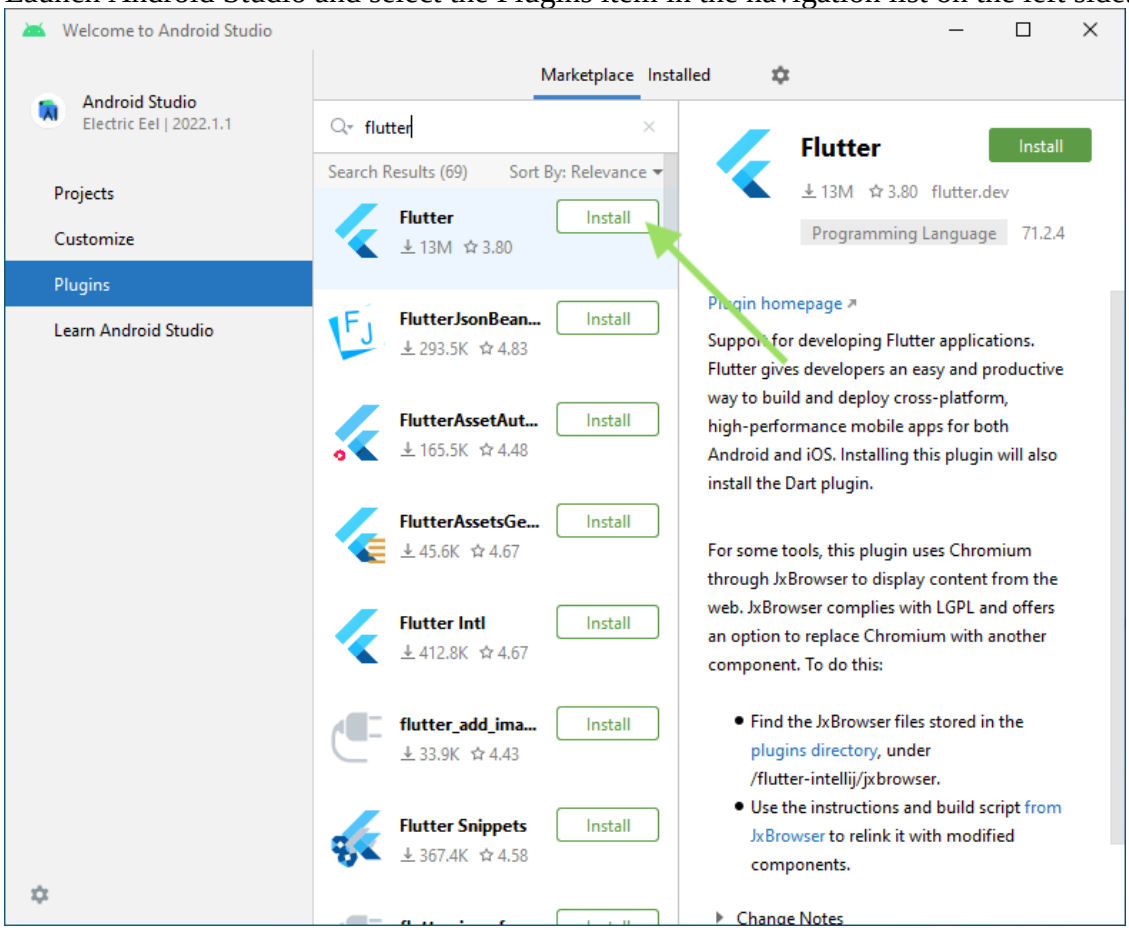
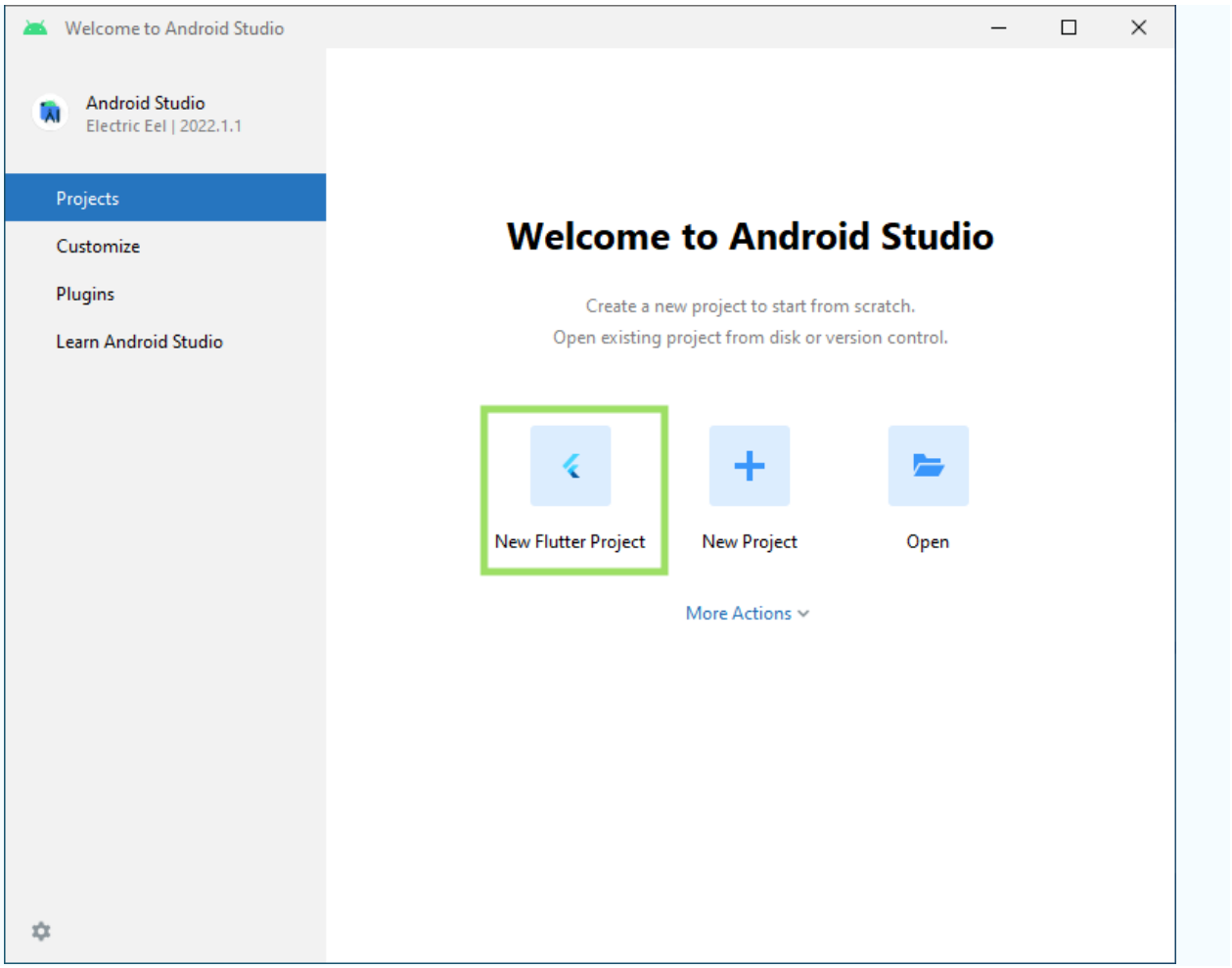
Week 1:
Installation of Android studio & Flutter.
Step 1 - System Requirements
The required tools to develop Android applications are open source and can be downloaded from the Web. Following is the list of software's you will need before you start your Android application programming.

- Java JDK8 or later version
- Java Runtime Environment (JRE)8
- Android Studio latest Version

We need to install the plugins to allow for Flutter development.
Launch Android Studio and select the Plugins item in the navigation list on the left side.



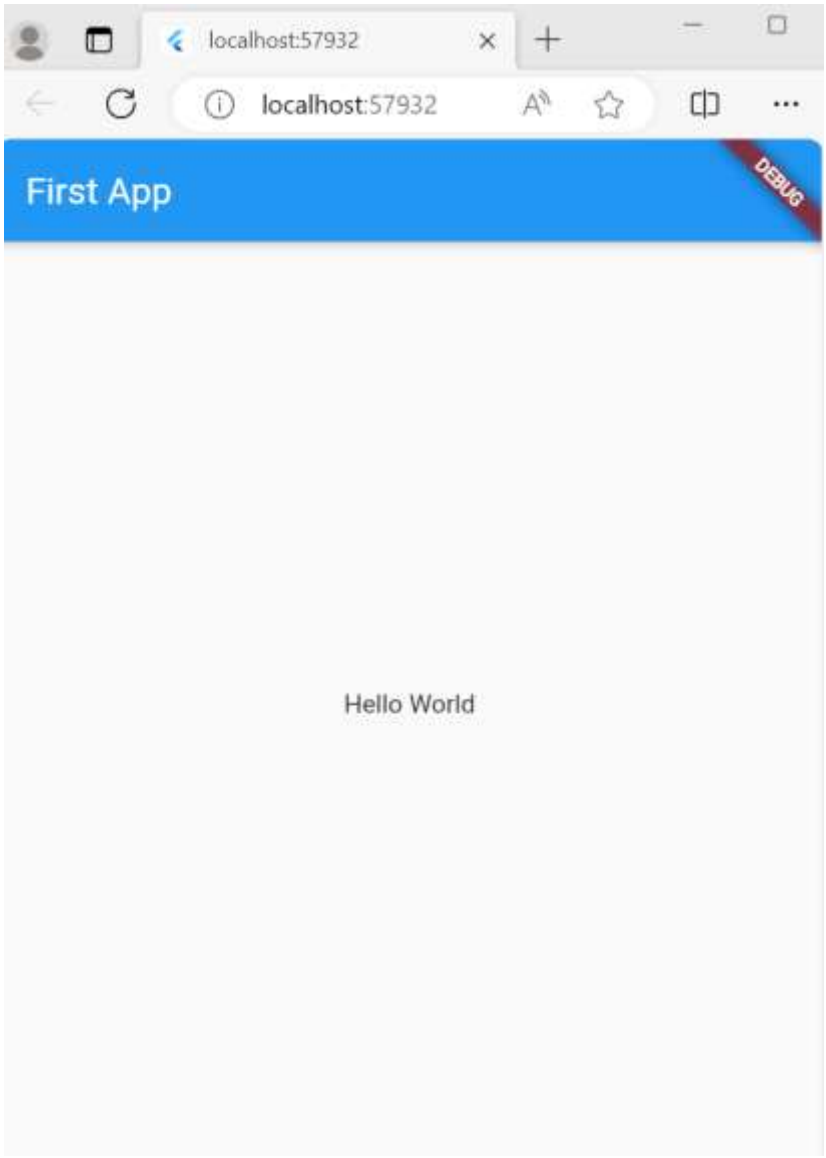
After selecting Plugins, we enter flutter as a search term in the Marketplace. We will install the Flutter plugin. Note that this will also install the Dart plugin as it is required.
Once the plugin has finished installing, Android Studio will restart, and we will now see an option create a Flutter app!


Welcome to Android Studio
Android Studio
Electric Eel | 2022.1.1
Projects
Customize
Plugins
Learn Android Studio
Welcome to Android Studio
Create a new project to start from scratch.
Open existing project from disk or version control.
New Flutter Project
New Project
Open
More Actions

Week 2:

Create an application using Flutter to print hello world.

```
/*   Flutter hello world app */
import 'package:flutter/material.dart';
void main() {
  runApp(const FirstApp());
}
class FirstApp extends StatelessWidget {
  const FirstApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    // Hello App
    return HelloApp(
      // Scaffold Widget
        home: Scaffold(
          appBar: AppBar(
            // AppBar takes a Text Widget in it's title parameter
            title: const Text('First App'),
          ),
          body: const Center(child: Text('Hello World')),
        ));
  }
}
```


localhost:57932
localhost:57932
First App
DEBUG
Hello World

Week 3:

Create an application to implement Decision making and loops using Dart.

Decision Making and Loops

The **decision-making** is a feature that allows you to evaluate a condition before the instructions are executed. The Dart language supports the following types of decision-making statements:

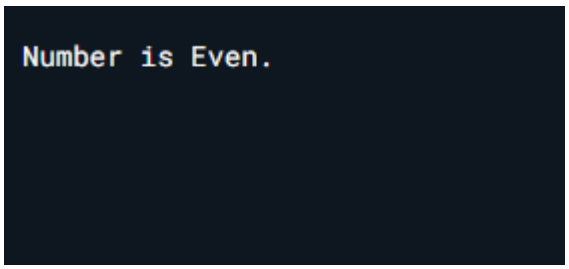
- If statement
- If-else statement
- Switch statement

Loops are used to execute a block of code repeatedly until a specified condition becomes true. Dart language supports the following types of loop statements:

- for
- for..in
- while
- do..while

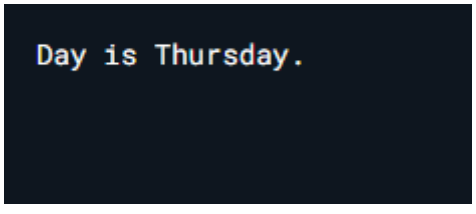
1) To find the given number is even or not

```
void main() {

  var num = 12;

  if (num%2 == 0) {

     print("Number is Even.");

  } else {        print("Number is Odd.");

  }

}
```



2) Switch case example

```
import dart.io;
void main(){
   var dayOfWeek = 5;
or
//print(“enter the choice”);
//int? dayOfWeek = int.parse(stdin.readLineSync()!);

if (dayOfWeek == 1) {
        print("Day is Sunday.");
  }
else if (dayOfWeek == 2) {
       print("Day is Monday.");
     }
else if (dayOfWeek == 3) {
      print("Day is Tuesday.");
     }
else if (dayOfWeek == 4) {
        print("Day is Wednesday.");
     }
else if (dayOfWeek == 5) {
        print("Day is Thursday.");
   }
else if (dayOfWeek == 6) {
        print("Day is Friday.");
    }
else if (dayOfWeek == 7) {
        print("Day is Saturday.");
}else{
        print("Invalid Weekday.");
     }
 }
```



Loops

1) **For each loop**

```
void main() {
  var name = ["Peter", "Rinky Ponting", "Abhishek"];

  for (var prop in name) {
    print(prop);
  }
}
```

```
Peter
Rinky Ponting
Abhishek
```

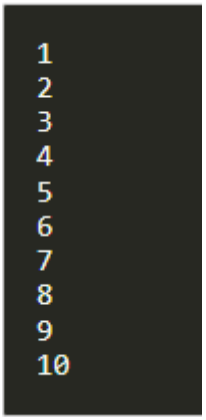
2) **For loop**

```
void main() {

 int sum=0;

 for(int i=1;i<=10;i++){

 sum=sum+i;

 }

   print("sum of numbers $sum");

}
```

```
sum of numbers 55
```

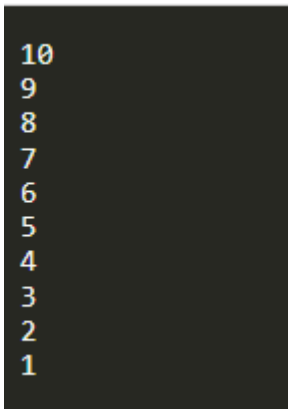
3) While loop

```
void main() {
  int i = 1;
  while (i <= 10) {
    print(i);
    i++;
  }
}
```



4) Do..while loop

```
void main() {
  int i = 10;
  do {
    print(i);
    i--;
  } while (i >= 1);
}
```



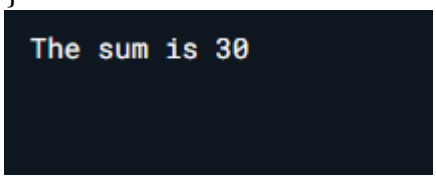
Week 4:
Create an application to demonstrate user defined functions using Dart.

Function That Find Simple Interest

```
// function that calculate interest
void calculateInterest(double principal, double rate, double time) {
  double interest = principal * rate * time / 100;
  print("Simple interest is $interest");
}
void main() {
  double principal = 5000;
  double time = 3;
  double rate = 3;
  calculateInterest(principal, rate, time);
}
```

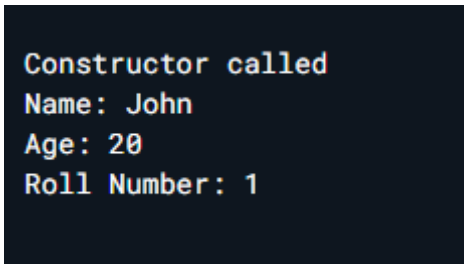
```
Simple interest is 450.0
```

```
//To find the sum of 2  numbers
// Here num1 and num2 are parameters
void add(int num1, int num2){
  int sum;
  sum = num1 + num2;

  print("The sum is $sum");
}

void main(){
// Here 10 and 20 are arguments
  add(10, 20);
}
```



Week 5:
Create an application to implement object oriented programming using Dart

```
//Constuctors
class Student {
  String? name;
  int? age;
  int? rollNumber;

  // Constructor
  Student(String name, int age, int rollNumber) {
    print(
        "Constructor called"); // this is for checking the constructor is called or not.
    this.name = name;
    this.age = age;
    this.rollNumber = rollNumber;
  }
}

void main() {
  // Here student is object of class Student.
  Student student = Student("John", 20, 1);
  print("Name: ${student.name}");
  print("Age: ${student.age}");
  print("Roll Number: ${student.rollNumber}");
}
```



Inheritance example

```
class Laptop {
  // Constructor
  Laptop({String name, String color}) {
    print("Laptop constructor");
    print("Name: $name");
    print("Color: $color");
  }
}

class MacBook extends Laptop {
  // Constructor
```

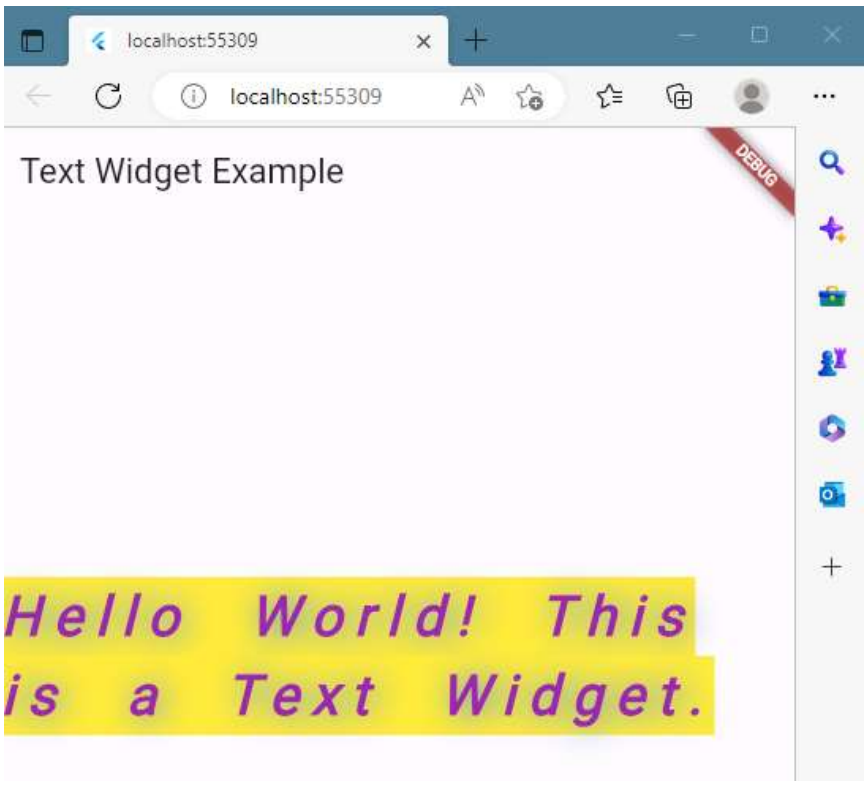
```
  MacBook({String name, String color}) : super(name: name, color: color) {
    print("MacBook constructor");
  }
}

void main() {
  var macbook = MacBook(name: "MacBook Pro", color: "Silver");
}
```

```
Laptop constructor
Name: MacBook Pro
Color: Silver
MacBook constructor
```

Week 6:

Create an application for platform basic widgets (Text, Image, and Icon).

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
        theme: ThemeData(
          primarySwatch: Colors.green,
        ),
        home: MyTextPage()
    );
  }
}
class MyTextPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
          title:Text("Text Widget Example")
      ),
      body: Center(
          child:Text(
            "Hello World! This is a Text Widget.",
            style: TextStyle(
                fontSize: 35,
                color: Colors.purple,
                fontWeight: FontWeight.w700,
                fontStyle: FontStyle.italic,
                letterSpacing: 8,
                wordSpacing: 20,
                backgroundColor: Colors.yellow,
                shadows: [
                  Shadow(color: Colors.blueAccent, offset: Offset(2,1),
blurRadius:10)
                ]
            ),
          )
      ),
    );
  }
}
```



Image

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Flutter Image Demo'),
        ),
        body: Center(
          child: Column(
            children: <Widget>[
              Image.asset('assets/tablet.png'),
              Text(
                'A tablet is a wireless touch screen computer that is smaller than a notebook but larger than a smartpho
ne.',
                style: TextStyle(fontSize: 20.0),
              )
            ],
          ),
```

```
        ),
      ),
    );
  }
}
```


4:57
Flutter Image Demo
A tablet is a wireless touch screen computer that is smaller than a notebook but larger than a smartphone.

Icon

```
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(
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: MyIconPage(),
    );
  }
}

class MyIconPage extends StatefulWidget {
  @override
  _MyIconPageState createState() => _MyIconPageState();
}

class _MyIconPageState extends State<MyIconPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Flutter Icon Tutorial'),
      ),
      body: Row(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
          children: <Widget>[
            Icon(Icons.camera_enhance),
            Icon(Icons.camera_front),
            Icon(Icons.camera_rear),
      ]),
    );
  }
}
```


2:51
Flutter Icon Tutorial

Week 7:
Create an application for Layout widgets (Single child, Multiple Child).

Layout a widget

Let us learn how we can create and display a simple widget. The following steps show how to layout a widget:

Step 1: First, you need to select a Layout widget.

Step 2: Next, create a visible widget.

Step 3: Then, add the visible widget to the layout widget.

Step 4: Finally, add the layout widget to the page where you want to display.

Single Child Widgets

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // It is the root widget of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Multiple Layout Widget',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        // This is the theme of your application.
        primarySwatch: Colors.green,
      ),
      home: MyHomePage(),
    );
  }
}
class MyHomePage extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
       appBar: AppBar(title: Text("FittedBox Widget")),
       body: Center(
       child: FittedBox(child: Row(
         children: <Widget>[
           Container(
             child: Image.asset('assets/computer.png'),
             ),
             Container(
               child: Text("This is a widget"),
             )
           ],
         ),
```

```
          fit: BoxFit.contain,
        )
      ),
    );
  }
}
```

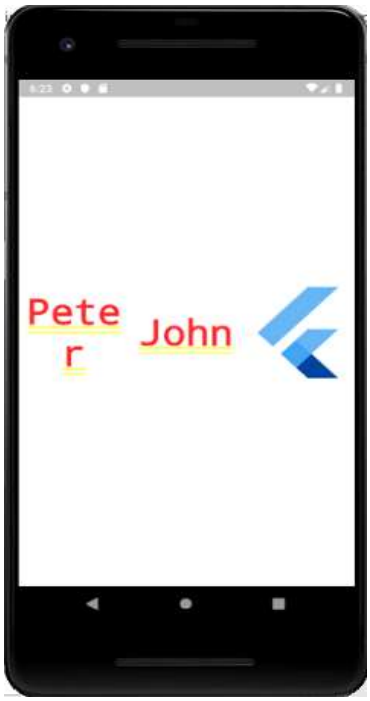
Output:

Multiple Child widgets

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // It is the root widget of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Multiple Layout Widget',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        // This is the theme of your application.
        primarySwatch: Colors.blue,
      ),
      home: MyHomePage(),
    );
```

```
  }
}
class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(
      child: Container(
        alignment: Alignment.center,
        color: Colors.white,
        child: Row(
          children: <Widget>[
            Expanded(
              child: Text('Peter', textAlign: TextAlign.center),
            ),
            Expanded(
              child: Text('John', textAlign: TextAlign.center ),

            ),
            Expanded(
              child: FittedBox(
                fit: BoxFit.contain, // otherwise the logo will be tiny
                child: const FlutterLogo(),
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

Output:



Week 8:

Create an application to demonstrate Gesture Detector.

```
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 Application', theme: ThemeData(
      primarySwatch: Colors.green,),
      home: MyHomePage(),
    );
  }
}
class MyHomePage extends StatefulWidget {
  @override
  MyHomePageState createState() => new MyHomePageState();
}

class MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: new AppBar(
        title: new Text('Gestures Example'),
        centerTitle: true,
      ),
      body: new Center(child: GestureDetector(
          onTap: () {
            print('Box Clicked');
          },
          child: Container(
            height: 60.0,
            width: 120.0,
            padding: EdgeInsets.all(10.0),
            decoration: BoxDecoration(
              color: Colors.blueGrey,
              borderRadius: BorderRadius.circular(15.0),
            ),
            child: Center(child: Text('Click Me')),
          )
      )),
    );
  }
}
```

Output:

Week 9 & 10:

Create an application for Registration form.

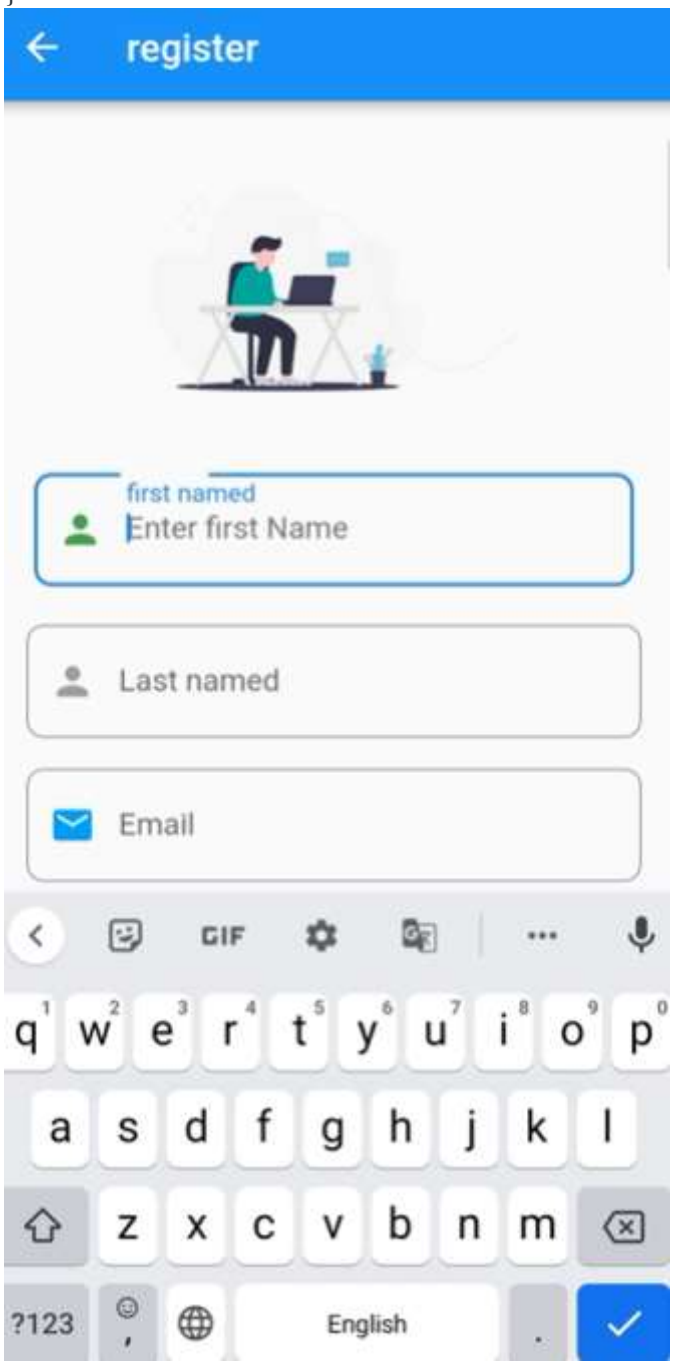
```
import 'package:flutter/material.dart';
import 'package:form_field_validator/form_field_validator.dart';
import 'package:flutter/foundation.dart';

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

@override
State<Register> createState() => _RegisterState();
}

class _RegisterState extends State<Register> {
Map userData = {};
final _formkey = GlobalKey<FormState>();
@override
Widget build(BuildContext context) {
return Scaffold(
        appBar: AppBar(
        title: Text('register'),
        ),
        body: SingleChildScrollView(
        child: Padding(
                padding: const EdgeInsets.all(12.0),
                child: Form(
                        key: _formkey,
                        child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: <Widget>[
                                Padding(
                                padding: const EdgeInsets.only(top: 20.0),
                                child: Center(
                                        child: Container(
                                        width: 200,
                                        height: 150,
                                        //decoration: BoxDecoration(
                                        //borderRadius: BorderRadius.circular(40),
                                        //border: Border.all(color: Colors.blueGrey)),
                                        child: Image.asset('assets/logo.png'),
                                        ),
                                ),
                                ),
                                Padding(
                                padding: const EdgeInsets.all(12.0),
                                child: TextFormField(
                                        // validator: ((value) {
                                        // if (value == null || value.isEmpty) {
                                        //         return 'please enter some text';
                                        // } else if (value.length < 5) {
                                        //         return 'Enter atleast 5 Charecter';
                                        // }

                                        // return null;
                                        // }),
```

```
validator: MultiValidator([
RequiredValidator(errorText: 'Enter first named'),
MinLengthValidator(3,
errorText: 'Minimum 3 charecter filled name'),
]),

decoration: InputDecoration(
hintText: 'Enter first Name',
labelText: 'first named',
prefixIcon: Icon(
Icons.person,
color: Colors.green,
),
errorStyle: TextStyle(fontSize: 18.0),
border: OutlineInputBorder(
borderSide: BorderSide(color: Colors.red),
borderRadius:
BorderRadius.all(Radius.circular(9.0)))),
),
),
Padding(
padding: const EdgeInsets.all(8.0),
child: TextFormField(
validator: MultiValidator([
RequiredValidator(errorText: 'Enter last named'),
MinLengthValidator(3,
errorText:
'Last name should be atleast 3 charater'),
]),
decoration: InputDecoration(
hintText: 'Enter last Name',
labelText: 'Last named',
prefixIcon: Icon(
Icons.person,
color: Colors.grey,
),
errorStyle: TextStyle(fontSize: 18.0),
border: OutlineInputBorder(
borderSide: BorderSide(color: Colors.red),
borderRadius:
BorderRadius.all(Radius.circular(9.0)))),
),
),
Padding(
padding: const EdgeInsets.all(8.0),
child: TextFormField(
validator: MultiValidator([
RequiredValidator(errorText: 'Enter email address'),
EmailValidator(
errorText: 'Please correct email filled'),
]),
decoration: InputDecoration(
hintText: 'Email',
labelText: 'Email',
prefixIcon: Icon(
Icons.email,
```

```
                color: Colors.lightBlue,
                ),
                errorStyle: TextStyle(fontSize: 18.0),
                border: OutlineInputBorder(
                        borderSide: BorderSide(color: Colors.red),
                        borderRadius:
                                BorderRadius.all(Radius.circular(9.0)))),
),
),
Padding(
padding: const EdgeInsets.all(8.0),
child: TextFormField(
        validator: MultiValidator([
        RequiredValidator(errorText: 'Enter mobile number'),
        PatternValidator(r'(^[0,9]{10}$)',
                errorText: 'enter vaid mobile number'),
        ]),
        decoration: InputDecoration(
                hintText: 'Mobile',
                labelText: 'Mobile',
                prefixIcon: Icon(
                Icons.phone,
                color: Colors.grey,
                ),
                border: OutlineInputBorder(
                        borderSide: BorderSide(color: Colors.red),
                        borderRadius:
                                BorderRadius.all(Radius.circular(9)))),
),
),
Center(
        child: Padding(
padding: const EdgeInsets.all(18.0),
child: Container(
        // margin: EdgeInsets.fromLTRB(200, 20, 50, 0),
        child: RaisedButton(
        child: Text(
                'Register',
                style: TextStyle(color: Colors.white, fontSize: 22),
        ),
        onPressed: () {
                if (_formkey.currentState!.validate()) {
                print('form submiitted');
                }
        },
        shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(30)),
        color: Colors.blue,
        ),

        width: MediaQuery.of(context).size.width,

        height: 50,
),
)),
Center(
```

```
                    child: Padding(
                        padding: EdgeInsets.only(top: 20),
                        child: Center(
                        child: Text(
                            'Or Sign Up Using',
                            style: TextStyle(fontSize: 18, color: Colors.black),
                        ),
                        ),
                    ),
                    ),
                    Center(
                    child: Padding(
                        padding: EdgeInsets.only(top: 20, left: 90),
                        child: Row(
                        children: [
                            Container(
                                height: 40,
                                width: 40,
                                child: Image.asset(
                                'assets/google.png',
                                fit: BoxFit.cover,
                                )),
                            Container(
                            height: 70,
                            width: 70,
                            child: Image.asset(
                                'assets/vishal.png',
                                fit: BoxFit.cover,
                            ),
                            ),
                            Container(
                            height: 40,
                            width: 40,
                            child: Image.asset(
                                'assets/google.png',
                                fit: BoxFit.cover,
                            ),
                            ),
                        ],
                        ),
                    ),
                    ),
                    Center(
                    child: Container(
                        padding: EdgeInsets.only(top: 60),
                        child: Text(
                        'SIGN IN',
                        style: TextStyle(
                            fontSize: 20, fontWeight: FontWeight.bold),
                        ),
                    ),
                    )
                ],
                )),
    ),
    ));
```

}
}


register
first named
Enter first Name
Last named
Email
GIF
English

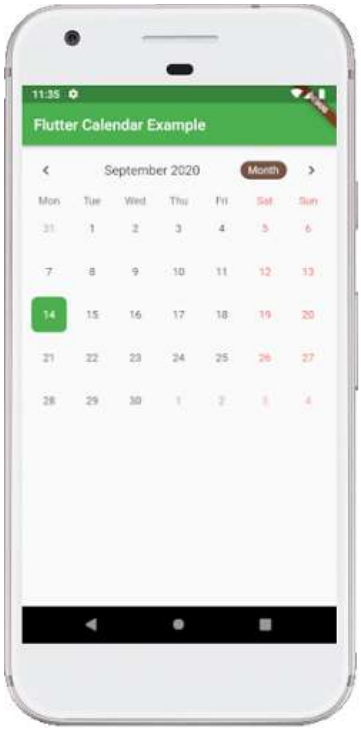
Week 11:
Create an application to implement flutter calendar.

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        primarySwatch: Colors.green,
      ),
      home: HomeCalendarPage(),
    );
  }
}

class HomeCalendarPage extends StatefulWidget {
  @override
  _HomeCalendarPageState createState() => _HomeCalendarPageState();
}

class _HomeCalendarPageState extends State<HomeCalendarPage> {
  CalendarController _controller;

  @override
  void initState() {
    super.initState();
    _controller = CalendarController();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Flutter Calendar Example'),
      ),
      body: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            TableCalendar(
              initialCalendarFormat: CalendarFormat.month,
```

```
    calendarStyle: CalendarStyle(
      todayColor: Colors.blue,
      selectedColor: Theme.of(context).primaryColor,
      todayStyle: TextStyle(
          fontWeight: FontWeight.bold,
          fontSize: 22.0,
          color: Colors.white)
    ),
    headerStyle: HeaderStyle(
     centerHeaderTitle: true,
     formatButtonDecoration: BoxDecoration(
      color: Colors.brown,
      borderRadius: BorderRadius.circular(22.0),
     ),
     formatButtonTextStyle: TextStyle(color: Colors.white),
     formatButtonShowsNext: false,
    ),
    startingDayOfWeek: StartingDayOfWeek.monday,
    onDaySelected: (date, events) {
     print(date.toUtc());
    },
    builders: CalendarBuilders(
     selectedDayBuilder: (context, date, events) => Container(
        margin: const EdgeInsets.all(5.0),
        alignment: Alignment.center,
        decoration: BoxDecoration(
            color: Theme.of(context).primaryColor,
            borderRadius: BorderRadius.circular(8.0)),
        child: Text(
          date.day.toString(),
          style: TextStyle(color: Colors.white),
        )),
     todayDayBuilder: (context, date, events) => Container(
        margin: const EdgeInsets.all(5.0),
        alignment: Alignment.center,
        decoration: BoxDecoration(
            color: Colors.blue,
            borderRadius: BorderRadius.circular(8.0)),
        child: Text(
          date.day.toString(),
          style: TextStyle(color: Colors.white),
        )),
    ),
    calendarController: _controller,
  )
],
```

```
        ),
      ),
    );
  }
}
```

Output


11:35
Flutter Calendar Example
September 2020
Month
Mon
Tue
Wed
Thu
Fri
Sat
Sun

Week 12:
Create an application to implement Animated Text in Flutter

Step1: Add animated_text_kit package to the *pubspec.yaml* file:

```
dependencies:
  flutter:
    sdk: flutter


  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  animated_text_kit: ^4.2.1
```

```
C:/> flutter pub add animated_text_kit
```

Rotate

```
import 'package:animated_text_kit/animated_text_kit.dart';
AnimatedTextKit(
                        animatedTexts: [
                        RotateAnimatedText('AWESOME',
                                textStyle: TextStyle(
                                        fontSize: 30,
                                        color: Colors.white,
                                        backgroundColor: Colors.blue)),
                        RotateAnimatedText('OPTIMISTIC',
                                textStyle: TextStyle(
                                        letterSpacing: 3,
                                        fontSize: 30,
                                        fontWeight: FontWeight.bold,
                                        color: Colors.orange)),
                        RotateAnimatedText(
                                'DIFFERENT',
                                textStyle: TextStyle(
                                fontSize: 30,
                                decoration: TextDecoration.underline,
                                ),
                        ),
                        ],
                        isRepeatingAnimation: true,
                        totalRepeatCount: 10,
                        pause: Duration(milliseconds: 1000),
```

```
                    ),    ),
  ],
);
```

Output: