

--- تامر مدحت عبدالحميد محمد 198058

--- احمد السيد سعد سلام 198003

--- عبدالرحمن عرفه السيد 198121

## Contents

Assignment 3-1 :	2
Assignment 3-2 :	3
Assignment 3-3 :	4
Assignment 3-4 :	5
Assignment 4-1 :	6
Assignment 4-2 :	7
Assignment 4-3 :	8
Assignment 6-1 :	9
Assignment 6-2 :	12
Assignment 6-3 :	16
Assignment 7-1 :	18
Assignment 7-2 :	20
Assignment 8-1 :	24
Assignment 8-2 :	27
Assignment 8-3 :	31

### Assignment 3-1 :

```
main(){  
    double tempFahrenheit = 90;  
    // calculate the temperature in Celsius using the formula  
    double tempCelsius = (tempFahrenheit - 32) / 1.8;  
    // print the result with one decimal place  
    print('${tempFahrenheit}F =  
    ${tempCelsius.toStringAsFixed(1)}C');  
}
```



The screenshot shows a code editor with a dark theme. On the left, a code file is open with the following Dart code:

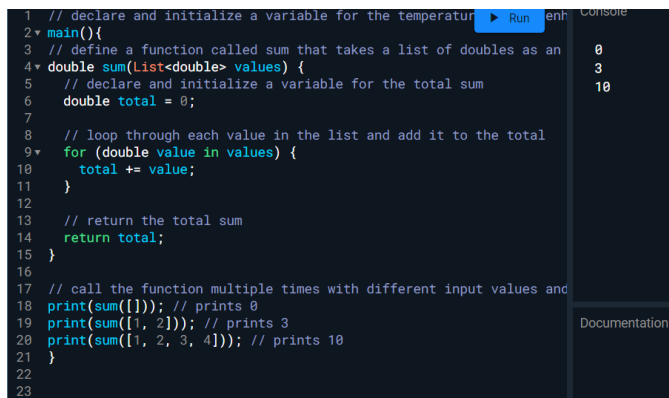
```
1 main(){  
2   double tempFahrenheit = 90;  
3   // calculate the temperature in Celsius using the formula  
4   double tempCelsius = (tempFahrenheit - 32) / 1.8;  
5   // print the result with one decimal place  
6   print( '${tempFahrenheit}F = ${tempCelsius.toStringAsFixed(1)}C' );  
7 }  
8
```

A blue "Run" button is visible next to the code. On the right, a "Console" panel shows the output of the program:

```
90F = 32.2C
```

## Assignment 3-2 :

```
main(){  
    double sum(List<double> values) {  
        double total = 0;  
        for (double value in values) {  
            total += value;  
        } return total;  
    }  
    print(sum([])); // prints 0  
    print(sum([1, 2])); // prints 3  
    print(sum([1, 2, 3, 4])); // prints 10  
}
```



```
1 // declare and initialize a variable for the temperature  
2 main(){  
3 // define a function called sum that takes a list of doubles as an  
4 double sum(List<double> values) {  
5 // declare and initialize a variable for the total sum  
6 double total = 0;  
7  
8 // loop through each value in the list and add it to the total  
9 for (double value in values) {  
10 total += value;  
11 }  
12  
13 // return the total sum  
14 return total;  
15 }  
16  
17 // call the function multiple times with different input values and  
18 print(sum([])); // prints 0  
19 print(sum([1, 2])); // prints 3  
20 print(sum([1, 2, 3, 4])); // prints 10  
21 }  
22  
23
```

### Assignment 3-3 :

```
import 'dart:io';

main(){
  bool isPrime(int n) {
    if (n < 2) {
      return false;
    }
    for (int i = 2; i * i <= n; i++) {
      if (n % i == 0) {
        return false;
      }
    }
    return true;}

  stdout.write('Enter a number: ');
  int number = int.parse(stdin.readLineSync()!);
  if (isPrime(number)) {
    print('$number is prime.');
```

```
  } else {
```

```
    print('$number is not prime.');
```

```
    return false;
  }
  // loop from 2 to the square root of the number
  for (int i = 2; i * i <= n; i++) {
    // if the number is divisible by any i, it is not prime
    if (n % i == 0) {
      return false;
    }
  }
  // if the loop finishes without finding any divisor, the number
  return true;
}

// ask the user for a number and store it in a variable
// stdout.write('Enter a number: ');
// int number = int.parse(stdin.readLineSync()!);
```

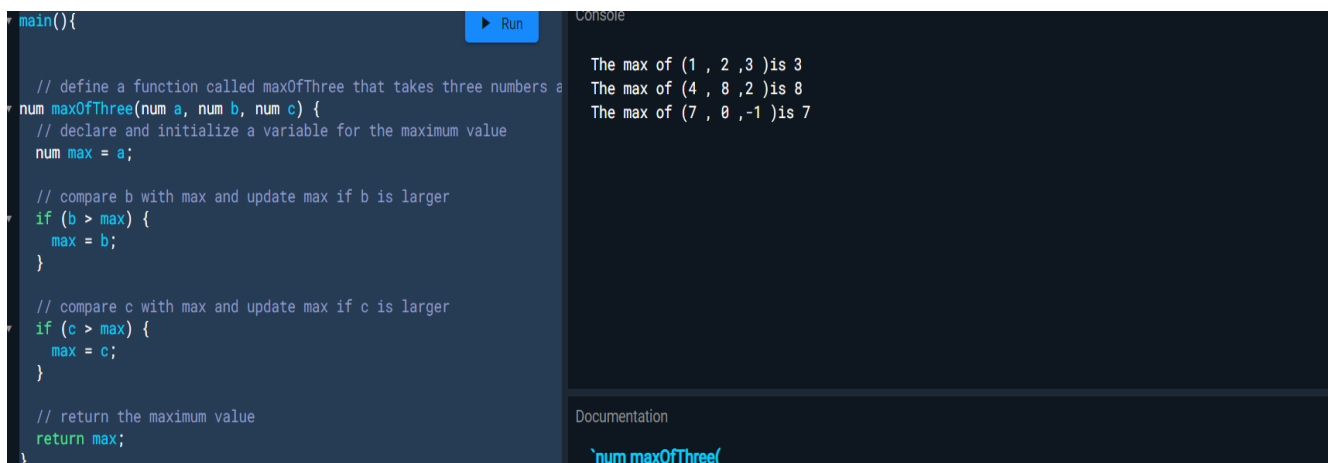
Run

10 is not prime.  
12 is not prime.  
5 is prime.

Assignment 3-4 :

```
main(){
    num maxOfThree(num a, num b, num c) {
        num max = a;
        if (b > max) {
            max = b;
        }
        if (c > max) {
            max = c;
        }
        return max;
    }

    print( 'The max of (1 , 2 ,3 )is ${maxOfThree(1, 2, 3)}'); // prints 3
    print( 'The max of (4 , 8 ,2 )is ${maxOfThree(4, 8, 2)}'); // prints 8
    print( 'The max of (7 , 0 ,-1 )is ${maxOfThree(7 , 0 ,-1 )}'); // prints 7
}
```



The screenshot shows a code editor with a dark theme. On the left, the code for the `main()` function is written in a light blue font. It defines a function `maxOfThree` that takes three arguments `a`, `b`, and `c`. The function initializes `max` to `a`, then compares `b` and `c` to `max`, updating `max` if either is larger. Finally, it returns `max`. The `main` function calls `maxOfThree` with three different sets of arguments and prints the results. On the right, the console output shows the results of these calls: 'The max of (1 , 2 ,3 )is 3', 'The max of (4 , 8 ,2 )is 8', and 'The max of (7 , 0 ,-1 )is 7'. Below the console, there is a section for documentation, which shows the function signature `num maxOfThree(`.

```
main(){
    // define a function called maxOfThree that takes three numbers a, b, and c
    num maxOfThree(num a, num b, num c) {
        // declare and initialize a variable for the maximum value
        num max = a;

        // compare b with max and update max if b is larger
        if (b > max) {
            max = b;
        }

        // compare c with max and update max if c is larger
        if (c > max) {
            max = c;
        }

        // return the maximum value
        return max;
    }

    print( 'The max of (1 , 2 ,3 )is ${maxOfThree(1, 2, 3)}'); // prints 3
    print( 'The max of (4 , 8 ,2 )is ${maxOfThree(4, 8, 2)}'); // prints 8
    print( 'The max of (7 , 0 ,-1 )is ${maxOfThree(7 , 0 ,-1 )}'); // prints 7
}
```

Console

```
The max of (1 , 2 ,3 )is 3
The max of (4 , 8 ,2 )is 8
The max of (7 , 0 ,-1 )is 7
```

Documentation

```
num maxOfThree(
```

## Assignment 4-1 :

```
main(){  
    double sum(List<double> values) {  
        double total = 0;  
        for (double value in values) {  
            total += value;  
        }  
        return total;  
    }  
    print(sum([])); // prints 0  
    print(sum([1, 2])); // prints 3  
    print(sum([1, 2, 3, 4])); // prints 10  
}
```

```
1 // declare and initialize a variable for the temperature  
2 main(){  
3 // define a function called sum that takes a list of doubles as an argument  
4 double sum(List<double> values) {  
5 // declare and initialize a variable for the total sum  
6 double total = 0;  
7  
8 // loop through each value in the list and add it to the total  
9 for (double value in values) {  
10 total += value;  
11 }  
12  
13 // return the total sum  
14 return total;  
15 }  
16  
17 // call the function multiple times with different input values and print the results  
18 print(sum([])); // prints 0  
19 print(sum([1, 2])); // prints 3  
20 print(sum([1, 2, 3, 4])); // prints 10  
21 }  
22  
23
```

0  
3  
10

Documentation

#### Assignment 4-2 :

```
import 'dart:io';

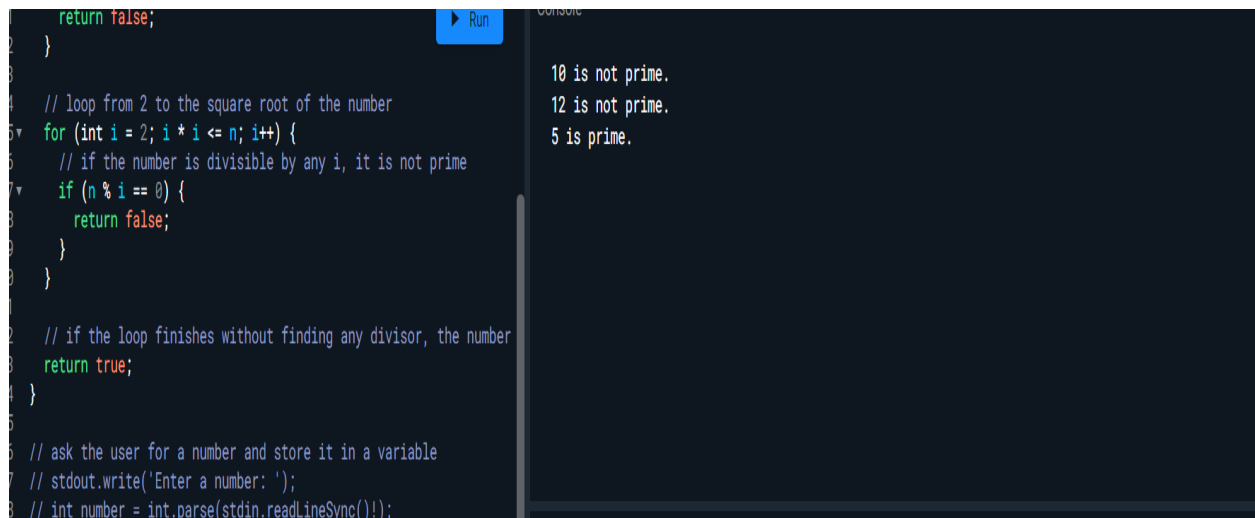
main(){
  bool isPrime(int n) {
    if (n < 2) {
      return false;
    }
    for (int i = 2; i * i <= n; i++) {
      if (n % i == 0) {
        return false;}}
    return true;}

  stdout.write('Enter a number: ');

  int number = int.parse(stdin.readLineSync(!));

  if (isPrime(number)) {
    print('$number is prime.');
```

```
} else {
  print('$number is not prime.');
```



The screenshot shows a code editor with a dark theme. On the left, the Dart code is displayed with line numbers 1 through 8. The code defines a function `isPrime` and uses it in the `main` function to prompt the user for a number and check if it is prime. A blue 'Run' button is visible above the code. On the right, a console window shows the output of the program: '10 is not prime.', '12 is not prime.', and '5 is prime.'

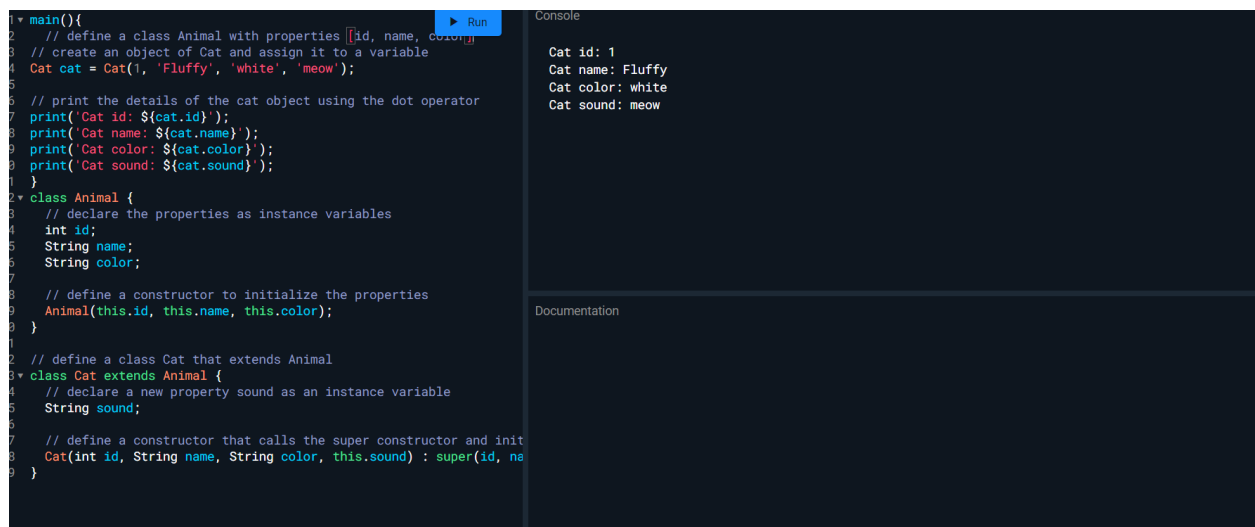
```
1 return false;
2 }
3
4 // loop from 2 to the square root of the number
5 for (int i = 2; i * i <= n; i++) {
6   // if the number is divisible by any i, it is not prime
7   if (n % i == 0) {
8     return false;
9   }
10 }
11
12 // if the loop finishes without finding any divisor, the number
13 return true;
14 }
15
16 // ask the user for a number and store it in a variable
17 // stdout.write('Enter a number: ');
18 // int number = int.parse(stdin.readLineSync(!));
```

Console

```
10 is not prime.
12 is not prime.
5 is prime.
```

### Assignment 4-3 :

```
main(){  
  
    Cat cat = Cat(1, 'Fluffy', 'white', 'meow');  
  
    print('Cat id: ${cat.id}');  
    print('Cat name: ${cat.name}');  
    print('Cat color: ${cat.color}');  
    print('Cat sound: ${cat.sound}');  
}  
  
class Animal {  
    int id;  
    String name;  
    String color;  
    Animal(this.id, this.name, this.color);  
}  
  
class Cat extends Animal {  
    String sound;  
    Cat(int id, String name, String color, this.sound) : super(id, name,  
color);};
```



The screenshot shows an IDE with a code editor on the left and a console on the right. The code editor contains the following Java code:

```
1 main(){  
2     // define a class Animal with properties [id, name, color]  
3     // create an object of Cat and assign it to a variable  
4     Cat cat = Cat(1, 'Fluffy', 'white', 'meow');  
5  
6     // print the details of the cat object using the dot operator  
7     print('Cat id: ${cat.id}');  
8     print('Cat name: ${cat.name}');  
9     print('Cat color: ${cat.color}');  
10    print('Cat sound: ${cat.sound}');  
11 }  
12  
13 class Animal {  
14     // declare the properties as instance variables  
15     int id;  
16     String name;  
17     String color;  
18  
19     // define a constructor to initialize the properties  
20     Animal(this.id, this.name, this.color);  
21 }  
22  
23 // define a class Cat that extends Animal  
24 class Cat extends Animal {  
25     // declare a new property sound as an instance variable  
26     String sound;  
27  
28     // define a constructor that calls the super constructor and init  
29     Cat(int id, String name, String color, this.sound) : super(id, ne  
30 }  
31 }
```

The console on the right shows the output of the program:

```
Cat id: 1  
Cat name: Fluffy  
Cat color: white  
Cat sound: meow
```

Below the console, there is a section for documentation, which is currently empty.



### Assignment 6-1 :

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

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

class ApplicationRoot extends StatelessWidget {
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Application',
      home: HomeScreen(),
    );
  }
}

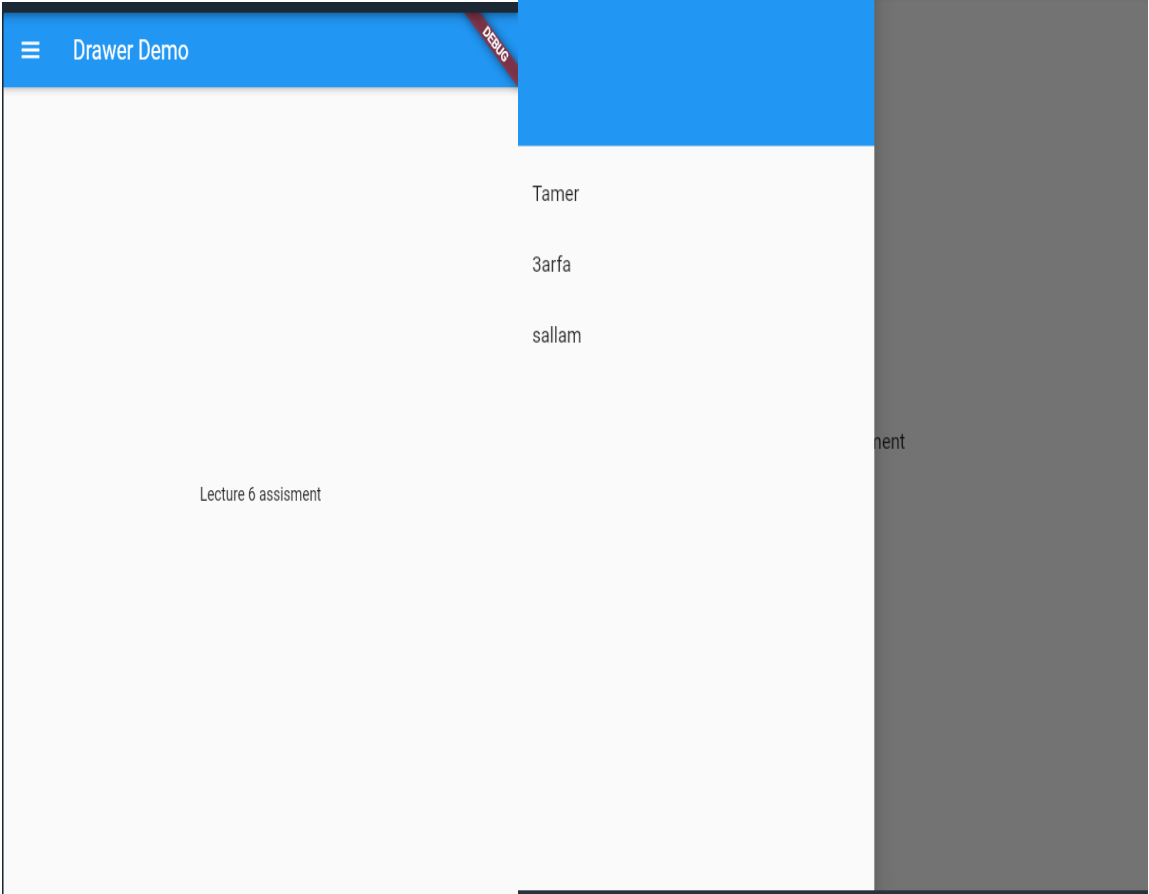
class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Drawer Demo'),
      ),
      body: Center(
        child: Text('Lecture 6 assisment'),
      ),
      drawer: Drawer(
        child: ListView(
          padding: EdgeInsets.zero,
          children: [
            const DrawerHeader(
```

```

        decoration: BoxDecoration(
          color: Colors.blue,
        ),
        child: Text('Drawer Header'),
      ),
      ListTile(
        title: const Text('Tamer'),
        onTap: () {
          print("Done");
        },
      ),
      ListTile(
        title: const Text('3arfa'),
        onTap: () {
          print("Done");
        },
      ),

      ListTile(
        title: const Text('sallam'),
        onTap: () {
          print("Done");
        },),),),
    ),);}}

```



## Assignment 6-2 :

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

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

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: BottomNavigationBarExample(),
    );
  }
}

class BottomNavigationBarExample extends StatefulWidget {
  const BottomNavigationBarExample({super.key});

  @override
  State<BottomNavigationBarExample> createState() =>
    _BottomNavigationBarExampleState();
}

class _BottomNavigationBarExampleState
  extends State<BottomNavigationBarExample> {
  int _selectedIndex = 0;

  static const TextStyle optionStyle =
    TextStyle(fontSize: 30, fontWeight: FontWeight.bold);
  static const List<Widget> _widgetOptions = <Widget>[
    Text(
```

```
        'Index 0-> Home',  
        style: optionStyle,  
    ),  
    Text(  
        'Index 1-> Business',  
        style: optionStyle,  
    ),  
    Text(  
        'Index 2-> School',  
        style: optionStyle,  
    ),  
    Text(  
        'Index 3-> Settings',  
        style: optionStyle,  ),,];
```

```
void _onItemTapped(int index) {  
    setState(() {  
        _selectedIndex = index;  
    });  
}
```

```
@override
```

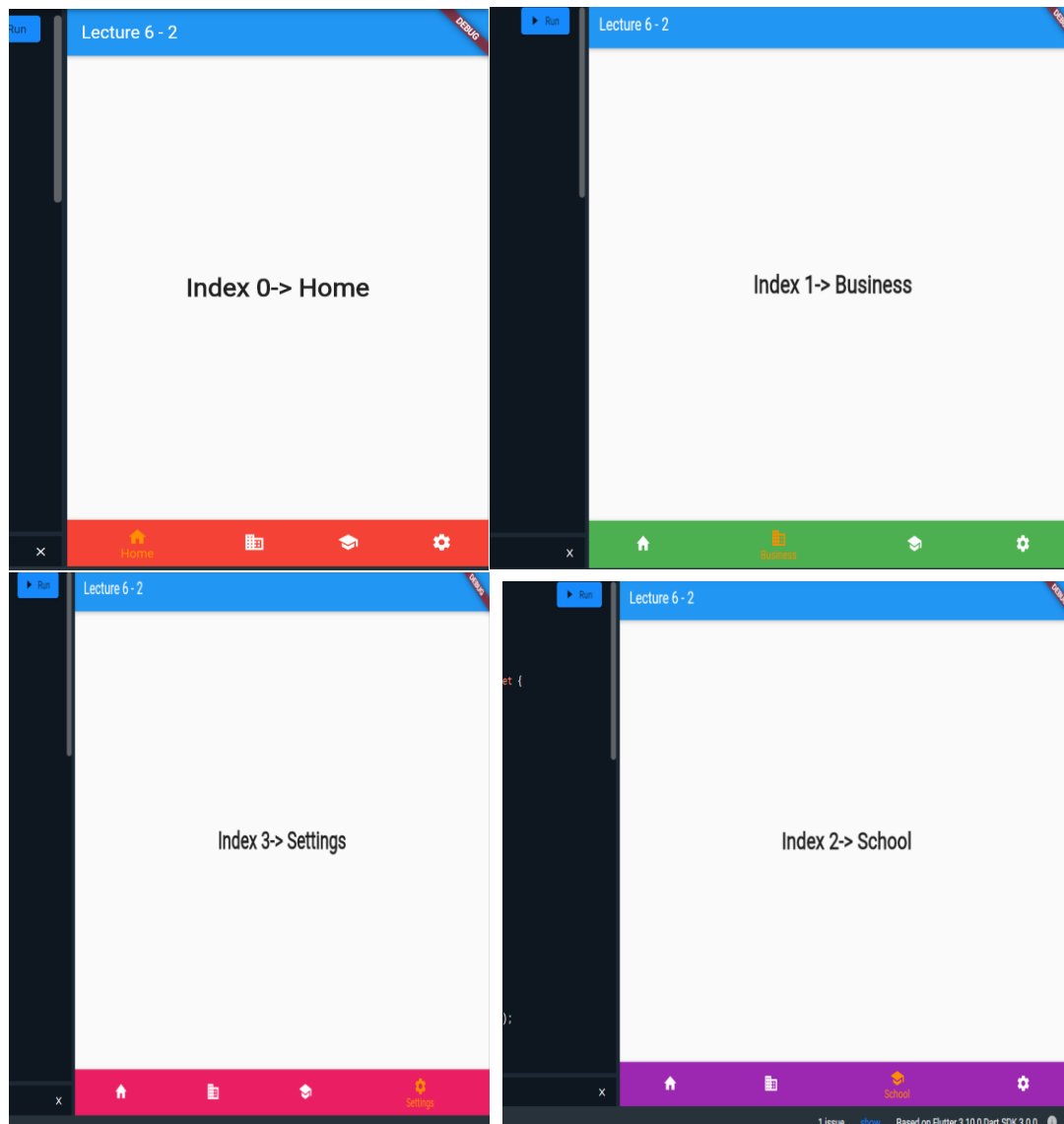
```
Widget build(BuildContext context) {  
    return Scaffold(  
        appBar: AppBar(  
            title: Text('Index 0-> Home'),  
            backgroundColor: Colors.white,
```

```
title: const Text('Lecture 6 - 2'),
),
body: Center(
  child: _widgetOptions.elementAt(_selectedIndex),
),
bottomNavigationBar: BottomNavigationBar(
  items: const <BottomNavigationBarItem>[
    BottomNavigationBarItem(
      icon: Icon(Icons.home),
      label: 'Home',
      backgroundColor: Colors.red,
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.business),
      label: 'Business',
      backgroundColor: Colors.green,
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.school),
      label: 'School',
      backgroundColor: Colors.purple,
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.settings),
      label: 'Settings',
```

```

        backgroundColor: Colors.pink,
      ),
    ],
    currentIndex: _selectedIndex,
    selectedItemColor: Colors.amber[800],
    onTap: _onItemTapped,)),);}}

```



### Assignment 6-3 :

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

void main() => runApp(const
FloatingActionButtonExampleApp());

class FloatingActionButtonExampleApp extends
StatelessWidget {

  const FloatingActionButtonExampleApp({super.key});

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: FabExample(),
    );}}

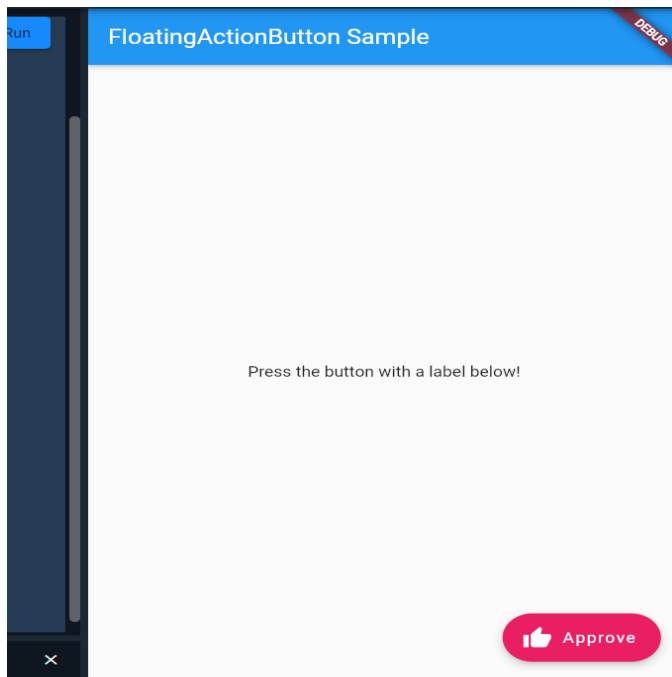
class FabExample extends StatelessWidget {

  const FabExample({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('FloatingActionButton Sample'),
      ),
      body: const Center(
        child: Text('Press the button with a label below!'),
      ),
      floatingActionButton: FloatingActionButton.extended(
        onPressed: () {
```



```
        print("hi");  
    },  
    label: const Text('Approve'),  
    icon: const Icon(Icons.thumb_up),  
  
    backgroundColor: Colors.pink,  
  ),);  
}  
}
```



### Assignment 7-1 :

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

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

class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Home(),
    );
  }
}

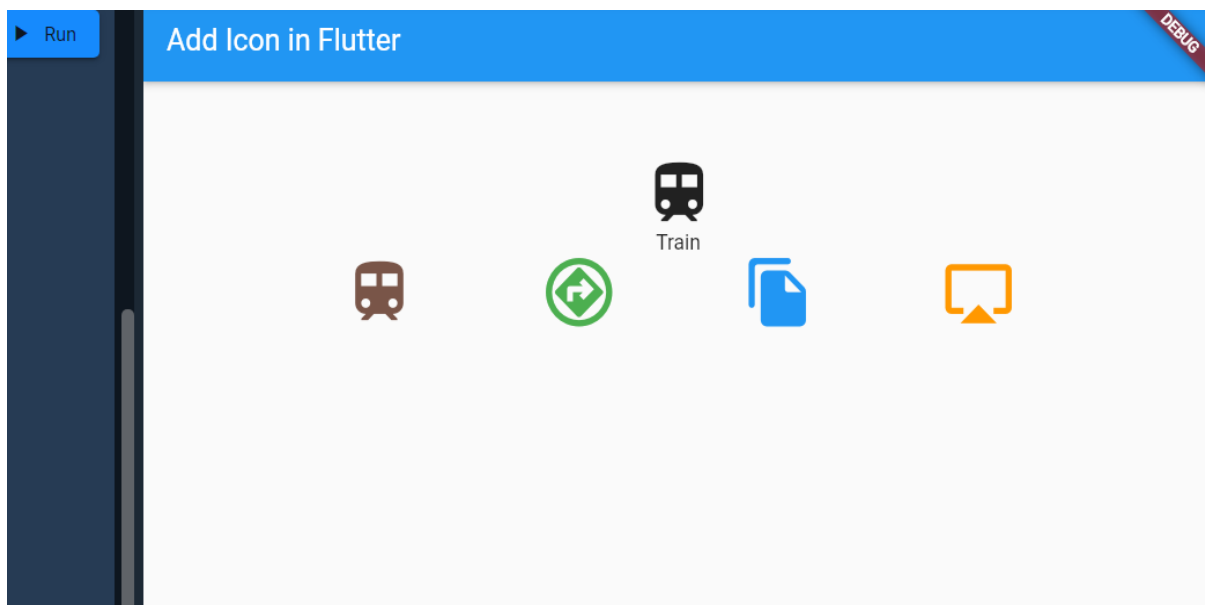
class Home extends StatefulWidget{
  @override
  _HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Add Icon in Flutter"),
      ),
      body: Container(
        padding: EdgeInsets.all(50),
```

```

alignment: Alignment.center,
child: Column(
  children:[
    Icon(Icons.train,
      size: 50,
    ), Text('Train'),
    Row(
mainAxisAlignment: MainAxisAlignment.spaceEvenly,
children: [
  Icon(Icons.train,size: 50,color: Colors.brown,),
  Icon(Icons.assistant_direction_outlined,size:
50,color:Colors.green),
  Icon(Icons.file_copy_rounded,size: 50,color:Colors.blue),
  Icon(Icons.airplay,size: 50,color:Colors.orange),
],),)],),);}}

```



### Assignment 7-2 :

```
import 'package:flutter/material.dart';
import 'HomePage.dart';
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: LoginDemo(),
    );
}
class LoginDemo extends StatefulWidget {
  @override
  _LoginDemoState createState() => _LoginDemoState();
}
class _LoginDemoState extends State<LoginDemo> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        title: Text("Login Page"),
      ),
    ),
  ),
}
```

```

body: SingleChildScrollView(
  child: Column(
    children: <Widget>[
      Padding(
        padding: const EdgeInsets.only(top: 60.0),
        child: Center(
          child: Container(
            width: 200,
            height: 150,
            /*decoration: BoxDecoration(
              color: Colors.red,
              borderRadius: BorderRadius.circular(50.0)),*/
            child: Image.asset('asset/images/flutter-logo.png')),
          ),
        ),
      Padding(
        //padding: const EdgeInsets.only(left:15.0,right:
15.0,top:0,bottom: 0),
        padding: EdgeInsets.symmetric(horizontal: 15),
        child: TextField(
          decoration: InputDecoration(
            border: OutlineInputBorder(),
            labelText: 'Email',
            hintText: 'Enter valid email id as abc@gmail.com'),
          ), ),
      Padding(

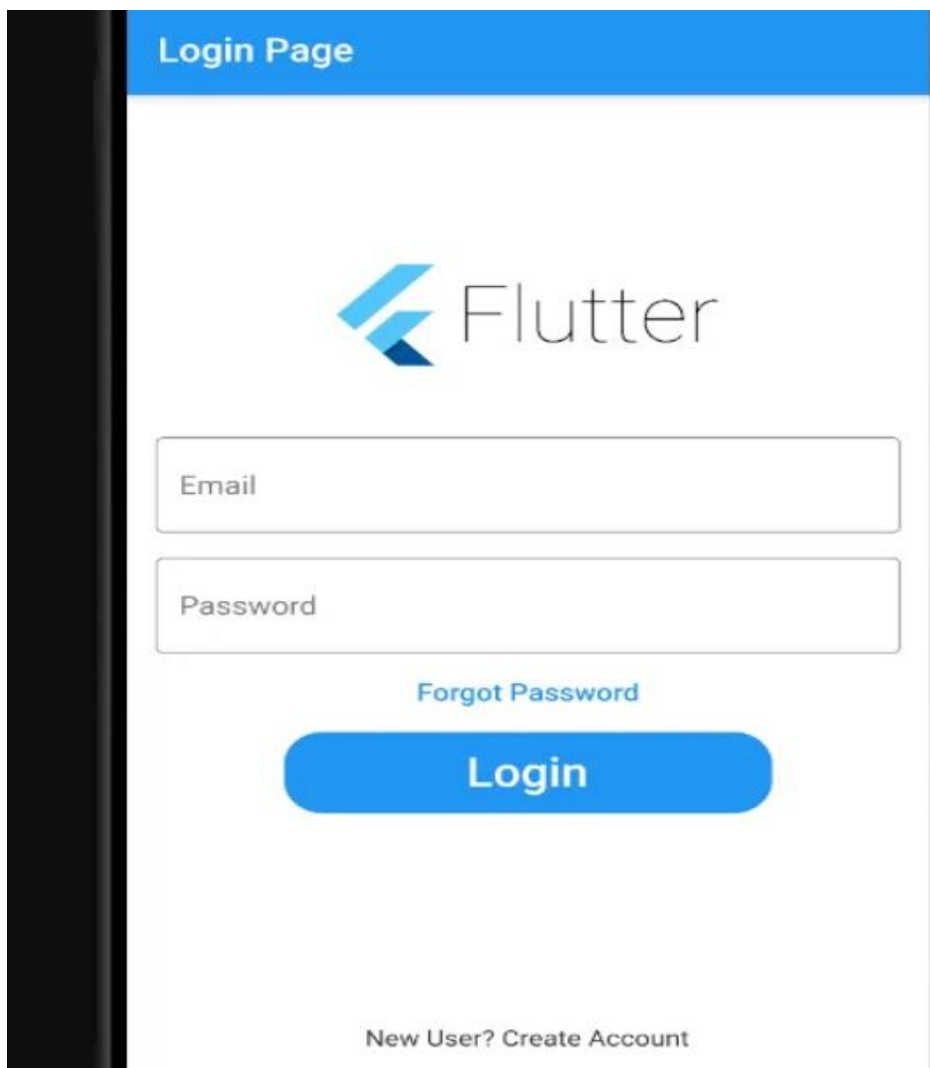
```

```
padding: const EdgeInsets.only(
  left: 15.0, right: 15.0, top: 15, bottom: 0),
//padding: EdgeInsets.symmetric(horizontal: 15),
child: TextField(
  obscureText: true,
  decoration: InputDecoration(
    border: OutlineInputBorder(),
    labelText: 'Password',
    hintText: 'Enter secure password'),
  ),
),
FlatButton(
  onPressed: (){      },
  child: Text(
    'Forgot Password',
    style: TextStyle(color: Colors.blue, fontSize: 15),
  ), ),
Container(
  height: 50,
  width: 250,
  decoration: BoxDecoration(
    color: Colors.blue, borderRadius:
BorderRadius.circular(20)),
  child: FlatButton(
    onPressed: () {
      Navigator.push(
```

```

context, MaterialPageRoute(builder: (_) =>
HomePage()));
    },
    child: Text(
      'Login',
      style: TextStyle(color: Colors.white, fontSize: 25),
    ), ),
    SizedBox( height: 130,),
    Text('New User? Create Account')
  ],),),);}}

```



### Assignment 8-1 :

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

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

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

  @override
  Widget build(BuildContext context) {
    const appTitle = 'Form Validation Demo';
    return MaterialApp(
      title: appTitle,
      home: Scaffold(
        appBar: AppBar(
          title: const Text(appTitle),
        ),
        body: const MyCustomForm(),
      ));
}

class MyCustomForm extends StatefulWidget {
  const MyCustomForm({super.key});

  @override
  MyCustomFormState createState() {
    return MyCustomFormState();
  }
}

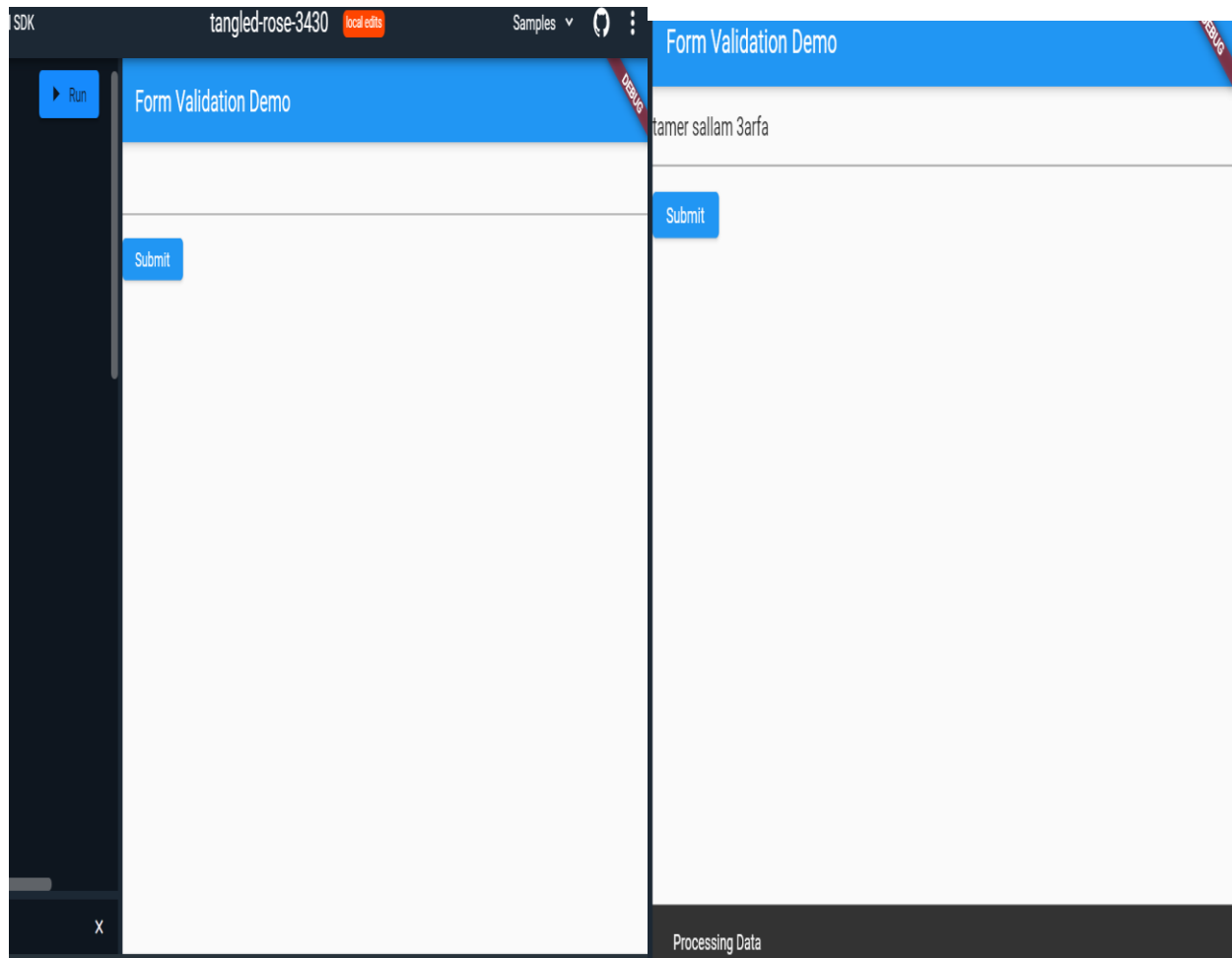
class MyCustomFormState extends State<MyCustomForm> {
  final _formKey = GlobalKey<FormState>();
```



```
@override
Widget build(BuildContext context) {
  return Form(
    key: _formKey,
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        TextFormField(
          validator: (value) {
            if (value == null || value.isEmpty) {
              return 'Please enter some text';
            }
            return null;
          },
        ),
        Padding(
          padding: const EdgeInsets.symmetric(vertical: 16.0),
          child: ElevatedButton(
            onPressed: () {
              if (_formKey.currentState!.validate()) {
                ScaffoldMessenger.of(context).showSnackBar(
                  const SnackBar(content: Text('Processing Data')),
                );
              }
            },
          ),
        ),
      ],
    ),
  );
}
```

```

        );
    }
},
child: const Text('Submit'),)
),],
),)
;}}
```



## Assignment 8-2 :

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

void main() => runApp(const SwitchApp());

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(useMaterial3: true),
      home: Scaffold(
        appBar: AppBar(title: const Text('Accessibility Test')),
        body: const Center(
          child: SwitchExample(),
        ),
      ),);}}

class SwitchExample extends StatefulWidget {
  const SwitchExample({super.key});

  @override
  State<SwitchExample> createState() =>
    _SwitchExampleState();
}

class _SwitchExampleState extends State<SwitchExample> {
  bool light0 = true;
  bool isChecked = true;
```

```

final MaterialStateProperty<Icon?> thumbIcon =
    MaterialStateProperty.resolveWith<Icon?>{
        (Set<MaterialState> states) {
            if (states.contains(MaterialState.selected)) {
                return const Icon(Icons.check);
            }
            return const Icon(Icons.close);
        }, );
@override
Widget build(BuildContext context) {
    return Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: <Widget>[
            Row(
                children: <Widget>[
                    Text('    Switch '),
                    Container(
margin: const EdgeInsets.fromLTRB(500,10,10,10),
padding: const EdgeInsets.all(10.0),
decoration: BoxDecoration(
    border: Border.all(color: Colors.white)
),
child: Switch(
    value: light0,
    onChanged: (bool value) {

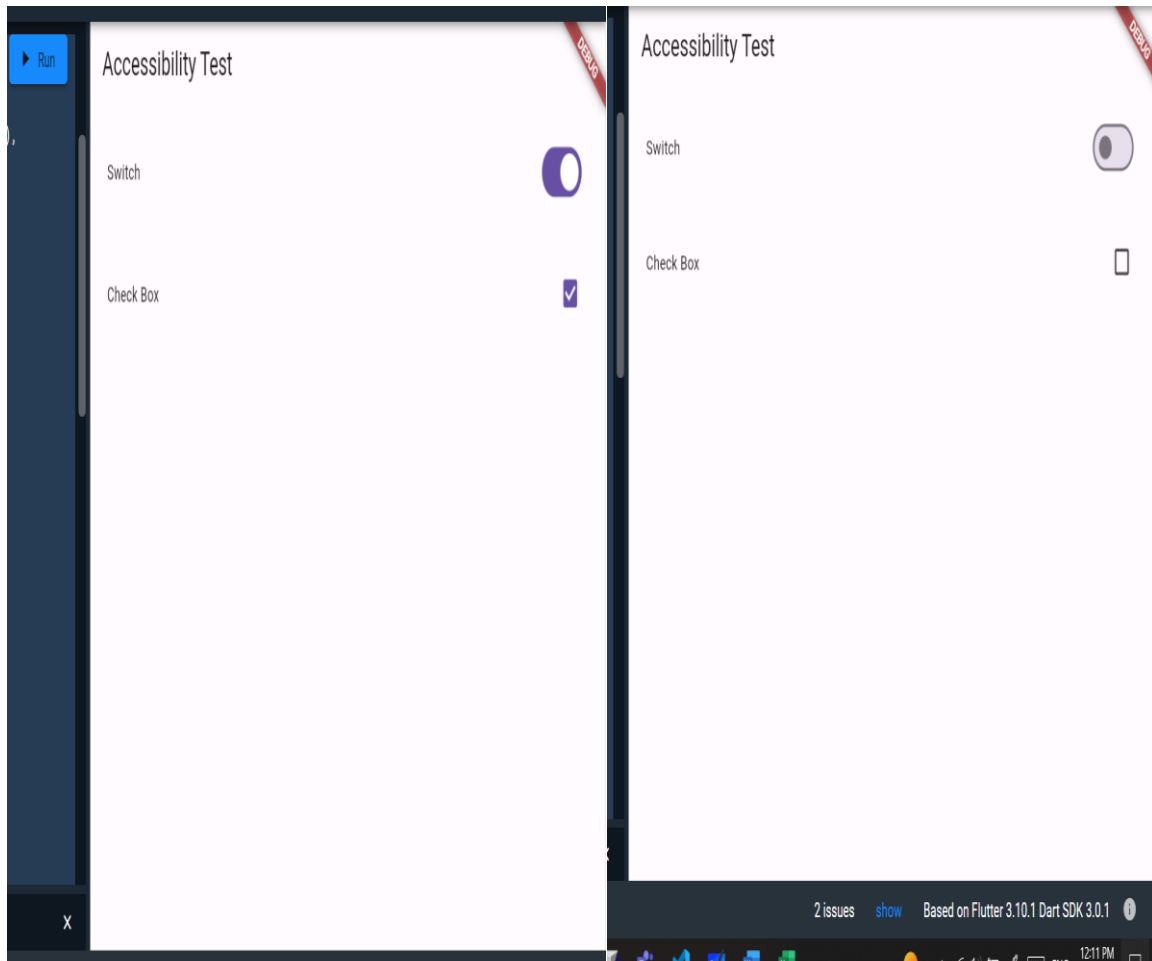
```

```

        setState(() {
          light0 = value;
        });
      },
    ),)
  ]),
  Row(
    children: <Widget>[
      Text('    Check Box  '),
      Container(
margin: const EdgeInsets.fromLTRB(500,10,10,10),
padding: const EdgeInsets.all(10.0),
decoration: BoxDecoration(
  border: Border.all(color: Colors.white)
),
child: Checkbox(
  value: isChecked,
  onChanged: (bool? value) {
    setState(() {
      isChecked = value!;
    });
  }),
)
]
),

```

1,):}}



### Assignment 8-3 :

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'My App',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: Scaffold(
        appBar: AppBar(title: Text('ListTile guide')),
        body: BodyWidget(),
      ),
    );
  }
}

String horseUrl = 'https://i.stack.imgur.com/Dw6f7.png';
String cowUrl = 'https://i.stack.imgur.com/XPOr3.png';
String camelUrl = 'https://i.stack.imgur.com/YN0m7.png';
String sheepUrl = 'https://i.stack.imgur.com/wKzo8.png';
String goatUrl = 'https://i.stack.imgur.com/Qt4JP.png';

class BodyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return ListView(
```

```
children: <Widget>[
  ListTile(
    leading: CircleAvatar(
      backgroundImage: NetworkImage(horseUrl),
    ),
    title: Text('Horse'),
    subtitle: Text('A strong animal'),
    trailing: Icon(Icons.keyboard_arrow_right),
    onTap: () {
      print('horse');
    },
    selected: true,
  ),
  ListTile(
    leading: CircleAvatar(
      backgroundImage: NetworkImage(cowUrl),
    ),
    title: Text('Cow'),
    subtitle: Text('Provider of milk'),
    trailing: Icon(Icons.keyboard_arrow_right),
    onTap: () {
      print('cow');
    },
  ),
  ListTile(
```



```

leading: CircleAvatar(
  backgroundImage: NetworkImage(camelUrl),
),
title: Text('Camel'),
subtitle: Text('Comes with humps'),
trailing: Icon(Icons.keyboard_arrow_right),
onTap: () {
  print('camel');
},
enabled: false,
),
],
);
}
}

```

