Name: Niyati V. Gaonkar
Class: D15B   Roll no.: 17

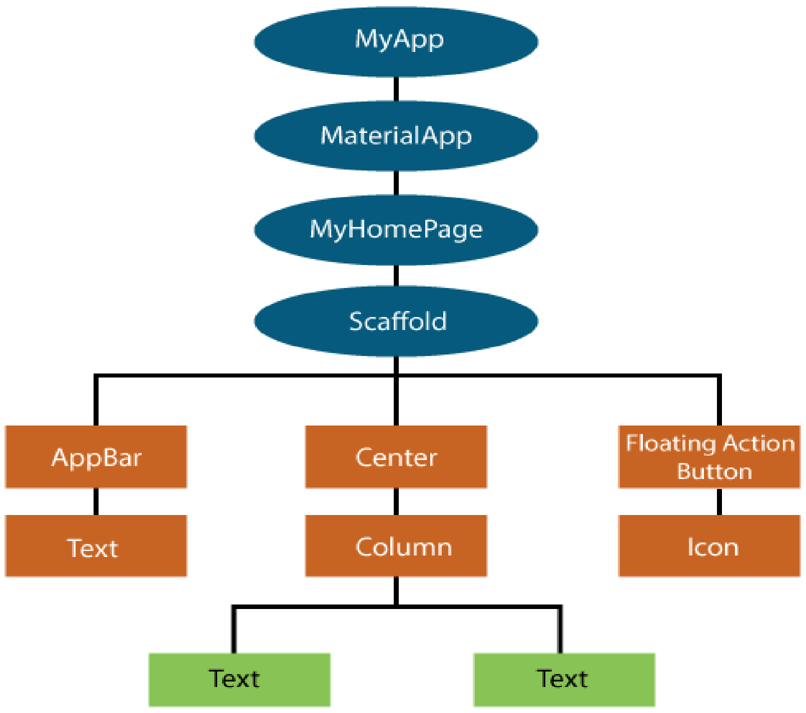
Experiment - 2

AIM: To design Flutter UI by including common widgets.

THEORY:
Whenever you are going to code for building anything in Flutter, it will be inside a widget. The central purpose is to build the app out of widgets. It describes how your app view should look like with their current configuration and state. When you made any alteration in the code, the widget rebuilds its description by calculating the difference of previous and current widget to determine the minimal changes for rendering in UI of the app.

Widgets are nested with each other to build the app. It means the root of your app is itself a widget, and all the way down is a widget also. For example, a widget can display something, can define design, can handle interaction, etc.

The below image is a simple visual representation of the widget tree.



Types of Widget

We can split the Flutter widget into two categories:

- Visible (Output and Input)
- Invisible (Layout and Control)

Visible widget

The visible widgets are related to the user input and output data. Some of the important types of this widget are:

- Text: A Text widget holds some text to display on the screen.
  Syntax:
```
new Text(
        'Hello, Javatpoint!',
        textAlign: TextAlign.center,
        style: new TextStyle(fontWeight: FontWeight.bold),
)
```

- Button: This widget allows you to perform some action on click. Flutter does not allow you to use the Button widget directly; instead, it uses a type of buttons like a FlatButton and a RaisedButton.
  Syntax:
```
//FlatButton Example
new FlatButton(
  child: Text("Click here"),
  onPressed: () {
    // Do something here
  },
),

//RaisedButton Example
new RaisedButton(
  child: Text("Click here"),
  elevation: 5.0,
  onPressed: () {
    // Do something here
  },
),
```

Invisible widget

The invisible widgets are related to the layout and control of widgets. It provides controlling how the widgets actually behave and how they will look onto the screen. Some of the important types of these widgets are:

- Column: A column widget is a type of widget that arranges all its children's widgets in a vertical alignment. It provides spacing between the widgets by using the mainAxisAlignment and crossAxisAlignment properties.

  Syntax:

  ```
  new Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      new Text(
        "VegElement",
      ),
      new Text(
        "Non-vegElement"
      ),
    ],
  ),
  ```

- Row:The row widget is similar to the column widget, but it constructs a widget horizontally rather than vertically.

  Syntax:

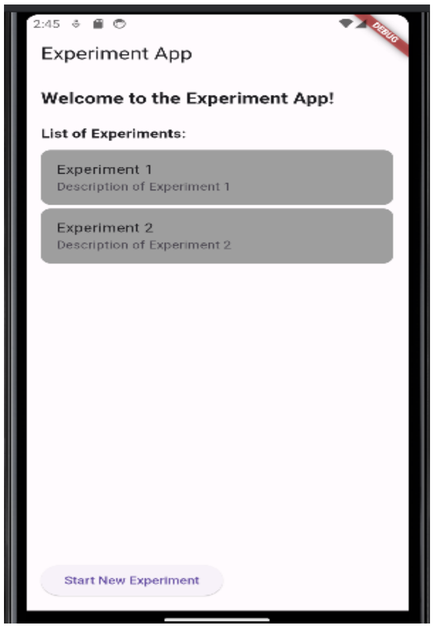
  ```
  new Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: <Widget>[
      new Text(
        "VegElement",
      ),
      new Text(
        "Non-vegElement"
      ),
    ],
  ),
  ```

CODE:

main.dart :

```dart
import 'package:flutter/material.dart';
void main() {
 runApp(MyApp());
}
class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     title: 'Experiment App',
     theme: ThemeData(
       primarySwatch: Colors.purple,
       hintColor: Colors.black,
     ),
     home: MyHomePage(),
   );
 }
}
class MyHomePage extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       title: Text('Experiment App'),
     ),
     body: Padding(
       padding: const EdgeInsets.all(16.0),
       child: Column(
         crossAxisAlignment: CrossAxisAlignment.start,
         children: [
           Text(
             'Welcome to the Experiment App!',
             style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
           ),
           SizedBox(height: 20),
           Text(
             'List of Experiments:',
             style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
           ),
           SizedBox(height: 10),
           Expanded(
             child: ListView(
               children: [
                 ListTile(
                   title: Text('Experiment 1'),
                   subtitle: Text('Description of Experiment 1'),
                   tileColor: Colors.grey,
                   shape: RoundedRectangleBorder(
                     borderRadius: BorderRadius.circular(10),
                   ),
                   onTap: () {},
                 ),
                 SizedBox(height: 5),
                 ListTile(
                   title: Text('Experiment 2'),
                   subtitle: Text('Description of Experiment 2'),
                   tileColor: Colors.grey,
                   shape: RoundedRectangleBorder(
                     borderRadius: BorderRadius.circular(10),
                   ),
                   onTap: () {},
                 ),
               ],
             ),
           ),
           SizedBox(height: 20),
           ElevatedButton(
             onPressed: () {
             },
             child: Text('Start New Experiment'),
           ),
         ],
       ),
     ),
   );
 }
}
```

Output:



Welcome to the Experiment App!

List of Experiments:

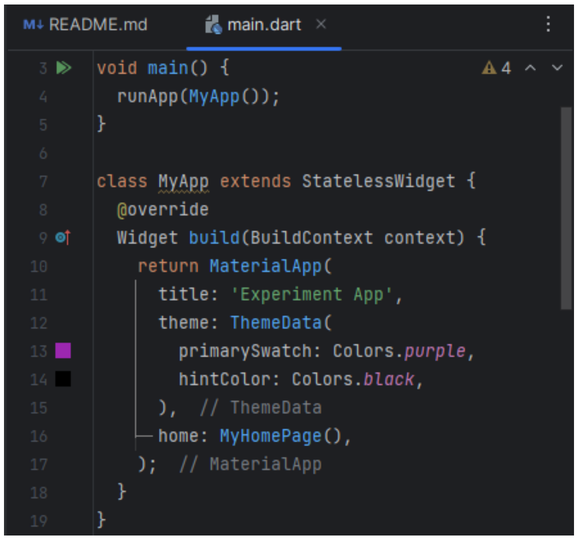
Experiment 1
Description of Experiment 1

Experiment 2
Description of Experiment 2

EXPLANATION :

Step 1: Create a new flutter project and setup with initial code.

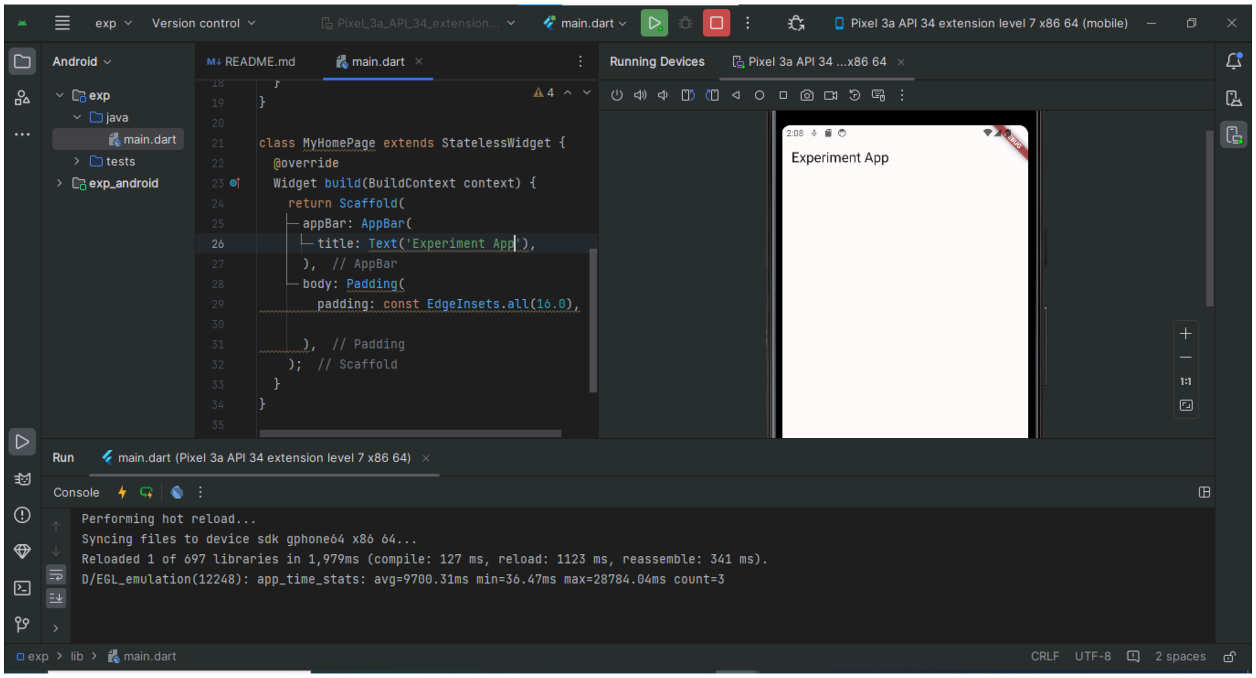
- MaterialApp: This is the root widget for your Flutter application. It sets up the default material design visual elements, such as the primary swatch color.



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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Experiment App',
      theme: ThemeData(
        primarySwatch: Colors.purple,
        hintColor: Colors.black,
      ),  // ThemeData
      home: MyHomePage(),
    );  // MaterialApp
  }
}
```

Step 2: Set the title of the app as Experiment app:

- Scaffold: This widget provides the basic structure of the visual interface. It typically includes an AppBar, a body, and other optional elements like a FloatingActionButton.
- AppBar: This is the top app bar that displays the title of the app.
- Column: This widget is used to arrange its children in a vertical column.

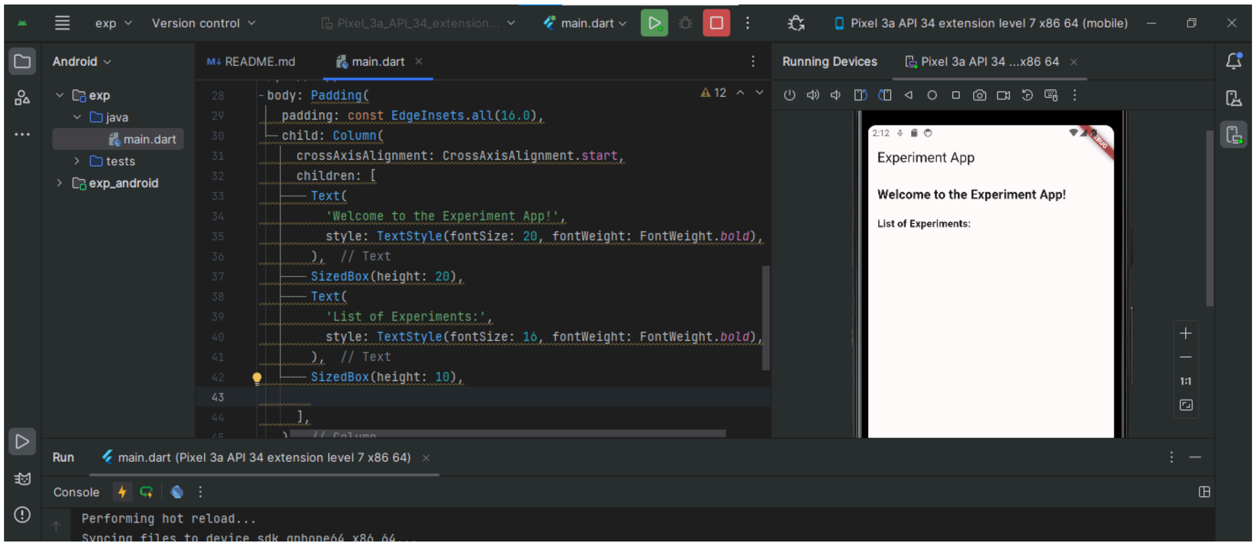


```dart
}

class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Experiment App'),
      ),  // AppBar
      body: Padding(
        padding: const EdgeInsets.all(16.0),

      ),  // Padding
    );  // Scaffold
  }
}
```

```
Performing hot reload...
Syncing files to device sdk gphone64 x86 64...
Reloaded 1 of 697 libraries in 1,979ms (compile: 127 ms, reload: 1123 ms, reassemble: 341 ms).
D/EGL_emulation(12248): app_time_stats: avg=9700.31ms min=36.47ms max=28784.04ms count=3
```

Step 3: Use the “text” widget to write an introductory message.

- Text: It displays a short piece of text. In your code, you have used it for displaying welcome messages and headings.
- SizedBox: This widget is used to introduce space between widgets. In your code, it's used to add vertical spacing between different sections.



```
body: Padding(
  padding: const EdgeInsets.all(16.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'Welcome to the Experiment App!',
        style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
      ),  // Text
      SizedBox(height: 20),
      Text(
        'List of Experiments:',
        style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
      ),  // Text
      SizedBox(height: 10),

    ],
```

```
Performing hot reload...
Syncing files to device sdk gphone64 x86 64...
```

Experiment App

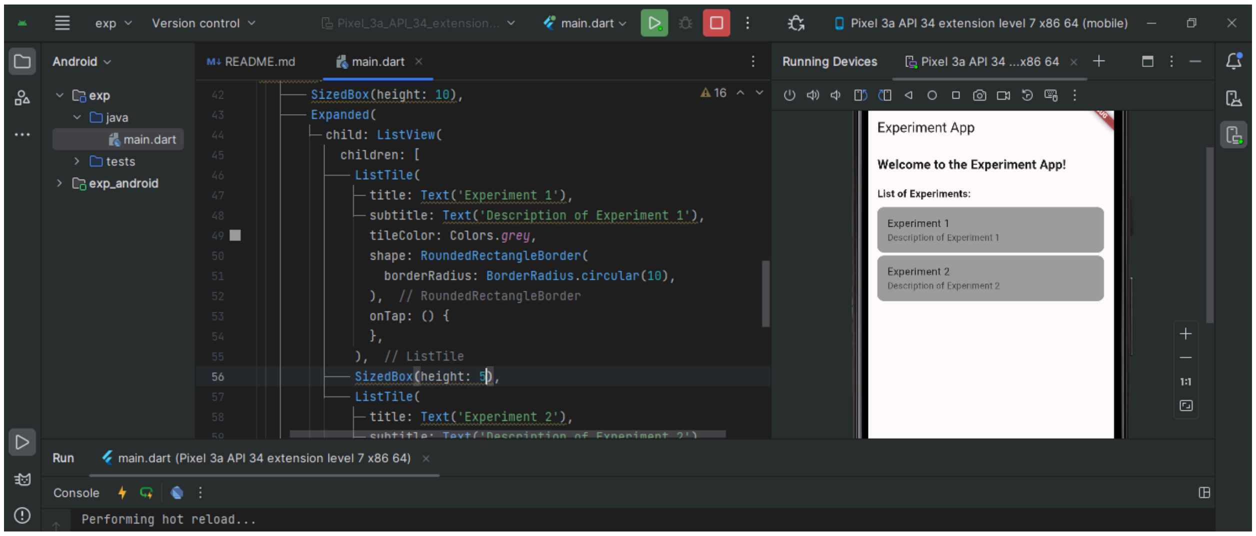
Welcome to the Experiment App!

List of Experiments:

exp
Version control
main.dart
Android
README.md
java
tests
exp_android
Running Devices
Pixel 3a API 34 extension level 7 x86 64 (mobile)
Run
main.dart (Pixel 3a API 34 extension level 7 x86 64)
Console

Step 4: Use the “listView” widget to make a list of items.

- Expanded: This widget is used to make a child of a Row, Column, or Flex expand to fill the available space along the main axis.
- ListView: This is a scrollable list of widgets. In your code, it is used to create a list of experiments.
- ListTile: It's a single fixed-height row that typically contains some text as well as optional icons. You have used it to represent each experiment in the list.
- SizedBox (inside ListView): This is used to add vertical spacing between ListTiles.



```
      SizedBox(height: 10),
      Expanded(
        child: ListView(
          children: [
            ListTile(
              title: Text('Experiment 1'),
              subtitle: Text('Description of Experiment 1'),
              tileColor: Colors.grey,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10),
              ),  // RoundedRectangleBorder
              onTap: () {
              },
            ),  // ListTile
            SizedBox(height: 5),
            ListTile(
              title: Text('Experiment 2'),
              subtitle: Text('Description of Experiment 2'),
```

```
Performing hot reload...
```

Experiment App

Welcome to the Experiment App!

List of Experiments:

Experiment 1
Description of Experiment 1

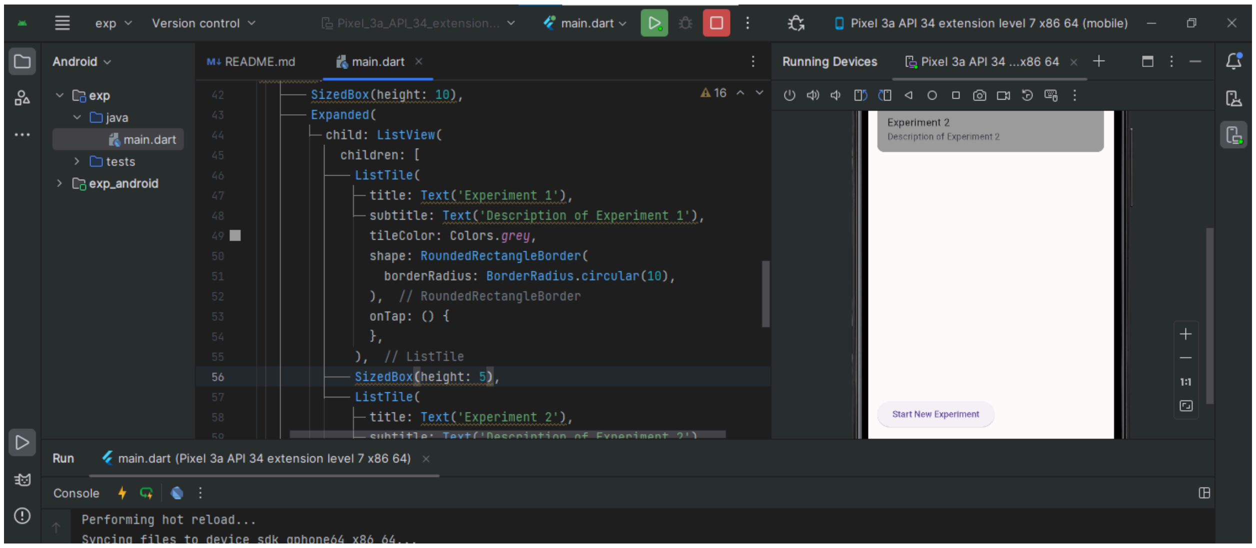
Experiment 2
Description of Experiment 2

exp
Version control
main.dart
Android
README.md
java
tests
exp_android
Running Devices
Pixel 3a API 34 extension level 7 x86 64 (mobile)
Run
main.dart (Pixel 3a API 34 extension level 7 x86 64)
Console

Note: Step 3 and Step 4 can be used multiple times too create different lists serving various purposes.

Step 5: Create a button at the bottom to add more experiments using the "ElevatedButton" widget.

- ElevatedButton: This is a Material Design raised button. In your code, it is used as a button to start a new experiment.



```
              SizedBox(height: 10),
              Expanded(
                child: ListView(
                  children: [
                    ListTile(
                      title: Text('Experiment 1'),
                      subtitle: Text('Description of Experiment 1'),
                      tileColor: Colors.grey,
                      shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(10),
                      ),  // RoundedRectangleBorder
                      onTap: () {
                      },
                    ),  // ListTile
                    SizedBox(height: 5),
                    ListTile(
                      title: Text('Experiment 2'),
```

```
Performing hot reload...
Syncing files to device sdk gphone64 x86 64...
```

CONCLUSION: Thus we have created a basic flutter UI . We have used different widgets of flutter like: Column, Text, ElevatedButton, Scaffold, MaterialApp, etc and we have understood their usage