

Introduction to Flutter

Developing a simple mobile app
By Wan Muzaffar Wan Hashim



Good morning. Get the slide here

<https://tinyurl.com/flutterutem03>

Youtube video:

https://www.youtube.com/watch?v=_OhvwhT2Tks&list=PLeEjv6GjeKWdlKWowAmvDsrxGC6fncE7H

<https://us02web.zoom.us/j/84448034259?pwd=ODV5VGxEMzVvd21MRmtWTXZwazJadz09>

<https://forms.gle/VTQYUzoM7DURknGq5>

Installation

Download Android Studio : <https://developer.android.com/studio> (Install right Away)

Download Flutter SDK: <https://docs.flutter.dev/get-started/install/windows> (Download and Unzip)

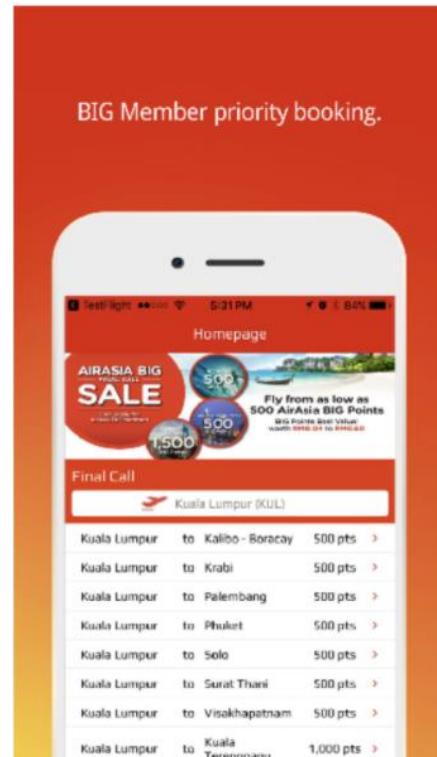
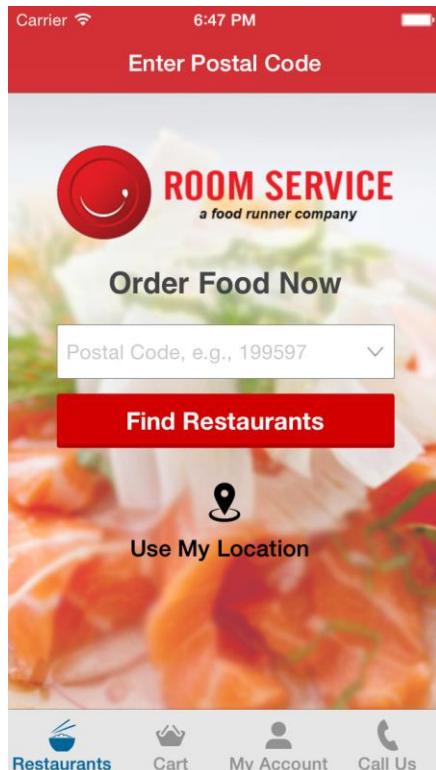
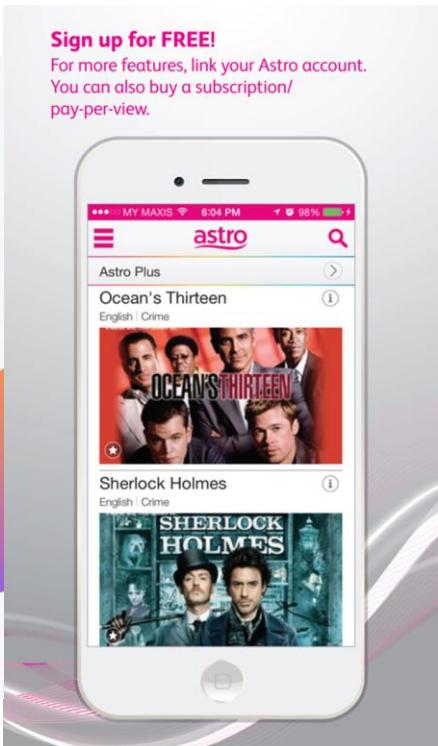
Once done let me know on the chat that you are done

Book Link

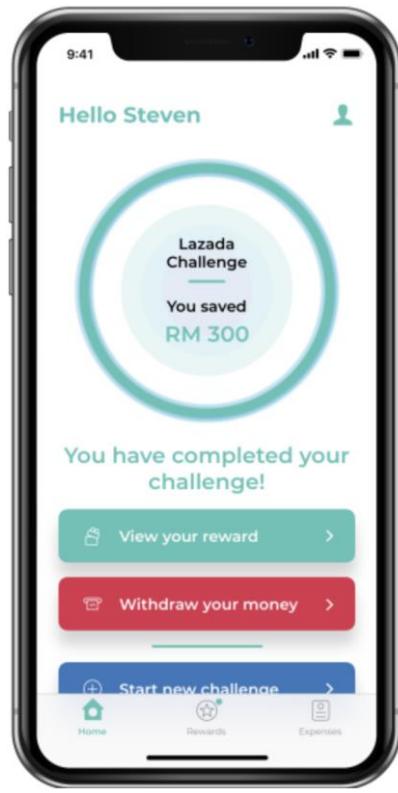
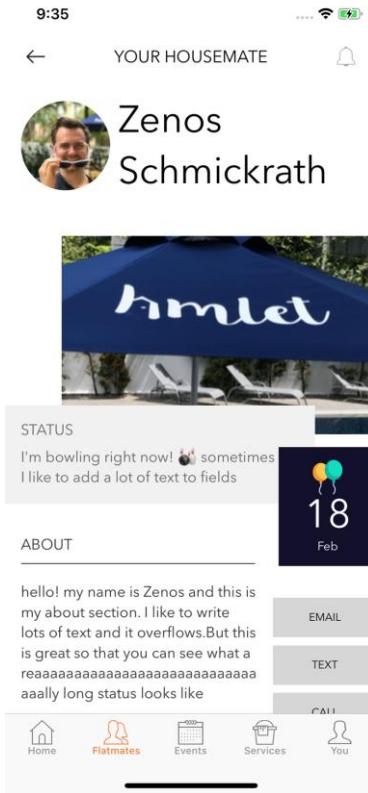
https://drive.google.com/file/d/1Xs_xS8hJ0vh1g0QiA-L7e3HwymDlFrX/view?usp=drive_link
https://drive.google.com/file/d/1bjEoCb8e6j5_71JdkQ9YRyF4lo9XQK73/view?usp=sharing

Bahasa video:

<https://www.youtube.com/playlist?list=PLeEjv6GjeKWdlKWowAmvDsrxGC6fn>
cE7H



<https://usepod.com/>



AIRBUS

Mobile App Development

- A mobile application is a software application designed to run on smartphones, tablet computers and other mobile devices.
- Users on smartphones typically check the news, weather, email or their social networks. They have a choice between the mobile web version or a specially-created mobile app.

Mobile App Dev: Current State

Native Development	Crossplatform Development
<ul style="list-style-type: none">• Android SDK - Kotlin or Java• iOS - Swift or Objective C (Mac)	<ul style="list-style-type: none">• Flutter - Dart -> Transforming a CP code into native code• React Native - JS• Ionic - JS -> Website inside app (<i>Hybrid</i>) - MySejahtera• Xamarin - .NET

Some abbreviation

.exe : windows

.dmg : Mac

.apk: android package / app bundle

.ipa: iOS app bundle

Mobile App Types

- **Native**
 - Programmed using Swift/Objective C on the iPhone or using Java/Kotlin on Android devices.
- **Crossplatform**
 - Mix between these two types of mobile applications.
 - Normally based on web programming language, eg: HTML, CSS, Javascript, Dart
 - Built once to be run on Android and iOS.
- **Web Apps / Progressive Web Apps.**
 - Web based.
Runs in the phone's browser.
 - Can have native features based on HTML5 -> take picture, location , video...

When to choose what?

Native	Crossplatform
I need to have 2 developers Or I need to have 1 person that knows two languages	1 developer only..
4 months	$\frac{2}{3}$ to $\frac{3}{4}$ of Native - 3 months <ul style="list-style-type: none">- We need to buffer times to do individual configuration for iOS and Android
Stable, spend less time to look for support	Still new, less resources, you might spend more time to look for support
RM 60 000 (Initial cost)	RM 45 000 (Initial cost)
Maintenance cost is lower	Maintenance cost is higher
Long term project...	Short term project

Native	Crossplatform
You have a business you want a new channel You build something that is compulsory	You don't have a business you want to create an MVP (Minimum Viable Product)
You need to use latest or complex device features..	You use normal or popular device features
you need to connect with the hardware (eg : Security,)	--
A lot of device features/ integration 5 <ul style="list-style-type: none"> - Camera - Push notification - Real time notification (driver location) - Payment provider .. 	Less than 5 device features <ul style="list-style-type: none"> - picture

APPLICATIONS

Home

Contacts

Phone

Browser

...

APPLICATION FRAMEWORK

Activity Manager

Window Manager

Content Providers

View System

Package Manager

Telephony Manager

Resource Manager

Location Manager

Notification Manager

LIBRARIES

Surface Manager

Media Framework

SQLite

OpenGL | ES

FreeType

WebKit

SGL

SSL

libc

ANDROID RUNTIME

Core Libraries

Dalvik Virtual Machine

LINUX KERNEL

Display Driver

Camera Driver

Flash Memory Driver

Binder (IPC) Driver

Keypad Driver

WiFi Driver

Audio Drivers

Power Management

Quiz

1. AirAsia - Native.. / It's a channel
2. Astro Go - Native , security..
3. Astro Awani - Crossplatform / Xamarin / simple no need security
4. MAE - Cross platform ? React Native? / [I think they are wrong]
5. MySejahtera - Crossplatform / It takes a template, short term, they need it fast
6. Selangkah - PWA
7. Setel - Crossplatform / / New business
8. GSC/TGV - Native , It's a channel
9. CIMB - - Native, Stable, long term

Supply Demand

HTML / CSS / JS -> Almost everyone can do it, from certificate to degree

PHP Developer -> Can hire from diploma level

Ionic Developer, PWA ->Some Diploma Mostly Degree , UITM, UTEM
(Laravel)

Android -> Some Diploma Mostly Degree ,UTM, UTEM

VueJS -> Easy to learn..

Flutter -> Not in school official syllabus, only those who self learn will know, got demand , can self learn.. Might have some self learned student

React.JS -> It's hard... not many people learn it from

React Native - > Too hard for university student to understand.., demand not as big as flutter

iOS - Supply low Demand high , you need to invest to buy a mac to learn

Cost

Developer Account...

Google Play - 25USD per lifetime..

Phone / tablet

Apple iOS Developer Account

- 99 USD per year [company account] / 299 USD per year [enterprise account]
- Macbook
- iphone/ ipad/ ipad mini
- D-U-N-S (<https://developer.apple.com/support/D-U-N-S/>)

Stack...

	JS	Dart	Python	Java	C++	R	PHP	Kotlin	Ruby	Swift	C#
Frontend website	VueJS, ReactJS, Angular Vanilla	Flutter	Flask/Django	JSF, Angular			Pure PHP, Laravel, CakePHP, CodeIgniter		Ruby on Rails		YES .NET framework!
Backend / API	Express		Flask/Django	Spring / Spring Boot			Pure PHP, Laravel, CakePHP, CodeIgniter		Ruby on Rails	Can	
Database	MongoDB		MySQL or Postgresql	Postgresql or Oracle			MySQL		MySQL		MSSQL
Mobile app	React Native, Ionic	Flutter	NO	Android app with Java				YES		iOS App	Xamarin
Desktop app	Electron	Flutter	Qtinker	YES, JavaFX	YES					YES / Mac OS	YES
Data science	Yes not widely used		YES	NO		YES					

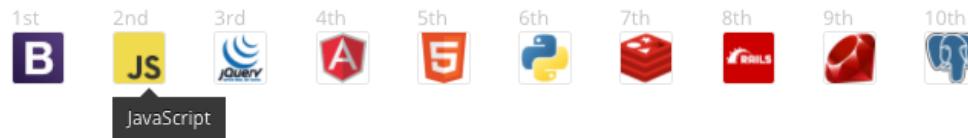
Application and Data

Application and Data

Sponsored

**Datadog**
See metrics from all of your apps, tools and services in...
[Visit Website](#)

TOP 10 TOOLS & SERVICES



STACK LAYER



DevOps

DevOps

Sponsored



Datadog

See metrics from all of your apps, tools and services in...

[Visit Website](#)

TOP 10 TOOLS & SERVICES



STACK LAYER



Categories of a stack in Airbnb

- Database - MySQL, Redis, RDS, elasticcache, EBS
- Frontend Framework - React
- Operating System/ Runtime environment - Amazon EC2,
- Backend Framework / Web Server: - Nginx, Rails
- Programming language - Javascript, Java, Ruby
Big data processing: Hadoop, Airflow
- Mobile app : Java
- Object Storage : Amazon S3
- CSS Framework - SASS
- CDN - Cloudfront

Uber Stack case study

- Database: MySQL , Postresql, MongoDB, Redis,
- Operating System/ Runtime Environment : Node.js, Amazon EC2
- Frontend: jQuery, React,
- Backend Framework / Web Server: Nginx, Backbone.js
- Programming language - Python, Golang
- Big data processing: Kafka, Cassandra, Spark, Hadoop, Thrift, RIBs, AresDB
- Mobile app : Swift, Java (Android), Objective C

<https://www.youtube.com/watch?v=umWABit-wbk>

LAMP Stack

Database - MySQL

Frontend Framework - PHP, Perl, Python

Operating System/ Runtime environment - Linux

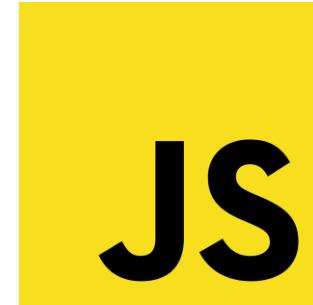
Web Server - Apache / Nginx



Content Management System

- Wordpress
- Drupal
- Magento
- WooCommerce by Wordpress
- Joomla
- Zencart
- Shopify

Bahasa pengaturcaraan membina API



Framework untuk membina API



Spring
Boot



What is an API

- API stands for Application Programming Interface.
- An API is a set of rules and protocols that allows different software applications to communicate and interact with each other.
- APIs define how different components of software systems should interact, specifying the methods, data formats, and protocols that should be used.
- APIs enable developers to access the functionality of existing software components, services, or platforms without needing to understand the underlying implementation details.
- APIs provide a layer of abstraction, allowing developers to focus on using the functionality provided by an API rather than worrying about the internal complexities.

CMS- (Java, PHP
(Laravel, CI,
CakePHP) Python,
ASP, .Net, Javascript
- ReactJS, VueJS,
Angular)

Payment - Paypal
Chat - Twilio,
Firebase
Facerecognition -
wyse.ai, Firebase,
Fakuti AI
Email, SMS
Push notification

Front Website (HTML, CSS, JS)
-Pure CSS, Bootstrap, Material,
- Vanilla, VueJS, ReactJS, AngularJS

Database
SQL and NoSQL
-MySQL, SQL,
MSSQL,
-Firebase,
MongoDB,
Memcached,
Neo4J

API (Java
Server,
PHP,
Python,
ASP, .Net,
Javascript)

JSON or
XML

iOS

Android



What is Flutter

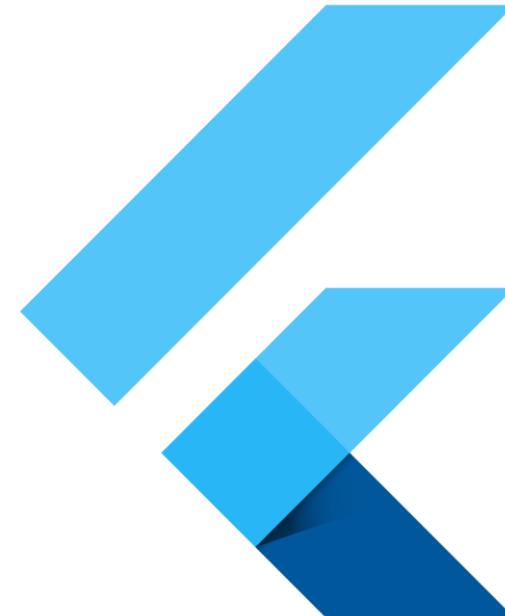
Open source UI Framework by Google

Able to create iOS, Android and web application using Dart

High performance, high fidelity, low latency, as it renders the Native UI.

Use DART as main programming language

Open source / github.



About Dart

Dart is a programming language developed by Google

Learning it isn't hard if you have experience with Java or JavaScript. You will quickly get it.

You can use dartpad as an online compiler of Dart

<https://dartpad.dev/>



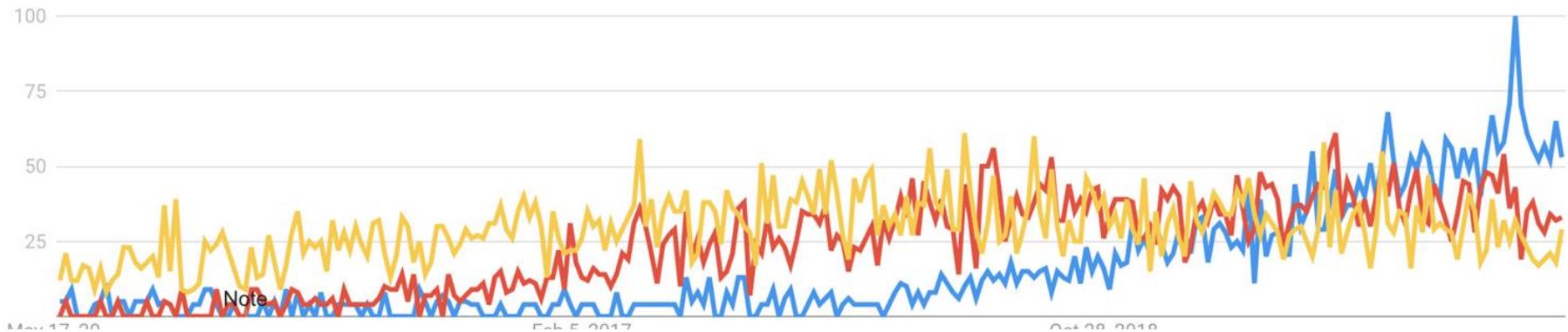
Dart

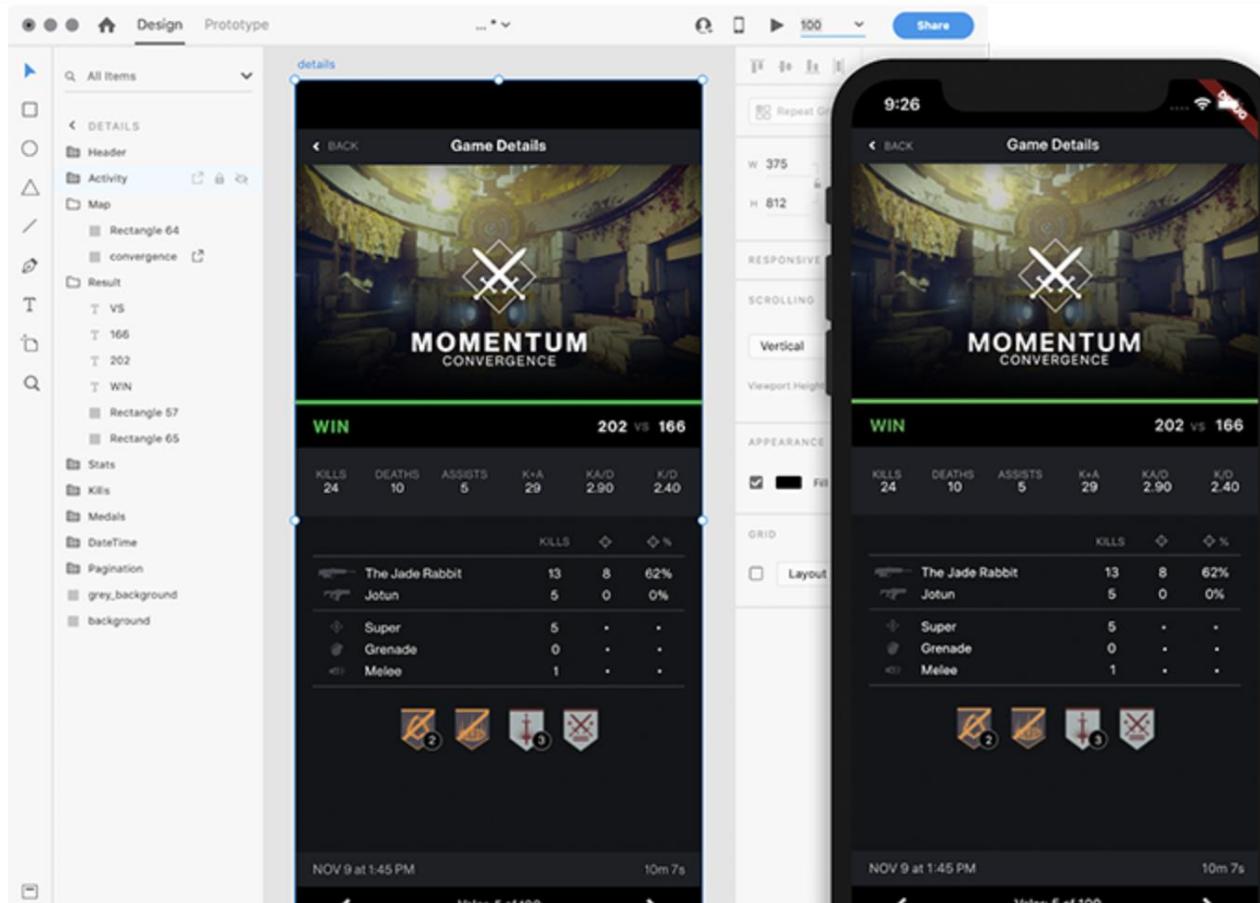
Who uses Flutter



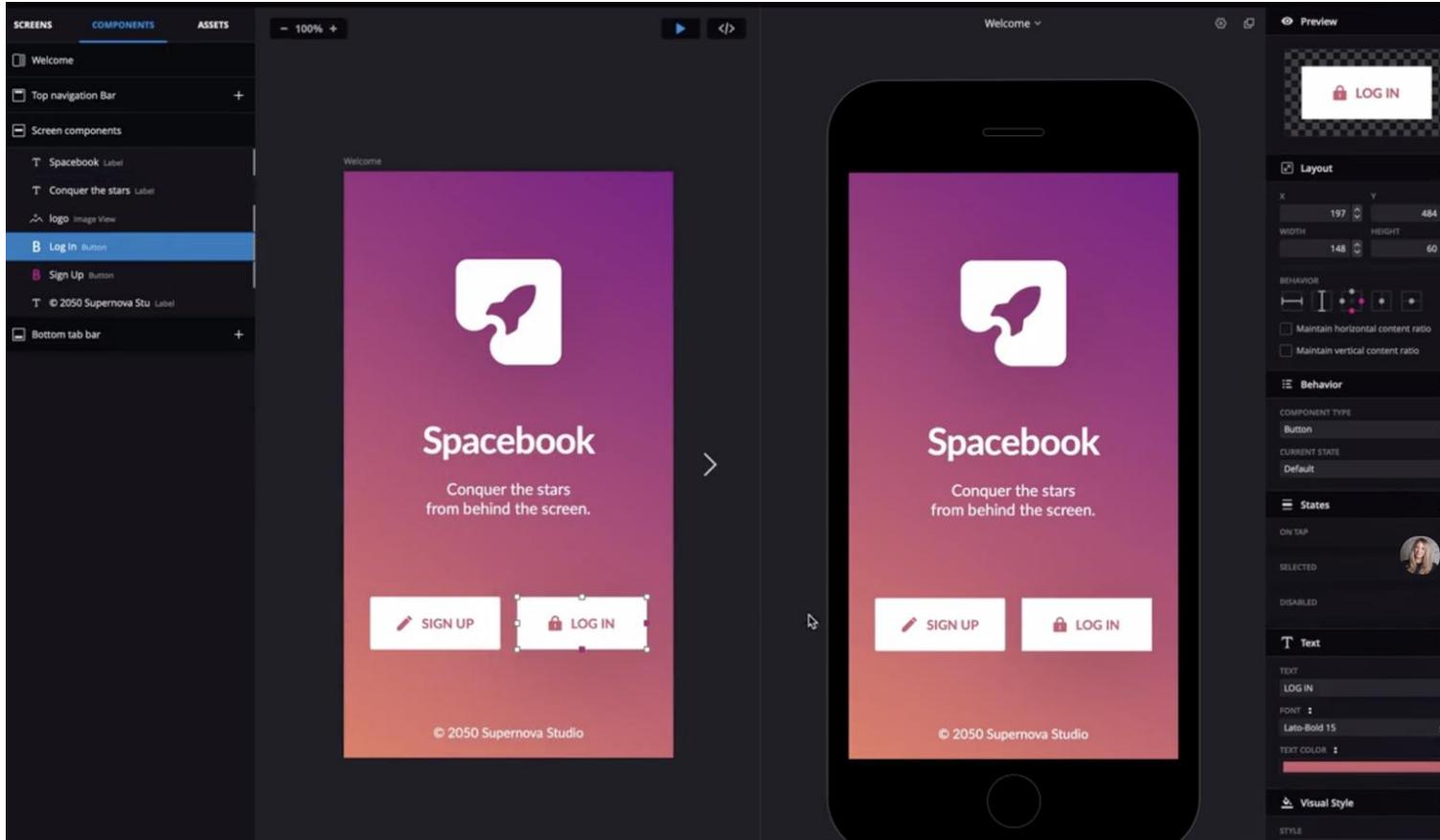
<https://flutter.dev/showcase>

Malaysia Google Trend (over 5 years)



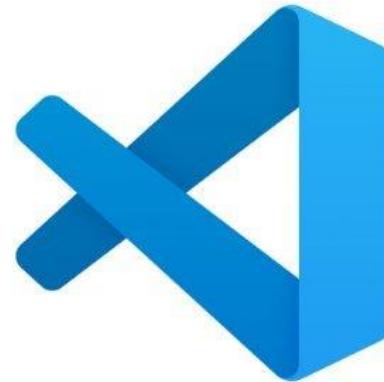


Bridge gap between designer and developer - XD Flutter integration



Bridge gap between designer and developer - Supernova.io

Setup your Editor



<https://flutter.dev/docs/get-started/editor>

You will need to configure an emulator after setting up the SDK.

Online Editor (Demo purposes - no setup)

The screenshot shows a web-based code editor interface for Flutter development. The left pane displays the Dart code for a StatelessWidget named MyWidget, which contains a Scaffold with a central 'Hello World!' text. The right pane shows a preview of the app titled 'BMI Calculator' with the text 'Hello World!'. The top bar includes a 'Save' button, 'Settings' button, 'Change View' button, and an 'astro GO' logo. A watermark for 'CodePen: Unlock all of CodePen' is visible at the bottom.

```
8  );
9 }
10
11 class MyWidget extends StatelessWidget {
12   @override
13   Widget build(BuildContext context) {
14     return Scaffold(
15       appBar: AppBar(
16         title: Text("BMI Calculator"),
17         body: Center(
18           child: Text(
19             'Hello World!',
20             style:
21               Theme.of(context).textTheme.headline4,
22             ),
23           );
24     );
25   }
26 }
```

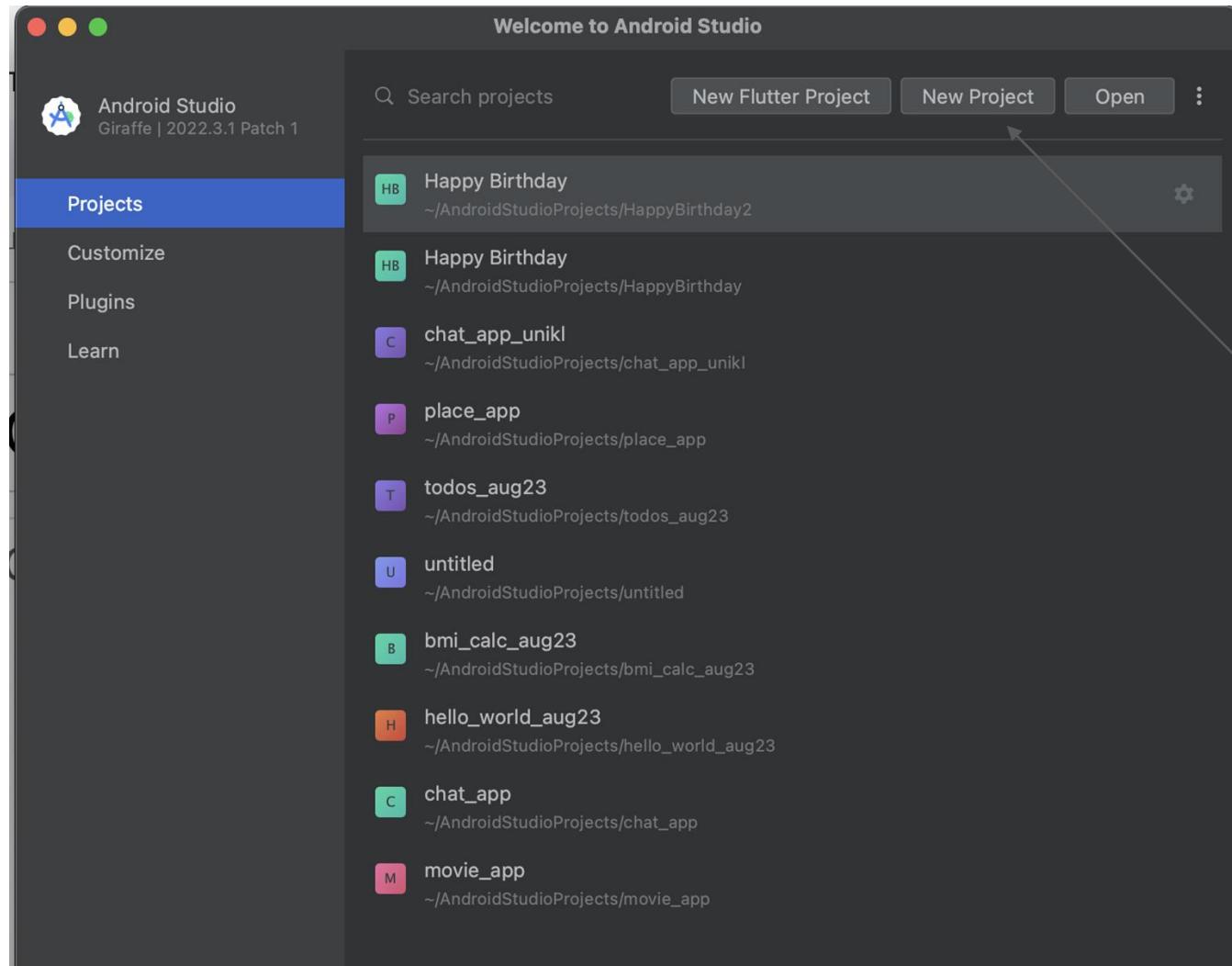
CodePen: Unlock all of CodePen

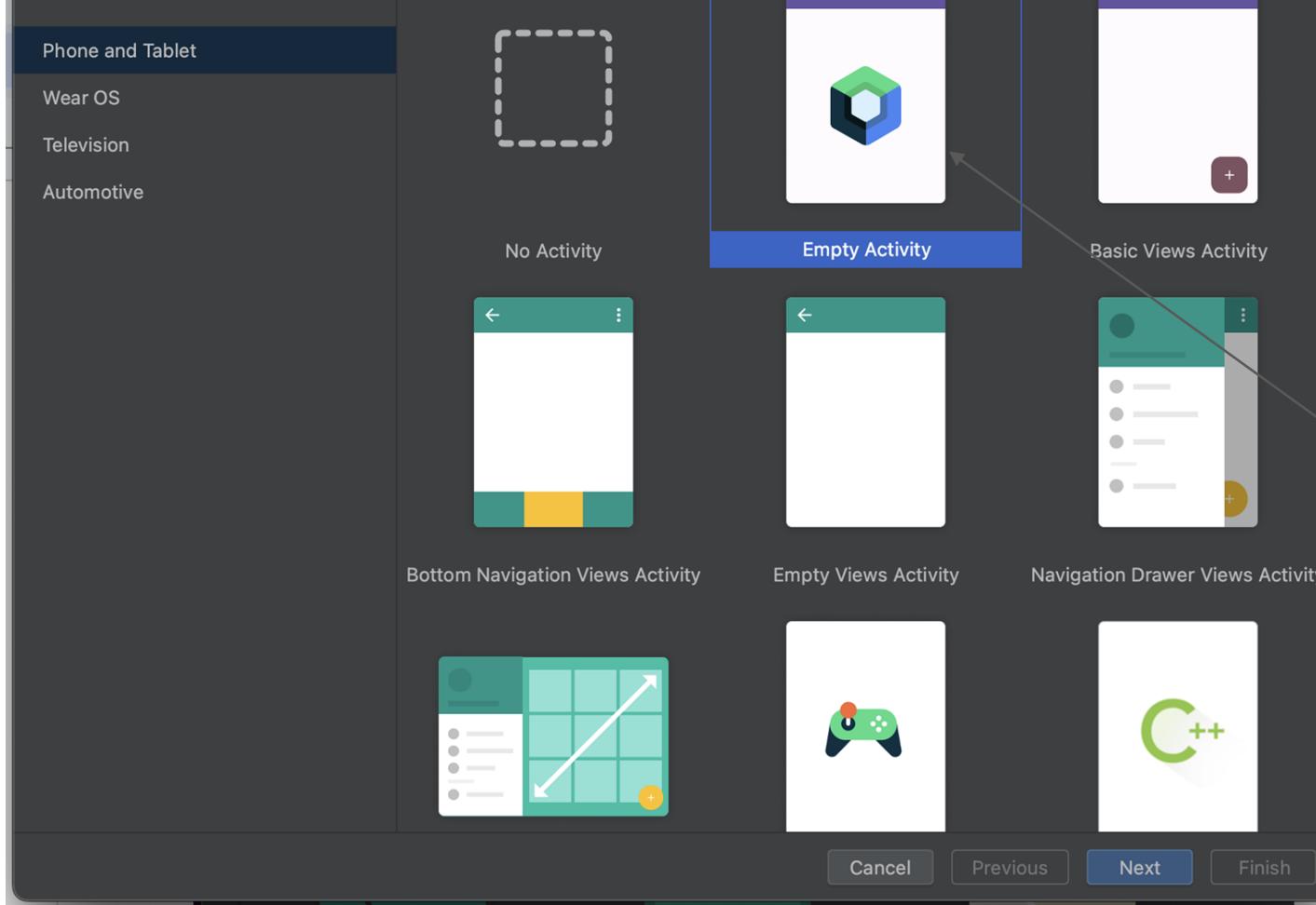
<https://codepen.io/pen/editor/flutter>

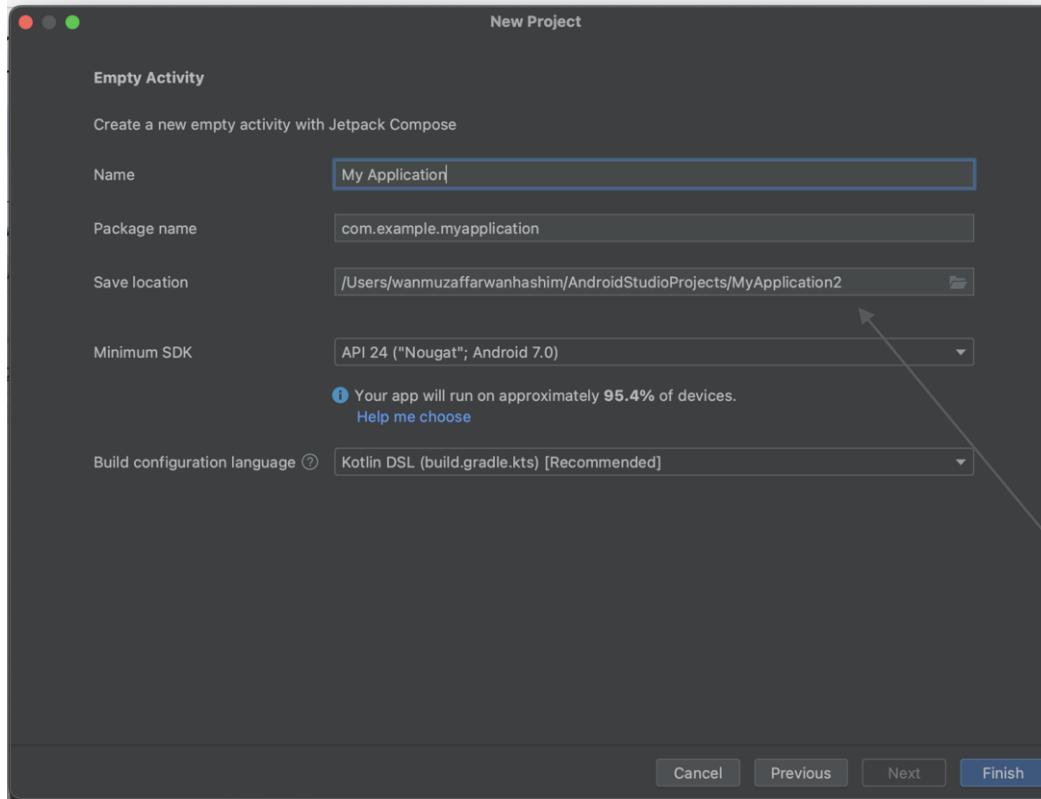
Flutter Installation

Download and Install

- 1) Flutter sdk : <https://docs.flutter.dev/get-started/install> (unzip the .zip folder)
- 1) Android Studio IDE: <https://developer.android.com/studio>
 - 1) Install Flutter plugin for Android Studio







Project name

- My Application (no need change)

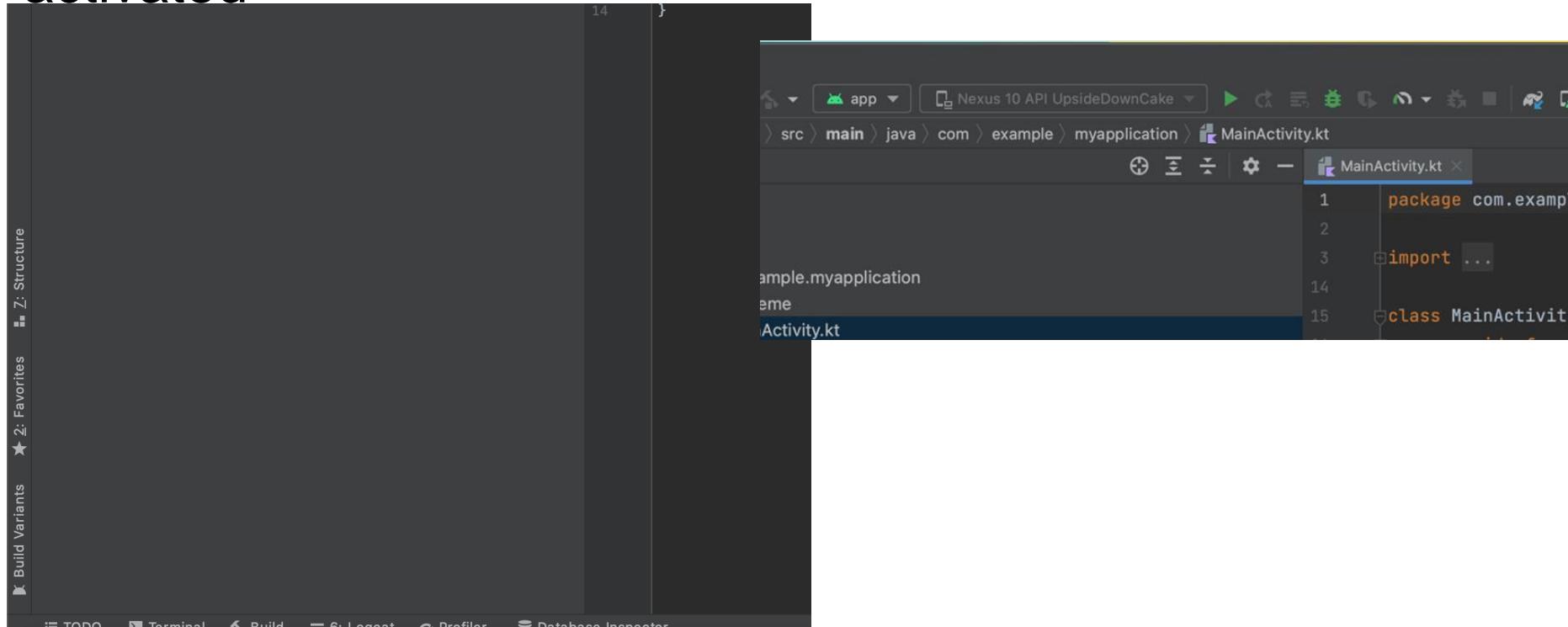
Save location

- Select a save location

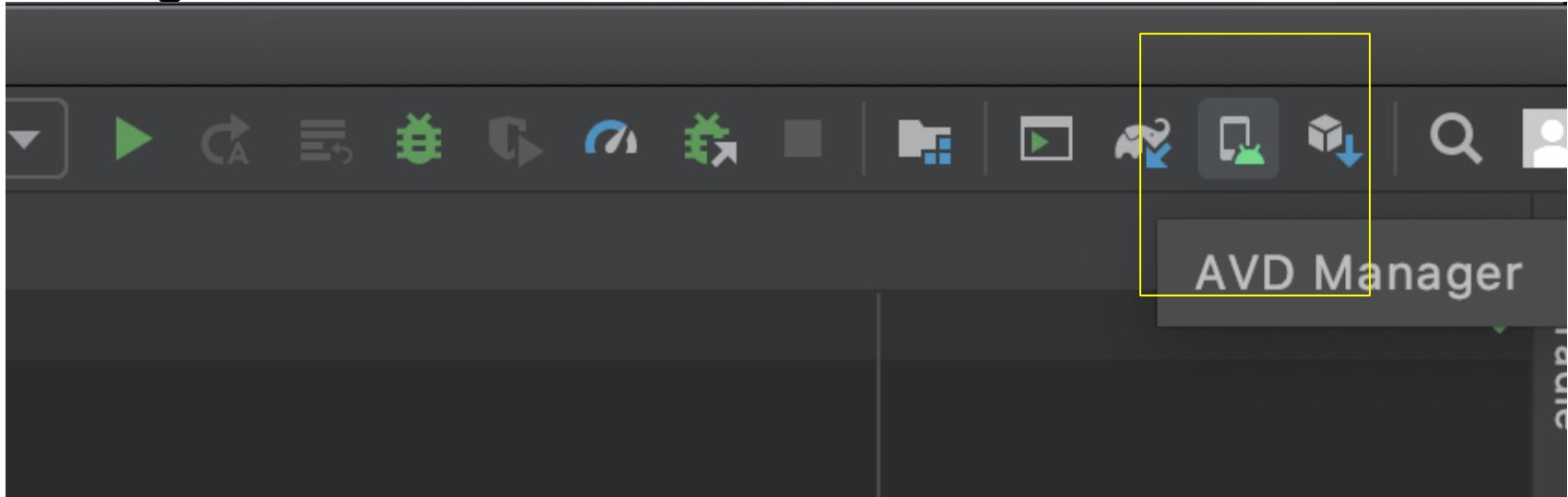
Other than that,
dont do anything

Remember your save
location to retrieve your
project later

If everything is OK you should see Gradle sync finished at the bottom left of Android Studio, and the play button is activated



Look at the toolbar for AVD (Android Virtual Device) Manager



Your virtual device should appear here



Your Virtual Devices

Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
Pixel	Pixel C API 27	2	2560 × 1800: xhdpi	27	Android 8.1 (Google ...)	x86	4.0 GB	  
Pixel	Pixel 2 API R		1080 × 1920: 420dpi	R	Android 11.0 (Google ...)	x86	9.6 GB	  
Pixel	Nexus S API 27		480 × 800: hdpi	27	Android 8.1 (Google ...)	x86	3.9 GB	  

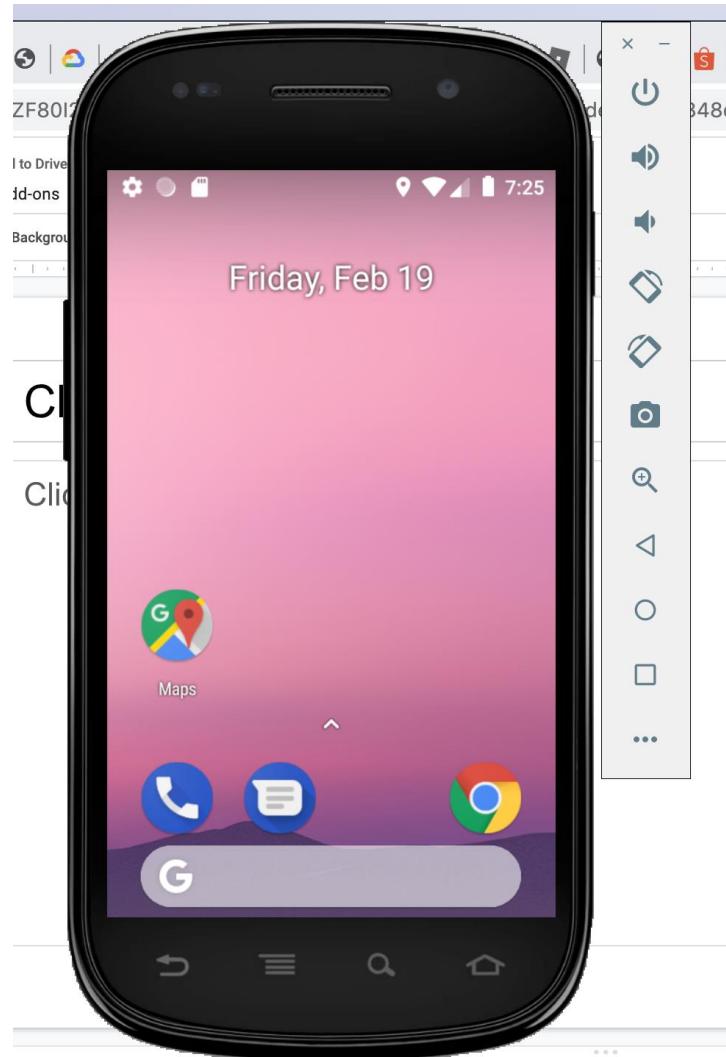
Press Play to open the device

Your Virtual Devices

Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
Pixel	Pixel C API 27 2		2560 × 1800: xhdpi	27	Android 8.1 (Google ...)	x86	4.0 GB	▶ ✎ ▼
Pixel	Pixel 2 API R	►	1080 × 1920: 420dpi	R	Android 11.0 (Google ...)	x86	9.6 GB	▶ ✎ ▼
Pixel	Nexus S API 27		480 × 800: hdpi	27	Android 8.1 (Google ...)	x86	3.9 GB	▶ ✎ ▼

This should come out



My Application – MainActivity.kt [My_Application.app.main]

View Navigate Code Refactor Build Run Tools VCS Window Help

Pixel 2 API UpsideDownCake

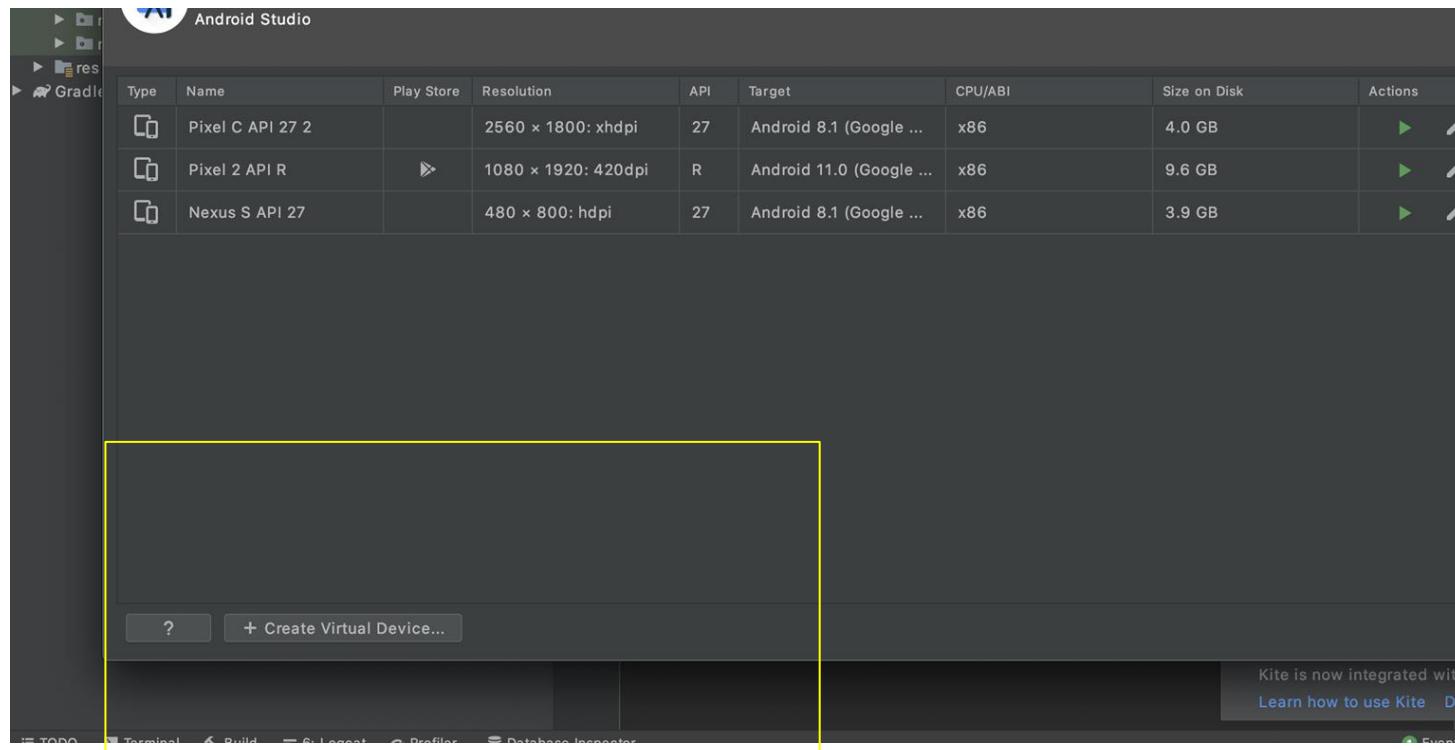
java > com > example > myapplication > MainActivity.kt

```
1 package com.example.myapplication
2
3 import ...
4
5 class MainActivity : ComponentActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContent {
9             MyApplicationTheme {
10                 // A surface container using the 'background' color from the
11             }
12         }
13     }
14 }
```

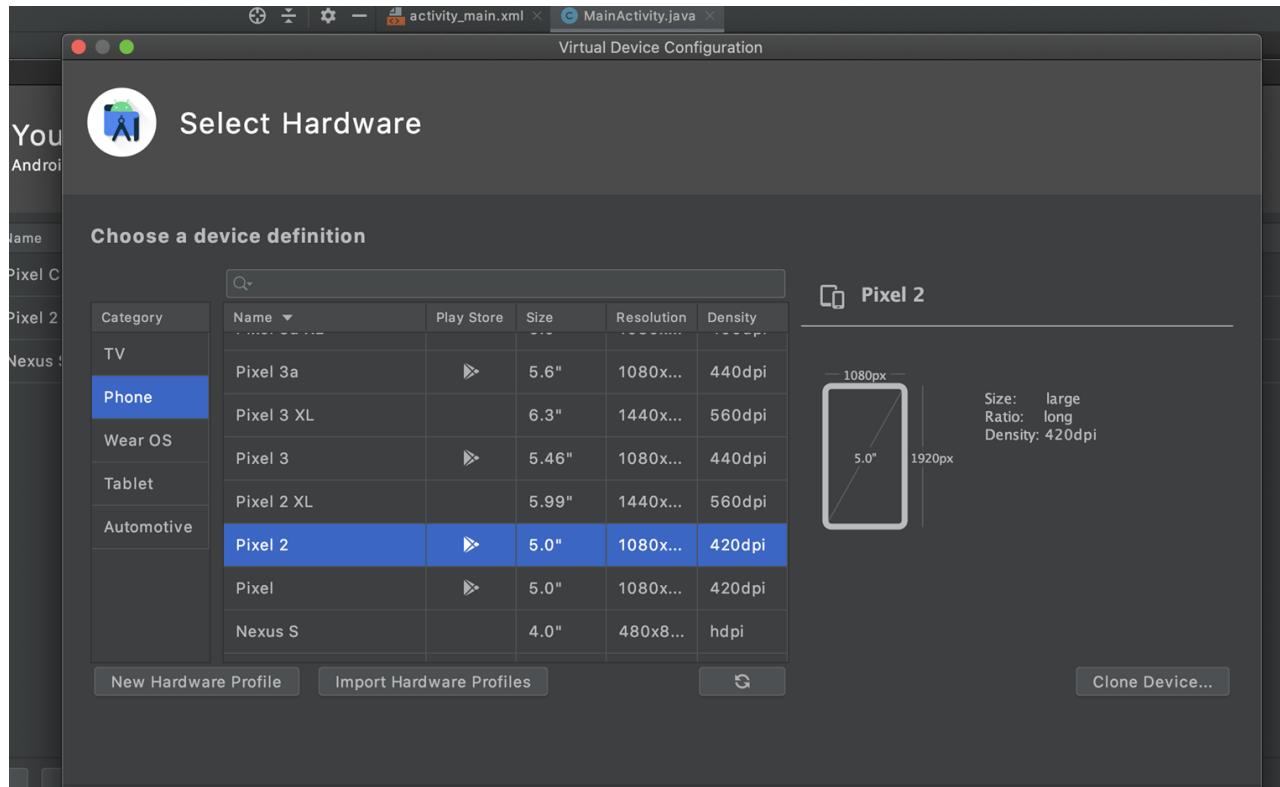
Hello Android should come out



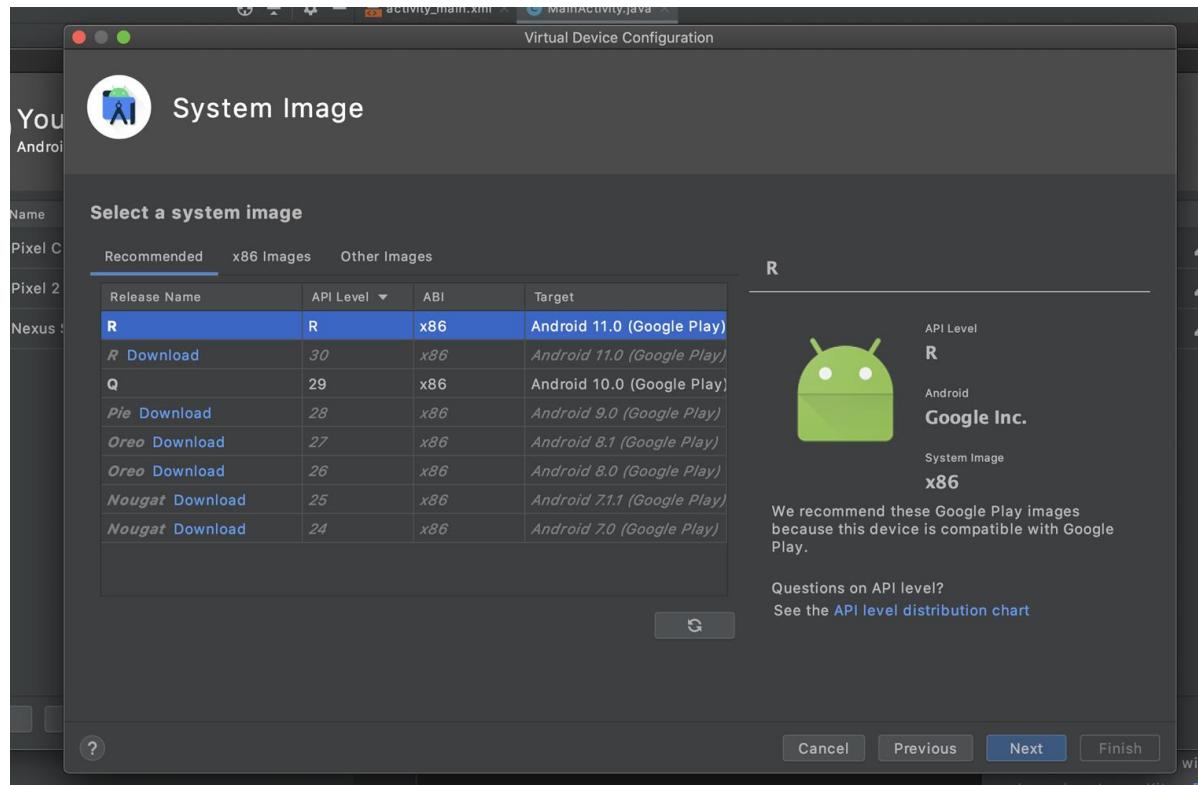
If you don't have Virtual Device, Click Create Virtual Device at the bottom



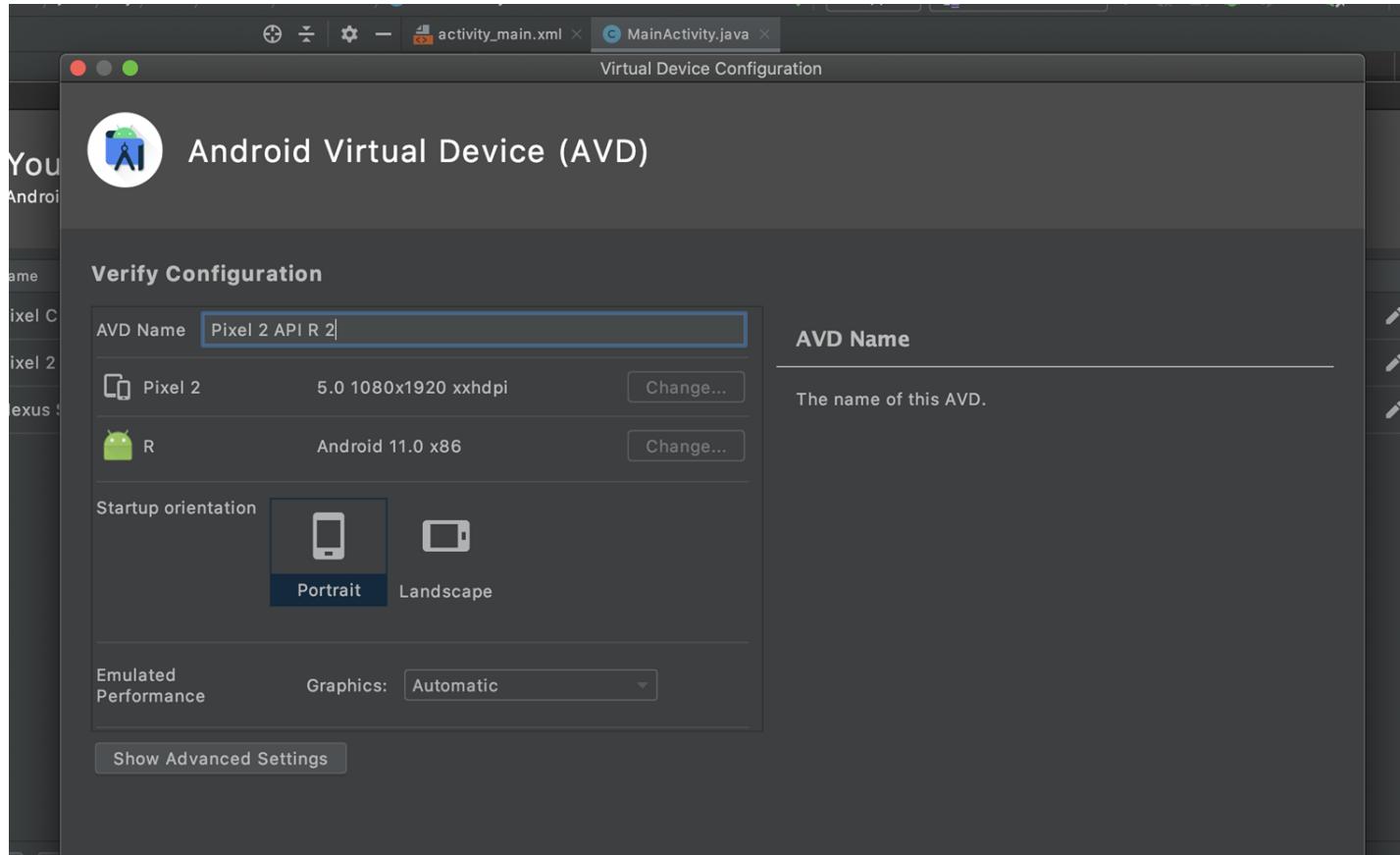
Select a device (eg, Pixel 2)



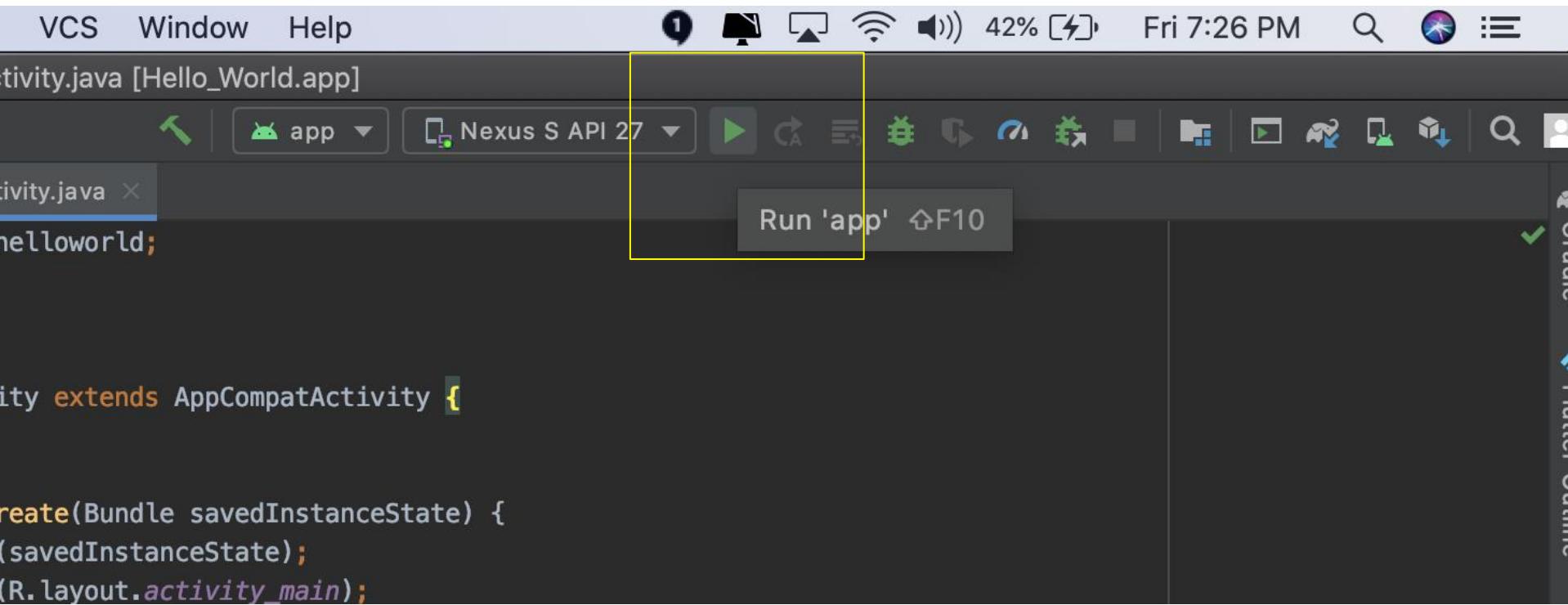
Press Download on R, After that select R and Press Next

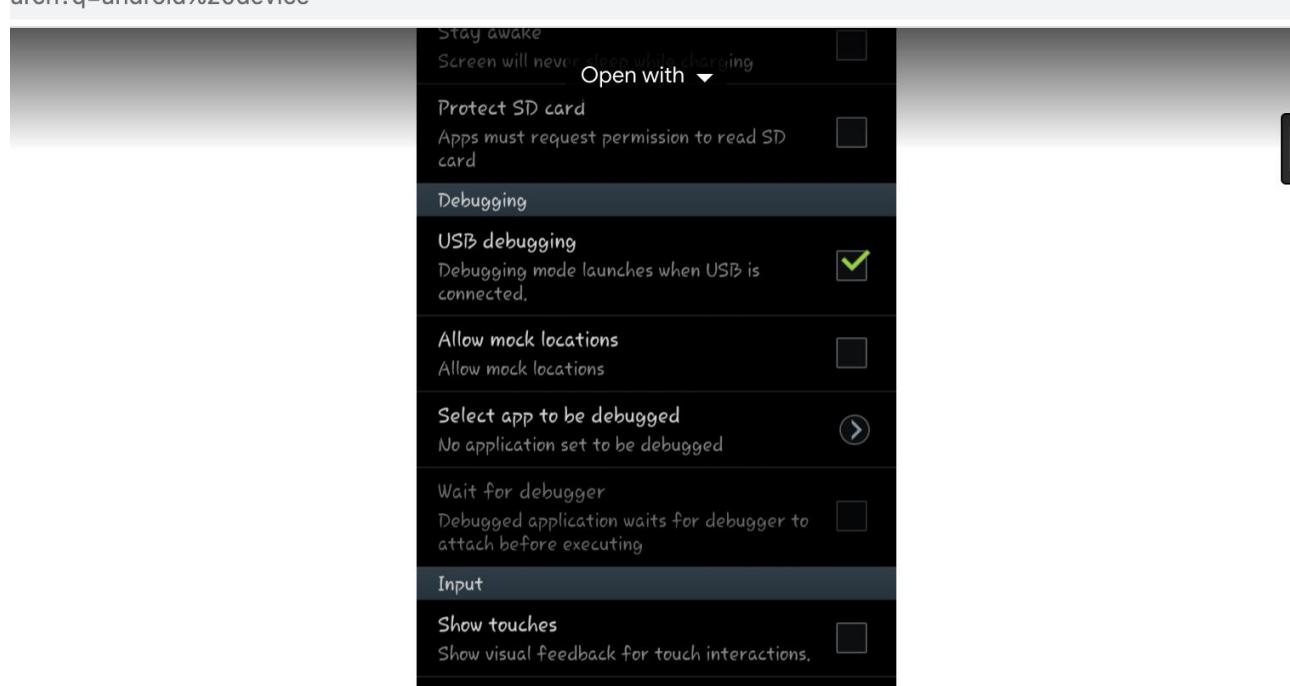


This page should appear, no need to change anything



Press Play (Run App)





You now need to enable debug mode in your Phone, this is done by Clicking “Settings”, “Developer Options”, under Debugging tick “USB Debugging”. To see the option for USB debugging mode in Nexus 4 or Android 4.2 or higher OS within System Settings scroll to the bottom and tap “About phone” or “About tablet” then scroll to the bottom and tap on “Build number” seven times. If you see a “No need, you are already a developer!” message pop up, then you have done it. By tapping on “Build number” seven times, you have unlocked USB debugging mode on Android 4.2 and higher. Now

you can enable/disable it whenever you desire by going to “Settings”, “Developer Options”, “Debugging”, ” USB Debugging”. Some mobile devices needs to have their specific drivers, if your mobile device was unsuccessful you might need to download and install its drives listed in this URL:

<http://developer.android.com/tools/extras/oem-usb.html#Drivers>

Android phone + Cable

- 1) Open phone go to **About Phone/ About tablet**
- 2) **Look for Build number -> Tap it 7 times ...**
- 3) After tap 7 times, a popup should come out “You are now a developer”
- 4) Look for **developer options** menu, look for **USB Debugging**, tick (enabled)
- 5) Once enabled, link the USB to the pc select your phone, Press play

Google - How to enable developer options in xxx xx (device name)

OEM	Driver URL	Open with ▾
Acer	http://www.acer.com/worldwide/support/mobile.html	
ALCATEL ONE TOUCH	http://www.alcatelonetouch.com/global-en/support/faq/usbdriver.html	
Asus	http://support.asus.com/download/	
Dell	http://support.dell.com/support/downloads/index.aspx? c=us&cs=19&l=en&s=dhs&~ck=anavml	
Foxconn	http://drivers.cmcs.com.tw/	
Fujitsu	http://www.fmworld.net/product/phone/sp/android/develop/	
Garmin-Asus	https://www.garminasus.com/en_US/support/pcsync/	
Hisense	http://app.hismarttv.com/dss/resourcecontent.do? method=viewResourceDetail&resourceId=16&type=5	
HTC	http://www.htc.com Click on the support tab to select your products/device. Different regions will have different links.	
Huawei	http://www.huaweidevice.com/worldwide/downloadCenter.do?method=index	
Intel	http://www.intel.com/software/android	
KT Tech	http://www.kttech.co.kr/cscenter/download05.asp for EV-S100 (Take)	
Kyocera	http://www.kyocera-wireless.com/support/phone_drivers.htm	
Lenovo	http://developer.lenovomm.com/developer/download.jsp	
LGE	http://www.lg.com/us/mobile-phones/mobile-support/mobile-lg-mobile-phone-support.jsp	
Motorola	http://developer.motorola.com/docstools/USB_Drivers/	
MTK	http://onlinemediatek.com/Public%20Documents/MTK_Android_USB_Driver.zip	
Oppo	http://www.oppo.com/index.php?q=software/view&sw_id=631	
Pantech	http://www.isky.co.kr/cs/software/software.sky?fromUrl=index	
Pegatron	http://www.pegatroncorp.com/download/New_Duke_PC_Driver_0705.zip (ZIP)	

Download and unzip Flutter

<https://flutter.dev/docs/get-started/install>

PowerShell.

🔗 Get the Flutter SDK

1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

[flutter_windows_1.22.6-stable.zip](#)

For other release channels, and older builds, see the [SDK releases](#) page.

2. Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `C:\src\flutter`).

⚠️ Warning: Do not install Flutter in a directory like `C:\Program Files\` that requires elevated privileges.

If you don't want to install a fixed version of the installation bundle, you can skip steps 1 and 2. Instead, get the source code from the [Flutter repo](#) on GitHub, and change branches or tags as needed. For example:

Contents

[System requirements](#)

[Get the Flutter SDK](#)

[Update your path](#)

[Run flutter doctor](#)

[Android setup](#)

[Install Android Studio](#)

[Set up your Android de](#)

[Set up the Android emi](#)

[Web setup](#)

[Next step](#)

Setting up Flutter

Upon restart new menu will appear Create a new Flutter project Go to Flutter.dev -> Docs -> Getting started

- 1) Select your OS and Download the installer file
- 2) **Unzip the installer folder to a proper folder**
- 3) Install Android Studio Flutter plugin
 - a) File -> Settings -> Plugins -> Flutter / (Android Studio -> Preference -> Plugin)
 - b) A pop up will appear for confirmation to install Dart as well. Select Yes
 - c) Restart IDE

If you don't have JDK install:

*https://drive.google.com/file/d/15VPFSI_ehq0JtULVveiD4DgId-sJXdDZ/view?usp=sharing (Install this) -> Run the file

- 1) Look for the flutter folder, open bin subfolder
- 2) Open this folder in CMD
- 3) Run the following command line:

`flutter doctor`

Inside Android Studio, go to File -> Settings

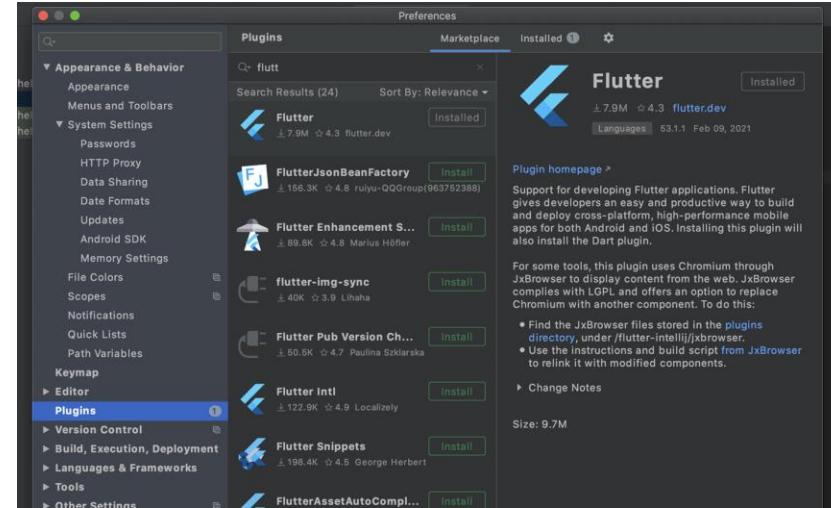
Look for Plugins, and search for Flutter

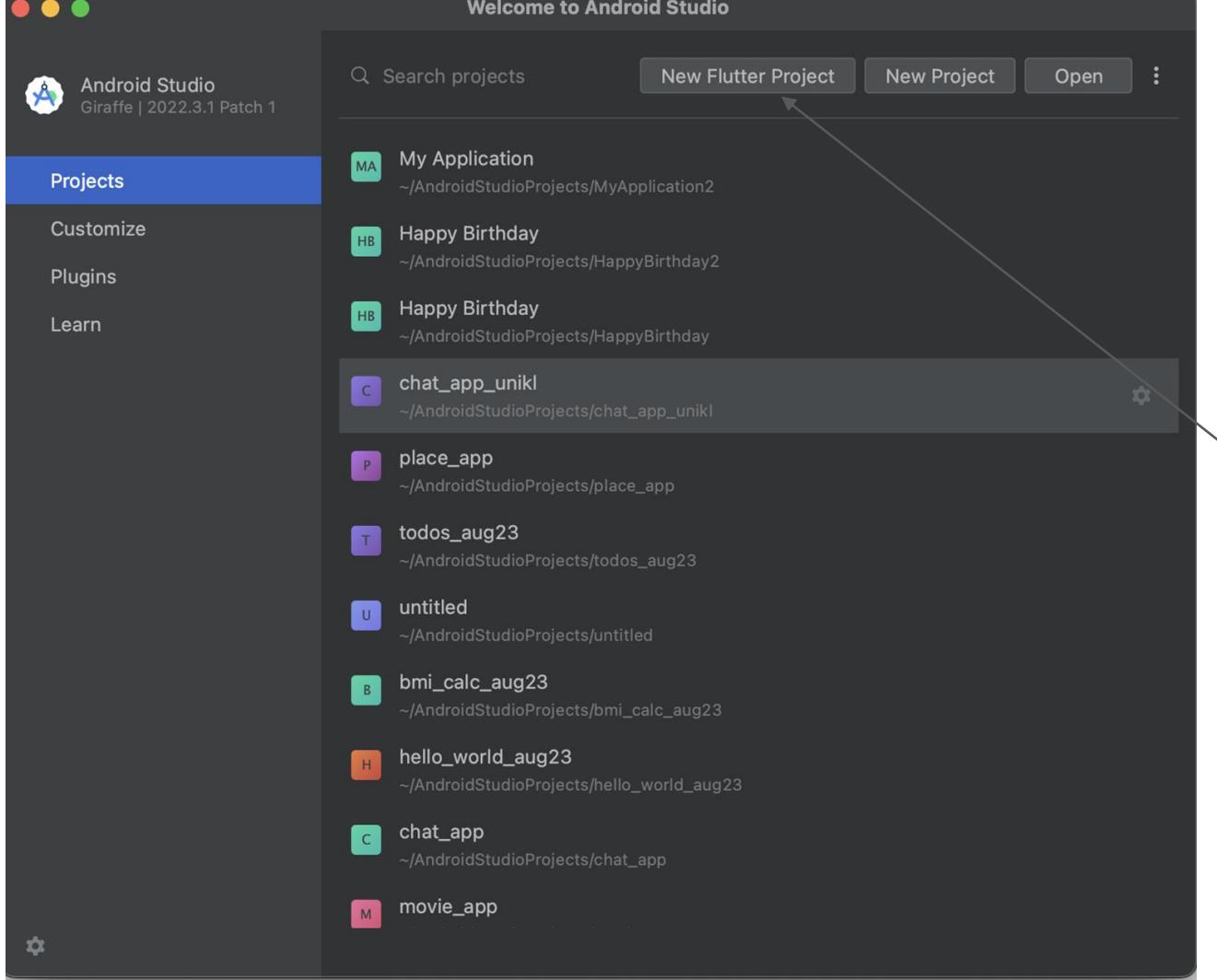
After you press install, there will be a popup “You need to install Dart as well” ->
Press OK

A popup to ask you to restart -> OK

If after restart, project is opened

press close project





New Project

Project name: hello_world

Project location: ~/AndroidStudioProjects/hello_world ...

Description: A new Flutter project.

Project type: Application ▾

Organization: com.example

Android language: Java Kotlin

iOS language: Objective-C Swift

Platforms: Android iOS Linux MacOS Web Windows

When created, the new project will run on the selected platforms (others can be added later).

Create project offline

► More Settings



Cancel

Previous

Create



Flutter SDK path: /Users/wanmuzaffarwanhashim/Downloads/flutter



New Project

Empty Project

Generators

Compose Multiplatform

IDE Plugin

Android

Flutter

Dart



Point to the place where you unzip the flutter project



Cancel

Next

The screenshot shows the Android Studio interface with a Flutter project named "flutter_app". The main window displays the code for `main.dart`. The code defines the application's entry point and its UI structure.

```
import 'package:flutter/material.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(
        // This is the theme of your application.
        //
        // Try running your application with "flutter run". You'll see the
        // application has a blue toolbar. Then, without quitting the app, try
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the application
        // is not restarted.
        primarySwatch: Colors.blue,
        // This makes the visual density adapt to the platform that you run
        // the app on. For desktop platforms, the controls will be smaller and
        // closer together (more dense) than on mobile platforms.
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      // ThemeData
      home: MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  MyHomePage({Key key, this.title}) : super(key: key);

  // This widget is the home page of your application. It is stateful,
  // that it has a State object (defined below) that contains fields th
  // how it looks.
}
```

Welcome to the future
Kite is now integrated
Learn how to use Kite

```
ter run". You'll see  
quitting the app,  
ow to Colors.green  
→ press the "hot  
s "r" if you used
```

```
the application  
state, use hot
```

```
le changes can be
```

Running Devices: Pixel 2 API UpsideDownCake



Flutter Inspector
Flutter Performance
Flutter Outline
Running Devices
Device Explorer

3 AM

21 PM

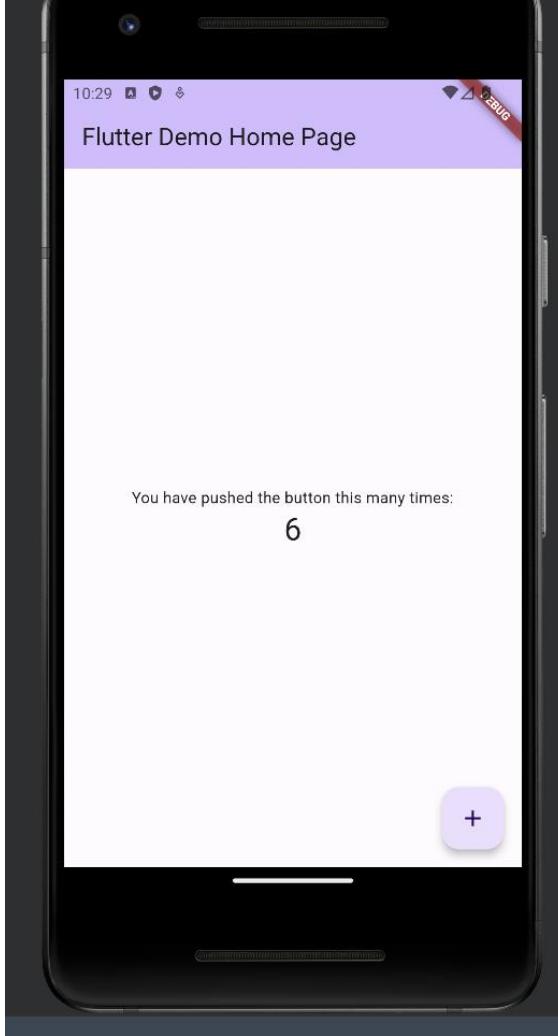
6 AM

3 PM

7 PM

3 PM

9 PM



Press Play this app should come out

```
void main() {  
  runApp(const MyApp());  
}
```

Starter code of a Flutter Project

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ), // ThemeData
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    ); // MaterialApp
  }
}
```

① 1

Remove the code that is related
to the counter app

..
-38-126

-16-30

The coded related to Starter code

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

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.

  @override
  Widget build(BuildContext context) {
```

In line 24, type `stless` and press tab

Type HomePage directly

Press esc

```
class HomePage extends StatelessWidget {  
    const HomePage({Key? key}) : super(key: key);  
  
    @override  
    Widget build(BuildContext context) {  
        return const Placeholder();  
    }  
}
```

Update the starter code with our HomePage() - Our class name

Save and check on emulator

```
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.blue,  
  
                visualDensity: VisualDensity.adaptivePlatformDensity,  
            ), // ThemeData  
            home: HomePage(),  
        ); // MaterialApp  
    }  
}
```

Run and see the app - Placeholder



Scaffold vs HTML structure tag

```
<html>
<head>
    <title>Hello World</title>
</head>
<body>
    <h1>Hello World</h1>
</body>
</html>
```

The diagram illustrates the mapping of an HTML structure to Scaffold code. It shows an HTML file with various tags and annotations. Arrows point from specific parts of the HTML to corresponding parts of the Scaffold code.

- An arrow points from the `<title>Hello World</title>` block in the HTML to the `appBar:AppBar(title:Text("Hello World"))` line in the Scaffold code.
- An arrow points from the `<body>` tag in the HTML to the `body:Text("Hello World")` line in the Scaffold code.
- An arrow points from the closing brace of the `<h1>Hello World</h1>` block in the HTML to the closing brace of the `Scaffold()` function in the Scaffold code.

The code for scaffold

```
}

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Hello Flutter"),),
      body:Text("This is my first app!")
    ); // Scaffold
  }
}
```

```
Widget build(BuildContext context) {  
  
  return Scaffold(  
  
    backgroundColor: Colors.yellow,  
  
    appBar: AppBar(title: Text("Hello World"),  
backgroundColor: Colors.red,) ,  
  
    body: Center(child: Text("Hello World")) ,  
  
  );  
  
}
```

11:42

DEBUG

Hello World

Hello World

Changing the size and color of the text

```
Widget build(BuildContext context) {  
  
  // The structure of a mobile app (Scaffold)  
  
  return Scaffold(  
  
    appBar: AppBar(title: Text("Hello World"), backgroundColor: Colors.red),  
  
    body: Text("Hello World", style: TextStyle(fontSize: 32  
      , color: Colors.blue))  
  );  
}
```

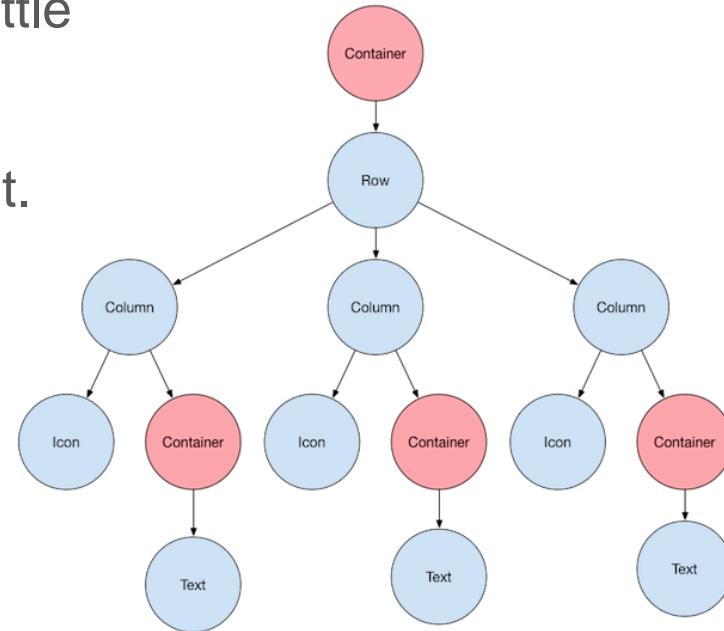
Everything is a widget

You build widget upon widget.

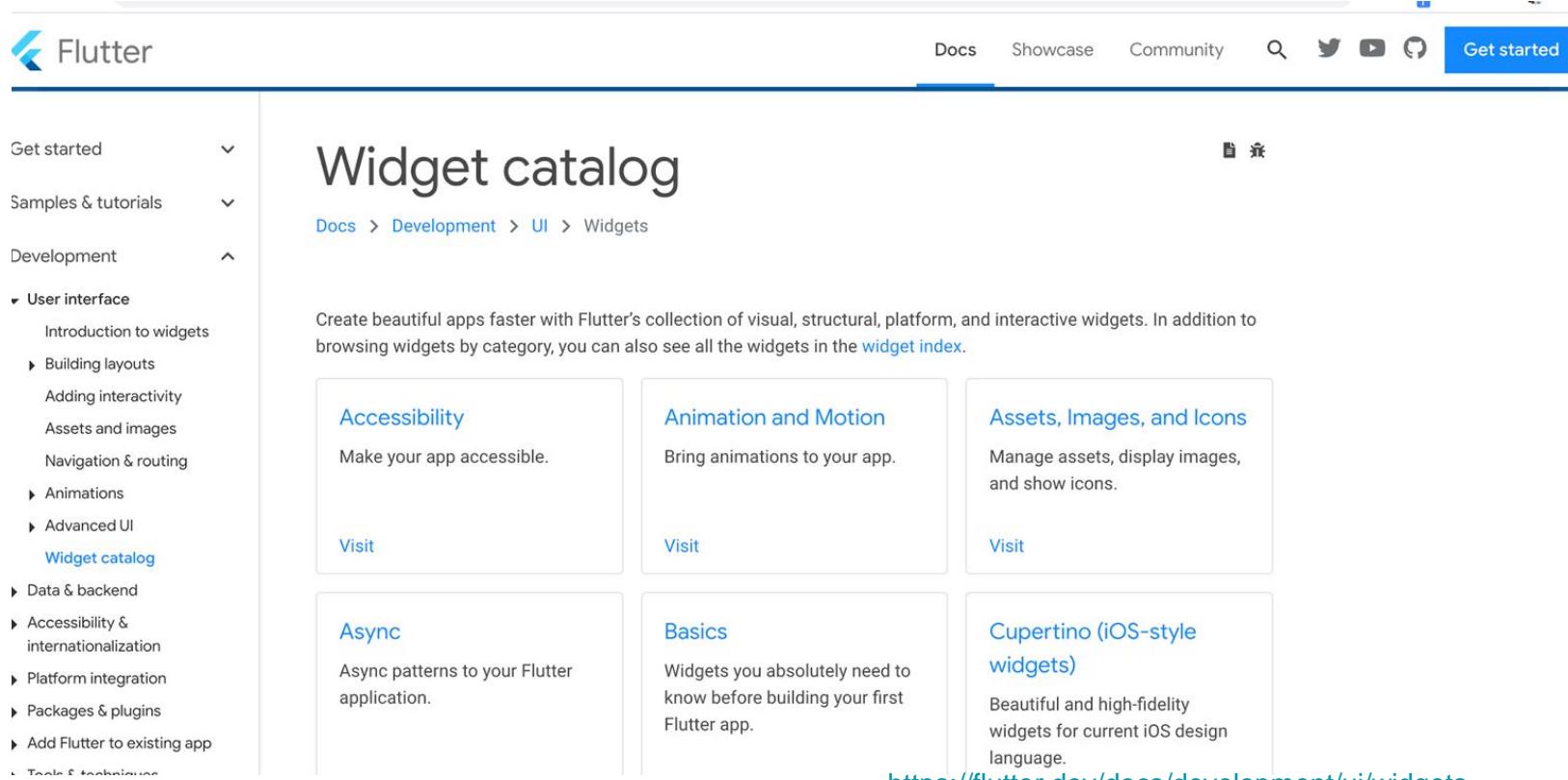
Your screen, a section in a screen, a tiny little section is also a Widget.

You create and customize your own widget.

- UI Element : Text, TextButton, Image
- Style: TextStyle, Color, SizedBox
- Layout and Position widget : Center, Row, Column



Widget catalog



The screenshot shows the Flutter documentation website with the URL <https://docs.flutter.dev/ui/widgets>. The page title is "Widget catalog". The navigation bar includes links for "Docs", "Showcase", "Community", a search icon, and social media links for Twitter, YouTube, and GitHub. A "Get started" button is also present. On the left, there's a sidebar with categories like "Get started", "Samples & tutorials", "Development", and "User interface" (which is expanded to show "Introduction to widgets", "Building layouts", "Adding interactivity", "Assets and images", "Navigation & routing", "Animations", "Advanced UI", and "Widget catalog"). Other collapsed sections include "Data & backend", "Accessibility & internationalization", "Platform integration", "Packages & plugins", and "Add Flutter to existing app". The main content area features a large heading "Widget catalog" and a breadcrumb trail: "Docs > Development > UI > Widgets". Below this, a text block says: "Create beautiful apps faster with Flutter's collection of visual, structural, platform, and interactive widgets. In addition to browsing widgets by category, you can also see all the widgets in the [widget index](#)". There are six cards with titles and descriptions: "Accessibility" (Make your app accessible), "Animation and Motion" (Bring animations to your app), "Assets, Images, and Icons" (Manage assets, display images, and show icons), "Async" (Async patterns to your Flutter application), "Basics" (Widgets you absolutely need to know before building your first Flutter app), and "Cupertino (iOS-style widgets)" (Beautiful and high-fidelity widgets for current iOS design language). Each card has a "Visit" button.

Flutter

Docs Showcase Community

Get started

Get started

Samples & tutorials

Development

User interface

- Introduction to widgets
- Building layouts
- Adding interactivity
- Assets and images
- Navigation & routing
- Animations
- Advanced UI
- Widget catalog**

Data & backend

Accessibility & internationalization

Platform integration

Packages & plugins

Add Flutter to existing app

Tools & techniques

Widget catalog

Docs > Development > UI > Widgets

Create beautiful apps faster with Flutter's collection of visual, structural, platform, and interactive widgets. In addition to browsing widgets by category, you can also see all the widgets in the [widget index](#).

Accessibility
Make your app accessible.
[Visit](#)

Animation and Motion
Bring animations to your app.
[Visit](#)

Assets, Images, and Icons
Manage assets, display images, and show icons.
[Visit](#)

Async
Async patterns to your Flutter application.

Basics
Widgets you absolutely need to know before building your first Flutter app.

Cupertino (iOS-style widgets)
Beautiful and high-fidelity widgets for current iOS design language.

<https://flutter.dev/docs/development/ui/widgets>

Scaffold

A scaffold is a ***basic structure of an application*** having the following property by default:

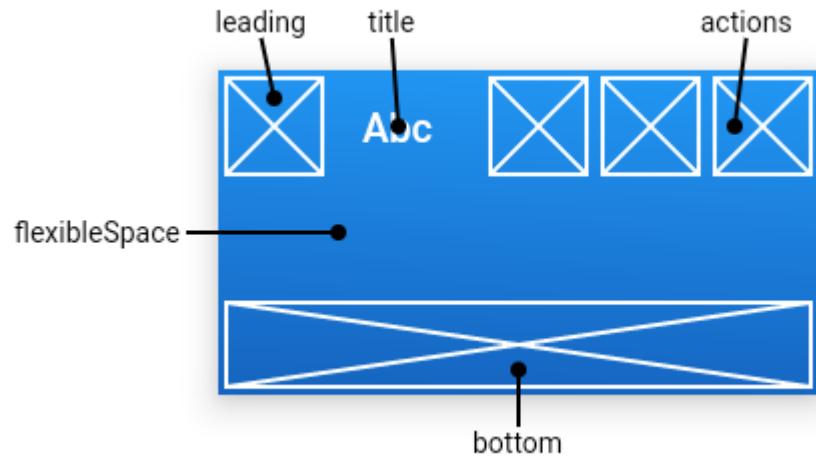
- appBar
- body
- floatingActionButton
- bottomNavigationBar
- drawer

Appbar

An app bar consists of a toolbar and potentially other widgets,

For example, if you would like to add a button on the left side you use leading and actions on the right side.

You may change the property `backgroundColor` to change the background color of the AppBar.



Changing appBar Background color

```
        }
    }
}

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Hello Flutter"), backgroundColor: Colors.yellow,),
      body:Text("This is my first app!")
    ); // Scaffold
}
}
```

Center a widget

Alt + enter on the word text

Select wrap with Center

The screenshot shows the Android Studio code editor with the following code:

```
class HomePage extends StatelessWidget {  
    @override  
    Widget build(BuildContext context) {  
        return Scaffold(  
            appBar: AppBar(title: Text("Hello Flutter"), back...  
            body: Text("This is my first app!"));  
    }  
}
```

A context menu is open over the word "text" in the "body" field of the Scaffold. The menu items are:

- Wrap with widget...
- Wrap with Center
- Wrap with Column
- Wrap with Container
- Wrap with Padding
- Wrap with Row
- Wrap with SizedBox
- Wrap with StreamBuilder

At the bottom of the menu, it says "Press ⌘I to open preview".

Code for center

```
return Scaffold(  
    appBar: AppBar(title: Text("Hello World"), backgroundColor: Colors.red,) ,  
    body:Center(child: Text("Hello World", style: TextStyle(color: Colors.blue),))  
) ;
```

```
Widget build(BuildContext context) {
```

```
    return Scaffold(
```

```
        backgroundColor: Colors.yellow,
```

```
        appBar: AppBar(
```

```
            title: Text("Hello World"),
```

```
            backgroundColor: Colors.red,
```

```
) ,
```

```
        body: Center(
```

```
            child: Column(
```

```
                children: [
```

Different between Child and Children

child - Center, Container (Box), Padding, SingleChildScrollView -> A widget

children - Row and Column , Stack-> [] -> Array (can have more than one widget)

Scaffold - body

Column , expect children, have more than one items

```
return Scaffold(  
    appBar: AppBar(title: Text("Hello Flutter"), backgroundColor: Colors.red,),  
    body: Column(  
        children: [  
            Text("Hello World"),  
            Text("This is my first app")  
        ],  
    ) // Column  
); // Scaffold  
}
```

Code for Column

```
Scaffold(  
    appBar: AppBar(title: Text("Hello World"), backgroundColor: Colors.red,) ,  
    body: Center(child: Column(  
        children: [  
            Text("Hello World", style: TextStyle(color: Colors.blue, fontSize: 32  
        ),),  
            Text("This is my app")  
        ],  
    )),  
);
```

Add another text: I hope you enjoy it (under This is my app)
Change the color of each text
Change the fontsize

```
return Scaffold(  
    appBar: AppBar(title: Text("Hello World"), backgroundColor: Colors.red,) ,  
    body: Center(child: Column(  
        children: [  
            Text("Hello World", style: TextStyle(color: Colors.blue, fontSize: 32),) ,  
            Text("This is my app", style: TextStyle(color:Colors.red, fontSize: 24),) ,  
            Text("I hope you enjoy the app", style: TextStyle(color:Colors.green, fontSize: 16),)  
        ] ,  
    ))  
) ;
```

Adding an Image

```
    SizedBox(height:10),
    Text("I hope you like it!"),
    SizedBox(height:10),
    Row(
      children: [
        Text("First column"),
        SizedBox(width:10),
        Text("Second column")
      ],
    ), // Row
    Image.network("http://static.asiawebdirect.com/m/kl/portals/kuala-lumpur-ws/homepage/pageProperties0gImage/kuala-lumpur.jpg.jpg"),
  ],
) // Column
); // Scaffold
}
```

Code to add image replace the URL with your image

```
Image.network("https://a.cdn-  
hotels.com/gdcs/production99/d1417/4da30a77-ecb2-  
4ead-a771-  
e61fac2a73dd.jpg?impolicy=fcrop&w=800&h=533&q=med  
ium")
```

Column + Row

```
class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Hello Flutter"), backgroundColor: Colors.red),
      body: Column(
        children: [
          Text("Hello World"),
          Text("This is my first app"),
          Text("I hope you like it!"),
          Row(
            children: [
              Text("First column"),
              Text("Second column")
            ],
          ) // Row
        ],
      ) // Column
    ); // Scaffold
}
```

Center, Row, Column,

```
Scaffold(  
    appBar: AppBar(title: Text("Hello World"), backgroundColor: Colors.red,) ,  
    body: Center(child: Column(  
        children: [  
            Text("Hello World", style: TextStyle(color: Colors.blue, fontSize: 32),) ,  
            Text("This is my app", style: TextStyle(color:Colors.red, fontSize: 24),) ,  
            Text("I hope you enjoy the app", style: TextStyle(color:Colors.green, fontSize: 16),) ,  
            Row(  
                mainAxisSize: MainAxisSize.center,  
                children:  
                    [Text("Row")]  
            )  
        ]  
    ))
```

SizedBox

SizedBox is margin
(distance between an
element with another
element) / spacing

So you specify either
height or width in the
property

```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(title: Text("Hello Flutter"), backgroundColor: Colors.red,),  
    body: Column(  
      children: [  
        Text("Hello World"),  
        SizedBox(height:10),  
        Text("This is my first app"),  
        SizedBox(height:10),  
        Text("I hope you like it!"),  
        SizedBox(height:10),  
        Row(  
          children: [  
            Text("First column"),  
            SizedBox(width:10),  
            Text("Second column")  
          ],  
        ) // Row  
      ],  
    ) // Column  
  ); // Scaffold  
}
```

SizedBox Source code

```
Center(child: Column(  
  
    children: [  
  
        Text("Hello World", style: TextStyle(color: Colors.blue, fontSize: 32),),  
  
        SizedBox(height:8),  
  
        Text("This is my app", style: TextStyle(color:Colors.red, fontSize: 24),),  
  
        SizedBox(height:8),  
  
        Text("I hope you enjoy the app", style: TextStyle(color:Colors.green, fontSize: 16),),  
  
        SizedBox(height:8),  
  
        Row(  
  
            mainAxisAlignment: MainAxisAlignment.center,  
  
            children: [  
  
            ]  
        ),  
    ]  
)  
)
```

Adding a TextButton

```
    Text("I hope you like it!"),
    Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
            Text("This is Left"),
            Text("This is right")
        ],
    ), // Row
    TextButton(onPressed: (){
        print("Hello World");
    }, child: Text("Press me!")) // TextButton
),
), // Column
) // Center
); // Scaffold
}
}
```

```
SizedBox(height: 8,) ,  
  
Image.network("https://a.cdn-hotels.com/gdcs/production99/d1417/4da30a77-ecb2-4ead-a771-  
e61fac2a73dd.jpg?impolicy=fcrop&w=800&h=533&q=medium") ,  
  
SizedBox(height: 8,) ,  
  
TextButton(onPressed: () {  
    print("Hello world");  
}, child: Text("Press me"))  
],
```

```
32     body: Center(child: Column(
33       children: [
34         Text("Hello World", style: TextStyle(
35           color: Colors.blue,
36           fontSize: 32
37         )),
38         Text("Hello World", style: TextStyle(
39           color: Colors.green,
40           fontSize: 24
41         )),
42         Text("Hello World", style: TextStyle(
43           color: Colors.red,
44           fontSize: 18
45         )),
46       ],
47     )),
48   ),
49   ),
50   ),
51   ),
52   ),
53   ),
54   ),
55   ),
56   ),
57   ),
58   ),
59   ),
60   ),
61   ),
62   ),
63   ),
64   ),
65   ),
66   ),
67   ),
68   ),
69   ),
70   ),
71   ),
72   ),
73   ),
74   ),
75   ),
76   ),
77   ),
78   ),
79   ),
80   ),
81   ),
82   ),
83   ),
84   ),
85   ),
86   ),
87   ),
88   ),
89   ),
90   ),
91   ),
92   ),
93   ),
94   ),
95   ),
96   ),
97   ),
98   ),
99   ),
100  ),
101  ),
102  ),
103  ),
104  ),
105  ),
106  ),
107  ),
108  ),
109  ),
110  ),
111  ),
112  ),
113  ),
114  ),
115  ),
116  ),
117  ),
118  ),
119  ),
120  ),
121  ),
122  ),
123  ),
124  ),
125  ),
126  ),
127  ),
128  ),
129  ),
130  ),
131  ),
132  ),
133  ),
134  ),
135  ),
136  ),
137  ),
138  ),
139  ),
140  ),
141  ),
142  ),
143  ),
144  ),
145  ),
146  ),
147  ),
148  ),
149  ),
150  ),
151  ),
152  ),
153  ),
154  ),
155  ),
156  ),
157  ),
158  ),
159  ),
160  ),
161  ),
162  ),
163  ),
164  ),
165  ),
166  ),
167  ),
168  ),
169  ),
170  ),
171  ),
172  ),
173  ),
174  ),
175  ),
176  ),
177  ),
178  ),
179  ),
180  ),
181  ),
182  ),
183  ),
184  ),
185  ),
186  ),
187  ),
188  ),
189  ),
190  ),
191  ),
192  ),
193  ),
194  ),
195  ),
196  ),
197  ),
198  ),
199  ),
200  ),
201  ),
202  ),
203  ),
204  ),
205  ),
206  ),
207  ),
208  ),
209  ),
210  ),
211  ),
212  ),
213  ),
214  ),
215  ),
216  ),
217  ),
218  ),
219  ),
220  ),
221  ),
222  ),
223  ),
224  ),
225  ),
226  ),
227  ),
228  ),
229  ),
230  ),
231  ),
232  ),
233  ),
234  ),
235  ),
236  ),
237  ),
238  ),
239  ),
240  ),
241  ),
242  ),
243  ),
244  ),
245  ),
246  ),
247  ),
248  ),
249  ),
250  ),
251  ),
252  ),
253  ),
254  ),
255  ),
256  ),
257  ),
258  ),
259  ),
259  ),
260  ),
261  ),
262  ),
263  ),
264  ),
265  ),
266  ),
267  ),
268  ),
269  ),
270  ),
271  ),
272  ),
273  ),
274  ),
275  ),
276  ),
277  ),
278  ),
279  ),
279  ),
280  ),
281  ),
282  ),
283  ),
284  ),
285  ),
286  ),
287  ),
288  ),
289  ),
289  ),
290  ),
291  ),
292  ),
293  ),
294  ),
295  ),
296  ),
297  ),
298  ),
299  ),
299  ),
300  ),
301  ),
302  ),
303  ),
304  ),
305  ),
306  ),
307  ),
308  ),
309  ),
309  ),
310  ),
311  ),
312  ),
313  ),
314  ),
315  ),
316  ),
317  ),
317  ),
318  ),
319  ),
320  ),
321  ),
322  ),
323  ),
324  ),
325  ),
326  ),
327  ),
328  ),
329  ),
329  ),
330  ),
331  ),
332  ),
333  ),
334  ),
335  ),
336  ),
337  ),
338  ),
339  ),
339  ),
340  ),
341  ),
342  ),
343  ),
344  ),
345  ),
346  ),
347  ),
348  ),
349  ),
349  ),
350  ),
351  ),
352  ),
353  ),
354  ),
355  ),
356  ),
357  ),
358  ),
359  ),
359  ),
360  ),
361  ),
362  ),
363  ),
364  ),
365  ),
366  ),
367  ),
368  ),
369  ),
369  ),
370  ),
371  ),
372  ),
373  ),
374  ),
375  ),
376  ),
377  ),
378  ),
379  ),
379  ),
380  ),
381  ),
382  ),
383  ),
384  ),
385  ),
386  ),
387  ),
388  ),
389  ),
389  ),
390  ),
391  ),
392  ),
393  ),
394  ),
395  ),
396  ),
397  ),
398  ),
399  ),
399  ),
400  ),
401  ),
402  ),
403  ),
404  ),
405  ),
406  ),
407  ),
408  ),
409  ),
409  ),
410  ),
411  ),
412  ),
413  ),
414  ),
415  ),
416  ),
417  ),
418  ),
419  ),
419  ),
420  ),
421  ),
422  ),
423  ),
424  ),
425  ),
426  ),
427  ),
428  ),
429  ),
429  ),
430  ),
431  ),
432  ),
433  ),
434  ),
435  ),
436  ),
437  ),
438  ),
439  ),
439  ),
440  ),
441  ),
442  ),
443  ),
444  ),
445  ),
446  ),
447  ),
448  ),
449  ),
449  ),
450  ),
451  ),
452  ),
453  ),
454  ),
455  ),
456  ),
457  ),
458  ),
459  ),
459  ),
460  ),
461  ),
462  ),
463  ),
464  ),
465  ),
466  ),
467  ),
468  ),
469  ),
469  ),
470  ),
471  ),
472  ),
473  ),
474  ),
475  ),
476  ),
477  ),
478  ),
479  ),
479  ),
480  ),
481  ),
482  ),
483  ),
484  ),
485  ),
486  ),
487  ),
488  ),
489  ),
489  ),
490  ),
491  ),
492  ),
493  ),
494  ),
495  ),
496  ),
497  ),
498  ),
499  ),
499  ),
500  ),
501  ),
502  ),
503  ),
504  ),
505  ),
506  ),
507  ),
508  ),
509  ),
509  ),
510  ),
511  ),
512  ),
513  ),
514  ),
515  ),
516  ),
517  ),
518  ),
519  ),
519  ),
520  ),
521  ),
522  ),
523  ),
524  ),
525  ),
526  ),
527  ),
528  ),
529  ),
529  ),
530  ),
531  ),
532  ),
533  ),
534  ),
535  ),
536  ),
537  ),
538  ),
539  ),
539  ),
540  ),
541  ),
542  ),
543  ),
544  ),
545  ),
546  ),
547  ),
548  ),
549  ),
549  ),
550  ),
551  ),
552  ),
553  ),
554  ),
555  ),
556  ),
557  ),
558  ),
559  ),
559  ),
560  ),
561  ),
562  ),
563  ),
564  ),
565  ),
566  ),
567  ),
568  ),
569  ),
569  ),
570  ),
571  ),
572  ),
573  ),
574  ),
575  ),
576  ),
577  ),
578  ),
579  ),
579  ),
580  ),
581  ),
582  ),
583  ),
584  ),
585  ),
586  ),
587  ),
588  ),
589  ),
589  ),
590  ),
591  ),
592  ),
593  ),
594  ),
595  ),
596  ),
597  ),
598  ),
599  ),
599  ),
600  ),
601  ),
602  ),
603  ),
604  ),
605  ),
606  ),
607  ),
608  ),
609  ),
609  ),
610  ),
611  ),
612  ),
613  ),
614  ),
615  ),
616  ),
617  ),
618  ),
619  ),
619  ),
620  ),
621  ),
622  ),
623  ),
624  ),
625  ),
626  ),
627  ),
628  ),
629  ),
629  ),
630  ),
631  ),
632  ),
633  ),
634  ),
635  ),
636  ),
637  ),
638  ),
639  ),
639  ),
640  ),
641  ),
642  ),
643  ),
644  ),
645  ),
646  ),
647  ),
648  ),
649  ),
649  ),
650  ),
651  ),
652  ),
653  ),
654  ),
655  ),
656  ),
657  ),
658  ),
659  ),
659  ),
660  ),
661  ),
662  ),
663  ),
664  ),
665  ),
666  ),
667  ),
668  ),
669  ),
669  ),
670  ),
671  ),
672  ),
673  ),
674  ),
675  ),
676  ),
677  ),
678  ),
679  ),
679  ),
680  ),
681  ),
682  ),
683  ),
684  ),
685  ),
686  ),
687  ),
688  ),
689  ),
689  ),
690  ),
691  ),
692  ),
693  ),
694  ),
695  ),
696  ),
697  ),
698  ),
699  ),
699  ),
700  ),
701  ),
702  ),
703  ),
704  ),
705  ),
706  ),
707  ),
708  ),
709  ),
709  ),
710  ),
711  ),
712  ),
713  ),
714  ),
715  ),
716  ),
717  ),
718  ),
719  ),
719  ),
720  ),
721  ),
722  ),
723  ),
724  ),
725  ),
726  ),
727  ),
728  ),
729  ),
729  ),
730  ),
731  ),
732  ),
733  ),
734  ),
735  ),
736  ),
737  ),
738  ),
739  ),
739  ),
740  ),
741  ),
742  ),
743  ),
744  ),
745  ),
746  ),
747  ),
748  ),
749  ),
749  ),
750  ),
751  ),
752  ),
753  ),
754  ),
755  ),
756  ),
757  ),
758  ),
759  ),
759  ),
760  ),
761  ),
762  ),
763  ),
764  ),
765  ),
766  ),
767  ),
768  ),
769  ),
769  ),
770  ),
771  ),
772  ),
773  ),
774  ),
775  ),
776  ),
777  ),
778  ),
779  ),
779  ),
780  ),
781  ),
782  ),
783  ),
784  ),
785  ),
786  ),
787  ),
788  ),
789  ),
789  ),
790  ),
791  ),
792  ),
793  ),
794  ),
795  ),
796  ),
797  ),
798  ),
799  ),
799  ),
800  ),
801  ),
802  ),
803  ),
804  ),
805  ),
806  ),
807  ),
808  ),
809  ),
809  ),
810  ),
811  ),
812  ),
813  ),
814  ),
815  ),
816  ),
817  ),
818  ),
819  ),
819  ),
820  ),
821  ),
822  ),
823  ),
824  ),
825  ),
826  ),
827  ),
828  ),
829  ),
829  ),
830  ),
831  ),
832  ),
833  ),
834  ),
835  ),
836  ),
837  ),
838  ),
839  ),
839  ),
840  ),
841  ),
842  ),
843  ),
844  ),
845  ),
846  ),
847  ),
848  ),
849  ),
849  ),
850  ),
851  ),
852  ),
853  ),
854  ),
855  ),
856  ),
857  ),
858  ),
859  ),
859  ),
860  ),
861  ),
862  ),
863  ),
864  ),
865  ),
866  ),
867  ),
868  ),
869  ),
869  ),
870  ),
871  ),
872  ),
873  ),
874  ),
875  ),
876  ),
877  ),
878  ),
878  ),
879  ),
880  ),
881  ),
882  ),
883  ),
884  ),
885  ),
886  ),
887  ),
888  ),
889  ),
889  ),
890  ),
891  ),
892  ),
893  ),
894  ),
895  ),
896  ),
897  ),
898  ),
898  ),
899  ),
900  ),
901  ),
902  ),
903  ),
904  ),
905  ),
906  ),
907  ),
908  ),
909  ),
909  ),
910  ),
911  ),
912  ),
913  ),
914  ),
915  ),
916  ),
917  ),
918  ),
919  ),
919  ),
920  ),
921  ),
922  ),
923  ),
924  ),
925  ),
926  ),
927  ),
928  ),
929  ),
929  ),
930  ),
931  ),
932  ),
933  ),
934  ),
935  ),
936  ),
937  ),
938  ),
939  ),
939  ),
940  ),
941  ),
942  ),
943  ),
944  ),
945  ),
946  ),
947  ),
948  ),
949  ),
949  ),
950  ),
951  ),
952  ),
953  ),
954  ),
955  ),
956  ),
957  ),
958  ),
959  ),
959  ),
960  ),
961  ),
962  ),
963  ),
964  ),
965  ),
966  ),
967  ),
968  ),
969  ),
969  ),
970  ),
971  ),
972  ),
973  ),
974  ),
975  ),
976  ),
977  ),
978  ),
978  ),
979  ),
980  ),
981  ),
982  ),
983  ),
984  ),
985  ),
986  ),
987  ),
988  ),
988  ),
989  ),
990  ),
991  ),
992  ),
993  ),
994  ),
995  ),
996  ),
997  ),
998  ),
999  ),
999  ),
1000  ),
```

8a77-ecb2-4ead-a771-e61fac2a73dd.jpg?impolicy=fcrop&w=800&h=533&q=medium"),



Run: main.dart (Pixel 2 API UpsideDownCake) <

Console

```
I/flutter (22143): Hello World
I/flutter (22143): Hello World
D/EGL_emulation(22143): app_time_stats: avg=5.65ms min=1.79ms max=23.96ms count=41
I/flutter (22143): Hello World
I/flutter (22143): Hello World
I/flutter (22143): Hello World
I/flutter (22143): Hello World
```

Version Control Run TODO Problems Services Profiler App Quality Insights App Inspection Locat Dart Analysis

Full code

```
    Text("I hope you like it!"),
    Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
            Text("This is Left"),
            Text("This is right")
        ],
    ), // Row
    TextButton(onPressed: (){
        print("Hello World");
    }, child: Text("Press me!")) // TextButton
),
), // Column
) // Center
); // Scaffold
}
}
```

Full code as of now

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

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.

  @override
```

Getting a value from TextField

- 1) Create the TextField UI with InputDecoration
- 2) Create TextEditingController, 1 textfield = 1 TextEditingController
- 3) Link each of the textfield to the controller, through **controller** property on the TextField
- 4) Retrieve the value of the textfield using .text property from the controller.

Adding TextField (step 1)

```
        ), // Row  
  
        TextField(decoration: InputDecoration(hintText: "Enter your name"),  
        TextButton(onPressed: (){  
            Fluttertoast.showToast(  
                msg: "Hello World welcome to my app",  
                toastLength: Toast.LENGTH_SHORT,  
                gravity: ToastGravity.CENTER,  
                timeInSecForIosWeb: 1,
```

```
TextField(decoration: InputDecoration(hintText: "Enter your  
name"),),
```

```
TextButton(onPressed: () {
```

Add a controller and link it to the TextField (2)

Retrieve the value from controller

```
    TextField(decoration: InputDecoration(hintText: "Enter your name"), controller: nameController),  
    TextButton(onPressed: (){  
        Fluttertoast.showToast(  
            msg: "Hello welcome to my app, ${nameController.text}",  
            toastLength: Toast.LENGTH_SHORT,  
            gravity: ToastGravity.CENTER,  
            timeInSecForIosWeb: 1,  
            backgroundColor: Colors.red,  
            textColor: Colors.white,  
            fontSize: 16.0  
    );  
}, child: Text("Press me")) // TextButton
```

Adding a Toast from a package

- 1) Go to pub.dev and looks for flutter toast package. Select FlutterToast :
<https://pub.dev/packages/fluttertoast>
- 2) Go to pubspec.yaml (the configuration file of the project), look for cupertino under dependencies and paste the line from documentation under it:

```
fluttertoast: ^8.2.2
```

Press pub get appearing on top of screen to download the library

- 1) From the documentation, get the import and example, and paste it to replace print("Hello World")

<https://pub.dev/packages/fluttertoast>

Fluttertoast code

```
TextButton(onPressed: () {
```

```
    Fluttertoast.showToast(
```

```
        msg: "Hello World",
```

```
        toastLength: Toast.LENGTH_SHORT,
```

```
        gravity: ToastGravity.CENTER,
```

```
        timeInSecForIosWeb: 1,
```

```
        backgroundColor: Colors.red,
```

```
        textColor: Colors.white,
```

```
        fontSize: 16.0
```

```
    );
```

```
}, child: Text("Press me")))
```

Adding a Google Font from a package

- 1) Go to pub.dev and looks for Google Font package. Select the first Google Font link : https://pub.dev/packages/google_fonts
- 2) Go to pubspec.yaml (the configuration file of the project), look for cupertino under dependencies and paste the line from documentation under it:

```
google_fonts: ^6.1.0
```

Press pub get appearing on top of screen to download the library

- 1) From the documentation, get the import and example.

Google Fonts code example

```
body:Center(child: Column(  
  
    children: [  
  
        Text("Hello World", style: GoogleFonts.oswald(textStyle: TextStyle(color: Colors.blue, fontSize: 32))),  
  
        SizedBox(height:8),  
  
        Image.network("https://res.klook.com/image/upload/f1_lossy.progressive,w_800,c_fill,q_85/destination/nm3w9yv9  
tvofc4oncbsd.jpg"),  
  
        SizedBox(height: 8,),  
  
        Text("This is my app", style: GoogleFonts.maShanZheng(textStyle: TextStyle(color:Colors.red, fontSize:  
24))),  
  
        SizedBox(height:8),
```

Before Widget build:

```
var nameEditingController = TextEditingController();
```

Inside controller:

```
var nameEditingController = TextEditingController();
```

On the toast:

```
msg: "Hello ${nameEditingController.text}",
```

Adding SingleChildScrollView

```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(title: Text("Hello Flutter"), backgroundColor: Colors.red,),  
    body:SingleChildScrollView(  
      child: Column(  
        children: [  
          Text("Hello World" ,style: TextStyle(fontSize: 40, color: Colors.red),),  
          SizedBox(height:10),  
          Text("This is my first app", style: TextStyle(fontSize: 32, color:Colors.blue),),  
          SizedBox(height:10),  
          Text("I hope you like it!!", style:TextStyle(fontSize: 18)),  
          SizedBox(height:10),  
          Row(  
            children: [
```

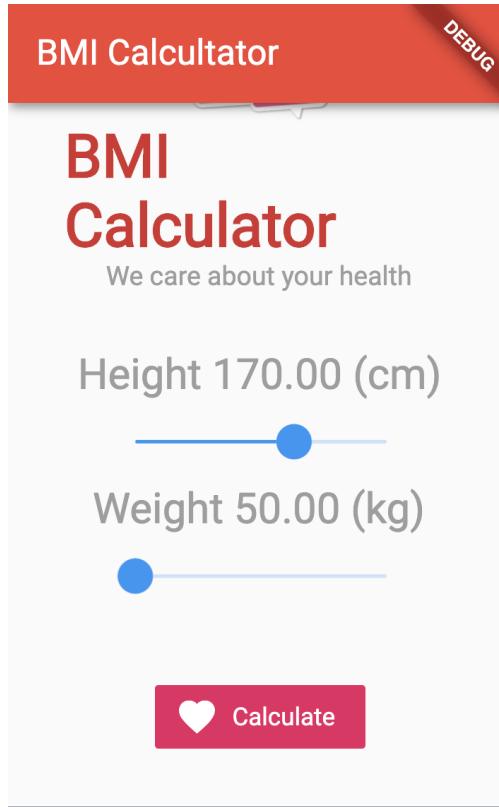
All Widgets saw in Hello World (Basic Widgets)

- UI Widget-> Text, Image, TextButton, ElevatedButton, Icon
- Styling Widget -> TextStyle, Color
- Layout and Positioning : Column, Row, Center, SingleChildScrollView, Padding, SizedBox, Scaffold
-

Day 1 (Part 2)

BMI Calculator exercise

Demo - BMI Calculator



<https://api.flutter.dev/flutter/material/Slider-class.html>

In between BMI Calculator and We care about your health, add an Image from local / files inside the app (<https://www.youtube.com/watch?v=aYGK-kZ1cKQ>)

Other than that it's almost the same except for **Slider**, where you need to find how to use it from the documentation

It's going to be a stateful widget, start building the UI in Widget build() method

stateful + tab

<https://www.cdc.gov/healthyweight/images/assessing/bmi-adult-fb-600x315.jpg>

How to change button color

```
TextButton.icon(onPressed: () {  
  
}, icon: Icon(Icons.favorite), label: Text("Calculate BMI"), style: ButtonStyle(  
  
foregroundColor: MaterialStateProperty.all(Colors.white),  
  
backgroundColor: MaterialStateProperty.all(Colors.red), // Set the background color to red  
  
),)
```

Then use `Image.asset` to retrieve the file

```
    Text("BMI Calculator", style:TextStyle(fontSize: 32, color:Color(0xFF0000)),  
        SizedBox(height: 8,),  
        Image.asset('assets/heart.jpg'),  
        SizedBox(height: 8,),  
        Text("We care about your health"),  
        SizedBox(height: 8,),  
        Text("Height: ${_heightSliderValue.round()} cm"),  
        Slider(  
            value: _heightSliderValue,
```

```
class _HomePageState extends State<HomePage> {

  @override

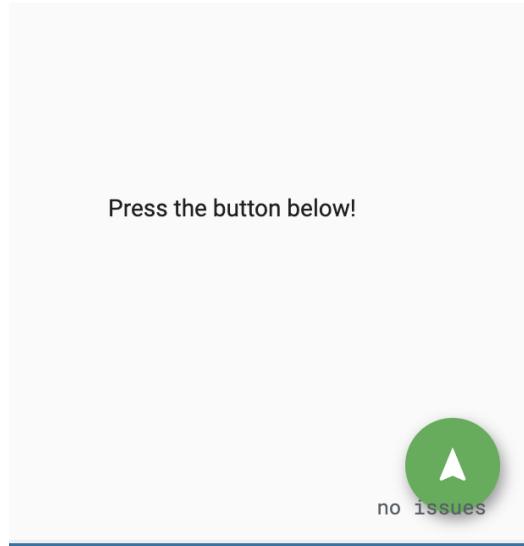
  Widget build(BuildContext context) {
    // To Do Scaffold with App title "BMI Calculator"

    return Scaffold(
      appBar: AppBar(title: const Text("BMI Calculator"), backgroundColor: Colors.red, ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Center(
          child: Column(
```

```
Column (  
  children: [  
    Text("BMI Calculator", style: TextStyle(color:  
      Colors.red, fontSize: 32), ),  
    SizedBox(height: 10, ),  
    Text("We care about our health"),  
    SizedBox(height: 10, ),  
    Image.asset("assets/bmi.png"),
```

Floating Action Button

A floating action button is a circular icon button that hovers over content to promote a primary action in the application



```
floatingActionButton:  
FloatingActionButton(  
    onPressed: () {  
        // Add your onPressed code  
        here!  
    },  
    child: Icon(Icons.navigation),  
    backgroundColor:  
Colors.green,
```

Body

This is where you build the content of your application.

Widgets for layouting

We will discover the widgets that are used to position items within a page. Here are some important/main widgets:

- Container
- Center
- Column
- Row
- SingleChildScrollView

Container

A container is a box! You can specify the width, height, color, padding and margin.
In the below example, EdgeInsets.all means all direction (top, bottom, left, right)

Center(

 child: Container(

 margin: EdgeInsets.all(10.0),

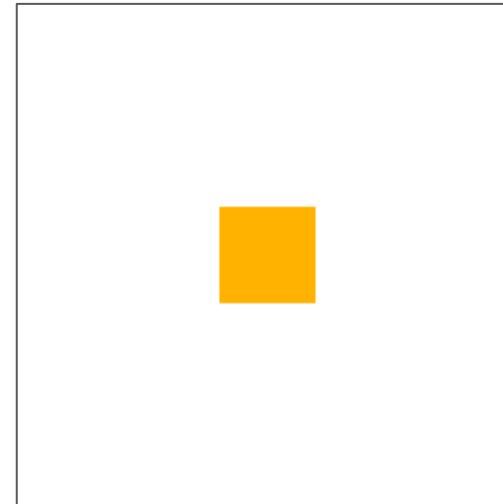
 color: Colors.amber[600],

 width: 48.0,

 height: 48.0,

 padding: EdgeInsets.all(10.0)

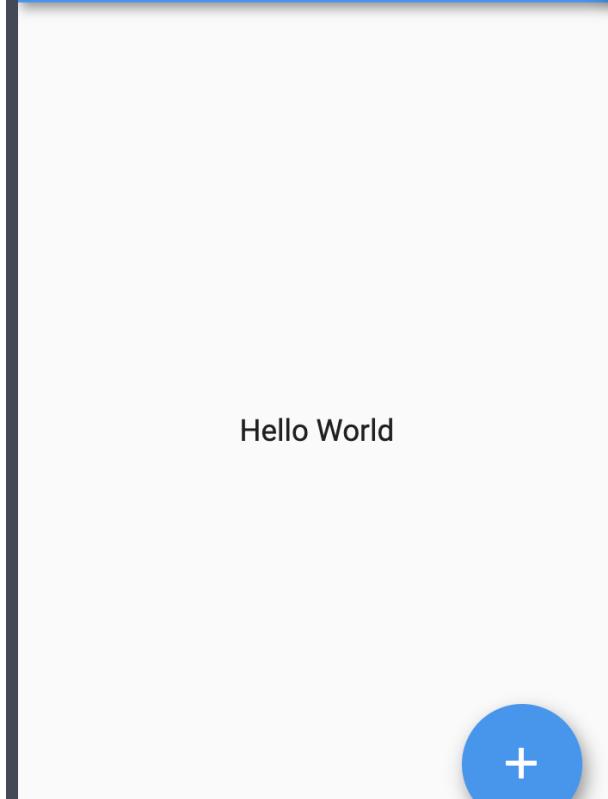
),



Center

A widget that centers its child within itself.

```
Center(child: Text('Hello World'),
```



A screenshot of a Flutter application demonstrating the `Center` widget. The screen has a light gray background. In the center, the text "Hello World" is displayed in a black sans-serif font. At the bottom right corner, there is a blue circular button with a white plus sign inside it. The entire code example from the previous slide is highlighted with a yellow background.

Hello World



Row

A widget that displays its children in a horizontal array.

```
Row(  
  children: <Widget>[  
    Expanded(  
      child: Text('Deliver features faster', textAlign:  
        TextAlign.center),  
    ),  
    Expanded(  
      child: Text('Craft beautiful UIs', textAlign:  
        TextAlign.center),  
    ),  
    Expanded(  
      child: FittedBox(  
        fit: BoxFit.contain,  
        child: const FlutterLogo(),  
      ),  
    ),  
  ],
```

Deliver features
faster

Craft beautiful UIs



Column

A widget that displays its children in a vertical array.

```
Column(  
  children: <Widget>[  
    Text('Deliver features faster'),  
    Text('Craft beautiful UIs'),  
    Expanded(  
      child: FittedBox(  
        fit: BoxFit.contain,  
        child: const FlutterLogo(),  
      ),  
    ),  
  ],  
)
```

Deliver features faster
Craft beautiful UIs



SingleScrollView

A box which allows a single widget to be scrolled.

You will use this when you have a single box that will normally be entirely visible, for example a clock face in a time picker, but you need to make sure it can be scrolled if the container gets too small in one axis

Visible widget in Flutter

Once you know how to position items on a page, we will see some of the widgets that you can use in your application. Here are some important/main widgets:

- Text
- Image
- Button
- Icon
- Slider

Text

This widget is used to displays a text with single style.

You might need to use TextStyle widget as well with this widget to add styling to the text, for example to add color, set to bold

```
Text(  
  'Hello World',  
  textAlign: TextAlign.center,  
  style: TextStyle(fontWeight: FontWeight.bold,  
  color: Colors.red),  
)
```

Image

To show an image. You may show an image from:

- Downloaded from a URL
- Stored locally in assets folder

```
Image(
```

```
    image:
```

```
        NetworkImage('https://flutter.github.io/assets-for-  
        api-docs/assets/widgets/owl.jpg'),
```

```
)
```

Icon

As per its name, an icon is a widget that is predefined, and can be used directly within your application.

You may refer to Icon documentation, to see all available icon ready to be used in your application

```
Icon(  
  Icons.audiotrack,  
  color: Colors.green,  
  size: 30.0,  
)
```

<https://api.flutter.dev/flutter/material/Icons-class.html>

RaisedButton

A raised button, follows Material design principle is a button that raises slightly, configurable via elevation property.

You will need to declare what should happen when the button is pressed via it's onPressed property.

Other type of button includes
FlatButton

```
RaisedButton(  
    child: Text('Color Changed'),  
    color: Colors.green,  
    onPressed: () {  
        print("Hello World");  
    },
```

Slider

A slider can be used to select from either a continuous or a discrete set of values.

We will use `onChanged` property to update the value of item, once the value of slider changed.

```
Slider(  
    value: _value.toDouble(),  
    min: 1.0,  
    max: 10.0,  
    onChanged: (double newValue) {  
        setState(() {  
            _value = newValue.round();  
        });  
    },
```

Stateless Widget

Stateless Widget is a widget that is immutable.

Stateless widgets cannot change their state during the runtime of the app, which means the widgets cannot be redrawn while the app is in action.

```
class MyApp extends StatelessWidget {  
  
  @override  
  
  Widget build(BuildContext context) {  
  
    return Container()  
  
  }  
}
```

Stateful Widget

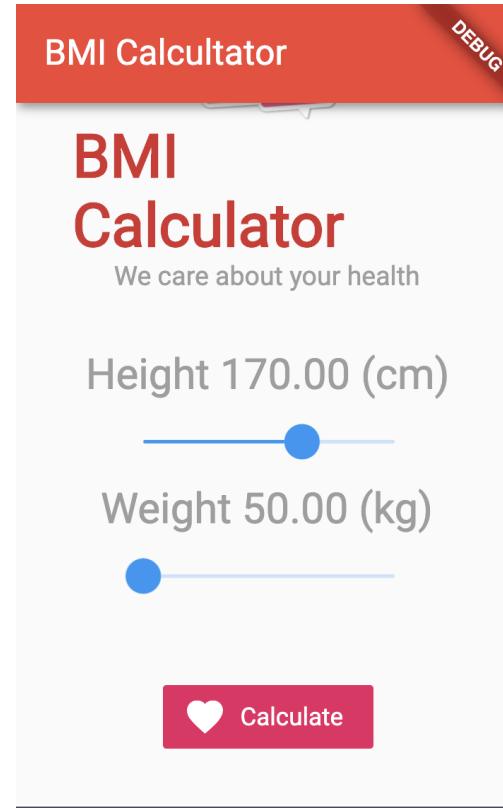
Stateful Widget is a widget that stores variable (state).

This widget will rebuild itself whenever there is a change of its state (variable)

For example when user interact with a button, you might change the state/variable within the widget => Widget will be refreshed.

```
class MyWidget extends StatefulWidget {  
  
  @override  
  
  _MyWidget createState() => _MyWidget();  
  
}  
  
class _MyWidget extends State<MyWidget>{  
  
  @override  
  
  Widget build(BuildContext context){  
  
    return Container();  
  
  }  
  
}
```

Demo - BMI Calculator



BMI Calculator steps

- 1) To get the starter code :
 - remove class MyHomePage below ~ line 38
 - Remove the comments ~ 16 -> 30
- 2) stless + tab (Stateless widget) **or** stful + tab (Stateful widget) ~ line 24 (call it HomePage)
- 3) Replace Placeholder with Scaffold (appBar, body)

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

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

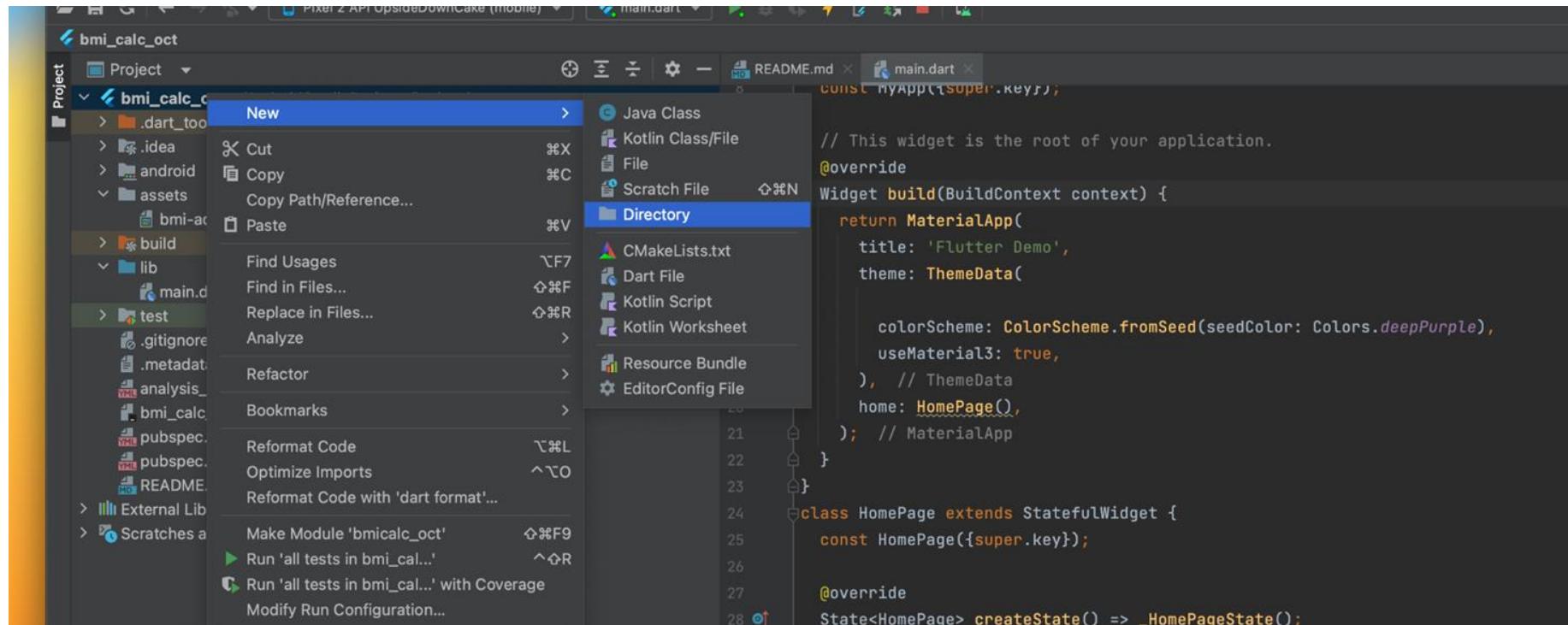
  // This widget is the root of your application.

  @override
```

```
Scaffold(  
    appBar: AppBar(title: Text("BMI Calculator")) ,  
    body: Center(  
        child: Column(  
            children: [  
                Text("BMI Calculator", style: TextStyle(fontSize: 32,  
color:Colors.red)) ,  
                SizedBox(height: 8,) ,
```

Adding image into local project

- 1) Create a new directory called “images” / “assets”
- 2) Drag the images into folder or copy it inside the folder
- 3) Add folder name inside pubspec.yaml under assets section (line 64)
- 4) Use Image.asset to and add the folder + file name



The screenshot shows the Android Studio interface with the project 'bmi_calc_oct' open. A context menu is displayed over the file 'main.dart' in the project tree. The menu path 'New > Directory' is highlighted. The main code editor on the right contains the following Dart code:

```
const MyApp({super.key});  
  
// This widget is the root of your application.  
@override  
Widget build(BuildContext context) {  
    return MaterialApp(  
        title: 'Flutter Demo',  
        theme: ThemeData(  
  
            colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),  
            useMaterial3: true,  
        ), // ThemeData  
        home: HomePage(),  
    ); // MaterialApp  
  
class HomePage extends StatefulWidget {  
    const HomePage({super.key});  
  
    @override  
    State<HomePage> createState() => _HomePageState();
```

```
54 flutter:
55
56     # The following line ensures that the Material Icons font is
57     # included with your application, so that you can use the icons in
58     # the material Icons class.
59     uses-material-design: true
60
61     # To add assets to your application, add an assets section, like this:
62 assets:
63     - assets/
64         #   - images/a_dot_ham.jpeg
65
66     # An image asset can refer to one or more resolution-specific "variants", see
67     # https://flutter.dev/assets-and-images/#resolution-aware
68
69     # For details regarding adding assets from package dependencies, see
70     # https://flutter.dev/assets-and-images/#from-packages
71
72     # To add custom fonts to your application, add a fonts section here,
73     # in this "flutter" section. Each entry in this list should have a
```

```
32 @override
33 Widget build(BuildContext context) {
34     return Scaffold(
35         appBar: AppBar(title: Text("BMI Calculator")),
36         body: Center(
37             child: Column(
38                 children: [
39                     Text("BMI Calculator", style: TextStyle(fontSize: 32, color: Colors.red)),
40                     SizedBox(height: 8),
41                     Text("We care about your health", style: TextStyle(fontSize: 24)),
42                     SizedBox(height: 8),
43                     Image.asset("assets/bmi-adult.jpeg"),
44                     Text("Height (170 cm)"),
45                     SizedBox(height: 8),
46                     Text("Weight (70kg)"),
47                     SizedBox(height: 8),
48                     TextButton.icon(onPressed: (){}, icon: Icon(Icons.favorite), label: Text("Calculate"))
49                 ],
50             ), // Column
51         ), // Center
52     ); // Scaffold
```

Remark

Everytime we change something in pubspec.yaml, you might have to press stop and play. Sometimes hot reload is not working

Quick notes

SizedBox (Margin) - Distance between an element and **another element**
(Jirannya)

Padding: Distance between an element and it's **inner element** (*anaknya*)

Adding local image

- 1) Create an assets folder inside bmi_calc
- 2) Drag the local image inside this folder ...
- 3) Declare the assets folder inside pubspec.yaml and press Pub Get.
- 4) We will use **Image.assets** widget to retrieve the image

```
12  # The following line ensures that the material icons are
13  # included with your application, so that you can use them
14  # the material Icons class.
15  uses-material-design: true
16
17  # To add assets to your application, add an assets section
18  assets:
19    - assets/
20      #   - images/a_dot_ham.jpeg
21
22  # An image asset can refer to one or more resolution-specific
23  # https://flutter.dev/assets-and-images/#resolution-aware
24
25  # For details regarding adding assets from package dependencies,
26  # https://flutter.dev/assets-and-images/#from-packages
```

Slider reference (doc)

<https://api.flutter.dev/flutter/material/Slider-class.html>

Height and Weight Slider

```
double _heightSliderValue = 170;  
  
double _weightSliderValue = 70;  
  
@override  
Widget build(BuildContext context) {  
    .....  
    Scaffold(appBar: AppBar(  
        title: Text("Height and Weight Slider"),  
        actions: [  
            IconButton(icon: Icon(Icons.add), onPressed: () {}),  
            IconButton(icon: Icon(Icons.remove), onPressed: () {}),  
        ],  
    ),  
    body: Column(  
        children: [  
            Row(  
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
                children: [  
                    Text("Height: " + _heightSliderValue.toString()),  
                    Slider(value: _heightSliderValue, onChanged: (value) {  
                        setState(() {  
                            _heightSliderValue = value;  
                        });  
                    }),  
                ],  
            ),  
            Row(  
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
                children: [  
                    Text("Weight: " + _weightSliderValue.toString()),  
                    Slider(value: _weightSliderValue, onChanged: (value) {  
                        setState(() {  
                            _weightSliderValue = value;  
                        });  
                    }),  
                ],  
            ),  
        ],  
    ),  
);  
}
```

BMI Calculation

```
import 'dart:math';

var tempBmi = _weightSliderValue /
pow(_heightSliderValue/100),2);

print(tempBmi);
```

Bringing out BMI to the UI

```
int _bmi = 0;
```

```
Text("Your BMI is ${_bmi}),
```

```
setState( () {
```

```
    _bmi = tempBmi.round();
```

```
});
```

tempBMI scenario

```
var tempMessage = "";

if (tempBmi < 18) {

    tempMessage = "You are underweight";

} else if (tempBmi < 25) {

    tempMessage = "You are normal";

}

else if (tempBmi < 30) {

    tempMessage = "You are overweight";

}

else {

    tempMessage = "You are obese";
}
```

Updating the message in the UI

```
int _bmi = 0;  
  
var _message = "";  
  
Text("Your BMI is $_bmi"),  
Text(_message),
```

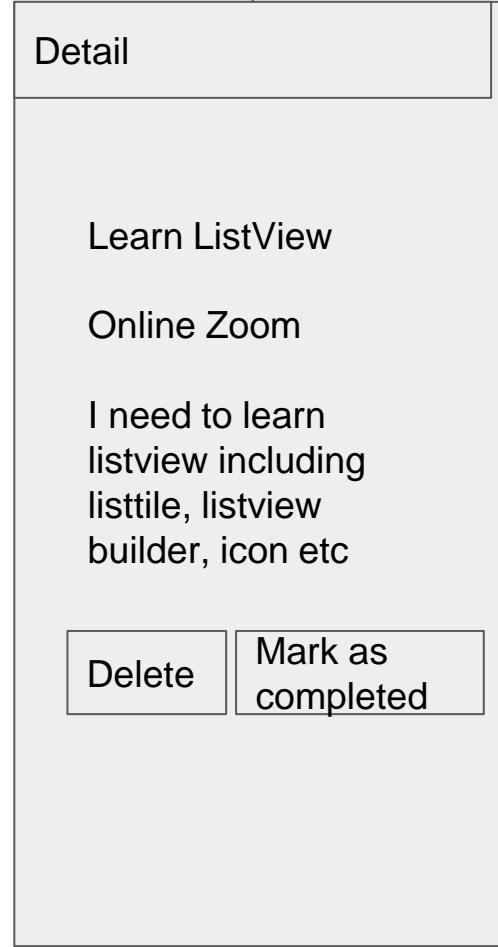
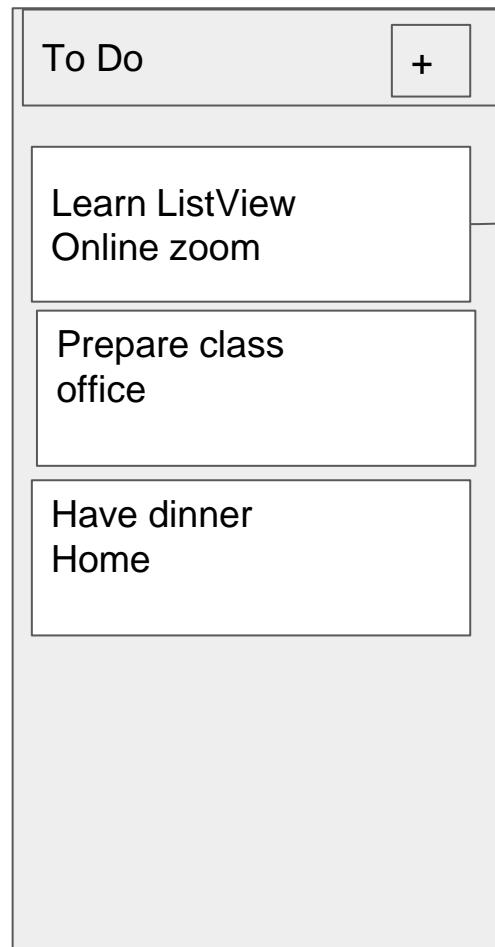
```
setState(() {  
    _bmi = tempBmi.round();  
  
    _message = tempMessage;  
});
```

Conditional Rendering

(condition) ? (true statement) : (false statement)

If (condition) then (true statement) if not (false statement)

```
_bmi != 0 ? Text("Your BMI is ${bmi}") : SizedBox(),
```



If you
you c
setSt
the U

Instruction before break

Create a new flutter project call it todos

Run the project first

Digest the dart code (dartpad.dev) ->Code in chat

See you at 11.05

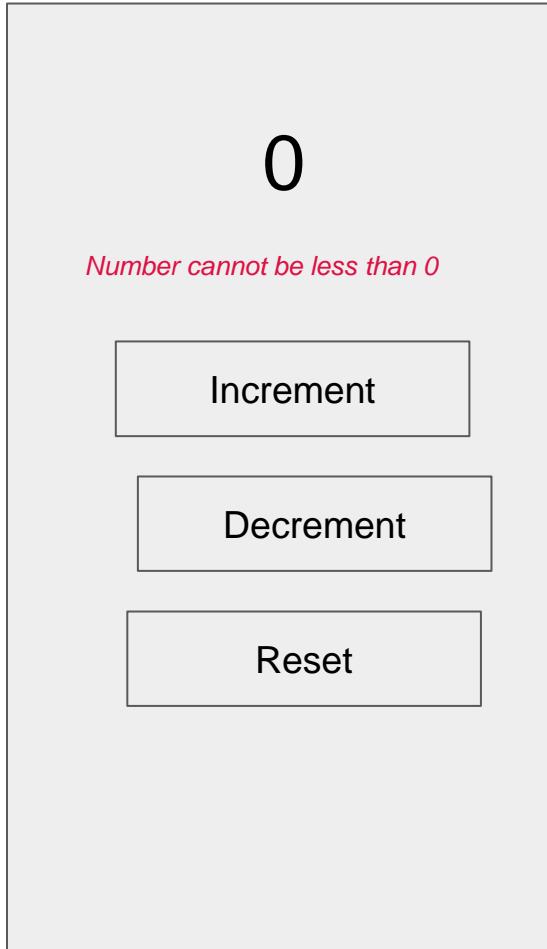
To do app

- Create a new project call it todos
- Inside todos/lib... you are going to create a directory, call it widgets.
- Inside widget, you create dart file (add.dart, detail.dart, home.dart)
- Import material.dart inside each of the files

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

- Clean up main.dart as per yesterday (38 - below, 16-30)

Stateful	Stateless
stful + tab	stless + tab
Show and updating the UI upon an event (eg: press on a button, slider changed) <i>setState()</i>	Only to show ... without updating the UI
Before the page load.. Do something, eg: retrieve the data from database, from API <i>initState()</i>	Cannot do any initialization ..
To start you can resume it like this: <ul style="list-style-type: none"> ● API Call ● Database ● Form Create, Update, Delete	Detail/Reading part: <ul style="list-style-type: none"> ● Read, getting data from other pages Read FutureBuilder, Streambuilder, State Management eg with Provider -> Intermediate/Advance topic



Counter app

Increment -> Increase the counter

Decrement -> Decrease the counter

Reset -> Reset back to 0

If user try to press decrement at 0, show “Number cannot be less than 0 in UI”

But if number cannot be less than

Data type revision

```
var numbers = [1,2,3,4,5];
```

```
numbers.add(6);
```

```
numbers.removeAt(2);
```

```
numbers.forEach((x){print(x);});
```

```
var info =
```

```
{"name":"luqman","email": "luqman@gmail.com", "phone" : "0123456789"};
```

```
print(info["name"]!);
```

Todo list

```
var _todos = [{  
  
  "name": "Multiple page",  
  
  "desc": "How to structure multiple page",  
  
  "place": "Amphi"},  
  
  {  
  
    "name": "Navigation",  
  
    "desc": "Simple , pass data forward, pass data backward",  
  
    "place": "Amphi"  
  
  },  
  
  {  
  
    "name": "ListView",  
  
    "desc": "ListView, ListTile and Card",  
  
  }]
```

HomePage

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

class HomePage extends StatefulWidget {

  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
```

DetailPage

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

class DetailPage extends StatelessWidget {

  const DetailPage({super.key});

  @override

  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Detail Page"),),
      body:Text("Detail page");
    );
  }
}
```

AddPage

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

class AddPage extends StatelessWidget {

  const AddPage({super.key});

  @override

  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Add Page")),
      body: Text("Add Page"),
    );
  }
}
```

Main.dart (Update home page)

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

import 'widgets/home.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.

  @override
```

Adding a button on the AppBar

```
return Scaffold(  
    appBar: AppBar(title: Text("Home Page"), actions: [  
        IconButton(onPressed: () {}, icon: Icon(Icons.add))  
    ]),  
    body: Text("Home Page")  
);
```

Opening new page (add.dart)

```
import 'add.dart';

...
Navigator.push(context,
MaterialPageRoute(builder: (context)=>AddPage()))
);
```

Add.dart

In Add.dart add 3 Textfields with hintText:

- Enter item name
- Enter item description
- Enter item place

Add an ElevatedButton , Once button pressed bring out the items entered by user in print

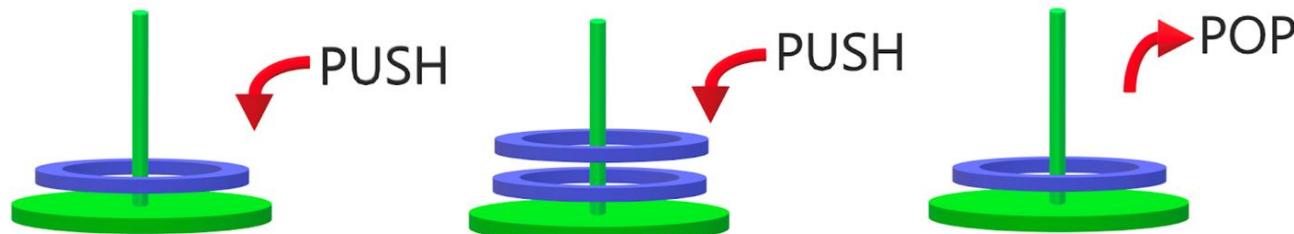
Add.dart forms

```
var nameEditingController = TextEditingController();  
  
var descEditingController = TextEditingController();  
  
var placeEditingController = TextEditingController();  
  
.....  
  
Padding(  
  padding: const EdgeInsets.all(8.0),  
  child: Column(  
    children: [  
      TextField(decoration: InputDecoration(hintText: "Enter item name"),  
      controller: nameEditingController,),
```

Navigator

- Navigator.push (opening a new page)
- Navigator.pop (go back from a page)

Navigation stack follows the term in Computer Science Data structure Stack



Simple Representation of a Stack with PUSH and POP operations

```
void main() {  
  
    // List  
  
    // Implicit declaration  
  
    // List<Map<String, String>>  
  
    var todos = [  
  
        {  
  
            "name": "Learn ListView",  
  
            "place": "Microsoft Teams",  
  
            "desc": "Learn ListView, ListTile, Card, passing data"  
  
        },  
  
        {
```

ListView

<https://api.flutter.dev/flutter/widgets/ListView-class.html>

ListView

itemCount -> How many rows are there => Normally it is the length of the list with data

itemBuilder -> What to show on each row

```
ListView.builder(  
  
    padding: const EdgeInsets.all(8),  
  
    itemCount: _todos.length, // number of items in the list  
  
    itemBuilder: (BuildContext context, int index) {  
  
        return Container(  
  
            height: 50,  
  
            color: Colors.amber,  
  
            child: Center(child: Text(_todos[index]["name"]!)), // index represent the row, starts with 0  
        );  
    }  
)
```

ListTile

Is the component that will help you to create a row in a ListView. It has 5 properties:

- title
- subtitle
- leading
- trailing
- onTap

<https://api.flutter.dev/flutter/material/ListTile-class.html>

ListTile

```
ListView.builder(  
  
    padding: const EdgeInsets.all(8),  
  
    // How many rows are there = the length of your data/List..  
  
    itemCount: _todos.length,  
  
    // What to show for each row (index)  
  
    // For each row, I am going to show a Container of height 50, of color amber  
  
    // Inside each row, I will show a Text that will show the todos for the row and value name  
  
    itemBuilder: (BuildContext context, int index) {  
  
        return ListTile(  
            title: Text(_todos[index]["name"]!),  
            subtitle: Text(_todos[index]["value"]!));  
    }  
);
```

The code as of now

<https://github.com/wanmuz86/todos-silverlake-oct>

```
],), // AppBar
body: ListView.builder(
    itemCount: itemList.length,
    itemBuilder: (BuildContext context, int position){
        return ListTile(
            title: Text(itemList[position]["name"]),
            subtitle: Text(itemList[position]["place"]),
            trailing: Icon(Icons.arrow_forward_ios),
            onTap: (){
                Navigator.push(context, MaterialPageRoute(builder: (context)=>DetailPage()));
            },
        ); // ListTile
    }), // ListView.builder
); // Scaffold
```

ListTile code

```
ListView.builder(  
  
    padding: const EdgeInsets.all(8),  
  
    // How many rows are there <- 90 percent it will be the length of your datasource  
  
    itemCount: _todos.length,  
  
    // What to show on each row <- index on the second argument refer to the row of the list  
  
    // Normally we will retrieve datasource[index]  
  
    itemBuilder: (BuildContext context, int index) {  
  
        return Card(  
  
            child: ListTile(  
  
                title: Text(_todos[index]["name"])),
```

Passing data to second page

- 1) You are going to create a variable to receive the item in the ***receiver*** (detail.dart)
- 2) You create a ***constructor of the receiver:*** (Refer to the 3rd chapter of the PDF)

```
class DetailPage extends StatelessWidget {  
  
    // 1) Create a property/variable to store passed item  
    final Map<String, String> todo;  
  
    // 2) Create the constructor of the class to receive the item  
  
    DetailPage({required this.todo});
```

Passing data to second page

3) In **sender** , which is home.dart will send the item in Navigator.push in constructor

```
Navigator.push(  
    context,  
    MaterialPageRoute(builder: (context)=>DetailPage(  
        // 3) Pass the data in the constructor of DetailPage  
        todo: todos[index],  
    ))  
);
```

4) Bring out the value in the UI of **receiver**

Detail.dart

```
Scaffold(  
  appBar: AppBar(title: Text("Detail")),  
  body: Center(child: Column(  
    children: [  
      Text(todo["name"]!),  
      Text(todo["desc"]!),  
      Text(todo["place"]!),  
    ],  
  )),  
) ;
```

MainAxisAlignment and Cross AxisAlignment

Column... From top to the bottom

- mainAxisAlignment - top to the bottom
- crossAxisAlignment - left to right

Row ... From left to right

- mainAxisAlignment: left to right
- crossAxisAlignment : top to bottom

Get data from add.dart

<https://flutter.dev/docs/cookbook/navigation/returning-data>

- 1) From the **receiver** (home.dart) , you will add a variable on the navigator.push.

Add await async on navigator push

```
return Scaffold(  
  appBar: AppBar(title: Text("To do app"), actions: [  
    IconButton(icon: Icon(Icons.add), onPressed: () async {  
      final result = await Navigator.push(context, MaterialPageRoute(builder: (context)=>AddPage()));  
    }) // IconButton  
  ]), // AppBar  
  body: ListView.builder(  
    itemCount: itemList.length,  
    itemBuilder: (BuildContext context, int index) {
```

```
AppBar(title: Text("Home Page"),  
  
actions: [  
  
    IconButton(onPressed: () async {  
  
        // 1) In receiver, I am going to WAIT for AddPage to finish, and receive the passed item  
  
        // If you use await, you need to add async at the nearest function {}  
  
        -- var itemPassed = await Navigator.push(context, MaterialPageRoute(builder: (context)=>AddPage()));  
  
    }, icon: Icon(Icons.add))  
  
],),
```

2) In the **sender** (add.dart) you will pass back the data as the second argument using Navigator.pop

```
ElevatedButton(onPressed: () {  
    //2) In add.dart, the item need to passed back is sent as the second parameter of navigator.pop  
    var itemToPassed = {  
        "name":nameEditingController.text,  
        "desc":descEditingController.text,  
        "place":placeEditingController.text  
    };  
    Navigator.pop(context, itemToPassed);  
  
}, child: Text("Add new item"))
```

3) Update the code in home.dart to receive the passed item and add it into the list

```
IconButton(onPressed: () async {
// 1) Create variable to retrieve the item from new Page, and wait for the page to
finish
    // If await is used, need to add async to the nearest function {}

    var itemPassed = await Navigator.push(
        context,
        MaterialPageRoute(builder: (context)=> AddPage())
    );

    if (itemPassed != null){
        _todos.add(itemPassed);
        setState(() {
            _todos;
        });
    }
}

), icon: Icon(Icons.add)],
```

Code with verification

```
if (nameEditingController.text == "" || placeEditingController.text == "" || descEditingController.text == "") {  
  
    print("Later add toast to remind user to fill in all info");  
  
}  
  
else {  
  
    var newItem = {  
  
        "name": nameEditingController.text,  
  
        "place": placeEditingController.text,  
  
        "desc": descEditingController.text  
  
    };
```

Homework

1) Add local image in app: <https://youtu.be/aYGK-kZ1cKQ?si=FvtpmB3n6OU7S6qx>

2) How to bring out the correct keyboard:

<https://www.youtube.com/watch?v=HQUV2RLh8V0&list=PLeEjv6GjeKWdIKWowAmvDsrxGC6fncE7H>

3) ListView

4) Pass forward data

5) Pass backward data

6) Do your API through Sheety:

<https://www.youtube.com/watch?v=F9m9w1EeXJM&list=PLeEjv6GjeKWdIKWowAmvDsrxGC6fncE7H&index=22>

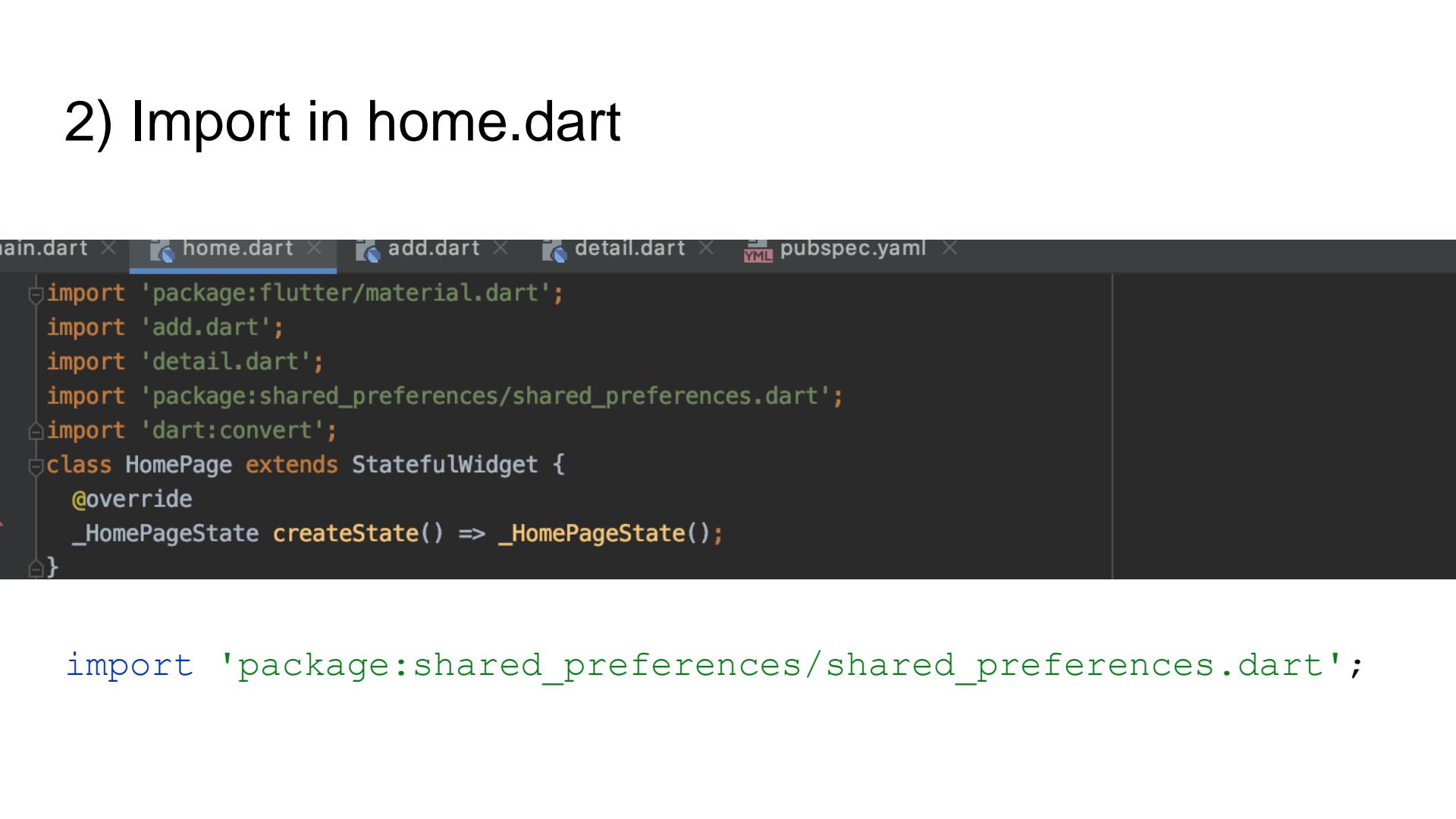
Adding Local Storage (SharedPreferences)

<https://flutter.dev/docs/cookbook/persistence/key-value>

1) Add on pubspec.yaml (get it from pub.dev)

```
# flutter commands
20   # The latest version available on pub.dev. To see which dependencies have newer
21   # versions available, run `flutter pub outdated`.
22
23   dependencies:
24     flutter:
25       sdk: flutter
26
27
28
29   # The following adds the Cupertino Icons font to your application.
30   # Use with the CupertinoIcons class for iOS style icons.
31   cupertino_icons: ^1.0.2
32
33   shared_preferences: ^2.1.1
34
35
36
37   dev_dependencies:
38     flutter_test:
39       sdk: flutter
40
41
42
43
44   # The "flutter_lints" package below contains a set of recommended lints to
```

2) Import in home.dart



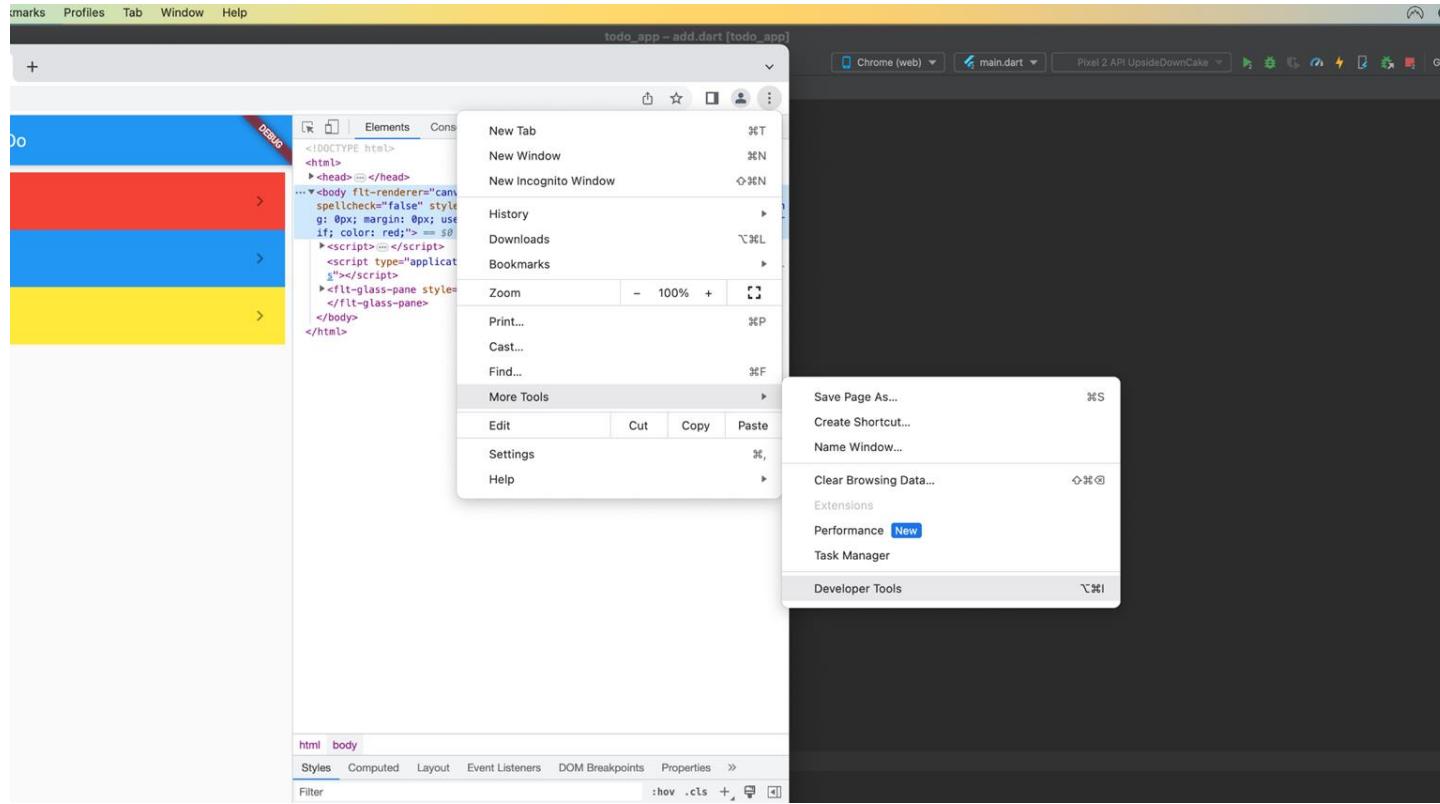
```
main.dart × home.dart × add.dart × detail.dart × pubspec.yaml ×
import 'package:flutter/material.dart';
import 'add.dart';
import 'detail.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'dart:convert';
class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}
```

```
import 'package:shared_preferences/shared_preferences.dart';
```

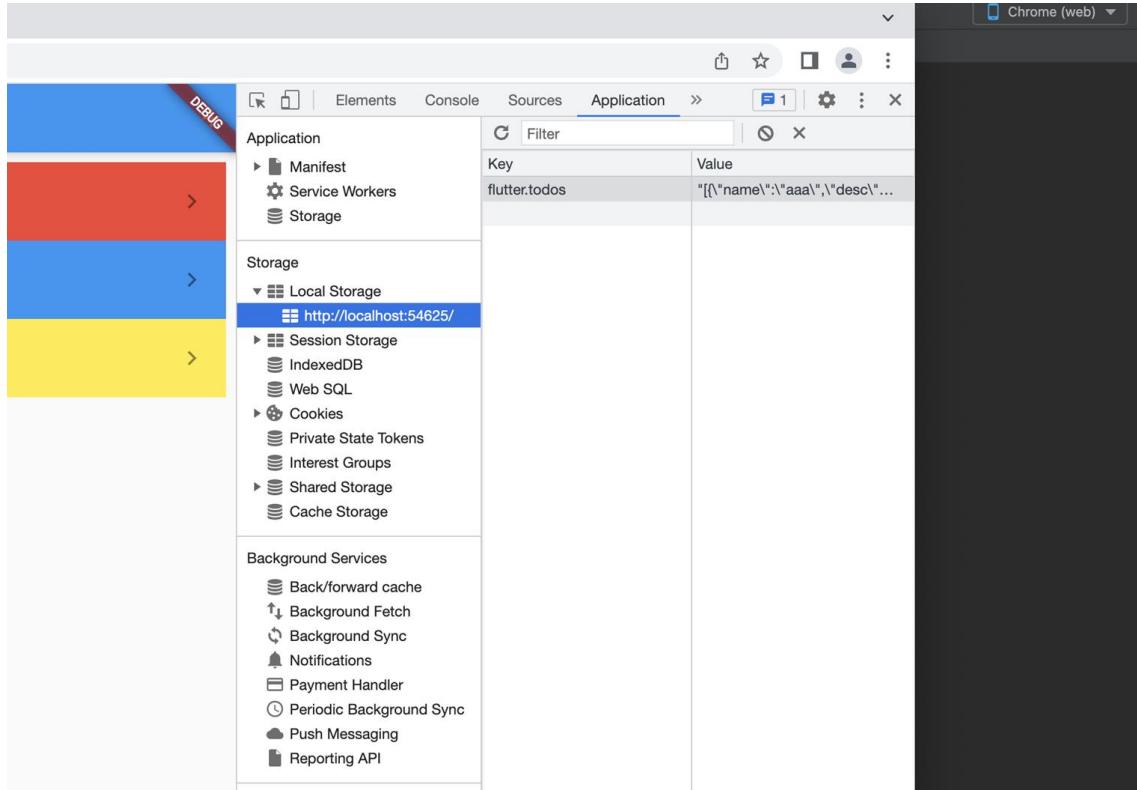
3) Save the todo using shared preference

```
if (newItem != null) {  
  
    _todos.add(newItem);  
  
    // SAVE IN SHARED PREFERENCES  
  
    // GET THE SHARED PREFERENCE MANAGER (AS IF DOING File in words)  
  
    final SharedPreferences prefs = await SharedPreferences.getInstance();  
  
    // SAVE...  
  
    // Transform List of Map <String, String> into String using jsonEncode  
  
    // Save it as a String  
  
    // first parameter is the file name,  
  
    // second parameter data that's going to be saved
```

If you use chrome you can clear data like this



Select Application tab and localstorege.. Remove flutter:todos



4) Retrieve the item When the page is loaded (initState)

```
    @override
    void initState() {
        retrieveData();
        super.initState();
    }

    void retrieveData() async{
        final prefs = await SharedPreferences.getInstance();
        var items = [];
        if (prefs.getString("items") != null){
            items = json.decode(prefs.getString("items"));
        }
        setState(() {
            itemList= items;
        });
    }
}
```

```
void loadData() async {

final SharedPreferences prefs = await SharedPreferences.getInstance();

var todoString = prefs.getString("todos");

if (todoString != null){

    // We need to transform the String (stored inside our shared preference)

    // into List<dynamic>

    var todoList = jsonDecode(todoString);

    setState(() {

        _todos = todoList;

    });

}
```

5) Remove the mock data

```
class _HomePageState extends State<HomePage> {  
    List itemList = [  
        ..  
    ];  
}
```

6) In detail.dart change data type to Map <String,dynamic>

```
class DetailPage extends StatelessWidget {

    final Map<String,dynamic> item;

    DetailPage({required this.item});

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(title: Text("Detail Page")),
            body: Center(
                child: Column(
```

Passing forward index (to use for Edit and Delete) - detail.dart

```
// 1) Create a property/variable to store passed item

final Map<String,dynamic> todo;

final int index;

// 2) Create the constructor of the class to receive the item

DetailPage({required this.todo, required this.index});
```

Home.dart will pass index as second parameter

```
Navigator.push(  
  context,  
  MaterialPageRoute(builder: (context)=>DetailPage(  
    // 3) Pass the data in the constructor of DetailPage  
    todo: _todos[index],  
    index: index,  
  ))
```

When delete pressed, pass back to home the deleted index and action

```
ElevatedButton.icon(onPressed: () {  
  
    var userAction = {  
  
        "action":1, // Assuming 0 is edit completed and 1 is delete  
  
        "index":index  
  
    };  
  
    Navigator.pop(context, userAction);  
  
},
```

In home page, onTap of ListTile will expect something to be returned from detail.dart

```
onTap: () async {
```

```
    var action = await Navigator.push(
```

```
        context,
```

```
        MaterialPageRoute(builder: (context)=>DetailPage(
```

```
            // 3) Pass the data in the constructor of DetailPage
```

```
            todo: _todos[index],
```

```
            index: index,
```

```
        ))
```

```
);
```

```
    if (action != null) {
```

When edit pressed, pass back to home the edited index and action

```
ElevatedButton.icon(onPressed: () {  
  
    var userAction = {  
  
        "action":0, // Assuming 0 is edit completed and 1 is delete  
  
        "index":1  
  
    };  
  
    Navigator.pop(context, userAction);  
  
}).
```

In home page, onTap of ListTile will expect something to be returned from detail.dart

```
if (action["action"] == 0) {
```

```
// edit
```

```
// change the status of completed
```

```
_todos[index]["completed"] = !_todos[index]["completed"];
```

```
saveInStorage();
```

```
setState(() {
```

```
    _todos;
```

```
});
```

```
}
```

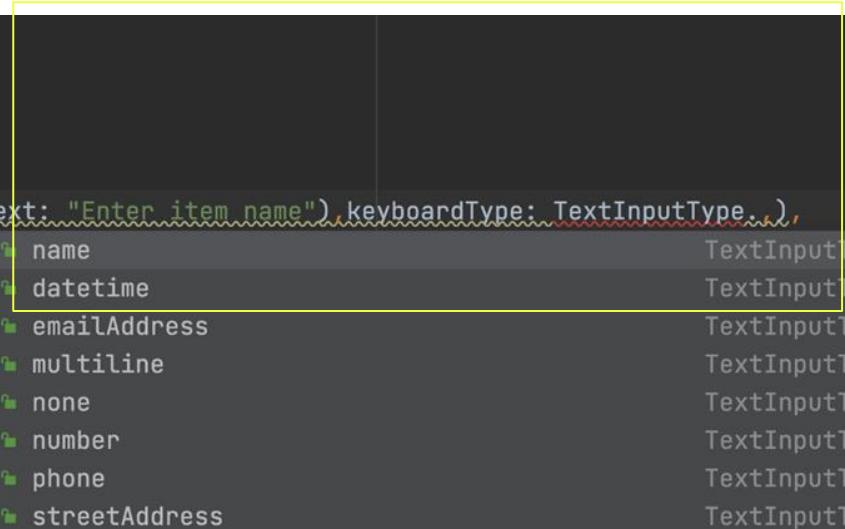
Change the ListTile to show tick on the completed item using conditional rendering

```
ListTile(  
  leading: _todos[index] ["completed"] ? Icon(Icons.check) : SizedBox(),  
  title: Text(_todos[index] ["name"]!),  
  subtitle: Text(_todos[index] ["place"]!),  
  trailing: Icon(Icons chevron_right),  
  onTap: () async {
```

- Conditional Rendering (BMI Calc)
- List, Map , dynamic
- Navigation
 - Push,pop
 - Pass forward
 - Pass backward data
- Showing List on a ListView, ListTile, Card
- SharedPreference
- Edit, Delete -> Recap of pass forward and passback data

Keyboard type

```
body: Padding(  
    padding: const EdgeInsets.all(8.0),  
    child: Column(  
        children: [  
            TextField(decoration: InputDecoration(hintText: "Enter item name"), keyboardType: TextInputType...),  
            SizedBox(height: 10,),  
            TextField(decoration: InputDecoration(hintText: "Enter item description"), keyboardType: TextInputType...),  
            SizedBox(height: 10,),  
            TextField(decoration: InputDecoration(hintText: "Enter item price"), keyboardType: TextInputType...),  
            SizedBox(height: 10,),  
            TextButton(onPressed: (){}, child: Text("Add"))  
        ],  
    ) // Column
```



- name
- datetime
- emailAddress
- multiline
- none
- number
- phone
- streetAddress

Add password field

```
    edgeInsets.all(8.0),  
  
    decoration: InputDecoration(hintText: "Enter item name"), obscureText: true),  
    height: 10,),  
    decoration: InputDecoration(hintText: "Enter item description"), ),  
    height: 10,),  
    decoration: InputDecoration(hintText: "Enter item place"), ),  
    height: 10,),  
    onPressed: (){}, child: Text("Add new Item"))
```

todo ~/Documents/fluttermimos/todo

- .dart_tool
- .idea
- android [todo_android]
- ios
- lib
- test
- widgets
 - add.dart
 - detail.dart
 - home.dart

.gitignore
.metadata
.packages
pubspec.lock
pubspec.yaml
README.md

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   // This widget is the root of your application.
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      title: 'Flutter Demo',
13      theme: ThemeData(
14
15        primarySwatch: Colors.blue,
16
17        visualDensity: VisualDensity.adaptivePlatformDensity,
18      ), // ThemeData
19      home: MyHomePage(title: 'Flutter Demo Home Page'),
20    );
21 }
```

<https://flutter.dev/docs/cookbook/navigation/navigation-basics>

```
get built(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(title: Text("To do app"), actions: [  
      IconButton(icon: Icon(Icons.add), onPressed: (){  
        Navigator.push(context, MaterialPageRoute(builder: (context)=>AddPage()));  
      }) // IconButton  
    ], // AppBar  
    body: Center(child: Text("Hello World")),  
  );  
}
```

Revision part

Create the form page on add.dart

Insert Item Name

Insert Item Description

Insert Place

Button

When the button is pressed, print the value entered by user

Create a mock list

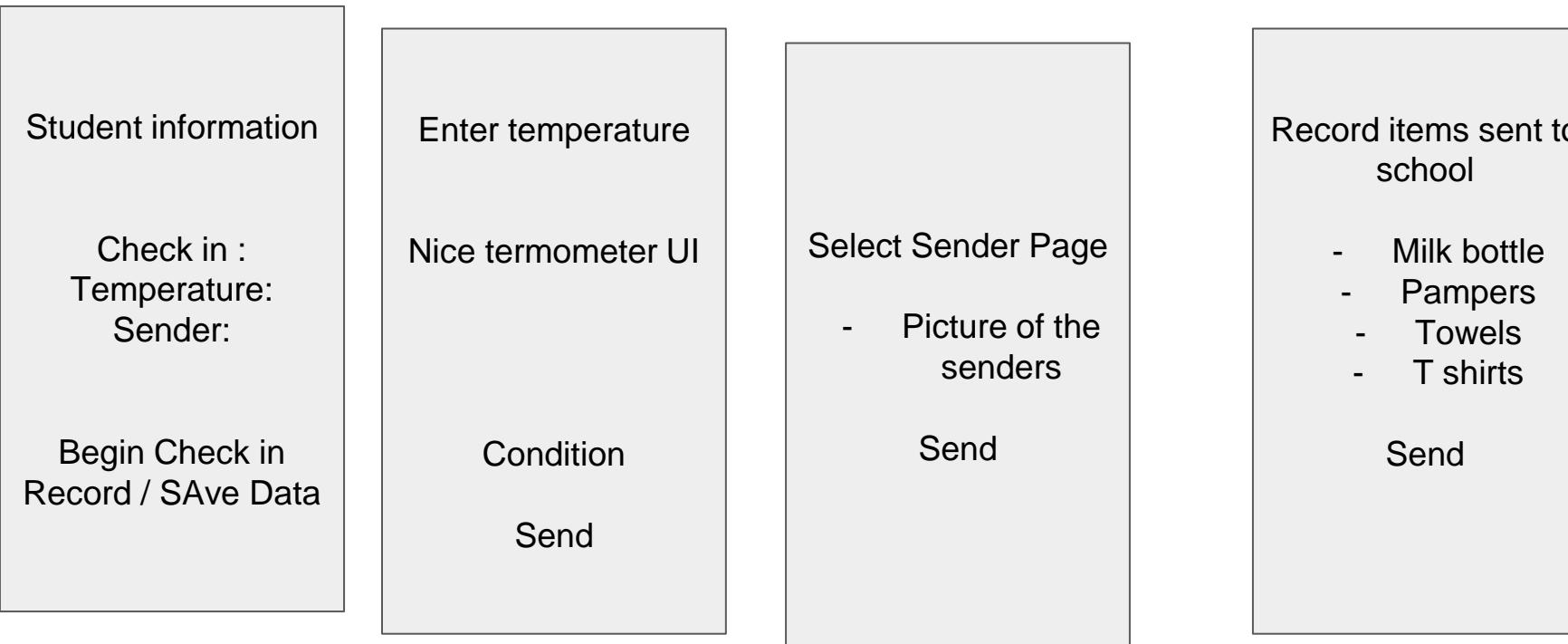
```
class _HomePageState extends State<HomePage> {
  List itemList = [
    {"name": "dinner", "desc": "dinner with friends", "place": "home"},  

    {"name": "bug fix", "desc": "fix bug on the app", "place": "home"},  

    {"name": "pay salary", "desc": "pay staffs salary", "place": "office"}  

  ];
  

  @override
  Widget build(BuildContext context) {
```



<https://docs.flutter.dev/data-and-backend/state-mgmt/simple>

Data Persistence in Mobile application

- Data persistence refers to the process of storing data in a way that it can be accessed and retrieved even after the application is closed or the device is restarted.
- An application can be persisted **locally** or **remotely**. Remote persistence is only available when user has connection to internet.

Local Persistence

1. **Shared Preferences:** SharedPreferences is a lightweight key-value storage mechanism provided by Android and iOS. It is ideal for storing small amounts of data such as user preferences, settings, or configuration values. (300 MB) - Like your website cache
2. Internal Storage: Every Android application has its own private internal storage directory, which can be used to store private data in the form of files. Internal storage is suited for storing sensitive or private data that should not be accessible to other applications. - File inside phone
3. External Storage: Android allows applications to read and write data to the device's external storage such as an SD card, suitable for storing large files or data that can be shared with other applications or accessed by the user. -- File inside SD Card

Local Persistence

4. **SQLite Database:** Android and iOS provides support for a built-in relational database management system called SQLite, to store structured data, and they offer efficient querying and manipulation of data. They are suitable for storing large amounts of structured data that require complex querying operations.
5. **Persistence Library:** Persistence Library is an Android library that provides an abstraction layer over SQLite, making it easier to work with databases in Android applications. It offers compile-time verification of SQL queries, convenient APIs for database operations, and integration with LiveData and ViewModel components.

Remote Persistence

1. Network Persistence: Android or IOS applications often need to interact with remote servers or APIs to fetch or send data. In such cases, data can be persisted using network persistence techniques such as caching, offline data synchronization, or using remote databases.
2. Backend as a service:Backend as a service , eg Firebase is a platform offered by Google that provides a suite of tools and services for mobile and web app development. It includes Firebase Realtime Database and Firebase Firestore, which are NoSQL databases hosted in the cloud. They offer real-time synchronization, offline support, and scalable data storage for Android applications.

<https://www.digitalocean.com/community/tutorials/gangs-of-four-gof-design-patterns>

MVC, ORM, Singleton, Factory

Python

JS

Java

Dart

Django

NodeJS

Spring Boot

Flutter

CMS- (Java, PHP
(Laravel, CI,
CakePHP) Python,
ASP, .Net, Javascript
- ReactJS, VueJS,
Angular)

Payment - Paypal
Chat - Twilio,
Firebase
Facerecognition -
wyse.ai, Firebase,
Fakuti AI
Email, SMS
Push notification

Front Website (HTML, CSS, JS)
-Pure CSS, Bootstrap, Material,
- Vanilla, VueJS, ReactJS, AngularJS

Database
SQL and NoSQL
-MySQL, SQL,
MSSQL,
-Firebase,
MongoDB,
Memcached,
Neo4J

API (Java
Server,
PHP,
Python,
ASP, .Net,
Javascript)

JSON or
XML

iOS

Android

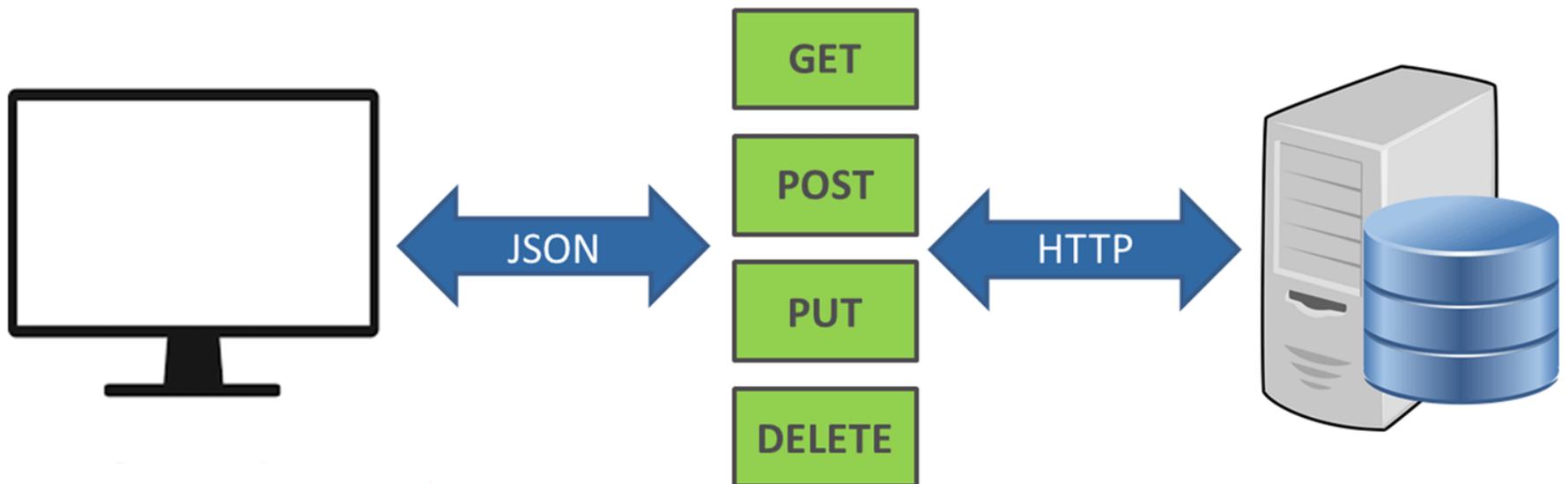


Integrating with Existing Open API

1. Existing or open APIs refer to application programming interfaces that are publicly available for developers to use in their applications.
2. These APIs are typically provided by companies, organizations, or service providers to allow developers to access and utilize their services, data, or functionality.
3. Existing APIs cover a wide range of domains and industries, including social media, weather, maps, payment gateways, messaging, and more.
4. Most API will require you to register and retrieve the key
5. Example <https://github.com/public-apis/public-apis>
6. Example of OMDB API:

<https://www.omdbapi.com/?s=Harry&apikey=87d10179>

<https://www.omdbapi.com/?i=tt6110504&apikey=87d10179>



Client sends a **request**

HTTP methods

Server sends a **response**

HTTP Methods

HTTP Methods	CRUD Operation	Description
POST	Add (Create)	Add Item inside database
PUT / POST	Edit (Update)	Change an Existing Resource
GET	Retrieve (Read)	Retrieve existing resource
DELETE	Delete (Delete)	Delete existing resource

[https://github.com/Babunashvili/Books-To-Read-Before-You-Die/blob/master/Ebooks/build-apis-you-wont-hate-everyone-phil-sturgeon\(www.ebook-dl.com\).pdf](https://github.com/Babunashvili/Books-To-Read-Before-You-Die/blob/master/Ebooks/build-apis-you-wont-hate-everyone-phil-sturgeon(www.ebook-dl.com).pdf)

HTTP Methods example - /hotels

Operation	METHOD	PATH	Receive or not receive data?
Create	POST	/hotels	YES
Read	GET	/hotels	NO
		/hotels/:id	NO
Update	PUT	/hotels/:id	YES
Delete	DELETE	/hotels/:id	NO

The combination of METHOD and PATH cannot be the same

```
void main(){

    String name = "Muzaffar";
    print(name);

    // A variable needs to have value assigned to it

    //String profession;
    //print(profession);

    // A nullable variable is declared with ? / Optional

    String? profession;
```

Insert the title

Search



Harry Potter
2001



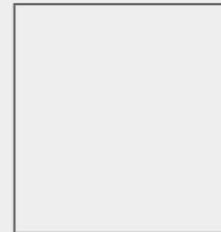
Harry Potter
2002



Harry Potter



Harry Potter



Harry Potter

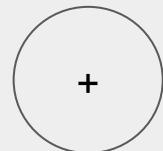
2001

Actor ..

Name
Department

Name
Department

Name
Department



Name

Email

Department

Address

Add new employee

Name

Department

Email

Address

home.dart

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

class HomePage extends StatefulWidget {

  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class HomePageState extends State<HomePage> {
```

detail.dart

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

class DetailPage extends StatefulWidget {

  const DetailPage({super.key});

  @override
  State<DetailPage> createState() => _DetailPageState();
}

class _DetailPageState extends State<DetailPage> {
  @override
```

Home.dart with scaffold

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

class HomePage extends StatefulWidget {

  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
```

Detail.dart with scaffold

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

class DetailPage extends StatefulWidget {

  const DetailPage({super.key});

  @override
  State<DetailPage> createState() => _DetailPageState();
}

class _DetailPageState extends State<DetailPage> {
  @override
```

Main.dart pointing to HomePage()

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

import 'widgets/home.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application

```

Add the mock data inside home.dart

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

class HomePage extends StatefulWidget {

  const HomePage({super.key});

  @override

  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {

  var _movies = [
    {
      "id": 1,
      "title": "The Shawshank Redemption",
      "year": 1994,
      "rating": 9.2
    },
    {
      "id": 2,
      "title": "The Godfather",
      "year": 1972,
      "rating": 9.1
    },
    {
      "id": 3,
      "title": "The Dark Knight",
      "year": 2008,
      "rating": 9.0
    },
    {
      "id": 4,
      "title": "The Godfather: Part II",
      "year": 1974,
      "rating": 8.9
    },
    {
      "id": 5,
      "title": "The Lord of the Rings: The Return of the King",
      "year": 2003,
      "rating": 8.8
    },
    {
      "id": 6,
      "title": "The Godfather: Part I",
      "year": 1972,
      "rating": 8.7
    },
    {
      "id": 7,
      "title": "The Pianist",
      "year": 2002,
      "rating": 8.6
    },
    {
      "id": 8,
      "title": "Schindler's List",
      "year": 1993,
      "rating": 8.6
    },
    {
      "id": 9,
      "title": "The Godfather: Part III",
      "year": 1990,
      "rating": 8.5
    },
    {
      "id": 10,
      "title": "The Thin Red Line",
      "year": 1998,
      "rating": 8.4
    }
  ];
}
```

Scaffold of the first page (without Expanded)

```
@override  
  
Widget build(BuildContext context) {  
  
  return Scaffold(  
  
    appBar: AppBar(title: Text("Home Page")) , body: Column(  
  
      children: [  
  
        Row(  
  
          children: [  
  
            TextField(decoration: InputDecoration(hintText:
```

UI is not there

ImageView, TextView , Button <- This UI normally has a default width or height => The size of the content , eg: “Hello”

Flutter does not know how to position these UI element in term of width and height

For example in Row , Which one is bigger? TextField or Button

In column which one is bigger? Row or ListView?

We have to specify the width but in a **relative** way

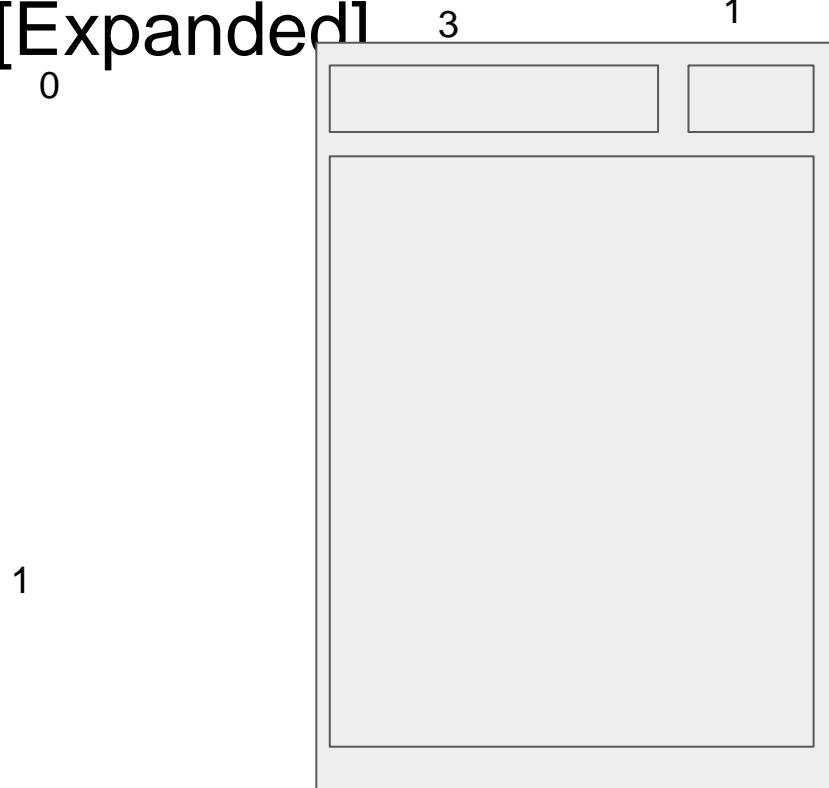
Error when Flutter does not know how to position the widget

```
constraints: BoxConstraints(unconstrained)
↑ size: MISSING <62 children>
=====
===== Exception caught by rendering library =====
The following assertion was thrown during performLayout():
RenderBox was not laid out: _RenderDecoration#ad34b relayoutBoundary=up8 NEEDS-PAINT NEEDS-COMPOSITING-BITS-UPDATE
'package:flutter/src/rendering/box.dart':
Failed assertion: line 1966 pos 12: 'hasSize'

Either the assertion indicates an error in the framework itself, or we should provide substantially more information in this error message to help you determine and fix the underlying cause.
In either case, please report this assertion by filing a bug on GitHub:
https://github.com/flutter/flutter/issues/new?template=2\_bug.md
```

Giving relative/ weightage to the UI element

[Expanded]



Width As big as the inner element

The rest of the page

Expanded

A widget that expands a child of a [Row](#), [Column](#), or [Flex](#) so that the child fills the available space

<https://api.flutter.dev/flutter/widgets/Expanded-class.html>

```
body: Padding(  
    padding: const EdgeInsets.all(8.0),  
    child: Column(  
        children: [  
            Row(  
                children: [  
                    Expanded(  
                        child: TextField(  
                            decoration: InputDecoration(  
                                hintText: "Insert movie to search"  
                            ), // InputDecoration  
                            ), // TextField  
                            ), // Expanded  
                            FlatButton(onPressed: (){},  
                                child: Text("Search"),  
                                color: Colors.blue,) // FlatButton  
                            ],  
                        ), // Row
```

```
SizedBox(height: 8.0),  
Expanded(  
    child: ListView.separated(  
        separatorBuilder: (BuildContext context, int index) => const Divider(),  
        itemCount: movies.length,  
        itemBuilder: (BuildContext context, int index){  
            return ListTile(  
                onTap: (){  
                    Navigator.push(context, MaterialPageRoute(builder:  
                        (context)=>DetailPage())); // MaterialPageRoute  
                },  
                title: Text(movies[index]["Title"]),  
                subtitle: Text(movies[index]["Year"]),  
                leading: Image.network(movies[index]["Poster"]),  
                trailing: Icon(Icons.chevron_right),  
            ); // ListTile  
        },
```

Source Code

```
Scaffold(  
    appBar: AppBar(title: Text("Home Page")), body: Column(  
        children: [  
            Row(  
                // flex 0 means as big as the inner element  
                children: [  
                    Expanded( flex: 2, child: TextField(decoration: InputDecoration(hintText: "Enter movie name"),)),  
                    Expanded( flex: 1, child: ElevatedButton(onPressed: (){}, child: Text("Search Movie")))  
                ],  
            ),  
            Expanded(  
                // Not specifying flex by default flex = 1 => The rest of the page
```

Fetching data from Internet

1. Add the `http` package.
2. Create a custom Dart Object (3a) - 4 or 5 steps
3. Make a network request using the `http` package.

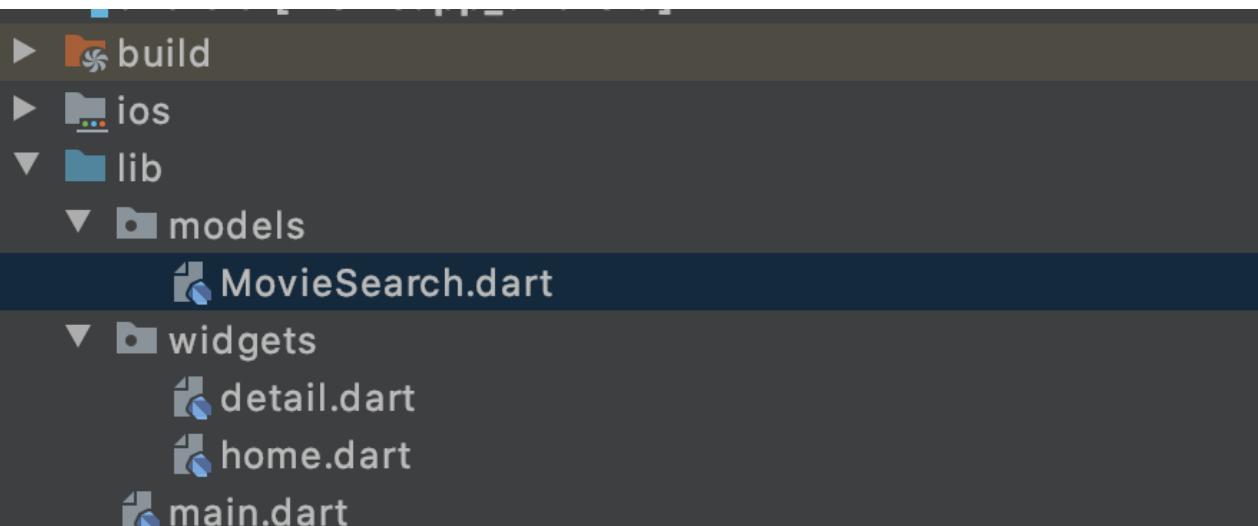
And convert the response into a custom Dart object. (Convert the `http.Response` to an `Album`)

1. Fetch and display the data with Flutter.

Step 2

- Create models folder
 - Create model dart, movie_search.dart , movie_detail.dart
-
- 1) Class MovieSearch
 - 2) Create the properties of the class (what data you want to show on UI)
 - 3) Create the constructor of the class
 - 4) Create JSON to Object transformer (from the doc, the one with factory)
 - 5) If you have a Array, you need to do Array JSON to List Object transformer

Create MovieSearch file inside movies package



main.dart × home.dart × pubspec.yaml × MovieSearch.dart × detail.dart ×

```
// 1) Create the class
class MovieSearch {

    // 2) Create the properties (normally same as the JSON file)
    final String title;
    final String imdbID;
    final String year;
    final String type;
    final String posterUrl;

    // 3) Create the constructor
    MovieSearch({this.title, this.imdbID, this.year, this.type, this.posterUrl});
```

JSON to Object transformer

```
factory MovieSearch.fromJson(Map<String,dynamic> json) {  
  
    return  
  
        MovieSearch(title: json["Title"], year: json["Year"],  
type: json["Type"],  
  
        poster: json["Poster"], imdbId: json["imdbID"]);  
  
}
```

If you have a List of data, need this transformer

```
static List<MovieSearch> moviesFromJson (dynamic json ) {  
    var searchResult = json[ "Search" ] ;  
  
    List<MovieSearch> results = List.empty(growable: true) ;  
  
    if (searchResult != null) {  
  
        searchResult.forEach( (v)=>{  
            results.add(MovieSearch.fromJson (v) )  
        } );  
        return results;  
    }  
    return results;  
}
```

```
// 5) If you are retrieving a List you need to add this step:
```

```
static List<MovieSearch> moviesFromJson(dynamic json){  
    var searchResult = json["Search"];  
    if (searchResult != null){  
        var results = new List<MovieSearch>();  
        searchResult.forEach((v){  
            results.add(MovieSearch.fromJson(v));  
        });  
        return results;  
    }  
    return List<MovieSearch>();  
}
```

MovieSearch class code

```
// 1 ) Create class

class MovieSearch {

// 2) List down the properties of the class

// Flutter convention (camelCase)

final String imdbId;

final String year;

final String title;

final String type;

final String poster;

// 3 ) Create the constructor of the class
```

The imports

```
import 'package:flutter/material.dart';
import 'detail.dart';
import 'package:movieapp/models/MovieSearch.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
```

The method

```
Future<List<MovieSearch>> fetchMovies() async {
    final response = await http.get('https://www.omdbapi.com/?s=Harry&apikey=87d10179');

    if (response.statusCode == 200) {
        // If the server did return a 200 OK response,
        // then parse the JSON.
        return MovieSearch.moviesFromJson(jsonDecode(response.body));
    } else {
        // If the server did not return a 200 OK response,
        // then throw an exception.
        throw Exception('Failed to load album');
    }
}
```

Code to retrieve movies

```
Future<List<MovieSearch>> fetchMovies() async {  
  
    final response = await http  
  
        .get(Uri.parse('https://www.omdbapi.com/?s=fast&apikey=87d10179'));  
  
    if (response.statusCode == 200) {  
  
        // If the server did return a 200 OK response,  
  
        // then parse the JSON.  
  
        // If am working with [] -> Call the 5th method  
  
        // If am working with {} -> call the 4th method  
  
        // Call the last method in your Class  
  
        return MovieSearch.moviesFromJson(jsonDecode(response.body));  
  
    } else {  
  
    }  
}
```

Retrieve the movie and set it on movies variable

```
) // Expanded
FlatButton(onPressed: (){

    fetchMovies().then((value) => setState((){
        movies = value;
    }));
},
child: Text("Search"),
color: Colors.blue,) // FlatButton
],
),
// Row
```

```
Expanded( flex: 1, child: ElevatedButton(onPressed: () {  
    // value is the movies returned from the fuction  
  
    fetchMovies().then((value) => setState(() {  
        _movies = value;  
    }));  
  
}, child: Text("Search Movie")))
```

Empty the list and set the List type

```
class _HomePageState extends State<HomePage> {  
    List<MovieSearch> movies = [];  
    @override  
    Widget build(BuildContext context) {  
        return Scaffold(  
            body: Center(  
                child: CircularProgressIndicator(),  
            ),  
        );  
    }  
}
```

Change the[""] notation by using . notation

```
itemCount: movies.length,  
itemBuilder: (BuildContext context, int index){  
    return ListTile(  
        onTap: (){  
            Navigator.push(context, MaterialPageRoute(builder:  
                (context)=>DetailPage(imdbID: movies[index].imdbID,))); // MaterialPageRoute  
        },  
        title: Text(movies[index].title),  
        subtitle: Text(movies[index].year),  
        leading: Image.network(movies[index].posterUrl),  
        trailing: Icon(Icons chevron_right),  
    ); // ListTile  
},
```

Change the[“”] notation by using . notation

```
ListView.builder(  
    itemCount: _movies.length,  
    itemBuilder: (context, index) {  
        return ListTile(  
            title: Text(_movies[index].title),  
            subtitle: Text(_movies[index].year),  
            leading: Image.network(_movies[index].poster),  
            trailing: Icon(Icons chevron right),
```

Revision to retrieve data from TextField

```
var searchEditingController = TextEditingController();
```

...

```
TextField(
```

```
    controller: searchEditingController,
```

```
    decoration:
```

```
        InputDecoration(hintText: "Enter movie to search"),
```

```
),
```

...

```
fetchMovies(searchEditingController.text).then((val)=>{
```

...

Retrieve searchText and pass in parameters

```
Future<List<MovieSearch>> fetchMovies(searchText) async {  
  
    final response = await http  
  
        .get(Uri.parse('https://www.omdbapi.com/?s=$searchText&apikey=87d10179'));  
  
  
  
    if (response.statusCode == 200) {  
  
        // If the server did return a 200 OK response,  
  
        // then parse the JSON.  
    }  
}
```

To cater for N/A poster Image

```
ListTile(  
    title: Text(_movies[index].title),  
    subtitle: Text(_movies[index].year),  
    leading: _movies[index].poster == "N/A" ? SizedBox() :  
    Image.network(_movies[index].poster),  
    trailing: Icon(Icons.chevron_right),  
);
```

Revision : pass forward data

1 and 2 (detail.dart)

```
final String imdbId;  
  
DetailPage({required this.imdbId});
```

3) home.dart

```
Navigator.push(context, MaterialPageRoute(builder: (context)=> DetailPage(imdbId:  
_movies[index].imdbId,))));
```

4) (detail.dart)

```
Text("Detail Page for ${widget.imdbId}") ,
```

```
class MovieDetail {  
  
    final String title;  
    final String year;  
    final String genre;  
    final String director;  
    final String plot;  
    final String actors;  
    final String poster;  
    final String writers;  
  
    MovieDetail({this.title, this.year, this.genre, this.director, this.plot, this.actors, this.poster, this.writers});  
  
    factory MovieDetail.fromJson(Map<String, dynamic> json){  
        return MovieDetail(  
            title: json["Title"],  
            year: json["Year"],  
            genre: json["Genre"],  
            director: json["Director"],  
            plot: json["Plot"],  
            actors: json["Actors"],  
            poster: json["Poster"],  
            writers: json["Writers"]  
        );  
    }  
}
```

✓

Flutter Outline

Flutter Inspector

Flutter Performance

MovieDetail code

```
// 1 ) Create the class
```

```
class MovieDetail {
```

```
// 2) Create the properties
```

```
final String title;
```

```
final String year;
```

```
final String released;
```

```
final String genre;
```

```
final String director;
```

```
final String writer;
```

```
final String actors;
```

Method to call API

```
// Future it means that it is an asynchronous method

// {} => ClassName, [] => List<ClassName>

Future<MovieDetail> fetchMovie() async {

    final response = await http

        .get(Uri.parse('https://omdbapi.com/?i=tt1669165&apikey=87d10179'));

    if (response.statusCode == 200) {

        // If the server did return a 200 OK response,
        // then parse the JSON.

        // Call the last method in Class
    }
}
```

initState in Detail page

```
MovieDetail? _movie;  
  
@override  
void initState() {  
    // TODO: implement initState  
    super.initState();  
    // value here is the movie returned and processed from API  
    fetchMovie().then((value) => setState(() {  
        _movie = value;  
    }));  
}
```

UI to show the movie

```
MovieDetail? movie;
```

```
@override
```

```
void initState() {
```

```
// TODO: implement initState
```

```
super.initState();
```

```
fetchMovie().then((val)=>{
```

```
    setState(() {
```

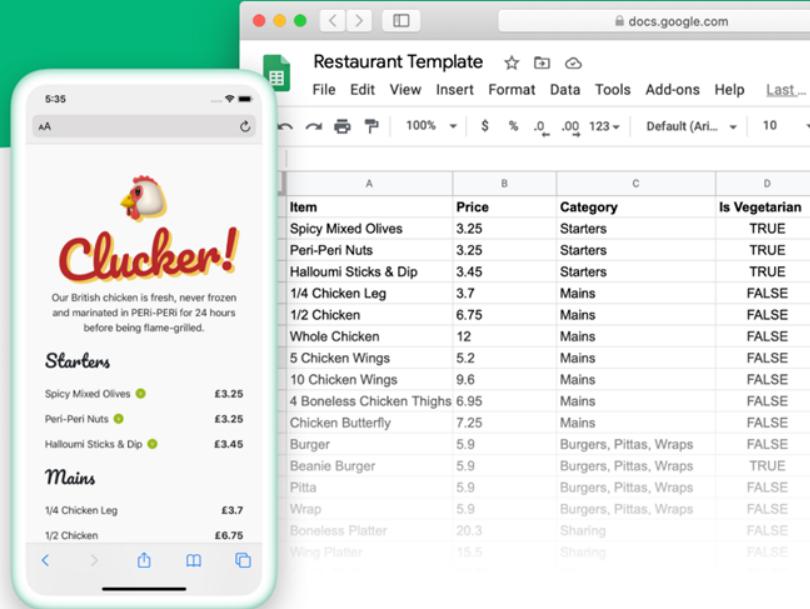
```
        movie=val;
```

```
Column(  
  children: [  
    Text(_movie!.title, style: TextStyle(color: Colors.red, fontSize: 32), textAlign: TextAlign.center,),  
    SizedBox(height: 8,),  
    Image.network(_movie!.poster),  
    SizedBox(height: 8,),  
    Text(_movie!.plot, textAlign: TextAlign.center),  
,  
)
```

Experiences from sheets

Thousands of people are using Sheety to turn their spreadsheets into powerful APIs to rapidly develop prototypes, websites, apps and more.

 Connect Google Sheet



The image shows a composite interface. On the left, a mobile phone screen displays a restaurant menu titled "Clucker!" with a chicken icon. The menu lists items like Spicy Mixed Olives, Peri-Peri Nuts, Halloumi Sticks & Dip, 1/4 Chicken Leg, 1/2 Chicken, Whole Chicken, 5 Chicken Wings, 10 Chicken Wings, 4 Boneless Chicken Thighs, Chicken Butterfly, Burger, Beanie Burger, Pitta, Wrap, Boneless Platter, and Wing Platter, each with a price. On the right, a Google Sheets document titled "Restaurant Template" is shown in a browser window. It contains a table with columns for Item, Price, Category, and Is Vegetarian. The data matches the items listed on the mobile app.

Item	Price	Category	Is Vegetarian
Spicy Mixed Olives	3.25	Starters	TRUE
Peri-Peri Nuts	3.25	Starters	TRUE
Halloumi Sticks & Dip	3.45	Starters	TRUE
1/4 Chicken Leg	3.7	Mains	FALSE
1/2 Chicken	6.75	Mains	FALSE
Whole Chicken	12	Mains	FALSE
5 Chicken Wings	5.2	Mains	FALSE
10 Chicken Wings	9.6	Mains	FALSE
4 Boneless Chicken Thighs	6.95	Mains	FALSE
Chicken Butterfly	7.25	Mains	FALSE
Burger	5.9	Burgers, Pittas, Wraps	FALSE
Beanie Burger	5.9	Burgers, Pittas, Wraps	TRUE
Pitta	5.9	Burgers, Pittas, Wraps	FALSE
Wrap	5.9	Burgers, Pittas, Wraps	FALSE
Boneless Platter	20.3	Sharing	FALSE
Wing Platter	15.5	Sharing	FALSE

Uber

SONOS

JUST EAT

ByteDance

Lemonade



Vox



Import < POST http:// | POST http:// [CONFLICT] | POST http:// | GET http:// | POST https:// | POST https: | GET http:// | GET http:// | PUT http:// | GET http:// | > +

HTTP https://rest-api-places.onrender.com/api/places Save

POST <https://api.sheetly.co/4db58997dd33ab7eaa3d621c48bdea06/employees/sheet1> Send ▾

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary **JSON** ▾ Beautify

```
1 {
2   "sheet1": {
3     "name": "Muzaffar",
4     "email": "wanmuz86@gmail.com",
5     "phonenumber": "012356789",
6     "address": "Bandar Baru Bangi"
7   }
8 }
```

Body Cookies Headers (15) Test Results 200 OK 1039 ms 991 B Save Response ▾

Pretty Raw Preview Visualize JSON ▾ 🔗 🔍

```
1 {
2   "sheet1": {
3     "name": "Muzaffar",
4     "email": "wanmuz86@gmail.com",
5     "phonenumber": "012356789"
```

API Created with Sheety

Get all -

<https://api.sheety.co/4db58997dd33ab7eaa3d621c48bdea06/employees/sheet1>

POST -

<https://api.sheety.co/4db58997dd33ab7eaa3d621c48bdea06/employees/sheet1>

```
{  
  "sheet1":{  
    "name":"Muzaffar",  
    "email":"wanmuz86@gmail.com",  
    "phonenumber":"012356789",  
    "address":"Bandar Baru Bangi"  
  }  
}
```

Verify your data here : <https://docs.google.com/spreadsheets/d/1Iu9acLL8G8WYNPXB9mKm-5ILXiVzh8wuRrx11iHtMFg/edit#gid=0>

UI For POST exercise

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

void main() {
    runApp(const MyApp());
}

class MyApp extends StatelessWidget {
    const MyApp({super.key});

    // This widget is the root of your application.

    @override
    Widget build(BuildContext context) {
```

Sending Data to Internet

<https://docs.flutter.dev/cookbook/networking/send-data>

- 1) Add http in pubspec.yaml

```
http: ^1.1.0
```

Method to call API

```
import 'package:http/http.dart' as http;

import 'dart:convert';

Future<http.Response> createEmployee(String name, String email, String address, String phonenumber) {

    return http.post(
        Uri.parse('https://api.sheety.co/4db58997dd33ab7eaa3d621c48bdea06/employees/sheet1'),
        // To edit the headers.. If you need to manage authentication

        // Normally we need to modify and add Authorization and token

        headers: <String, String>{
            'Content-Type': 'application/json; charset=UTF-8',
        },
        body: jsonEncode({
            'name': name,
            'email': email,
            'address': address,
            'phone': phonenumber
        })
    );
}
```

Calling the API on onPressed

```
ElevatedButton(onPressed: () {  
  
    createEmployee(nameEditingController.text, emailEditingController.text, addressEditingController.text,  
  
        phoneEditingController.text).then((value) => print(jsonEncode(value.body)));  
  
}, child: Text("Add Employee"))
```

Using Flutter with Firebase



Firebase



email

password

Login

No account? Register now!

Forgot your password

email

password

Register

email

Reset password

Login code

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

class LoginPage extends StatelessWidget {
  const LoginPage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Login")),
      body: Center(
        child: Column(
          children: [
            TextField(decoration: InputDecoration(hintText: "Enter email")),
            TextField(decoration: InputDecoration(hintText: "Enter password"), obscureText: true),
            ElevatedButton(onPressed: (){}, child: Text("Login")),
            TextButton(onPressed: (){
              Navigator.push(context, MaterialPageRoute(builder: (context)=> RegisterPage()));
            },child: Text("No account? Register now!"),
            TextButton(onPressed: (){
              Navigator.push(context, MaterialPageRoute(builder: (context)=>ForgotPage()));
            },child: Text("Forgot Password"),
          ],
        ),
      ),
    );
  }
}
```

Register code

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

class RegisterPage extends StatelessWidget {
  const RegisterPage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Register")),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Center(
          child: Column(
            children: [
              TextField(decoration: InputDecoration(hintText: "Enter email")),
              TextField(decoration: InputDecoration(hintText: "Enter password"), obscureText: true),
              ElevatedButton(onPressed: () {}, child: Text("Register")),
            ],
          ),
        ),
      ),
    );
  }
}
```

Forgot Password

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

class ForgotPage extends StatelessWidget {
  const ForgotPage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Forgot Password")),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Center(
          child: Column(
            children: [
              TextField(decoration: InputDecoration(hintText: "Enter email")),
              ElevatedButton(onPressed: () {}, child: Text("Forgot Password")),
            ],
          ),
        ),
      ),
    );
  }
}
```

This will go to user
profile

AAA
marketing

BBB
IT

CCC
IT

Enter message

Send

AAA - 12.15 pm
Jom Makan

BBB- 12.15 pm
Jom! Makan mana?

HomePage

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

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {

  var _employees = [
    {
      "name": "Clint",
      "department": "Marketing"
    },
    {
      "name": "Alice",
      "department": "I.T."
    },
    {
      "name": "John",
      "department": "I.T."
    }
];
  @override
  Widget build(BuildContext context) {
    return Scaffold(

```

ChatPage

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

class DetailPage extends StatefulWidget {
  const DetailPage({Key? key}) : super(key: key);

  @override
  State<DetailPage> createState() => _DetailPageState();
}

class _DetailPageState extends State<DetailPage> {

  var _messages = [
    {
      "sender": "John",
      "receiver": "Alice",
      "message": "Let's have lunch",
      "dt": "1/10/03 10.30"
    },
    {
      "sender": "John",
      "receiver": "Alice",
      "message": "Are you there?",
      "dt": "1/10/03 10.33"
    }
  ];
}
```

App logic/ automation/ business rules

When Do ...

When a **new order is made (Firestore/ Realtime Database)**

- Send an email confirmation to the user (3rd party provider)
- Send a push notification/ ring the merchant (Cloud Notification)
- Send a push notification to rider who wants to accept the job (Cloud Notification)

Cloud Messaging

- 1) Send to all users
- 2) Segmentized based on preset value - ios/ android -> version of phone, version of app - out of box
- 3) Segmentized based on the value that I define (topics) - need to define and do some code
- 4) Individual

Go to <https://firebase.google.com/> and create new project

×

Create a project (Step 1 of 3)

Let's start with a name for
your project?

Project name

chat-app

 chat-app-ff3ea

Continue





Firebase

Project Overview



Build

Authentication

Cloud Firestore

Realtime Database

Storage

Hosting

Functions

Machine Learning

Release & Monitor

Crashlytics

Performance

Test Lab

App Distribution

Extensions

Spark

Free \$0/month

Upgrade

chat-app ▾

Go to docs



Receive email updates about new Firebase features, research, and events

Sign up



chat-app

Spark plan

Get started by adding Firebase to your app



Add an app to get started



Store and sync app data in milliseconds



Download the file and move it as per instruction

x Add Firebase to your Android app

1 Register app
Android package name: my.mimos.chatapp

2 Download config file

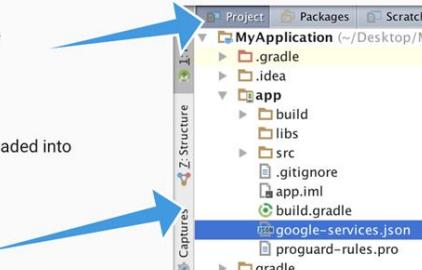
[Download google-services.json](#)

Instructions for Android Studio below | [Unity](#) [C++](#)

Switch to the Project view in Android Studio to see your project root directory.

Move the google-services.json file you just downloaded into your Android app module root directory.





[Next](#)

Android -> app -> build.gradle

The screenshot shows a code editor interface with a dark theme. On the left is a file tree for a Flutter project named 'chatapp' located at '/Documents/fluttermimos/chatapp'. The 'build.gradle' file under the 'app' directory is selected and highlighted with a blue bar at the top of the code area. The code itself is a Gradle configuration file for an Android application.

```
Flutter commands
compileSdkVersion 29
sourceSets {
    main.java.srcDirs += 'src/main/kotlin'
}
lintOptions {
    disable 'InvalidPackage'
}
defaultConfig {
    // TODO: Specify your own unique Application ID (https://developer.android.com/studio/bundle#gradle-build-file-application-id)
    applicationId "my.mimos.chatapp"
    minSdkVersion 16
    targetSdkVersion 29
    versionCode flutterVersionCode.toInt()
    versionName flutterVersionName
}
buildTypes {
    release {
        // TODO: Add your own signing config for the release build.
    }
}
```

The code includes several TODO comments for specifying application ID, min SDK version, target SDK version, and signing configurations for the release build.

Get `firebase_core`, `firebase_auth` and `cloud_firestore` library from pub.dev

A screenshot of the pub.dev search interface. The search bar at the top contains the query "firebase core". Below the search bar, there are filters for "Platforms" (Android, iOS, Linux, macOS, Web, Windows) and "SDKs" (Flutter, Platform, Android, iOS, macOS, Web, Windows). The search results show 144 packages. The first result is `firebase_core`, which is described as a "Flutter plugin for Firebase Core, enabling connecting to multiple Firebase apps." It has 2707 likes, 140 pub points, and 100% popularity. The second result is `flutter_firebase_chat_core`, described as "Actively maintained, community-driven Firebase BaaS for chat applications with an optional chat UI." It has 234 likes, 140 pub points, and 94% popularity.

firebase core

Platforms

- Android
- iOS
- Linux
- macOS
- Web
- Windows

RESULTS 144 packages

SORT BY SEARCH RELEVANCE

firebase_core

Flutter plugin for Firebase Core, enabling connecting to multiple Firebase apps.

v 2.13.0 (5 days ago) [firebase.google.com](#) BSD-3-Clause [Flutter Favorite](#) Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS MACOS WEB WINDOWS

API result: [firebase_core/firebase_core-library.html](#)

2707 LIKES 140 PUB POINTS 100% POPULARITY

SDKs

License

Advanced

flutter_firebase_chat_core

Actively maintained, community-driven Firebase BaaS for chat applications with an optional chat UI.

v 1.6.7 (5 days ago) [flyer.chat](#) Apache-2.0 Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS MACOS WEB

234 LIKES 140 PUB POINTS 94% POPULARITY

Import firebase_core & auth, firestore

```
26  # consider running `flutter pub upgrade --major-versions`. Alternatively,
27  # dependencies can be manually updated by changing the version numbers below to
28  # the latest version available on pub.dev. To see which dependencies have newer
29  # versions available, run `flutter pub outdated`.
30  dependencies:
31    flutter:
32      sdk: flutter
33
34
35    # The following adds the Cupertino Icons font to your application.
36    # Use with the CupertinoIcons class for iOS style icons.
37    cupertino_icons: ^1.0.2
38    firebase_core: ^2.13.0
39    firebase_auth: ^4.6.1
40    cloud_firestore: ^4.7.1
41
42  dev_dependencies:
43    flutter_test:
44      sdk: flutter
45
46  # The "flutter_lints" package below contains a set of recommended lints to
47  # encourage good coding practices. The lint set provided by the package is
```

Configuration Firebase (subject to version number)

```
# The following adds the Cupertino Icons font to your application.  
# Use with the CupertinoIcons class for iOS style icons.  
cupertino_icons: ^1.0.2  
firebase_core: ^2.15.0  
firebase_auth: ^4.7.2  
cloud_firestore: ^4.8.4
```

Initialize Firebase app (add this in main.dart file)

```
import 'package:firebase_core/firebase_core.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp();

  runApp(const MyApp());
}
```

Problem that commonly found and resolved abstractly.
Each language and framework can resolve the problem
with it's own implementation

MVC - Semua kod bercampur di dalam satu fail
(spagethi code)

Model - Related to data

View -> UI element

Controller - Logic of application

Programming language

(Javascript, Python, Dart, Java ..)

Framework

Django

Laravel

Spring
Boot

Ruby on
Rails

Extra configuration within android folder

- 1) Add multiDexEnabled true inside **app level build.gradle** (refer next page) - around line 54
- 2) Inside **app level build.gradle** add following code at the last line - apply plugin: 'com.google.gms.google-services'
- 3) Inside **android level build.gradle** add around line 11

```
classpath 'com.google.gms:google-services:4.3.3'
```

- 1) Set minSdkVersion to 19 inside **app level build.gradle**

```
minSdkVersion 19
```

Flutter commands

```
sourceSets {  
    main.java.srcDirs += 'src/main/kotlin'  
}  
  
lintOptions {  
    disable 'InvalidPackage'  
}  
  
defaultConfig {  
    // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/signed-apk.html#signing)  
    applicationId "my.mimos.chatapp"  
    minSdkVersion 16  
    targetSdkVersion 29  
    versionCode flutterVersionCode.toInt()  
    versionName flutterVersionName  
    multiDexEnabled true  
}  
  
buildTypes {  
    release {  
        // TODO: Add your own signing config for the release build.  
        // Signing with the debug keys for now, so `flutter run --release` works.  
        signingConfig signingConfigs.debug  
    }  
}  
android{ } > defaultConfig{ }
```

Open for Edit

Flutter commands

```
45     versionName flutterVersionName
46     multiDexEnabled true
47 }
48
49 buildTypes {
50     release {
51         // TODO: Add your own signing config for the release build.
52         // Signing with the debug keys for now, so `flutter run --release` works.
53         signingConfig signingConfigs.debug
54     }
55 }
56
57
58 flutter {
59     source '../..'
60 }
61
62 dependencies {
63     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
64
65 }
66 apply plugin: 'com.google.gms.google-services'
```

Flutter commands

```
3     repositories {
4         google()
5         jcenter()
6     }
7
8     dependencies {
9         classpath 'com.android.tools.build:gradle:3.5.0'
10        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
11        classpath 'com.google.gms:google-services:4.3.3'
12    }
13
14
15    allprojects {
16        repositories {
17            google()
18            jcenter()
19        }
20    }
21
22    rootProject.buildDir = '../build'
23    subprojects {
24        project.buildDir = "${rootProject.buildDir}/${project.name}"
25    }

```

```
    }

    defaultConfig {
        // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).
        applicationId "com.example.chat_app"
        // You can update the following values to match your application needs.
        // For more information, see: https://docs.flutter.dev/deployment/android#reviewing-the-gradle-build-configuration.
        minSdkVersion 19
        targetSdkVersion flutter.targetSdkVersion
        versionCode flutterVersionCode.toInt()
        versionName flutterVersionName
        multiDexEnabled true
    }

    buildTypes {
        release {
```

Firebase Authentication

- 1) Import firebase_auth library:

```
import 'package:firebase_auth/firebase_auth.dart';
```

- 1) Initialize Firebase Authentication Instance

```
FirebaseAuth _auth = FirebaseAuth.instance;
```

- 1) Add code to createUser (as per next page) when register Button is pressed

```
1 import 'package:firebase_auth/firebase_auth.dart';
2 import 'package:flutter/material.dart';
3
4 class RegisterPage extends StatefulWidget {
5     @override
6     _RegisterPageState createState() => _RegisterPageState();
7 }
8
9 class _RegisterPageState extends State<RegisterPage> {
10
11     var emailController = TextEditingController();
12     var passwordController = TextEditingController();
13
14     FirebaseAuth _auth = FirebaseAuth.instance;
15     @override
16     Widget build(BuildContext context) {
17         return Scaffold(
18             appBar: AppBar(title: Text("Register")),
19             body: Padding(
20                 padding: const EdgeInsets.all(8.0),
21                 child: Column(
22                     children: [
23                         Text("Email"),
24                         TextFormField(controller: emailController),
25                         Text("Password"),
26                         TextFormField(controller: passwordController),
27                         Text("Confirm Password"),
28                         TextFormField(),
29                         ElevatedButton(onPressed: () {
30                             Navigator.pushNamed(context, "/login");
31                         }, child: Text("Register"))
32                     ],
33                 ),
34             ),
35         );
36     }
37 }
```

Code to createUser

```
ElevatedButton(onPressed: () {  
  
    _auth.createUserWithEmailAndPassword(email: emailEditingController.text,  
        password: passwordEditingController.text).then((val) {  
  
        if (val != null) {  
  
            // TODO replace with toast  
  
            print("Successfully registered");  
  
        }  
  
        // TODO replace with toast  
  
        else {  
  
            print("Something is wrong");  
  
        }  
    });  
});
```

```
  appBar: AppBar(
    title: const Text("Register"),
  ), // AppBar
  body: Column(
    children: [
      TextField(
        decoration:
          const InputDecoration(hintText: "Enter email address"),
        controller: emailEditingController,
      ), // TextField
      const TextField(
        decoration: InputDecoration(hintText: "Enter password"),
        obscureText: true,
      ), // TextField
      TextButton(
        onPressed: () async {
          User? user = (await _auth.createUserWithEmailAndPassword(
            email: emailEditingController.text,
            password: passwordEditingController.text))
            .user;
          if (user != null) {
            print("Successfully registered");
          } else {
            print("Something went wrong");
          }
        },
        child: const Text("Login") // TextButton
      ],
    ); // Column, Scaffold
  },
}
```

Go to Firebase console and enable Email authentication

The screenshot shows the Firebase Authentication interface for a project named "chat-app". The "Sign-in method" tab is selected. A modal dialog is open, titled "Sign-in providers". It contains two provider configurations:

- Email/Password:** This section includes an icon of an envelope, the text "Email/Password", and an "Enable" toggle switch which is turned on (blue). Below this, a descriptive text states: "Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives." followed by a "Learn more" link.
- Email link (passwordless sign-in):** This section includes an icon of an envelope, the text "Email link (passwordless sign-in)", and an "Enable" toggle switch which is turned off (gray).

At the bottom right of the modal, there are "Cancel" and "Save" buttons. The "Save" button is highlighted with a blue background.

 **Firebase**

chat-app ▾

Go to docs  

Project Overview 

Build

 **Authentication**

 **Cloud Firestore**

 **Realtime Database**

 **Storage**

 **Hosting**

 **Functions**

 **Machine Learning**

Release & Monitor

 **Crashlytics**

 **Performance**

 **Extensions**

Spark
Free \$0/month 



google-services....json   

Failed to start monitoring emulator-5554 (12 minutes ago) 49 chars 11:57 LF UTF-8 4 spaces P masterervi

Authentication

Users Sign-in method Templates Usage

 Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication [Get started](#) 

 Search by email address, phone number, or user UID   

Identifier	Providers	Created	Signed In	User UID ↑
wanmuz86@gmail.com		Mar 10, 2021	Mar 10, 2021	A6gKMEDVzTMzInt6fc2LBnpyg272

Rows per page: 50  1 – 1 of 1  

Exercise

- Add code for error handler
- Add code to show success upon registration
- Redirect user to login page

Firebase Authentication (Login)

- 1) Import firebase_auth library:

```
import 'package:firebase_auth/firebase_auth.dart';
```

- 1) Initialize Firebase Authentication Instance

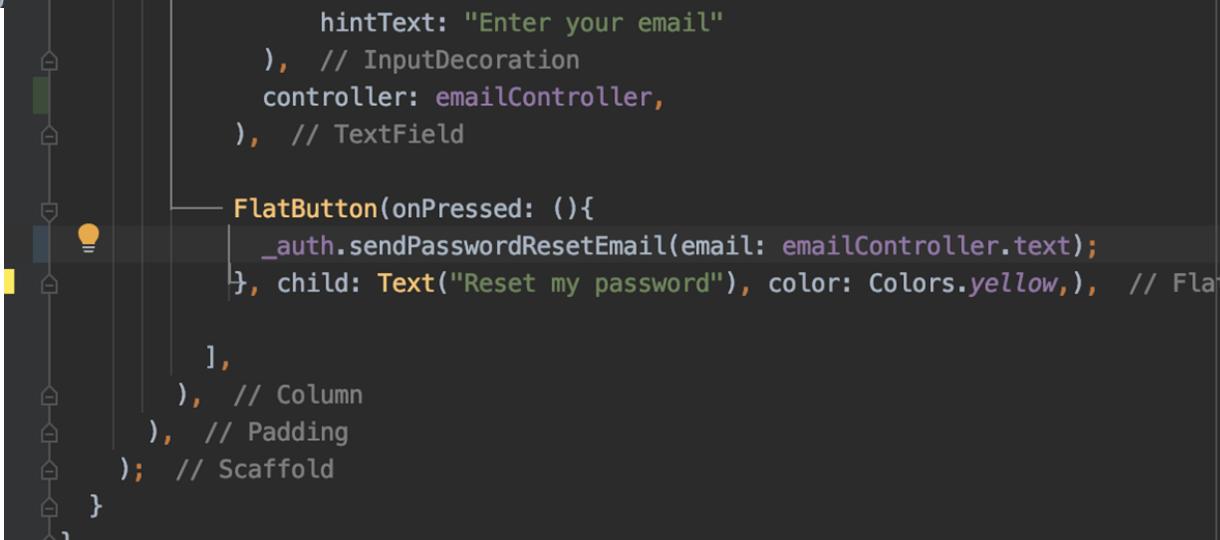
```
FirebaseAuth _auth = FirebaseAuth.instance;
```

- 1) Add code to signInWithEmailAndPassword (as per next page)
when login Button is pressed

Sign in code

```
_auth.signInWithEmailAndPassword(email: emailEditingController.text, password:  
passwordEditingController.text)  
  
.then((val){  
  
if (val != null){  
  
Navigator.push(context,  
  
MaterialPageRoute(builder: (context)=> HomePage()));  
  
}  
  
else {  
  
print("Something is wrong");  
  
}  
  
});
```

```
    TextButton(  
        onPressed: () async {  
            User? user = (await _auth.signInWithEmailAndPassword(  
                email: emailEditingController.text,  
                password: passwordEditingController.text))  
                .user;  
            print(user);  
            if (user != null) {  
                print("Successfully logged in!");  
                Navigator.push(  
                    context,  
                    MaterialPageRoute(  
                        builder: (context) => const ListPage())); // MaterialPageRoute  
            } else {  
                print("Error");  
            }  
        },  
        child: const Text("Login"), // TextButton
```

```
9
10
11 class _ResetpasswordPageState extends State<ResetpasswordPage> {
12     FirebaseAuth _auth = FirebaseAuth.instance;
13     var emailController = TextEditingController();
14     @override
15     Widget build(BuildContext context) {
16         return Scaffold(
17             appBar: AppBar(title: Text("Reset Password"),),
18             body: Center(

                child: Column(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: [
                        Padding(
                            padding: const EdgeInsets.all(16.0),
                            child: TextField(
                                decoration: InputDecoration(
                                    hintText: "Enter your email"
                                ),
                                controller: emailController,
                            )
                        ),
                        FlatButton(
                            onPressed: () {
                                _auth.sendPasswordResetEmail(email: emailController.text);
                            },
                            child: Text("Reset my password", style: TextStyle(color: Colors.yellow)),
                        )
                    ],
                ),
            );
        }
    }
}
```



Realtime updates, powerful queries, and automatic scaling

Create database

Secure rules for Cloud Firestore

2 Set Cloud Firestore location

Your location setting is where your Cloud Firestore data will be stored.

⚠ After you set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.

[Learn more](#)

Cloud Firestore location

asia-southeast2

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

Cancel

Enable

Watch later Share

upgrade



How much will Cloud Firestore cost?

[View pricing](#)

Firestore setup

The screenshot shows the Google Cloud Platform (GCP) interface for setting up a new Cloud Firestore database. The left sidebar lists various project settings like Authentication, Firestore, and Monitoring. The main area is titled "Cloud Firestore" and displays a "Create database" wizard.

Create database

1 Secure rules for Cloud Firestore 2 Set Cloud Firestore location

After you define your data structure, **you will need to write rules to secure your data.** [Learn more](#)

Start in **production mode**
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

Start in **test mode**
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2021, 4, 10);
    }
  }
}
```

! The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

Cancel **Next**

How much will Cloud Firestore cost? [View pricing](#)

SQL vs NoSQL

A table in relational database -> A collection in NoSQL

A row in relational database -> A document in NoSQL

A column -> A key in NoSQL

```
1 import 'package:firebase_auth/firebase_auth.dart';
2 import 'package:flutter/material.dart';
3 import 'package:cloud_firestore/cloud_firestore.dart';
4
5
6 class RegisterPage extends StatefulWidget {
7     @override
8     _RegisterPageState createState() => _RegisterPageState();
9 }
```

```
    .user,
10 if (user != null) {
11     FirebaseFirestore.instance.collection('users').doc(user.uid).set({
12         'email': user.email,
13         'id': user.uid,
14         'createdAt': DateTime.now(),
15         'chattingWith': null
16     });
17 } else {
18     print("Something went wrong");
19 }
20 },
```

Save the user info inside our database

```
_auth.createUserWithEmailAndPassword(email: emailEditingController.text,  
password: passwordEditingController.text).then((val){  
if (val != null){  
// TODO replace with toast  
  
// val is the response from the server, inside val there is a property user which  
// stores user information from firebase authentication  
  
User user = val.user!;  
  
// I save (set) the information inside firestore , collection users  
  
FirebaseFirestore.instance.collection('users').doc(user.uid).set(  
{
```

1) Create the userId and constructor, I need to know who is currently logged in

```
class ListPage extends StatelessWidget {  
  
    final String userId;  
    ListPage({super.key, required this.userId})  
        .user;  
  
    if (user != null) {  
  
        Navigator.push(  
            context,  
            MaterialPageRoute(  
                builder: (context) => ListPage(userId: user.uid,)); // MaterialPageRoute  
    } else {  
        // Handle user not found or other error  
    }  
}
```

2) Get the data and show it in the ListView

```
StreamBuilder<QuerySnapshot>(
    stream: FirebaseFirestore.instance.collection('users').snapshots(),
    builder:(context,snapshot) {
        if (!snapshot.hasData){
            return Center(child: CircularProgressIndicator());
        }
        else {
            final List<DocumentSnapshot> documents = snapshot.requireData.docs;
            return ListView.builder(
                itemCount: documents.length,
                itemBuilder: (context,index){
                    return documents[index]["id"] == widget.currentUserId ? Container(
                        child: ListTile(
                            title: Text(documents[index]["email"]),
                            subtitle: Text(documents[index]["department"]),
                            trailing: Icon(Icons.chevron_right),
                            onTap: (){
                                Navigator.push(context,
                                    MaterialPageRoute(builder: (context)=>DetailPage())
                                );
                            },
                        ),
                    );
                },
            );
        }
    }
)
```

<https://dev.to/kazuhideoki/get-data-from-firebase-and-show-it-in-listview-on-flutterbuilder-or-streambuilder-e05>

3) Pass the userId from Login page to Employee List

```
        password: passwordController.text))
    .user;
if (user != null) {
    print("User successfully logged in");
    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) => EmployeelistPage(userId: user.uid)); // MaterialPageRoute
    } else {
    print("error");
}
},
child: Text("Login"),
color: Colors.yellow,
) // FlatButton
```

4) Create an onTap that will open the new page

```
        else {
            return ListTile(
                title: Text(snapshot.data.docs[position]["email"]),
                trailing: Icon(Icons.keyboard_arrow_right),
                onTap: () {
                    Navigator.push(context, MaterialPageRoute(builder: (builder)=>ChatPage(userId:widget.userId,
                        friendId:snapshot.data.docs[position]["id"])));
                } // ChatPage, MaterialPageRoute
            );
        } // ListTile
    }
}

import 'package:flutter/material.dart';

class ChatPage extends StatefulWidget {

    final String userId;
    final String friendId;
    ChatPage({this.userId, this.friendId});
    @override
    _ChatPageState createState() => _ChatPageState();
}
```

1) Create the channel ID

```
];
@Override
Widget build(BuildContext context) {

    // 1) Create the chat room code
    if (widget.userId.hashCode < widget.friendId.hashCode){
        groupChatId = '${widget.userId}-${widget.friendId}';
    }
    else {
        groupChatId = '${widget.friendId}-${widget.userId}';
    }
    return Scaffold(
```

groupChatId - Room ID

John 1

John log in

Alice

Alice 2

Alice 1-2

1-2

Wan 3

Wan 1-3

James 4

James 1-4

2) Code to chat (save/pass data to firebase)

```
Row(  
    children: [  
        Expanded(  
            child: TextField(  
                controller: messageController,  
                decoration: InputDecoration(  
                    hintText: "Enter your message"  
                ), // InputDecoration  
            ), // TextField  
        ), // Expanded  
        FlatButton(onPressed: (){  
  
            FirebaseFirestore.instance.collection('messages')  
                .doc(groupId)  
                .collection(groupId)  
                .doc(DateTime.now().microsecondsSinceEpoch.toString())  
                .set({  
                    'idFrom':widget.userId,  
                    'idTo':widget.friendId,  
                    'timestamp':DateTime.now().microsecondsSinceEpoch.toString(),  
                    'content':messageController.text  
                })  
        })  
    ]  
)
```

```
var groupChatId = "";

if (widget.friendId.hashCode < widget.userId.hashCode) {

    groupChatId = '${widget.friendId}-${widget.userId}';

}

else {

    groupChatId = '${widget.userId}-${widget.friendId}';

}

FirebaseFirestore.instance.collection('messages').doc(groupChatId).collection(groupChatId)

.doc(DateTime.now().microsecondsSinceEpoch.toString()).set({

'idFrom':widget.userId,

'idTo':widget.friendId,
```

Cloud Firestore

[Data](#) [Rules](#) [Indexes](#) [Usage](#)

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication

[Get started](#)  [messages > wphSSVSNMddAhwqcuDyPVfq3CXc2 > wphSSVSNMddAhwqcuDyPVfq3CXc2 > 161546783872015](#) [wphSSVSNMddAhwqcuDyPVfq3CXc2](#) [Start collection](#) [wphSSVSNMddAhwqcuDyPVfq3CXc2](#) > [wphSSVSNMddAhwqcuDyPVfq3CXc2](#) [Add document](#)

1615467799078105

 [1615467838720015](#) [Start collection](#) [Add field](#)

content: "Are you there?"

idFrom: "wphSSVSNMddAhwqcuDyPVfq3CXc2"

idTo: "eA9mulvEl2eZvJ0D6dZYQCrkK983"

timestamp: "1615467838720075"

 [Add field](#)

5) Retrieve the chat, and show the message

```
        ),
    // Row
    Expanded(
        child: StreamBuilder(
            stream: FirebaseFirestore.instance.collection('messages')
                .doc(groupChatId)
                .collection(groupChatId).snapshots(),
            builder: (context, snapshots){
                if (!snapshots.hasData){
                    return CircularProgressIndicator();
                }
                else {
                    return ListView.builder(
                        itemCount: snapshots.data.docs.length,
                        itemBuilder: (context, position){
                            return ListTile(
                                title: Text(snapshots.data.docs[position]["idFrom"]),
                                subtitle: Text(snapshots.data.docs[position]["content"]))
                            ); // ListTile
                }
            );
        } // ListView.builder
    }
}
```

```
Expanded(  
  
    child: StreamBuilder(  
  
        stream:FirebaseFirestore.instance.collection('messages') .  
  
        doc(groupChatId)  
  
        .collection(groupChatId).snapshots(),  
  
        builder: (context, snapshot) {  
  
            if (!snapshot.hasData) {  
  
                return Center(child: CircularProgressIndicator());  
  
            }  
  
            else {  
  
                final List<DocumentSnapshot> documents = snapshot.requireData.docs;
```

/main



/detail



6) Modify your code to show the sender email ...

Later verify with my source code

Inside profile, you need current UserId, create the textediting controller as well

```
class ProfilePage extends StatefulWidget {  
  
    final String userId;  
    ProfilePage({this.userId});  
    @override  
    _ProfilePageState createState() => _ProfilePageState();  
}  
↑
```

```
class _ProfilePageState extends State<ProfilePage> {
```

```
    var emailController = TextEditingController();  
    var nameController = TextEditingController();
```

```
    @override
```

This is the code to get the data and assign it to the controller

```
}

// Get the user Info from firebase
void retrieveUserInfo() async {
    FirebaseFirestore.instance.collection('users').doc(widget.userId).get().then((ds){
        if (ds.exists){
            setState(() {
                nameController.text = ds.data()["name"];
                emailController.text = ds.data()["email"];
            });
        }
    });
}
```

Call the method created on initState

```
var nameController = TextEditingController(),  
  
@override  
void initState() {  
  
    super.initState();  
    this.retrieveUserInfo();  
}  
  
@override
```

```
47           hintText: "Phone Number"
48           ), // InputDecoration
49           ), // TextField
50           TextField(
51             maxLines: 3,
52             decoration: InputDecoration(
53               hintText: "Address"
54             ), // InputDecoration
55             ), // TextField
56             FlatButton(onPressed: (){
57
58               FirebaseFirestore.instance.collection('users').doc(widget.userId).update({
59                 'email':emailController.text,
60                 'name':nameController.text,
61               });
62
63               }, child: Text("Update User")) // FlatButton
64             ],
65           ), // Column
66           ), // Padding
67           ), // SingleChildScrollView
```

- 1) Complete the rest code for update profile, include phone number and address profile update (compulsory)
- 2) Add a View Profile page, for example on the top right of chat pagem you can have a view profile page that will bring you to your friend's profile page showing his or her info

If you want to do more, use this tutorial, which I simplified to teach you:

<https://medium.com/flutter-community/building-a-chat-app-with-flutter-and-firebase-from-scratch-9eaa7f41782e>

More info on Streambuilder

<https://flutter.dev/docs/development/ui/widgets/async>

Cloud Storage for Firebase

Store and serve content with ease

[Visit console](#)[View docs](#)

Store your users' photos and videos

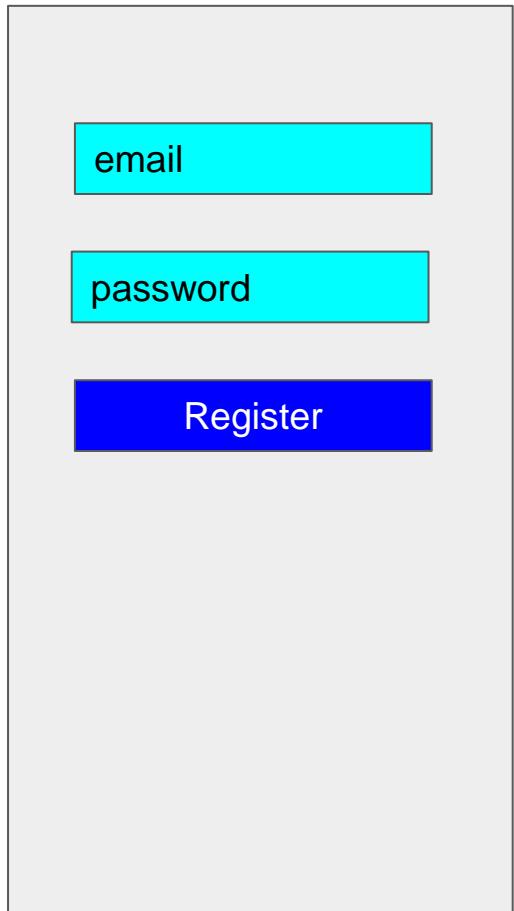
Cloud Storage is designed to help you quickly and easily store and serve user-generated content, such as photos and videos.

[iOS+ C++](#) 

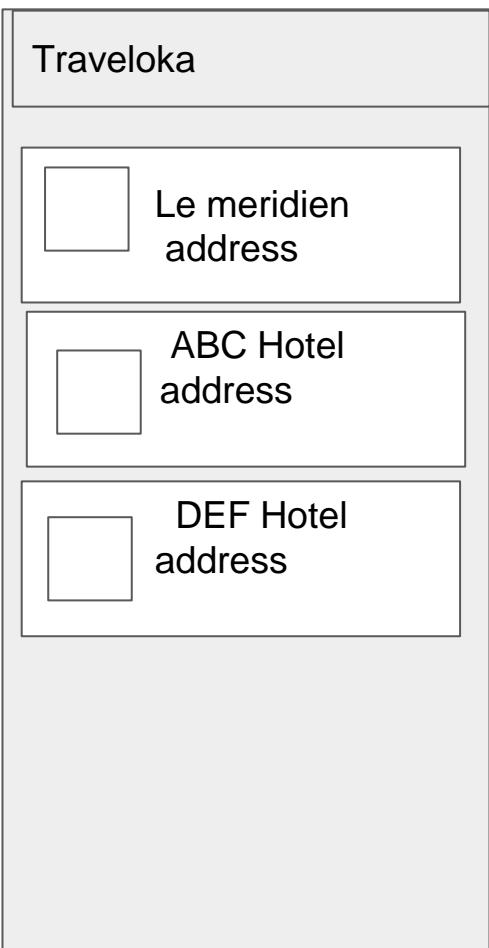
Upload to FirebaseStorage

```
firebase_storage.FirebaseStorage storage = firebase_storage.FirebaseStorage.instance;  
  
Upload to FirebaseStorage  
  
final ref = firebase_storage.FirebaseStorage.instance  
  
.ref('files')  
  
.child(destination);  
  
await ref.putFile(_photo!);  
  
print(ref.fullPath);
```

Stless (register.dart)



Stful (home.dart)



Stful (detail.dart)
Detail



If you
you c
setSt
the U

- 1) Add http package inside pubspec.yaml
- 2) Create the Model of our response
 - a) Create the class
 - b) Create the property
 - c) Create the constructor
 - d) Create the fromJson method
 - e) Create Array to List transformer (if applicable)
- 3) Create function to call the API inside widgets/home.dart
- 4) Call the function

<https://rest-api-places-9pgy.onrender.com/api/places>

If you have a List of data, need this transformer

```
static List<MovieSearch> moviesFromJson(dynamic json ) {  
    var searchResult = json["Search"];  
    List<MovieSearch> results = List.empty(growable: true);  
  
    if (searchResult != null){  
  
        searchResult.forEach( (v)=>{  
            results.add(MovieSearch.fromJson(v))  
        });  
        return results;  
    }  
    return results;  
}
```

```
FROM php:8.2.1-fpm

USER root

WORKDIR /var/www/html

RUN apt-get update && apt-get install -y \
libpng-dev \
zlib1g-dev \
libxml2-dev \
```

Data type in Javascript - JS, Python, PHP, PERL

String

```
var movie
```

Number

```
fetch(xxxxx).then(val=>{
```

Boolean

```
    movie = val;  
})
```

Array

Object

null / undefined

```
var myname
```

Data type in Javascript

String

```
var movie
```

Number

```
fetch('xxxxx').then(val=>{
```

Boolean

```
    movie = val;  
})
```

Array

Object

null / undefined

```
var myname
```

Dart - Type Script, Java, C++, Swift, Typedscript

String.. String name = “Walid”

Int, double, long , float - 1,2,3... , 2.3, longer int ,
longer double

2^16 - 65535 , or -32768 until -32768

long 2^32 ... ,

Boolean

List<>.. (Array)

Map -> Object

In Dart and Swift, the concept of null is represented with Optional

Normally, in JS we declare null as follows:

```
var name
```

```
print(name)
```

// I declare a string or null datatype variable
(optional)

String? Name

Int? age

An exclamation is a way for me to retrieve
the non null value of an optional datatype

dartpad.dev

```
void main() {  
  
  // 1) Implicit (imply = guessing) and Explicit declaration
```

```
    var name = "Muzaffar";
```

```
    // Implicit declaration => The variable will be
```

```
    // String based on the initial value
```

```
    print(name);
```

```
  var age = 30;
```

```
  // Implicit declaration => The variable will be integer
```

```
  // Based on the initial value
```

Source code

First day : <https://github.com/wanmuz86/hello-world-sept23>

BMI Calc: https://github.com/wanmuz86/bmi_calc_sep23

To do app: https://github.com/wanmuz86/todos_sept23.git

API Call (Movie) - <https://github.com/wanmuz86/omdb-api-sept23>

Camera code - https://github.com/wanmuz86/camera_demo_sept23

API Call - POST with Sheety : <https://github.com/wanmuz86/http-post-silverlake-oct>

Data Type explanation : https://github.com/wanmuz86/dart_datatype240323/tree/main

Optional explanation:

https://github.com/wanmuz86/optional_dart_aug/blob/main/index.dart

Open source code from git

- Unzip
- Open Project
- Open the folder
- Run pub get (pubspec.yaml)
- Run the project

Dartpad (<https://dartpad.dev/>)

```
void main() {
```

```
// 1) Variable declaration
```

```
double weight = 170; // Explicit declaration
```

```
print(weight);
```

```
var bmi = 0;
```

```
print(bmi); // Implicit declaration ,
```

```
// Dart imply that the variable type is integer
```

```
// Dart is a safe type language
```

```
// int + double == (wrong)
```

```
// We saw the example with bmi Formula
```

Optional

```
void main() {  
  
    // In Dart when we declare a variable normally  
  
    // It CANNOT be null  
  
    // If I declare a variable as null, syntax error  
  
    // A normal Variable in Dart needs to have a value  
  
  
  
    String name = "Muzaffar";  
  
    print(name);  
  
  
  
    // Load the page  
  
    // We retrieve data from the API
```

QnA :Themes and Design in Flutter

<https://codelabs.developers.google.com/codelabs/mdc-101-flutter>

<https://codelabs.developers.google.com/codelabs/mdc-102-flutter>

<https://codelabs.developers.google.com/codelabs/mdc-103-flutter> (Theming)

<https://codelabs.developers.google.com/codelabs/mdc-104-flutter>

QnA : Conditional Rendering on Style

```
_bmi != 0 ? Text("Your BMI is ${_bmi}", style: _bmi < 18 ?  
smallTextStyle()  
  
: _bmi < 25 ? normalTextStyle() : bigTextStyle(),)  
  
: const SizedBox(),
```

QnA : Camera

Use `image_picker`: https://pub.dev/packages/image_picker

Image will be returned as `Xfile`, transform to bytes

```
import 'package:image_picker/image_picker.dart';

ElevatedButton(onPressed: () async{

final picker = ImagePicker();

final pickedImage = await picker.pickImage(source: ImageSource.gallery);

if (pickedImage != null) {

var imageBytes = await pickedImage.readAsBytes();

setState(() {
```

- 1) Get the library from pub.dev
 - 2) There is a “Instance”/ Manager -> ImagePicker() / Geolocator
 - 3) Call the function upon the instance/manager , eg: pickImage,
getCurrentPosition
 - 4) You will receive something from the manager: retrieved the value and
process, Xfile or Position
-
- Normally you need to set permission to use User device features

Android -> app -> src -> main -> AndroidManifest.xml

my_camera

Project

Resource Manager

Emulator: Pixel 2 API 29

Flutter commands

Open for Editing in Android Studio Hide

```
17 <meta-data
18     android:name="io.flutter.embedding.android.NormalTheme"
19     android:resource="@style/NormalTheme"
20   />
21   <intent-filter>
22     <action android:name="android.intent.action.MAIN"/>
23     <category android:name="android.intent.category.LAUNCHER"/>
24   </intent-filter>
25 </activity>
26   <!-- Don't delete the meta-data below.
27       This is used by the Flutter tool to generate GeneratedPluginRegistrant.java -->
28   <meta-data
29     android:name="flutterEmbedding"
30     android:value="2" />
31   </application>
32   <uses-permission android:name="android.permission.INTERNET"/>
33 </manifest>
```

manifest uses-permission

Camera page

Get from photo library

Run: main.dart

Adding Permission in Android

The screenshot shows an Android Studio project titled "my_camera". The Project structure on the left includes files like .gradle, .idea, android [my_camera_android], app, build.gradle, gradle, .gitignore, build.gradle, gradle.properties, gradlew, gradlew.bat, local.properties, my_camera_android.iml, and settings.gradle. The main focus is on the "main\AndroidManifest.xml" tab in the code editor. The XML code includes meta-data for themes and intent-filters for the MAIN and LAUNCHER categories. A comment indicates it's used for GeneratedPluginRegistrant. Below this, there's a manifest node containing an application node with an INTERNET permission. The code editor has tabs for manifest and uses-permission. On the right, the Emulator shows a smartphone displaying a camera interface with a photo library.

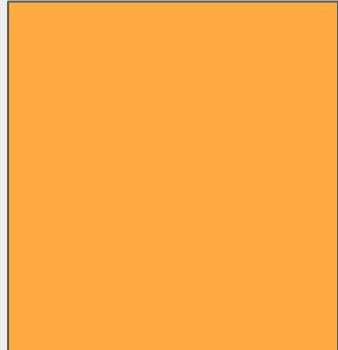
```
<meta-data
    android:name="io.flutter.embedding.android.NormalTheme"
    android:resource="@style/NormalTheme"
/>
<intent-filter>
    <action android:name="android.intent.action.MAIN"/>
    <category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>

<meta-data
    android:name="flutterEmbedding"
    android:value="2" />
</application>
<uses-permission android:name="android.permission.INTERNET" />
</manifest>
```

Emulator: Pixel 2 API 29

Camera page

Get from photo library



Get image from gallery

Get image from
camera

Upload profile image

Bring out the image

```
children: <Widget>[
    if (_imageBytes != null)
        Image.memory(
            _imageBytes!,
            width: 200,
            height: 200,
        )
    else
        Text('No image selected'),
        SizedBox(height: 20),
        ElevatedButton(
            onPressed: loadImage,
            child: Text('Select Image'),
        ),
],
```

Checklist for iOS submission

- 1) Add permission in Info.plist
- 2) Change Icon in Assets (use makeappicon/ <https://www.appicon.co/> to facilitate change of logo)
- 3) Change Launch Image in Assets
- 4) Give the app proper version name and code
- 5) CHange the app id
- 6) Sign in into iOS Teams in XCode
- 7) Archive the code

Checklist in iTunesConnect

- 1) Create an Application
- 2) Fill in the metadata
- 3) Prepare screenshots and logos
- 4) Upload on Xcode and find it in Binary
- 5) Go through app submission list , rejection reason and HIG
 - a) <https://developer.apple.com/app-store/review/guidelines/>
 - b) <https://developer.apple.com/design/human-interface-guidelines>

```
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ), // ThemeData
      home: HomePage(),
    ); // MaterialApp
  }
}
```

DEVELOPER PROGRAM

iOS Developer Program

Individual

\$99 / Year

For an individual developer who will be creating iOS apps for distribution on the App Store.

iOS Developer Program

Company

\$99 / Year

For a company with a development team who will be creating iOS apps for distribution on the App Store.

Note: A D-U-N-S Number is required.

iOS Developer Enterprise Program

\$299 / Year

For a company who will be creating proprietary, in-house iOS apps.

Note: A D-U-N-S Number is required.

iOS Developer University Program

Free

For higher education institutions looking to introduce iOS development into their curriculum.



App Submission Question

- Who pays the app?
- What is the business model?
- Who is the user?
- Is this an internal app?
- How much do you charge?
- What is the paid function within the app?
 - They want to determine do you need an enterprise or company account?
 - They want you to go for in app purchase vs paying outside the app

<https://www.youtube.com/watch?v=j9i7zMKkN-w>

Expand your scope

Use the following structure for your project

1) Title + One Liner

2) Problem

3) Solution

4) Traction

5) Business Model

6) Market

7) Competition

8) Vision [1 year plan]

9) Team / Who am I

10) Use of funds [N/A]

Title

Startup Name

Founders Names

Date

Taglines (if any)



Presented by:

Wan Muzaffar Bin
Wan Hashim



Anak2U

Easy Education through tech

How to create a one liner

“I am, _(insert name of company)_, is developing _(a defined offering)_ to help _(a defined audience)_ _(solve a problem)_ with _(secret sauce)_”.

I am from Anak2U, developing a mobile application to help teacher to reduce teachers workload by automating and digitalizing their day to day task.

CUSTOMER AVATAR

GOALS AND VALUES

Goals:

Values:

Age:

Gender:

Marital Status:

#/Age of Children:

Location:



CHALLENGES & PAIN POINTS

Challenges:

Pain points:

SOURCES OF INFORMATION

Books:

Magazines:

Blogs/Websites:

Conferences:

Gurus:

Other:

Quote:

Occupation:

Job Title:

Annual Income:

Level of Education:

Other:

OBJECTIONS & ROLE IN PURCHASE PROCESS

Objections to the sale:

Role in the Purchase Process:

CUSTOMER AVATAR

GOALS AND VALUES

Goals:

To understand the basic words for 6 years old based on kspk..

Values:

Age: 5 years old

Gender: boy

Marital Status: -

#/Age of Children: -

Location: Kuantan



SOURCES OF INFORMATION

Books:

Magazines:

Blogs/Websites:

Conferences:

Gurus:

Other:

Quote:

Occupation:

Job Title:

Annual Income:

Level of Education:

Other:

CHALLENGES & PAIN POINTS

Challenges:

Pain points:
Cannot concentrate learning with books
Prefer learn using video & music
Like to play games

OBJECTIONS & ROLE IN PURCHASE PROCESS

Objections to the sale:

Role in the Purchase Process:

Problem

- Why is this problem important to the world?
- Why this problem is important to you, the founder? Why are you passionate about this problem?

Problem

2

Price is an important concern for customers booking travel online.

Hotels leave you disconnected from the city and its culture.

No easy way exists to book a room with a local or become a host.

Problem

- The **amount of time** that teacher need to take to do their day to day compulsory task (2 hours)
- The use of Whatsapp group is a norm and it **is not effective** for managing class.
- Sharing of pictures and videos activities of students are done in Social Media or Whatsapp group which **has privacy and security issue**.

Solution

The solution needs to be concise and very clear. Especially if you are a tech startup, your solution needs to be scalable. Scalability is the capability of a system to increase its total output under an increased load when resources are added.

Is this a “painkiller” or a “vitamin”?

Showcase the product (screenshot)

PAINKILLER



VITAMIN



"Need to have"	"Nice to have"
Solves unmet customer needs	Improves an existing solution
Want to use	Should use
Google Maps Amazon Echo Lyft/Uber	To-Do List Apps Fitness Trackers Nextdoor

Solution

3

A web platform where users can rent out their space to host travelers to:

**SAVE
MONEY**

when traveling

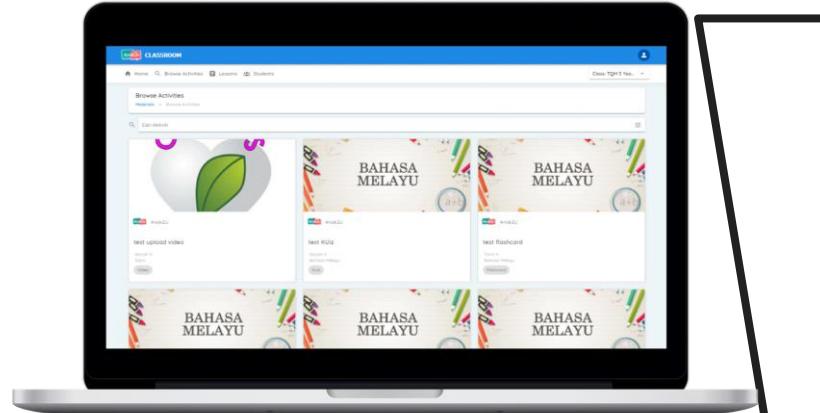
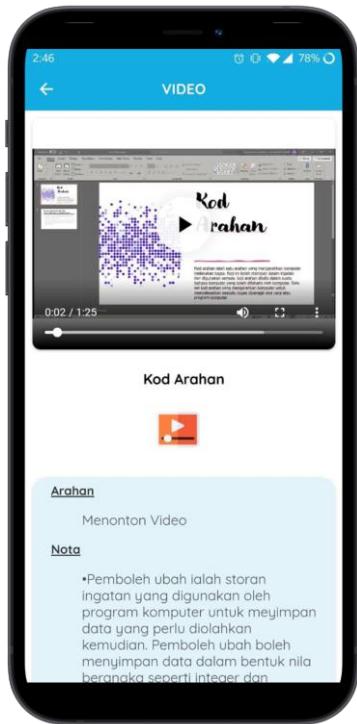
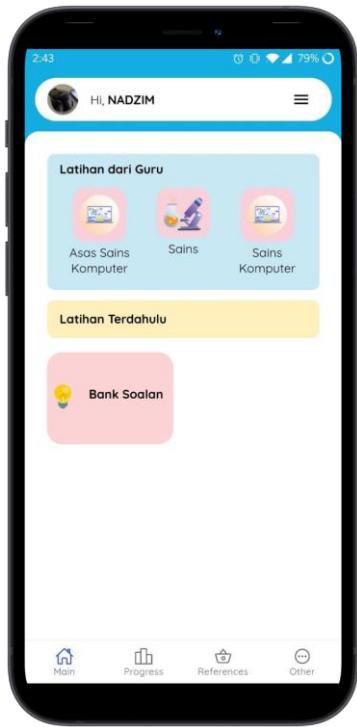
**MAKE
MONEY**

when hosting

**SHARE
CULTURE**

local connection to the city

Anak2U Classroom



Traction (Important slide)

Growth chart since your product launch

- One Metrics that matters
- Monetary value
- Show change over time (graph)

Traction

- 800 Paying Users
- \$150,000 annual revenue run rate
- 97% margins
- 55,000 users, growing 40% per month
- 1.5 million updates Buffered

Transaction for FYP

Is it on Google Play or App Store?

How many downloads within xxx months / weeks

Week by week chart of download

Is it deployed website?

How many people visiting your website?

How many people registered?

How many people purchased something on your website

Unit Economics & Business Model

Here's how I'm going to make money
(now or future)

Subscription? One time? Per transaction?

Here's how the math makes sense

Customer Acquisition Cost <<(3) Customer Lifetime Value

- Eg: If you create an ads, you spend RM300 to get 5 customers one month

Business Model

App

- Consumable product : In app purchase - you buy, you use and it's gone..
Gem in mobile is a consumable product
- Non consumable product : In app purchase - you buy, you use and you can use it forever - In game, you might be able to purchase hero, or level
- Recurring - You pay monthly and it is auto renewable - Netflix, spotify, zoom
- Non Recurring - You pay for a period and then it is gone..
- Freemium - Some features are free, some you need pay.. [Astro Go/Njoi], Evernotes, Zoom



What are a good business model for a IT quiz app targeted to high school and university students



A potential business model for an IT quiz app targeting high school and university students could include:



1. Freemium Model: Offer a basic version of the app for free, with limited features, and charge a fee for advanced features such as more questions, analytics, and personalized study plans.
2. Subscription Model: Offer a subscription-based model where users pay a monthly or annual fee to access the app's full range of features, including access to new content and updates.
3. In-app Advertising: Incorporate ads into the app to generate revenue from advertisers who are interested in targeting the high school and university student demographic.
4. Sponsorship: Partner with IT companies or organizations to sponsor quizzes, challenges, or scholarships. This can also help in branding and awareness.
5. Data Licensing: Collect user data and sell insights or anonymized data to third-party companies, which can use the data to improve their marketing efforts or product development.

Ultimately, the best business model will depend on various factors such as user demographics, app features, and competitors. It is important to conduct market research and gather feedback from potential users to determine the best approach for your app.



What are a good business model for a Supir Management app in travel industry targeted to Malaysia and Indonesian market



A potential business model for a driver management app in the travel industry, targeting the Malaysian and Indonesian market, could include:

1. Commission-based Model: Charge a commission fee for each ride booked through the app. The commission rate could be a percentage of the total fare, with a minimum or maximum fee per ride.
2. Subscription Model: Offer subscription-based plans to drivers, where they pay a monthly or yearly fee to access the app's full range of features such as booking management, trip analytics, and customer support.
3. Advertising Model: Incorporate ads into the app and generate revenue from advertisers who are interested in targeting the Malaysian and Indonesian market. This can include local businesses or travel-related companies.
4. Partnership Model: Partner with hotels, airlines, or other travel-related businesses to offer bundled services or discounts to users who book through the app. This can help in cross-promotion and improve user experience.
5. Data Licensing: Collect user data and sell insights or anonymized data to third-party companies, which can use the data to improve their marketing efforts or product development. This can be a potential revenue stream.

It is important to research the market and understand user behavior to determine the best business model for your app. Additionally, consider local laws and regulations, competition, and pricing strategies when selecting a model.



Costing

- Fixed Cost
- Rental?
- Salary?
- Server?
- Computer?
- Marketing?
- Per unit

Costing for an app

Fixed Cost

- Computer? = 7000
- Apple Account = 300
- Google Play = 100

Variable

- Rental = $1000 \times 12 = 12\ 000$
- Salary? = $3000 \times 12 = 36\ 000$
- Server? = $2000 \times 12 = 24\ 000$



The cost breakdown structure for creating a quiz app for high school and university students can vary depending on several factors such as app features, complexity, development platform, and geographical location of the development team. However, here is a general cost breakdown structure that can give you an idea of the potential costs involved in developing such an app:

1. Planning and Design: This includes the cost of market research, ideation, user interface (UI)/user experience (UX) design, and wireframing. The cost can vary between \$2,000 to \$10,000 or more, depending on the extent of the research and design required.
2. Development: This includes the cost of programming, testing, and quality assurance (QA) for the app. The cost can vary based on the app's complexity, development platform, and the number of features required. An estimate for a basic quiz app can range from \$10,000 to \$50,000 or more.
3. Content Creation: This includes the cost of creating and sourcing questions, answers, and other content for the quiz app. The cost can vary depending on the number of questions and the subject matter expertise required. An estimate for a basic quiz app can range from \$2,000 to \$10,000.
4. Server and Hosting: This includes the cost of hosting the app on a server and providing data storage, bandwidth, and security. The cost can vary based on the app's usage and the hosting provider selected. An estimate for a basic quiz app can range from \$2,000 to \$5,000 per year.
5. Marketing and Promotion: This includes the cost of promoting the app through social media, paid ads, content marketing, and other channels. The cost can vary based on the marketing budget and the promotional activities planned. An estimate for a basic quiz app can range from \$5,000 to \$20,000 or more.

Overall, the cost of developing a quiz app for high school and university students can range from \$20,000 to \$100,000 or more, depending on the app's features and requirements. It is essential to conduct thorough research, set clear goals, and work with a reputable development team to launch.

Regenerate response

Market

You can start with the addressable local market size, but need to show the larger ASEAN market roadmap

TRY NOT TO DEFINE UNIVERSITY , PEKAN, UMP AS YOUR MARKET.. THINK HOW CAN YOU CAN DO BEYOND UNIVERSITY

TAM SAM SOM Infographics

\$150M



TAM

Total
Addressable
Market

Mercury is the smallest planet in the Solar System

\$200M



SAM

Service
Addressable
Market

Jupiter is a gas giant and also the biggest planet of them all

\$15M



SOM

Service
Obtainable
Market

Despite being red, Mars is actually a very cold place

Ex SOM SAM and TAM for learning app

SOM - Service Observable Market : Number of 4-6 years old in Kuala Lumpur / Kuantan

SAM - Service Addressable Market: Number of children 4-6 years old in Malaysia- ~ 4 Million

TAM - Total Addressable Market: Number of children in the world that is 4-6 years old who are not from english speaking country, but learning english as second language.

Number of 4-6 years old in South East Asia (Regional)

Asset Management System for University

SOM - MTUN university / University in East Cost / public university
SAM - Number of universities in Malaysia -> How many are there?

TAM - Number of universities in South East Asia -> How many are there?

Private / Public university

Use ChatGPT to help you define your TAM/SAM & SOM



What is the tam, sam and som of turkish language learning app in Malaysia and World



To estimate the TAM, SAM, and SOM for a Turkish language learning app in Malaysia and globally, we need to consider the total addressable market, the serviceable available market, and the serviceable obtainable market.

- Total Addressable Market (TAM): The TAM for a Turkish language learning app in Malaysia would be the total number of people in Malaysia who are interested in learning the Turkish language, regardless of the platform they use. According to the Department of Statistics Malaysia, the population of Malaysia as of 2020 is around 32.7 million people. Assuming that a significant proportion of this population has an interest in learning Turkish, the TAM for your Turkish language learning app in Malaysia could be estimated to be a smaller number compared to the TAM for a more widely spoken language like English or Mandarin.

Globally, the TAM for a Turkish language learning app would be the total number of people worldwide who are interested in learning Turkish. According to Ethnologue, there are approximately 80 million Turkish speakers worldwide. Assuming that a significant proportion of these Turkish speakers are interested in learning the language, the TAM for a Turkish language learning app globally could be estimated to be a smaller number compared to the TAM for a more widely spoken language.

- Serviceable Available Market (SAM): The SAM for a Turkish language learning app in Malaysia would be the number of people in Malaysia who have access to a smartphone, the internet, and are interested in learning the Turkish language through an app. According to Statista, as of 2021, there are approximately 28.7 million smartphone users in Malaysia, and this number is projected to reach 31.3 million by 2025. Assuming that a significant proportion of these users are interested in learning Turkish, the SAM for a Turkish language learning app in Malaysia could be estimated to be around 28.7 million.

Regenerate response

Observation from judging FYP

Student Market Definition:

“This system is made to facilitate lecturer in FK to create bank of questions for their student in programming and it will help autocorrect and ease up teachers burden”

Actual Market

“To start we will test this in FK as pilot, however this system can be scalable to all institution that’s teaching Computer science and Programming, we are talking about xx xxx number of students, in addition, with the implementation of coding in school, it can benefit 15 000 student taking in SPM and xx xxx students in PT3. It can be enhanced to include questions on Science Computer for IGSCE which are around xxx xxxx in the world”

Competition

Why you're better than existing solutions?

Who are your competition?

Two way of showing competitors:

- Table
- Graph

Example Table

Competitive Landscape

	Pinterest	Telly	#waywire	VIDinterest
User Friendly	✓	✗	✗	✓
Privacy	✓	✗	✗	✓
Keyword follow	✗	✗	✗	✓
Playlist follow	✗	✗	✗	✓
Video Bookmarks	✓	✓	✓	✓
Video Autonext	✗	✓	✓	✓
Video Shuffle	✗	✗	✗	✓

Example graph



PRODUCT FEATURES	Sophia.my	Seesaw	Guruberbagi	Anak2U Classroom
Online content activity	✓	✗	✓	✓
Offline content activity	✗	✓	✓	✓
Emphasize 21st way learning	✗	✓	✓	✓
Class Management System	✗	✓	✗	✓
Recommendation & Path Creation	✗	✗	✗	✓

Customer Acquisition

What are the different channels you're going to use to acquire your customers and the costs for each. Are they scalable and sustainable?

19 Key Traction Channels



Targeting Blogs



Publicity/PR



Unconventional
PR



Search Engine
Marketing (SEM)



Social & Display
Ads



Offline Ads



Search Engine
Optimization (SEO)



Content
Marketing



Email
Marketing



Viral
Marketing



Engineering
as Marketing



Business
Development (BD)



Sales



Affiliate
Programs



Existing
Platforms



Trade Shows



Offline Events



Speaking
Engagements



Community
Building



READINGGRAPHICS
Ideas Come Alive

How to create “funded” project

- Select a SDG Goals and link it to your project
- Use 10 x 10 Matrix , select 1 domain and 1 technology of focus to be used in your project
- Select a domain or topic of research and focus on this.



SUSTAINABLE DEVELOPMENT GOALS

17 GOALS TO TRANSFORM OUR WORLD

1 NO
POVERTY



2 ZERO
HUNGER



3 GOOD HEALTH
AND WELL-BEING



4 QUALITY
EDUCATION



5 GENDER
EQUALITY



6 CLEAN WATER
AND SANITATION



7 AFFORDABLE AND
CLEAN ENERGY



8 DECENT WORK AND
ECONOMIC GROWTH



9 INDUSTRY, INNOVATION
AND INFRASTRUCTURE



10 REDUCED
INEQUALITIES



11 SUSTAINABLE CITIES
AND COMMUNITIES



12 RESPONSIBLE
CONSUMPTION
AND PRODUCTION



13 CLIMATE
ACTION



14 LIFE
BELOW WATER



15 LIFE
ON LAND



16 PEACE, JUSTICE
AND STRONG
INSTITUTIONS



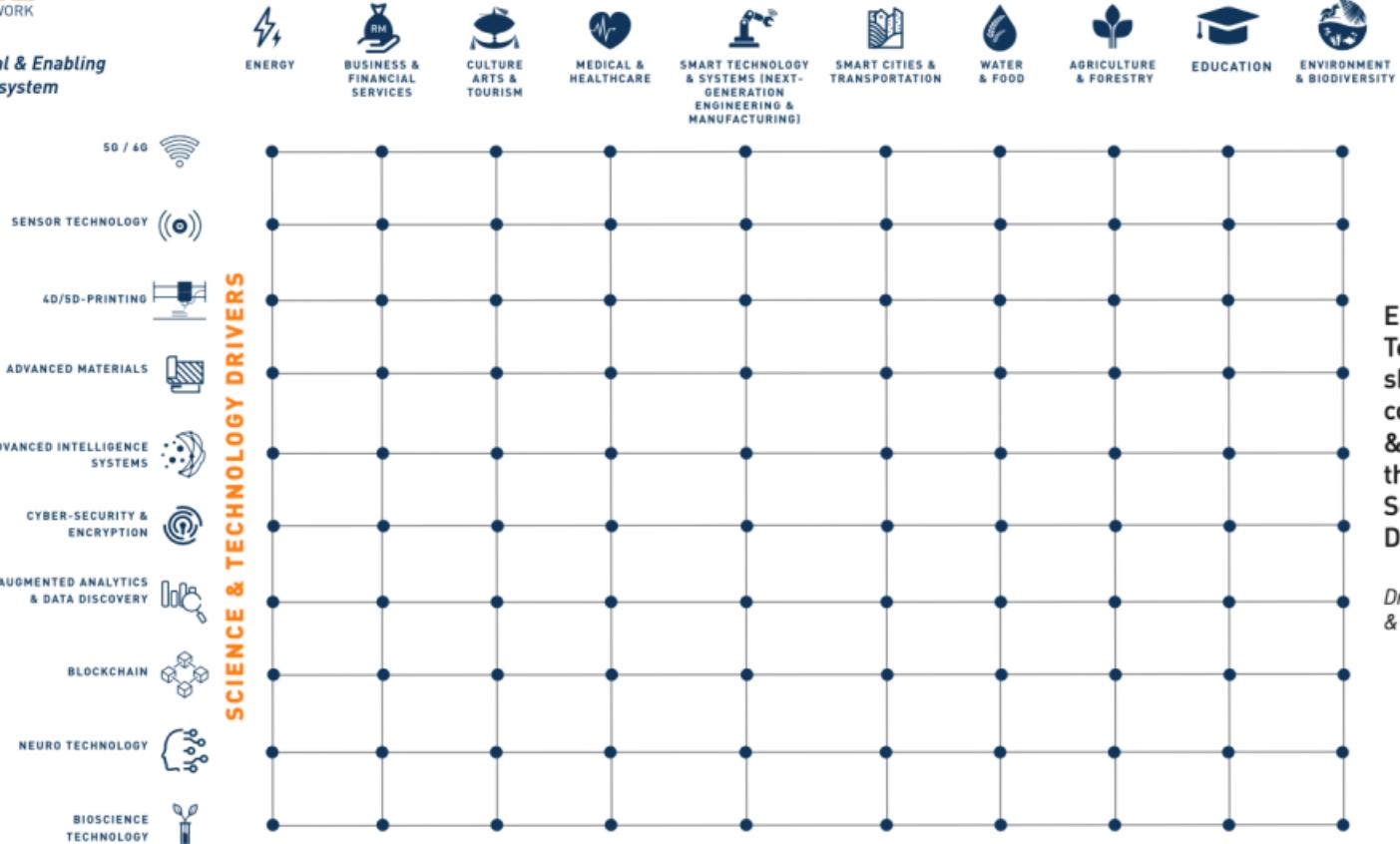
17 PARTNERSHIPS
FOR THE GOALS




SUSTAINABLE
DEVELOPMENT
GOALS

Integrating the Horizontal & Enabling Vertical in the Ecosystem

MALAYSIAN SOCIO-ECONOMIC DRIVERS



Each Science & Technology Driver should explore core technologies & applications for the 10 Malaysian Socio-economic Drivers

Driving Fundamental & Translational Research

Each Malaysian Socio-economic Driver should explore how the 10 Science & Technology Drivers will value-add and enhance their global competitiveness

What is considered the base of a PTA

- A system (app or web)
- Complete CRUD Operation
- Complete Authorization Operation
- Shows a scenario with different database relation: One to One, One to Many, Many to Many
- More than 5 database tables
- Plus point : Integration with API, eg: Google Maps, Google Text to Speech, Payment Provider etc

What is considered the base of a PSM

- An overall system (backend + web/app) - Frontend and backend
- More than 10 tables
- To show integration with other provider, eg: Google Maps, Google Text to Speech, Payment Provider etc
- Focus on one research topic , eg: IOT, Security, Gamification, Machine Learning
- Built with best practice : TDD, CI, CD, Architecture principle
- You may do “Improvement” project but need to do comparison of before and after the new project

Observation from previous FYP / PTA

- Understand what is a multi tenant system
- Understand what is a dynamic content, avoid hard coded everything inside an application.
- Do not do something that Chat GPT can do, eg To Do App
- Do not try to do everything , focus on one module that is significant enough as a research topic.
- Understand your target audience and competition. Do something that is in the “Blue Ocean”
- Use framework such as Bootstrap, Laravel, Django, Flutter, React

<https://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/>

-Status Points

-Badges (Achievement Symbols)

-Leaderboards (Earned Lunch)

-Leaderboard

-Progress Bar

-Quest Lists

-Dessert Oasis

-High Five

-Crowning

-Anticipation Parade

-Aura Effect

-Step-by-Step Overlay Tutorial

-Boss Fights

-Exchangeable Points

-Virtual Goods

-Build from Scratch

-Alfred Effect

-Collection Sets

-Avatar

-Protector Quest

-Pet Companion

-Observer Attachment

-Appointment Dynamics

-Magnetic Caps

-Dangling

-Prize Pacing

-Bootleg Quest

-Last Mile Drive

-Count Down Timer

-Torture Breaks

-Moats

-The Big Burn

-Narrative

-Elitism

-Humanity Hero

-Revealed Heart

-Beginners Luck

-Free Lunch

-Cap Switcher

-Co-creationist

-Milestone Unlocks

-Real-Time Control

-Chain Combos

-Dynamic Feedback

-Boosters

-Blank Fills

-Plant Pickers

-Poison Pickers

-Attribute Web Chart

-Double Edged Sword

-Friending

-Social Treasure/Gifting

-SeeSaw Bump

-Group Quests

-Trophy Shelf

-Brag Button

-Water Cooler

-Conformity Anchors

-Mentorship

-Social Prod

-Glowing Choice

-MiniQuests

-Visual Storytelling

-Easter Eggs

-Random Rewards

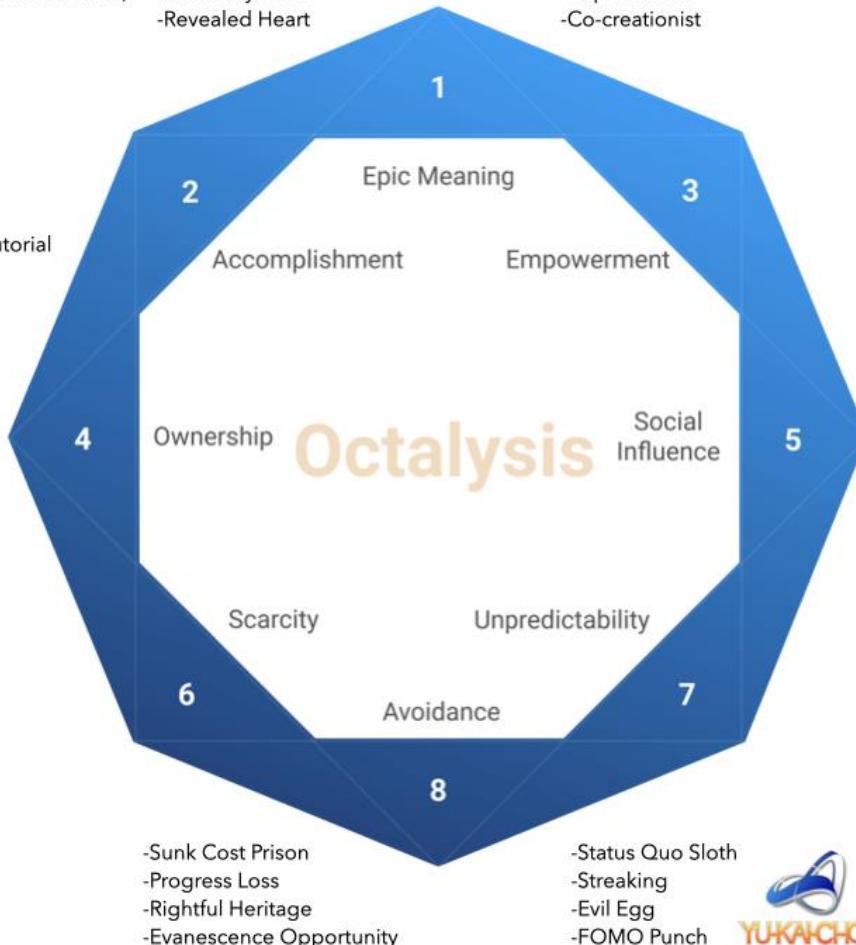
-Obvious Wonder

-Rolling Rewards

-Evolved UI

-Sudden Rewards

-Oracle Effect



Some of the challenge in Education Technology

- To build a quiz app which uses **NLP/Open AI** to mark answer when students answering subjective/open ended question in Malay Language
- Learn a language app for Malaysian/Indonesian user follows **CEFR standard.**
- Learn a language app for Malaysian/Indonesian that will provide vocabulary and text to read using **Space Repetition Principle.**
- Student data analytics platform: Create a software system that allows educators to collect, analyze, and visualize data on student performance, engagement, and behavior, to inform decision-making and improve teaching practices. (**Data Analytics**)
- Adaptive learning system: Design and develop a software system that uses



Here are some potential project ideas for computer science focusing on disabled people:



1. Augmented and Virtual Reality for Rehabilitation: Develop augmented and virtual reality applications that can help disabled individuals with rehabilitation and therapy. These apps can simulate real-world scenarios and provide a safe and controlled environment for therapy sessions.
2. Assistive Technologies: Develop assistive technologies that can help disabled individuals perform everyday tasks such as typing, browsing the internet, and communicating. Examples of such technologies include speech recognition software, eye-tracking devices, and exoskeletons.
3. Accessibility Testing and Auditing: Develop tools that can test the accessibility of websites and applications for disabled users. These tools can provide feedback on accessibility issues and suggest solutions for improvement.
4. Cognitive Assistance: Develop AI-powered tools that can assist people with cognitive disabilities such as autism and dementia. These tools can provide reminders, prompts, and assistance with decision making.
5. Smart Home Automation: Develop smart home systems that can be controlled using voice commands, gestures, or other assistive technologies. These systems can help disabled individuals with home automation tasks such as turning on lights, adjusting the thermostat, and locking doors.

Overall, computer science has the potential to significantly impact the lives of disabled individuals, and these project ideas are just a few examples of the many opportunities available. It's important to conduct thorough research and work with disabled individuals and experts in the field to ensure the projects are practical, effective, and have a positive impact on the disabled community.



Here are some potential project ideas for computer science focusing on the Islamic market:



1. Halal Certification and Verification: Develop a blockchain-based system that can track and verify the halal status of food, cosmetics, and other consumer goods. This can help ensure that products meet the strict halal requirements and regulations, and increase consumer confidence.
2. Islamic Finance Management: Develop a financial management system that adheres to Islamic principles of finance, such as interest-free lending, profit-sharing, and ethical investments. This can help Muslim investors manage their finances in accordance with their religious beliefs.
3. Islamic Education Technology: Develop educational technology applications that provide Islamic education and resources to Muslim students and adults. This can include Quran learning apps, Islamic history and culture games, and interactive prayer guides.
4. Hajj and Umrah Assistance: Develop applications that assist Muslim pilgrims during their Hajj and Umrah journeys. This can include travel planning tools, interactive maps of holy sites, and real-time prayer reminders.
5. Islamic Social Networking: Develop social networking platforms that cater to the needs of Muslim communities. These platforms can include features such as prayer group finders, halal restaurant locators, and forums for discussing Islamic topics.

Overall, computer science has the potential to significantly impact the Islamic market, and these project ideas are just a few examples of the many opportunities available. It's important to conduct thorough research and work with Islamic experts and leaders in the field to ensure the projects are practical, effective, and have a positive impact on the Muslim community.

Grant & Investment Opportunity





Anak2U Sdn Bhd

189A Jalan 8/1

Bandar Baru Bangi

43650 Bandar Baru Bangi

anak2uofficial@gmail.com