like accelerometer, compass, proximity sensor, Bluetooth, NFC, and more	Easy to write
<b>Hybrid</b>	PhoneGap, Cordova, Sencha, Ionic	Runs in a WebView so it can be slow. Nearly impossible to share code with the web app	Easier for web devs to learn because it uses HTML and JavaScript as its language and structure
<b>Compile-to-native solutions</b>	React Native, NativeScript, Flutter, Xamarin	Learning a framework may be difficult. Mastering the toolchain definitely is	Real apps that can be found in the stores and run fast

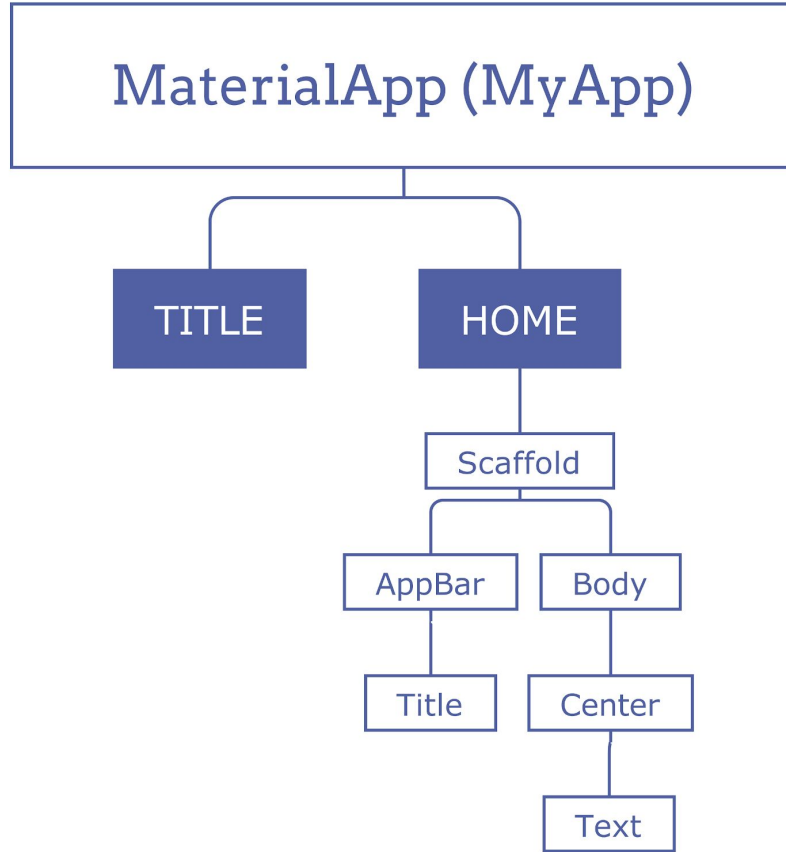
## Native solutions


As of today, there are four fairly popular compile-to-native solutions (Table 1-2).

**Table 1-2.** *Compile-to-native cross-platform frameworks*

	 Xamarin	 NativeScript	 React Native	 Flutter
<b>Year introduced</b>	2011	2014	2015	2018
<b>Backed by</b>	Microsoft	Telerik	Facebook	Google
<b>Presentation language</b>	XAML and/or xamarin.forms	Proprietary but looks like XML	Proprietary but looks like JSX	Dart
<b>Procedural language</b>	C#	JavaScript	JavaScript	Dart

<https://flutter.dev/clock>





```
import 'package:flutter/material.dart';
import 'package:myviu/home_page.dart';

void main() ⇒ runApp(MyApp());

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.white,
      ),
      home: HomePage(),
    );
  }
}
```





# Programming with Dart



# What is Programming?

Computer programming is the process of designing and building an **executable computer program** to accomplish a specific **computing** result.

In this case, we will use **Dart**.





```
//NB: you must download and import share preference package
//stores the usernames in a list
List<String> usernamesList = [];

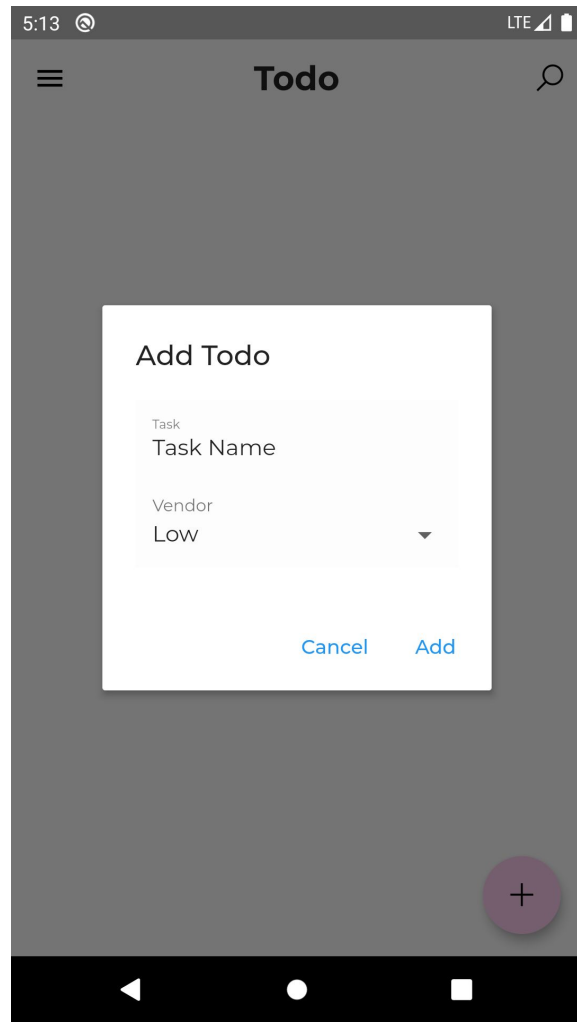
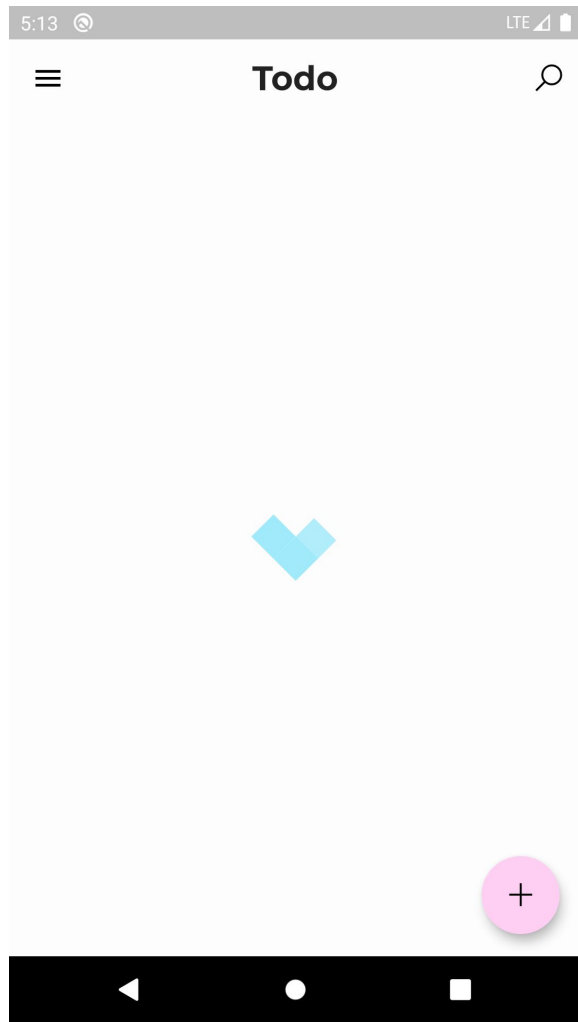
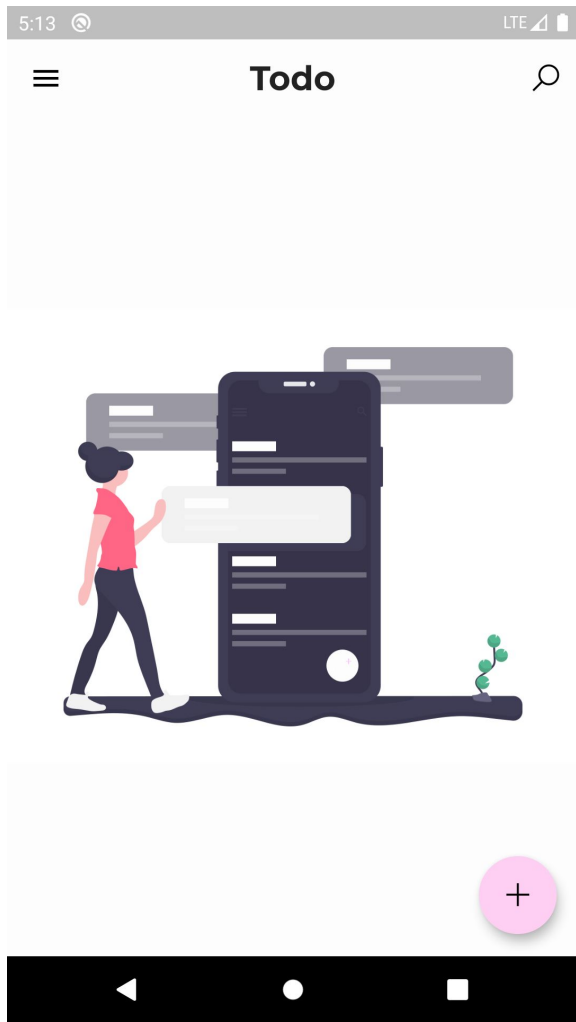
//stores the corresponding passwords in a password list
List<String> passwordList = [];

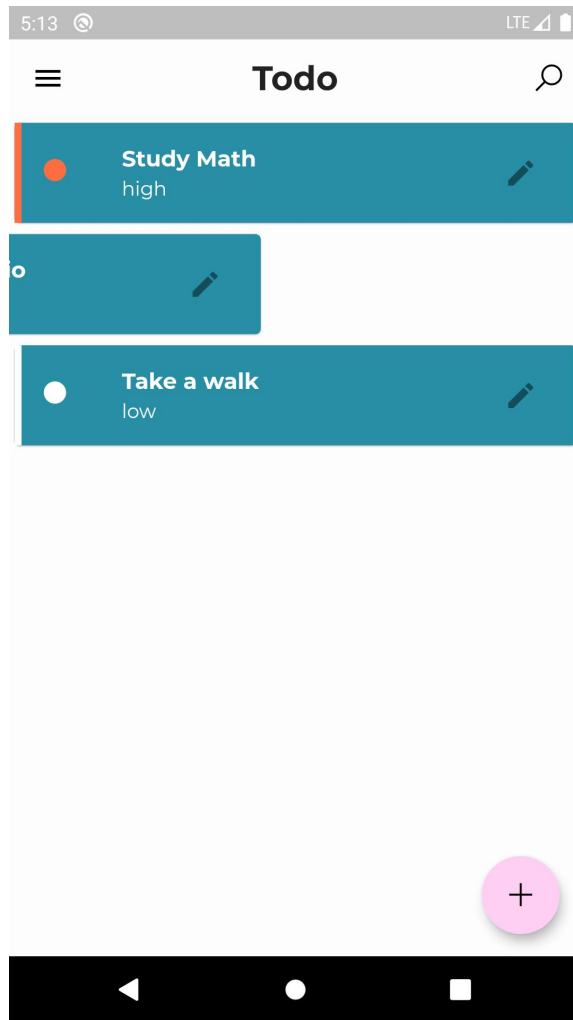
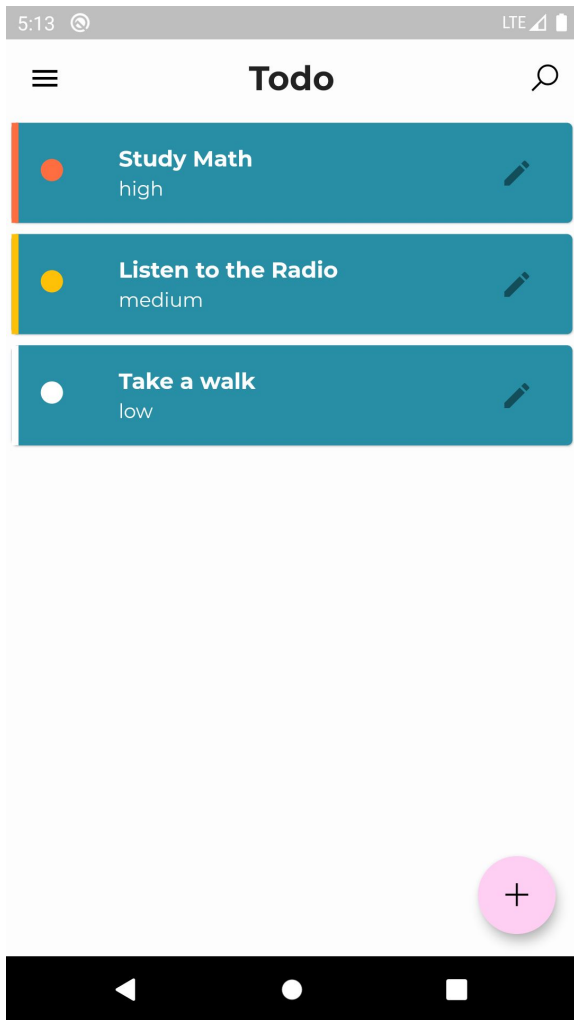
//use shared preferences to get the lists on login page
SharedPreferences prefs = await SharedPreferences.getInstance();
//get the usernames list
usernamesList = prefs.getStringList('usernames', []);
//get the passwords list
passwordList = prefs.setStringList('passwords', []);
```



# **Project Based Development**

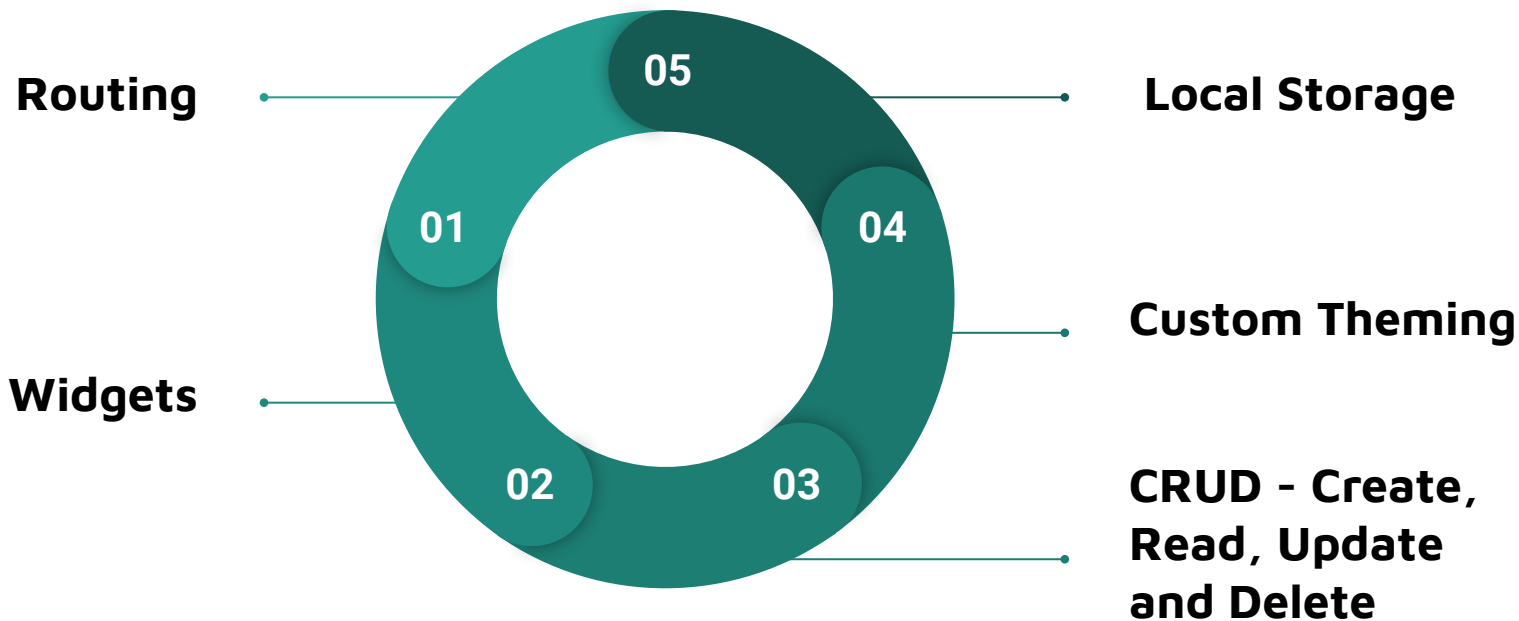








# Todo App Core features



My apps

Shop

Games

Family

Editors' Choice

Account

Payment methods

My subscriptions

Redeem

My wishlist

My Play activity

Parent Guide



## Time App

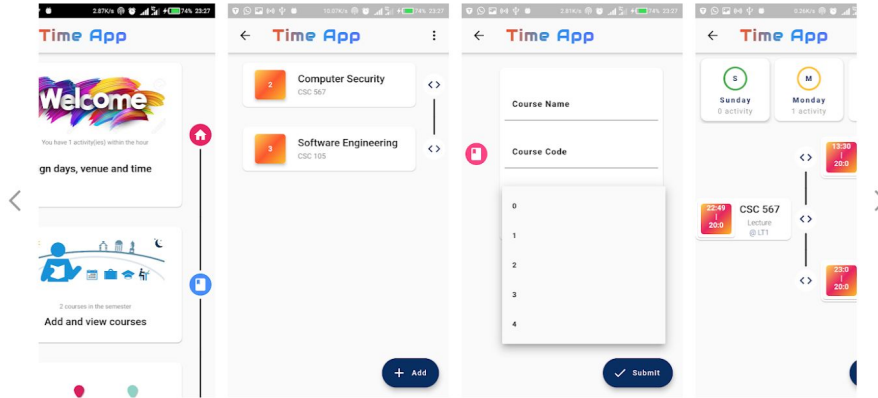
Francis Deh Education

3+

★★★★★ 31

This app is compatible with your device.

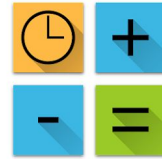
Installed



Time App targets students, especially in the tertiary institution who need to deal with managing

Similar

See more

tCalc - a time calcul  
Sergei Munovarov

★★★★★

Shift Calendar  
Skedlab

★★★★★



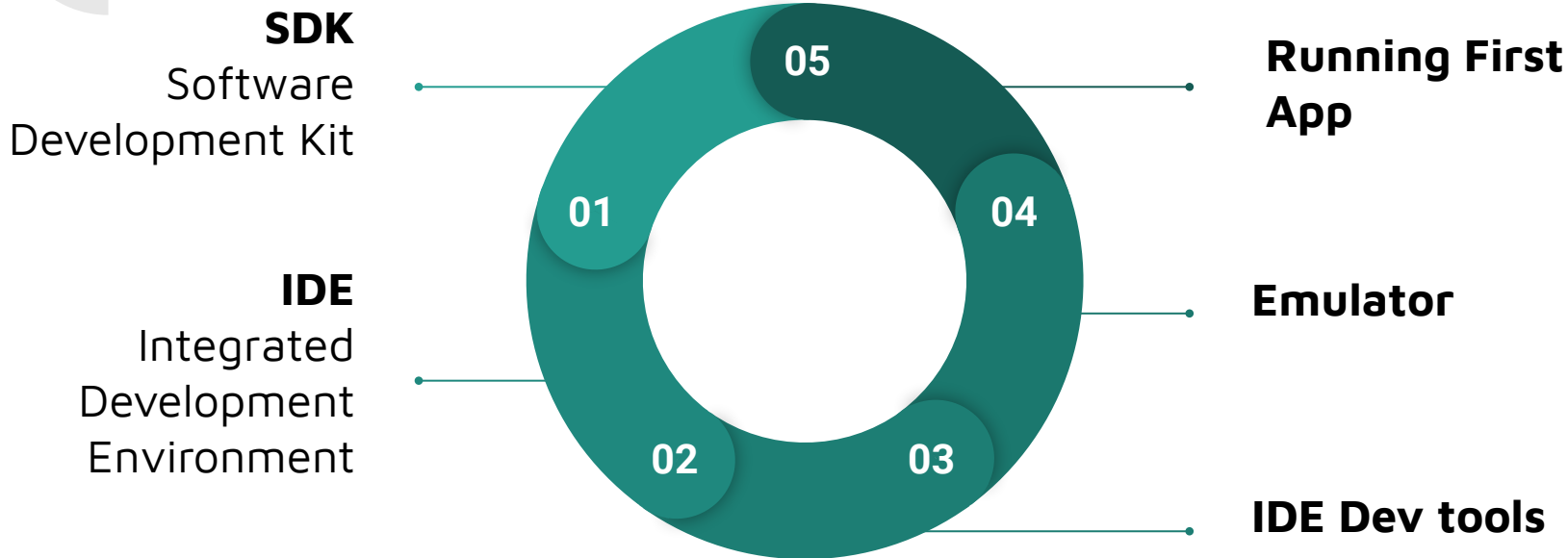


# Flutter Development Setup





# Core Setup Process - (refer to Docs and attached video)





# Using Emulator - (refer to Docs and attached video)

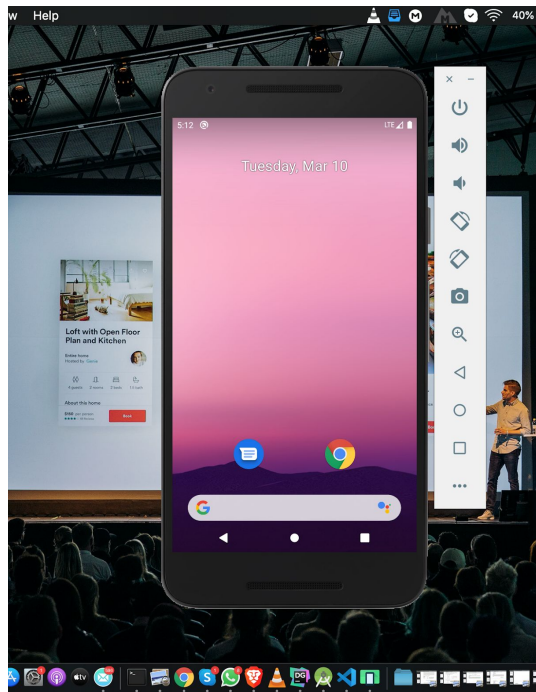
## Android Emulator

---

## Actual Device

The Real Device  
connected can be  
used as well

---





# Dart into Details



# Using the Dart Pad

1. Data types
2. Conditionals
3. Arrays/List
4. Loops
5. Functions
6. Classes



```
1 //Data types in Dart
2
3 int x = 10; // Integers
4 double y = 2.0; // IEEE754 floating point numbers
5 bool z = true; // Booleans
6 String s = "hello"; // Strings
7 dynamic d; // Dynamic variables can change types // at any time. Use sparingly!
8 d = x;
9 d = y;
10 d = z;
```

```
1 // Conditional expressions in Dart
2 // Traditional if/else statement int x = 10;
3 if (x < 100) {
4     print('Yes');
5 } else {
6     print('No');
7 }
8 // Would print "Yes"
9
10 // Dart also supports ternaries
11 String response = (x < 100) ? 'Yes' : 'No';
12
13 // If name is set, use it. Otherwise use 'No name given' String name;
14 String res = name ?? 'No name given';
```



```
1 // Arrays/lists in Dart
2 // Square brackets means a list/array
3 // In Dart, arrays and lists are the same thing.
4 List<dynamic> list = [1, "two", 3];
5 // Optional angle brackets show the type - Dart supports Generics
6 // How to iterate a list
7 for (var d in list) {
8     print(d);
9 }
10 // Another way to iterate a list
11 list.forEach((d) => print(d));
12 // Both of these would print "1", then "two", then "3"
```





```
1 // Looping in Dart
2 // A for loop
3 for (int i=1 ; i<10 ; i++) {
4     print(i);
5 }
6 // Would print 1 thru 9
7 // A while loop int i=1;
8 while(i<10) {
9     print(i++);
10 }
11 // Would print 1 thru 9
```



```
1 //Functions in Dart
2 int triple(int val) {
3     return val * 3;
4 }
5
6 //with Fat Arrow .
7 Function triple = (int val) => val * 3;
8
9
```



```
1 // Classes in Dart
2
3 class Person {
4   // Classes have properties
5   int id;
6   String email;
7   String phone;
8
9   // Typical constructor
10  Person({this.id, this.email, this.phone});
11
12  // Classes have methods
13  void summary() => this.email + this.phone;
14 }
15
```



# Resourceful Links

1. <https://flutter.dev/> - Official flutter website
2. <https://dart.dev/> - Official dart website
3. <https://dartpad.dev/> - Dart Padd
4. Accompanying Books and Youtube links
  - a. <https://b-ok.cc/book/1309331/196488> - Dart Up and Running
  - b. <https://b-ok.cc/book/5336752/c75e2a> - Flutter in Action
  - c. <https://b-ok.cc/book/5304545/f7e889> - Beginning App dev with Flutter
  - d. [https://youtu.be/l9ceqw5Ny-4?list=RDCMUcVD5Vh9LhLBxp3o1vRNyf\\_w](https://youtu.be/l9ceqw5Ny-4?list=RDCMUcVD5Vh9LhLBxp3o1vRNyf_w)  
- Flutter Video Setup
5. Project Github link
6. <https://flutter.dev/docs/get-started/install> - Flutter Setup
7. <https://play.google.com/store/apps/details?id=com.francisdeh.timeapp> - Time App
8. <https://mega.nz/#F!vPwTESiS> - Cloud link for all setups, books and videos
9. Github link for todo project

