

J.N.T.U.H. UNIVERSITY COLLEGE OF ENGINEERING HYDERABAD
(Autonomous)

KUKATPALLY, HYDERABAD – 500 085



Certificate

Certified that this is the bonafide record of the practical work done during

the academic year _____ by

Name _____

Roll Number _____ Class _____

in the Laboratory of _____

of the Department of _____

Signature of the Staff Member

Signature of the Head of the Department

Date of Examination _____

Signature of the Examiner/s

Internal Examiner

External Examiner

KUKATPALLY, HYDERABAD – 500 085

Class _____ *Year* _____ *Laboratory* _____

[illegible]

Page No. _____

1. a) Install Flutter and Dart SDK

Here are the steps to install Flutter with Android Studio:

1. Download and Install Android Studio:
 - Go to the [Android Studio download page](#) and download the latest version.
 - Follow the installation instructions for your operating system.
2. Install Flutter and Dart Plugins:
 - Open Android Studio.
 - Go to File > Settings (or Preferences on macOS).
 - Select Plugins from the sidebar.
 - Search for Flutter and click Install.
 - Click Yes when prompted to install the Dart plugin as well.
 - Restart Android Studio to apply the changes.
3. Set Up Flutter SDK:
 - Download the Flutter SDK from the [Flutter website](#).
 - Extract the downloaded file to a desired location on your system.
 - Add the Flutter bin directory to your system's PATH.
4. Create a New Flutter Project:
 - Open Android Studio.
 - Click on Start a new Flutter project from the welcome screen.
 - Select Flutter Application and click Next.
 - Enter your project name and location, then click Finish.
5. Run Your Flutter App:
 - Connect a physical device or start an emulator.
 - Click the Run button in Android Studio to build and run your Flutter app.

For more detailed instructions, you can refer to the [official Flutter documentation](#)

b) Write a simple Dart program to understand the language basics.

```
import 'dart:io';

void main() {
  print('Enter first number:');
  double num1 = double.parse(stdin.readLineSync());
  print('Enter second number:');
  double num2 = double.parse(stdin.readLineSync());
  print('Choose an operation (+, -, *, /):');
  String operation = stdin.readLineSync();
  double result;
  switch (operation) {
    case '+':
      result = num1 + num2;
      break;
    case '-':
      result = num1 - num2;
      break;
    case '*':
      result = num1 * num2;
      break;
    case '/':
      result = num1 / num2;
      break;
    default:
      print('Invalid operation');
      return;
  }
  print('Result: $result');
}
```

OUTPUT

Enter first number:

5

Enter second number:

3

Choose an operation (+, -, *, /):

+

Result: 8.0

2. a) Explore various Flutter widgets (Text, Image, Container, etc.).

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

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

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

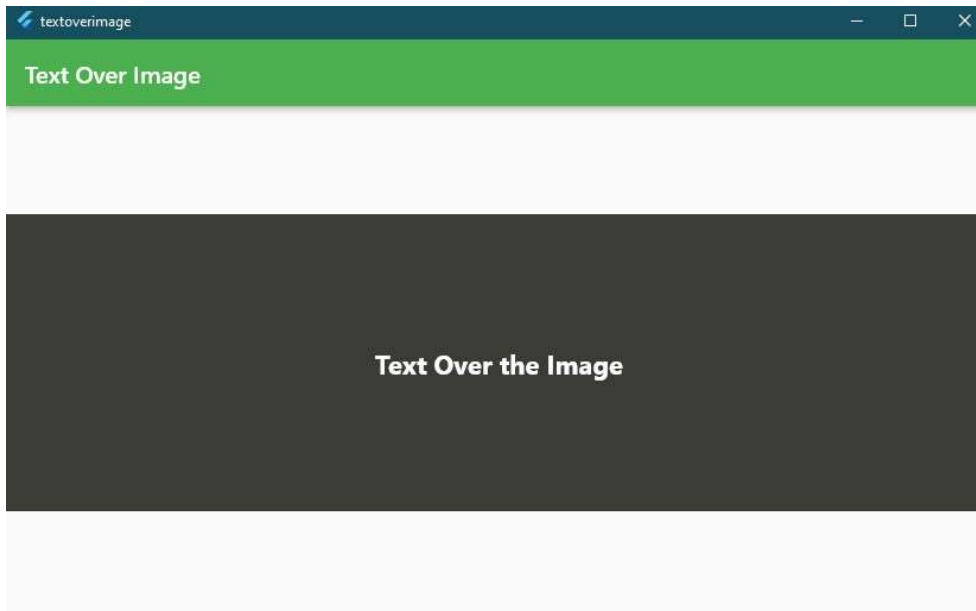
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(primarySwatch: Colors.green),
      home: Scaffold(
        appBar: AppBar(
          title: Text("Text Over Image"),
        ),
        body: Center(
          child: Stack(
            children: [
              Container(
                alignment: Alignment.center,
                child: Image.asset(
                  'assets/s2.png',
                  height: 200,
                  width: double.infinity,
                  fit: BoxFit.cover,
                ),
              ),
              Container(
                alignment: Alignment.center,
```

```

        child: Text(
          'Text Over the Image',
          style: TextStyle(color: Colors.white,
            fontWeight: FontWeight.bold,
            fontSize: 24.0),
        )),
      ],
    ),
  ),
);
}
}

```

OUTPUT



b) Implement different layout structures using Row, Column, and Stack widgets.

```
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('Layout Structures Example'),
        ),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: <Widget>[
                  Container(
                    color: Colors.red,
                    width: 50,
                    height: 50,
                  ),
                  Container(
                    color: Colors.green,
                    width: 50,
                    height: 50,
```

```

    ),
    Container(
      color: Colors.blue,
      width: 50,
      height: 50,
    ),
  ],
),
 SizedBox(height: 20),
 Column(
   mainAxisAlignment: MainAxisAlignment.center,
   children: <Widget>[
     Container(
       color: Colors.yellow,
       width: 50,
       height: 50,
     ),
     Container(
       color: Colors.orange,
       width: 50,
       height: 50,
     ),
     Container(
       color: Colors.purple,
       width: 50,
       height: 50,
     ),
   ],
 ),
  SizedBox(height: 20),

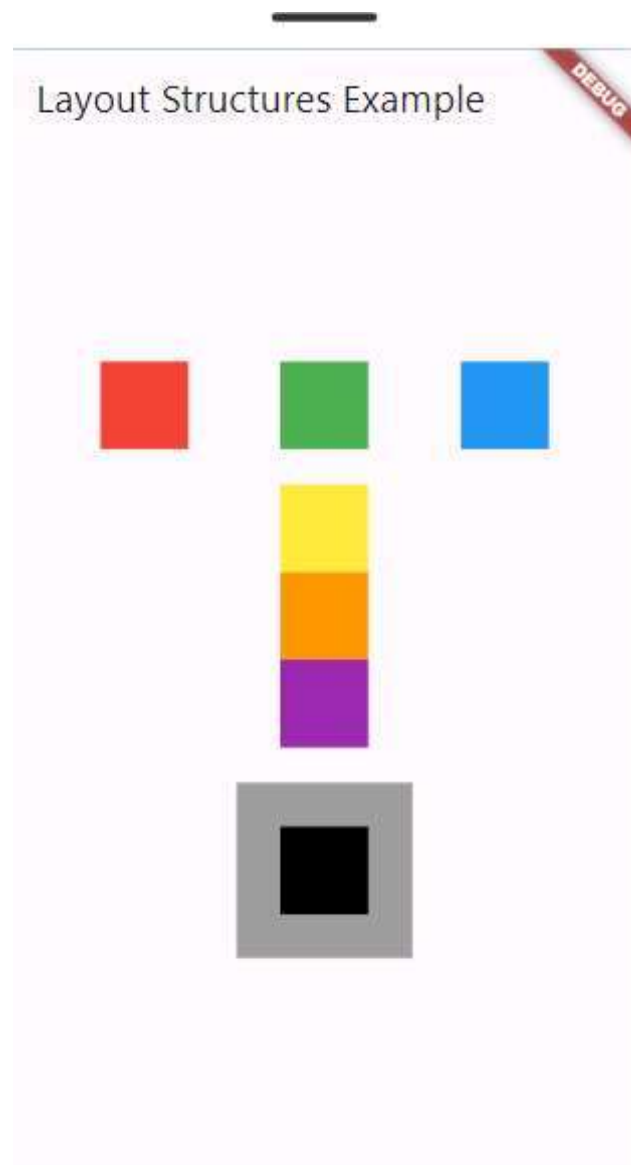
```

```

Stack(
  alignment: Alignment.center,
  children: <Widget>[
    Container(
      color: Colors.grey,
      width: 100,
      height: 100,
    ),
    Container(
      color: Colors.black,
      width: 50,
      height: 50,
    ),
  ],
),
],
),
),
),
);
}
}

```

OUTPUT



3. a) Design a responsive UI that adapts to different screen sizes.

```
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('Responsive UI Example'),
        ),
        body: ResponsiveLayout(),
      ),
    );
  }
}

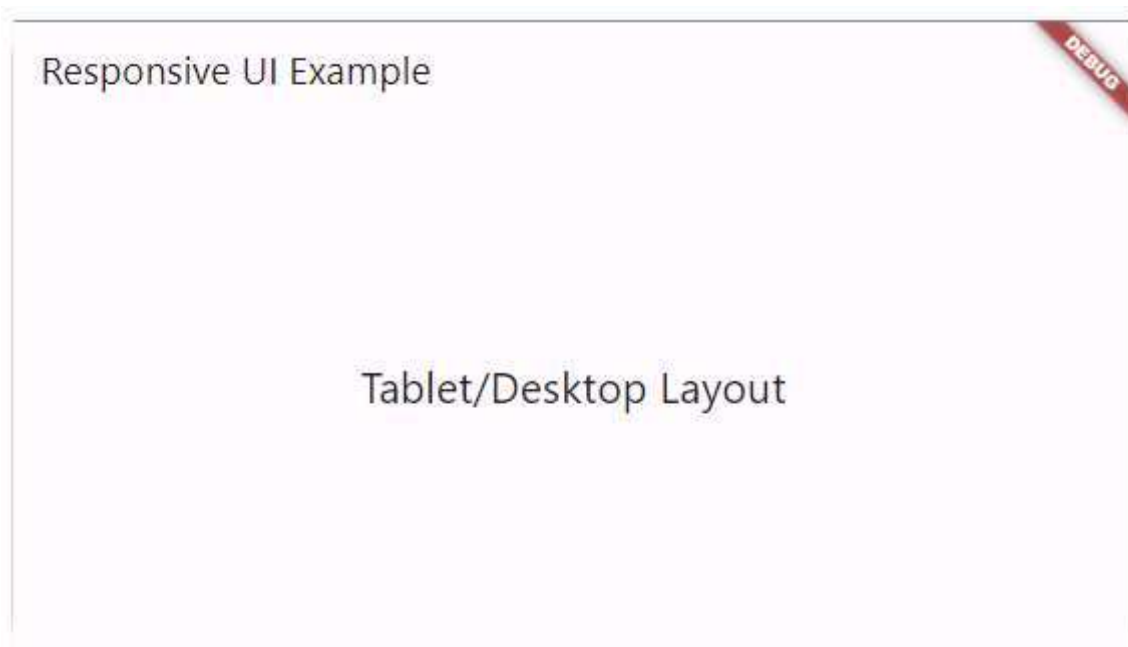
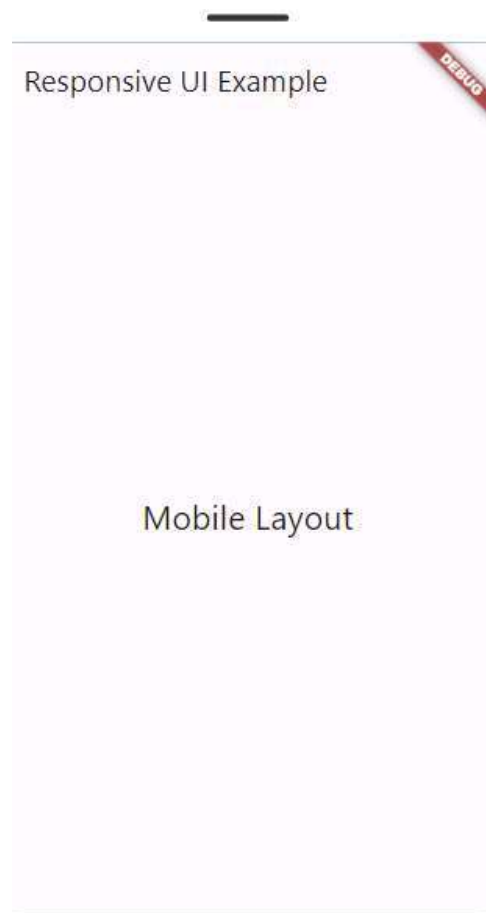
class ResponsiveLayout extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return LayoutBuilder(
      builder: (context, constraints) {
        if (constraints.maxWidth < 600) {
          return MobileLayout();
        } else {
          return TabletDesktopLayout();
        }
      },
    );
  }
}
```

```

    }
}
class MobileLayout extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(
      child: Text(
        'Mobile Layout',
        style: TextStyle(fontSize: 24),
      ),
    );
  }
}
class TabletDesktopLayout extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(
      child: Text(
        'Tablet/Desktop Layout',
        style: TextStyle(fontSize: 24),
      ),
    );
  }
}

```

OUTPUT



b) Implement media queries and breakpoints for responsiveness.

```
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('Responsive UI with MediaQuery'),
        ),
        body: ResponsiveLayout(),
      ),
    );
  }
}

class ResponsiveLayout extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var screenWidth = MediaQuery.of(context).size.width;
    if (screenWidth < 600) {
      return MobileLayout();
    } else if (screenWidth < 1200) {
      return TabletLayout();
    } else {
      return DesktopLayout();
    }
  }
}
```



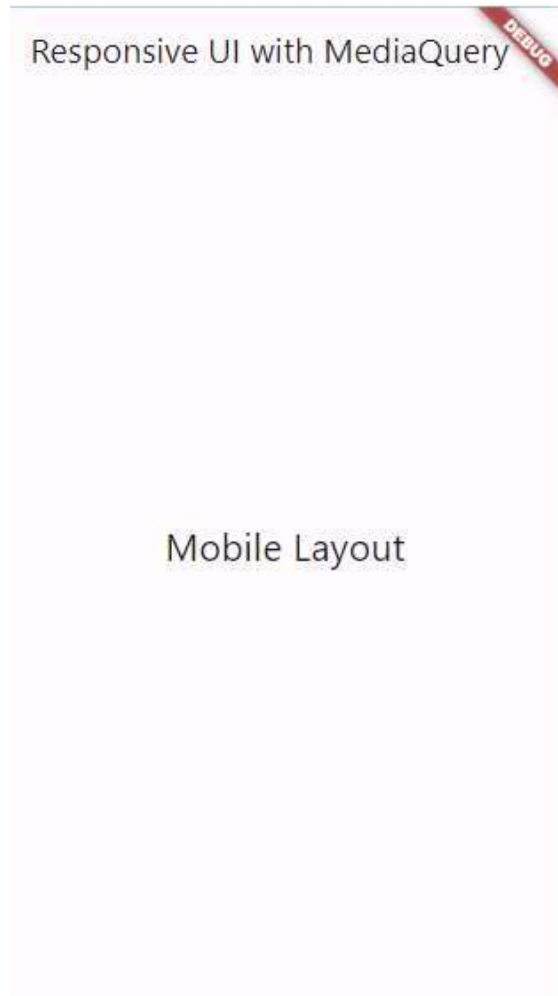
```

}
class MobileLayout extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(
      child: Text(
        'Mobile Layout',
        style: TextStyle(fontSize: 24),
      ),
    );
  }
}
class TabletLayout extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(
      child: Text(
        'Tablet Layout',
        style: TextStyle(fontSize: 24),
      ),
    );
  }
}
class DesktopLayout extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Center(
      child: Text(
        'Desktop Layout',
        style: TextStyle(fontSize: 24),

```

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

OUTPUT



4. a) Set up navigation between different screens using Navigator.

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Navigation Example',
      home: FirstScreen(),
    );
  }
}

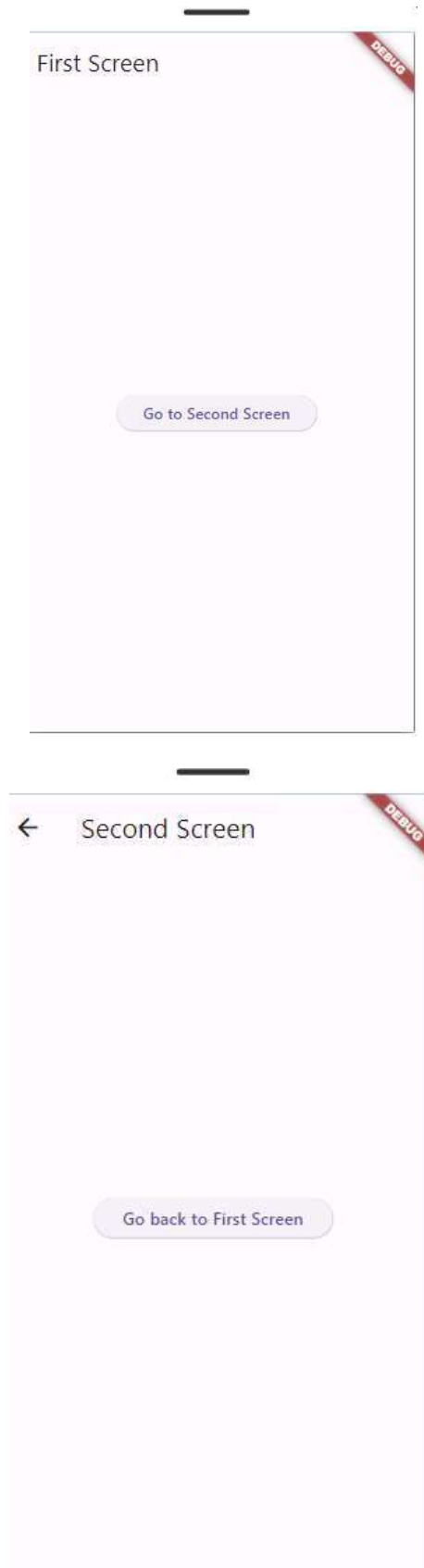
class FirstScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('First Screen'),
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => SecondScreen()),
            );
          },
          child: Text('Go to Second Screen'),
        ),
      ),
    );
  }
}
```

```

        ),
    ),
);
}
}
class SecondScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Second Screen'),
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: Text('Go back to First Screen'),
        ),
      ),
    );
  }
}

```

OUTPUT



b) Implement navigation with named routes.

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Named Routes Example',
      // Define the routes
      routes: {
        '/': (context) => FirstScreen(),
        '/second': (context) => SecondScreen(),
      },
      initialRoute: '/',
    );
  }
}

class FirstScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('First Screen'),
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Navigator.pushNamed(context, '/second');
          },
        ),
      ),
    );
  }
}
```

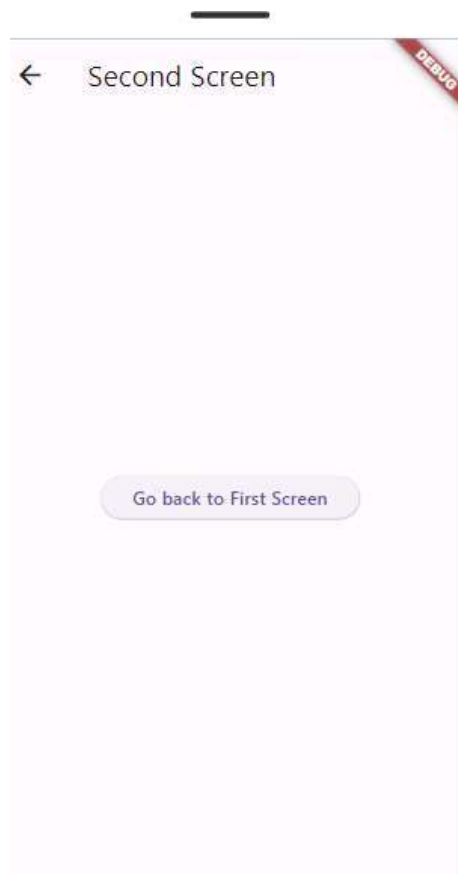
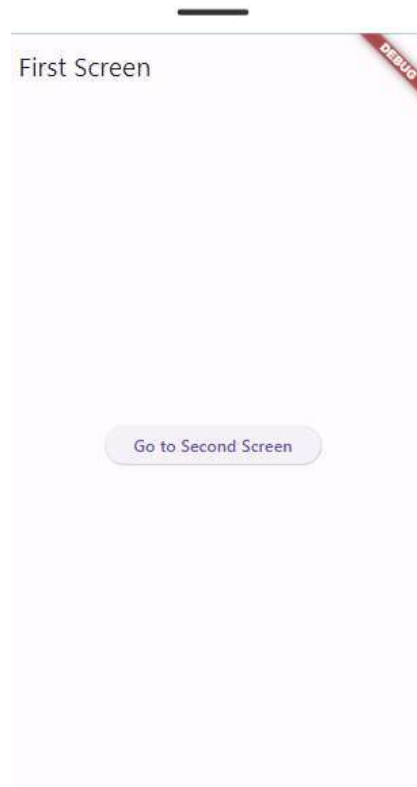
```

        },
        child: Text('Go to Second Screen'),
      ),
    ),
  );
}
}

class SecondScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Second Screen'),
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: Text('Go back to First Screen'),
        ),
      ),
    );
  }
}

```

OUTPUT



5. a) Learn about stateful and stateless widgets.

Stateless Widget

```
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('Stateless Widget Example'),
        ),
        body: Center(
          child: MyStatelessWidget(),
        ),
      ),
    );
  }
}

class MyStatelessWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Text(
      'Hello, I am a Stateless Widget!',
      style: TextStyle(fontSize: 24),
    );
  }
}
```

```
}
```

OUTPUT



Stateful Widget

```
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('Stateful Widget Example'),
        ),
        body: Center(
          child: MyStatefulWidget(),
        ),
      ),
    );
  }
}

class MyStatefulWidget extends StatefulWidget {
  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}

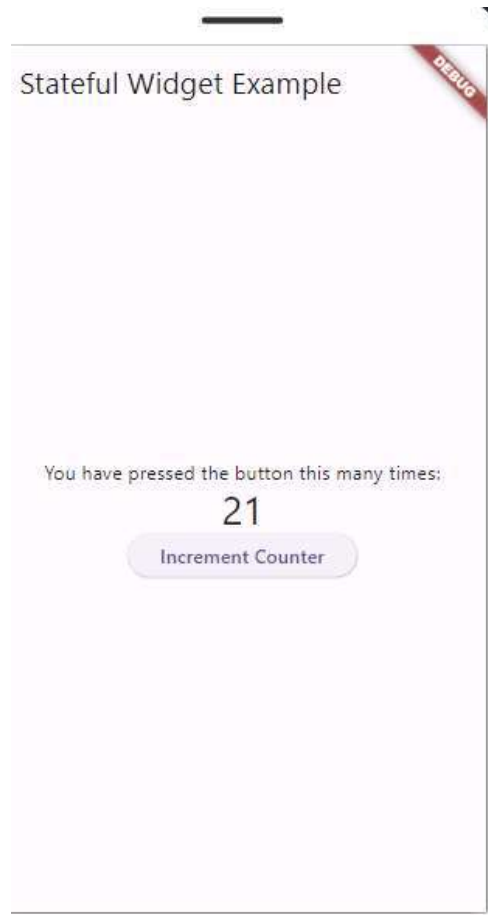
class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  int _counter = 0;
  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }
}
```

```

    }
    @override
    Widget build(BuildContext context) {
      return Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Text(
            'You have pressed the button this many times:',
          ),
          Text(
            '$_counter',
            style: Theme.of(context).textTheme.headline4,
          ),
          ElevatedButton(
            onPressed: _incrementCounter,
            child: Text('Increment Counter'),
          ),
        ],
      );
    }
  }
}

```

OUTPUT



b) Implement state management using set State and Provider.

Using set State

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

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

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

class CounterScreen extends StatefulWidget {
  @override
  _CounterScreenState createState() => _CounterScreenState();
}

class _CounterScreenState extends State<CounterScreen> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

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

```

appBar: AppBar(
  title: Text('Counter using setState'),
),
body: Center(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      Text(
        'You have pressed the button this many times:',
      ),
      Text(
        '$_counter',
        style: Theme.of(context).textTheme.headline4,
      ),
    ],
  ),
),
floatingActionButton: FloatingActionButton(
  onPressed: _incrementCounter,
  tooltip: 'Increment',
  child: Icon(Icons.add),
),
);
}
}

```

OUTPUT



Using Provider

pubsec.yaml

dependencies:

flutter:

 sdk: flutter

provider: ^6.0.0

main.dart

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

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

```
void main() {
```

```
  runApp(
```

```
    ChangeNotifierProvider(
```

```
      create: (context) => Counter(),
```

```
      child: MyApp(),
```

```
    ),
```

```
  );
```

```
}
```

```
class MyApp extends StatelessWidget {
```

```
  @override
```

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

```
    return MaterialApp(
```

```
      home: CounterScreen(),
```

```
    );
```

```
  }
```

```
}
```

```
class Counter with ChangeNotifier {
```

```
  int _count = 0;
```

```
  int get count => _count;
```

```
  void increment() {
```

```

        _count++;
        notifyListeners();
    }
}

class CounterScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final counter = Provider.of<Counter>(context);
    return Scaffold(
      appBar: AppBar(
        title: Text('Counter using Provider'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text(
              'You have pressed the button this many times:',
            ),
            Text(
              '${counter.count}',
              style: Theme.of(context).textTheme.headline4,
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: counter.increment,
        tooltip: 'Increment',
        child: Icon(Icons.add),
      ),
    );
  }
}

```

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

6. a) Create custom widgets for specific UI elements.

Custom Card Widget

```
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('Custom Widgets Example'),
        ),
        body: Center(
          child: CustomCard(
            title: 'Card Title',
            description: 'This is a custom card widget.',
          ),
        ),
      ),
    );
  }
}

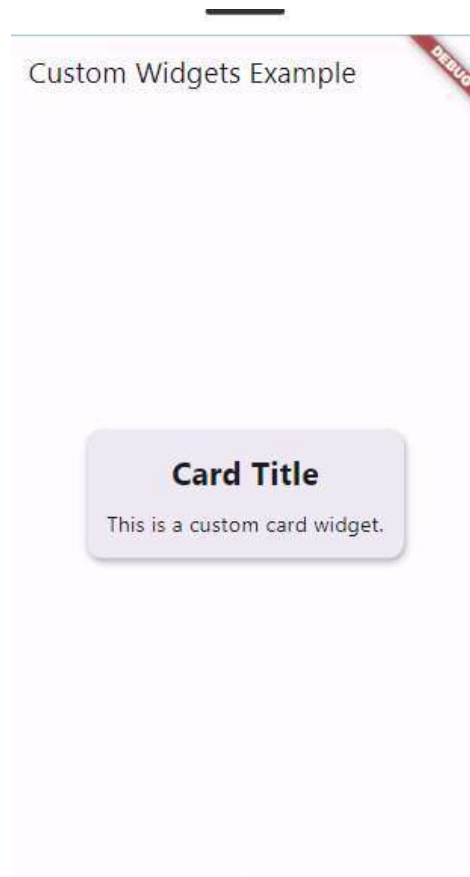
class CustomCard extends StatelessWidget {
  final String title;
  final String description;
  CustomCard({required this.title, required this.description});
```

```

@override
Widget build(BuildContext context) {
  return Card(
    elevation: 5,
    margin: EdgeInsets.all(10),
    child: Padding(
      padding: EdgeInsets.all(15),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.min,
        children: <Widget>[
          Text(
            title,
            style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
          ),
          SizedBox(height: 10),
          Text(
            description,
            style: TextStyle(fontSize: 16),
          ),
        ],
      ),
    ),
  );
}

```

OUTPUT



b) Apply styling using themes and custom styles.

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      // hide the debug banner
      debugShowCheckedModeBanner: false,
      title: 'KindaCode.com',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        // Override some of the default text styles
        textTheme: const TextTheme(
          titleLarge: TextStyle(fontSize: 50, color: Colors.purple),
          titleMedium: TextStyle(fontSize: 30, color: Colors.red),
          titleSmall: TextStyle(fontSize: 24, color: Colors.white),
          bodyMedium: TextStyle(fontSize: 18, color: Colors.green),
          bodySmall: TextStyle(
            fontSize: 14,
            color: Colors.indigo,
            fontWeight: FontWeight.bold,
            decoration: TextDecoration.underline)),
        ),
      home: const MyHomePage(),
    );
  }
}
```

```

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          'KindaCode.com',
          style: Theme.of(context).textTheme.titleSmall,
        ),
      ),
      body: Padding(
        padding: const EdgeInsets.all(30),
        child: Column(crossAxisAlignment: CrossAxisAlignment.start, children: [
          Text(
            'Title Large',
            style: Theme.of(context).textTheme.titleLarge,
          ),
          Text(
            'Title Medium',
            style: Theme.of(context).textTheme.titleMedium,
          ),
          Text(
            'Body Medium',
            style: Theme.of(context).textTheme.bodyMedium,
          ),
          Text(
            'Body Small',

```



```
        style: Theme.of(context).textTheme.bodySmall,  
      ),  
    ],  
  ),  
);  
}  
}
```

OUTPUT

