Flutter Course

***Coding Club, IIT Guwahati***

**Head:** *Varenya Tiwari (AficionadoXD#7443)*

**Coordinators:** *Pranav Nair(Pranav#7114) , Soumadip Das(SoumadipDas#4773), Harsh Bhakkad(The\_Lord\_Harsh#4494), Mansi (Mansi#6161), Aman Kumar(Aman Kumar#7963), Pranshu Pandya, Pratyush Ranjan, Shikhar Bind, Vignesh Deshpande(Delta Proxim#7619), Vishwesh S Warrier(Whatever00#8650)*

**WEEK 2**

**Getting started with Flutter UI**

It’s rightly said ‘the first impression is the last impression’. Similarly, for every app, its UI/UX is the first impression to the user. Thus, as a developer, it remains vital to power your apps with the best possible UI modeled with your design. Luckily, for us, Flutter provides us with in-built Material Component widgets that make UI/UX and designing much easier for Flutter developers.

* **Images & Assets, Buttons & Icons, Container & Padding, Rows, Columns, Expanded:**
  + [Flutter tutorial for beginners](https://www.youtube.com/watch?v=Hxh6nNHSUjo&list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ&index=8): Videos 8-14
* **Snackbar:**
  + [Brief Video Description](https://www.youtube.com/watch?v=zpO6n_oZWw0&list=PLjxrf2q8roU23XGwz3Km7sQZFTdB996iG&index=69)
  + [Official Documentation](https://api.flutter.dev/flutter/material/SnackBar-class.html)
  + Try implementing your own snackbar with the above 2 links first. It’s okay to copy the code as long as you understand. In case, you fail to make your own snackbar, keep calm and watch this [video description](https://www.youtube.com/watch?v=VlqKaAz8c1A)
* **ListTile and ListView**
  + [ListTile Brief Video Description](https://www.youtube.com/watch?v=l8dj0yPBvgQ&list=PLjxrf2q8roU23XGwz3Km7sQZFTdB996iG&index=54)
  + [ListTile Official Documentation](https://api.flutter.dev/flutter/material/ListTile-class.html)
  + ListTiles are very useful material widgets and easy to implement ones. Try out your own ListTile, won't take long
  + In case you need to display multiple ListTiles or multiple other widgets, use an in-built widget called ListView. It can be both static and dynamic. We shall dive deep in dynamic later in this course
  + [ListView Video Description](https://www.youtube.com/watch?v=KJpkjHGiI5A&list=PLjxrf2q8roU23XGwz3Km7sQZFTdB996iG&index=53)
  + [ListView Official Documentation](https://api.flutter.dev/flutter/widgets/ListView-class.html)
  + [Video Description 1](https://www.youtube.com/watch?v=Lf9DwdVbBuM)
  + [Video Description 2](https://www.youtube.com/watch?v=W-0eeS97xV4)
* **GridView:**
  + [Brief Video Description](https://www.youtube.com/watch?v=bLOtZDTm4H8&list=PLjxrf2q8roU23XGwz3Km7sQZFTdB996iG&index=89)
  + [Official Documentation](https://api.flutter.dev/flutter/widgets/GridView-class.html)
* **Scrolling:**
  + There are innumerable options to make your app scrollable. Many of which are given [here](https://flutter.dev/docs/development/ui/widgets/scrolling). However, the most simplest of all is SingleChildScrollView. Try it out.
* **TextField:**
  + [Video Description](https://www.youtube.com/watch?v=YGTCP5AsaGM)
  + [Official Documentation](https://api.flutter.dev/flutter/material/TextField-class.html)
  + See the various properties of textfield from the documentation and give a try
  + (Optional) As a challenge, if interested, try decorating a text field.

**Navigation in Flutter**

Every modern-day app consists of several pages. Navigation in Flutter exactly helps us by linking a widget to navigate from one page to another

* **Navigator**
  + [Video Description](https://www.youtube.com/watch?v=nyvwx7o277U)
  + [Navigator.push() and Navigator.pop() documentation and example implementation](https://flutter.dev/docs/cookbook/navigation/navigation-basics)
  + [Named routes](https://flutter.dev/docs/cookbook/navigation/named-routes)
  + [Advanced options and implementations](https://flutter.dev/docs/cookbook/navigation)
* **Bottom Navigation Bar:**
  + [Video Description](https://www.youtube.com/watch?v=elLkVWt7gRM)
  + [Official Documentation](https://api.flutter.dev/flutter/material/BottomNavigationBar-class.html)

**Custom Widget in Flutter**

Many times in Flutter, you require a certain combination of widgets again and again in your app. In such cases, one obvious solution is to copy and paste those widgets in all required places. But a good programmer/developer follows DRY principle i.e. Don't Repeat Yourself. So what next?

In case, the widget to be reused is exactly ditto everywhere, Flutter provides an option to refactor widgets by extracting them.

Simply hover the cursor over the widget, then in more options choose Refactor Widget / Extract Widget.

However, in case the widget to be reused requires subtle changes when used in different places in the app, we create a custom widget.

We do so by creating a new class that extends a stateless/stateful/any suitable class, creating appropriate constructors, and then defining the widget.

Here’s a [video description](https://www.youtube.com/watch?v=CyKFCLtOb_s&t=122s)

**Importing packages**

Quite often during the development of an app, you are required to implement a certain feature or UI in your app. However, it’s not a matter of surprise that what you wanted to implement was already implemented by other developers, and Flutter being an OpenSource platform lets you share your code and also use other developer’s code implementation as packages.

We shall install a very famous package called animated\_text\_kit in your app. Here is a [video demonstration](https://www.youtube.com/watch?v=VeHOciP8h5I)

Note: pubspec.yaml file is an indentation sensitive file and thus, any deviation from indentation as shown in video may lead to errors.

In particular, every subsection needs to be indented 2 spaces ahead of its section heading.

**Assignment**

Deadline: 14 EOD

<https://www.figma.com/file/mPCumGo8QiI7lldYsFIPTb/Flutter-Course-Week-2-Task?node-id=0%3A2>

Assignment details are provided along with UI template, you have to submit the link to your github repo for the same in the form below.

<https://forms.gle/ZqyBG3tdpjpLwEaMA>

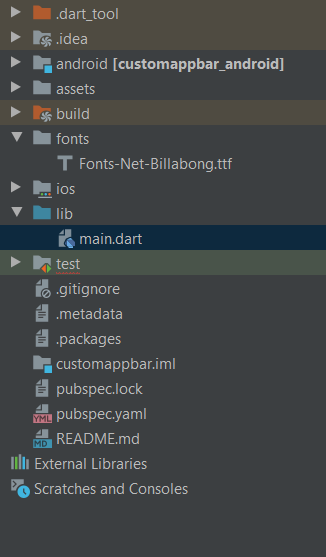
**NOTE: Remember you’re just supposed to make the UI for this assignment no need to worry about it’s actual functionality.**

**Creating fonts and switching theme in flutter**

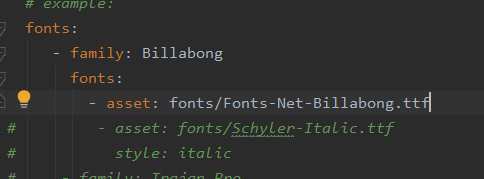
****

To use custom fonts, set up a fonts folder inside the root directory of the project. Download the Billabong font from [here](https://fontsfree.net/billabong-font-download.html).

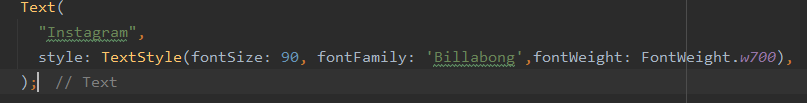
Add it in the fonts folder

****

Specify it in the pubspec file, run ‘flutter pub get’.

****

Use it in the Instagramtextwidget



**Themes in Flutter**

To specify the theme of the app, we use the ThemeData class of flutter inside the MaterialApp widget.

Refer these blogs for more info

[Part 1](https://medium.com/flutter-community/themes-in-flutter-part-1-75f52f2334ea), [Part 2](https://medium.com/flutter-community/themes-in-flutter-part-2-706382bc32c5), [Part 3](https://medium.com/flutter-community/themes-in-flutter-part-3-71361ffdc344).

So we can create themedata for various themes.

**Switching to dark theme**

1. To switch to dark theme need to change the value of the brightness inside materialapp. Refer this [code](https://github.com/oliver-gomes/flutter-unsplassh-ui/blob/master/lib/main.dart) to implement light and dark time switch. However if your app have many screens, it may be difficult to switch between themes as themedata is defined in the main.dart only. In order to solve this issue, use this [Phoenix](https://pub.dev/packages/flutter_phoenix) package from pub.dev to restart the app from main.dart once theme is changed.
2. Another way to switch themes is to use the provider package. Its more simple. Refer this video to implement it.

[Flutter: Dynamic Dark/Light Theme with Provider](https://www.youtube.com/watch?v=G7gV89hnooM)