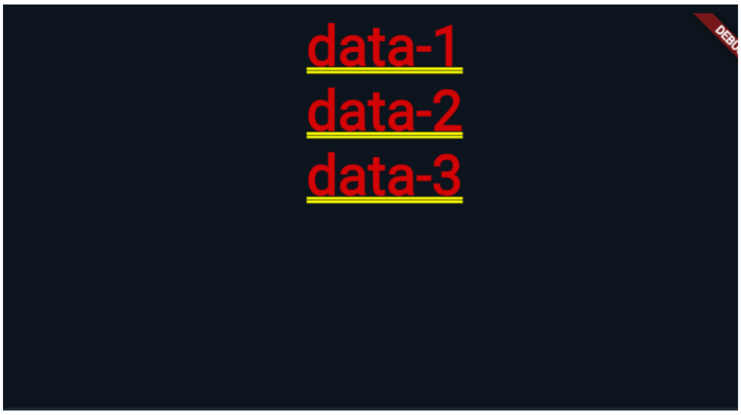
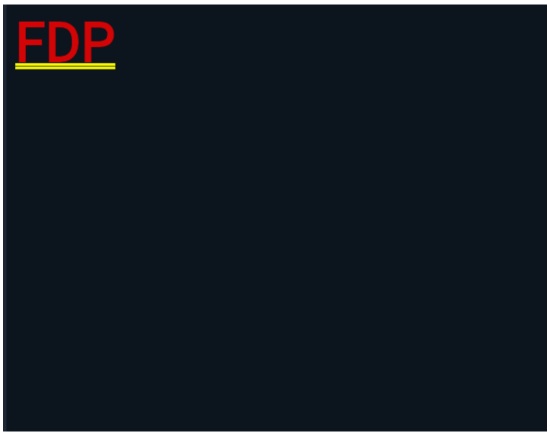
5 a) Learn about stateful and stateless widgets

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

//main function

void main() {

 runApp(MyApp());

}

class MyApp extends StatefulWidget {

 const MyApp({super.key});

 @override

 State<MyApp> createState() => _MyAppState();

}

class _MyAppState extends State<MyApp> {

 @override

 Widget build(BuildContext context) {

  return MaterialApp(

   home: Column(

    children: [Text('data-1'), Text('data-2'), Text('data-3')],

   ),

  );

 }

}
```



```
data-1
data-2
data-3
```

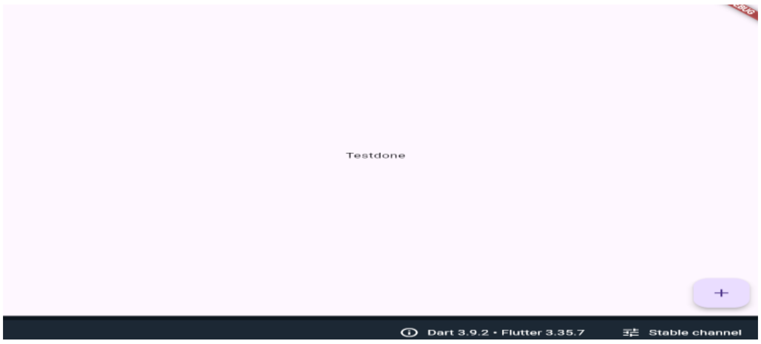
```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Text('FDP'),
    );
  }
}
```



FDP

5 b) Implement state management using set State and Provider.

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

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

class MyApp extends StatefulWidget {
  const MyApp({super.key});
  @override
  State<MyApp> createState() => _MyAppState();
}


class _MyAppState extends State<MyApp> {
  String text = "Test";
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        body: Center(
          child: Text(text),
        ),
        floatingActionButton: FloatingActionButton(
          child: Icon(Icons.add),
          onPressed: () => setState(() => text = "Testdone"),
        ),
      ),
    );
  }
}
```

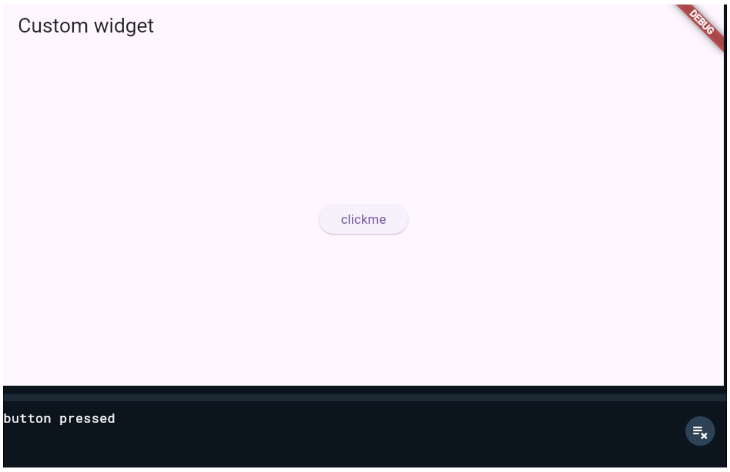


Testdone

6a) Create custom widgets for specific UI elements.

```
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(
      home: Scaffold(
        appBar: AppBar(
          title:Text("Custom widget"),
        ),
        body:Center(
          child: Mybutton("clickme"),
        ),
      ),
    );
  }
}

class Mybutton extends StatelessWidget {
  final String text;
Mybutton(this.text);

  @override
  Widget build(BuildContext context) {
    return ElevatedButton(
      onPressed: () {
        print("button pressed");
      },
     child: Text(text),
    );
  }
}
```

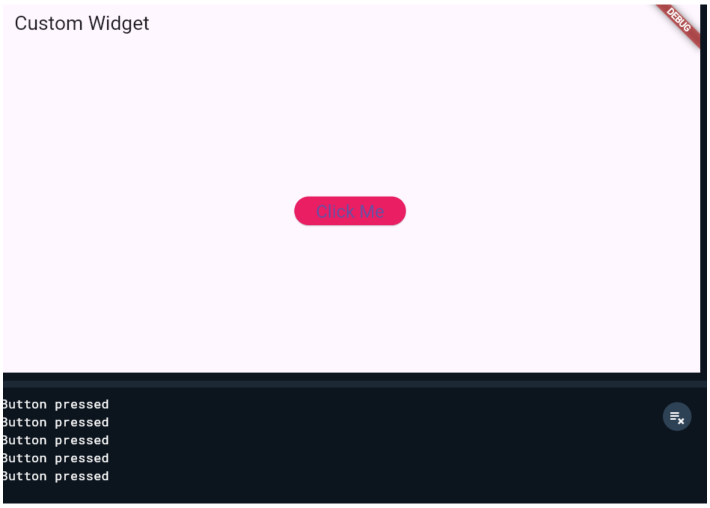


Custom widget

```
button pressed
```

6 b) Apply styling using themes and custom styles.

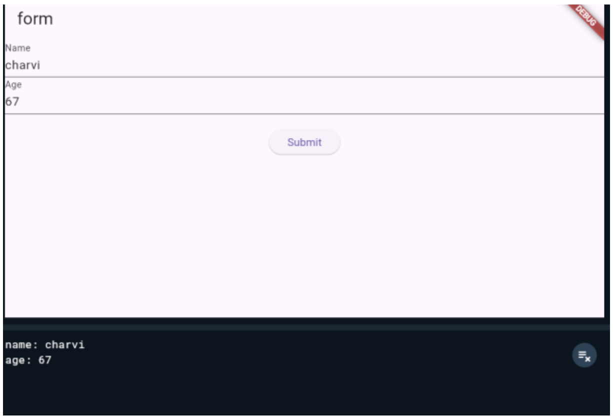
```
import 'package:flutter/material.dart';


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


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


  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        primaryColor: Colors.blue,
        elevatedButtonTheme: ElevatedButtonThemeData(
          style: ElevatedButton.styleFrom(
            backgroundColor: Colors.pink,
            textStyle:TextStyle(fontSize: 20),
          ),
        ),
      ),
      home: Scaffold(
        appBar: AppBar(title:Text("Custom Widget")),
        body: Center(
          child: Mybutton("Click Me"),
        ),
      ),
    );
  }
}

class Mybutton extends StatelessWidget {
  final String text;
  Mybutton(this.text);

  @override
  Widget build(BuildContext context) {
    return ElevatedButton(
      onPressed: () {
        print("Button pressed");
      },
      child: Text(text),
    );
  }
}
```



```
Button pressed
Button pressed
Button pressed
Button pressed
Button pressed
```

7 a) Design a form with various input fields.

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


void main() {

  runApp(const MyApp());

}


class MyApp extends StatefulWidget {

  const MyApp({super.key});


  @override

  State<MyApp> createState() => _MyAppState();

}


class _MyAppState extends State<MyApp> {

   final name =TextEditingController();

   final age  =TextEditingController();


  @override

  Widget build(BuildContext context) {

    return MaterialApp(
   home: Scaffold(
    appBar: AppBar(title: const Text('form')),
    body: Column(
     children: [
      TextField(
       controller: name,
       decoration: const InputDecoration(labelText: 'Name'),
      ),
      TextField(
       controller: age,
       decoration: const InputDecoration(labelText: 'Age'),
      ),
      const SizedBox(height: 20.0),
      ElevatedButton(
       onPressed: () {
        print("name: ${name.text}");
        print("age: ${age.text}");
       },
       child: Text("Submit"),
      ),
     ],
    ),
   ),
  );
 }
}
```



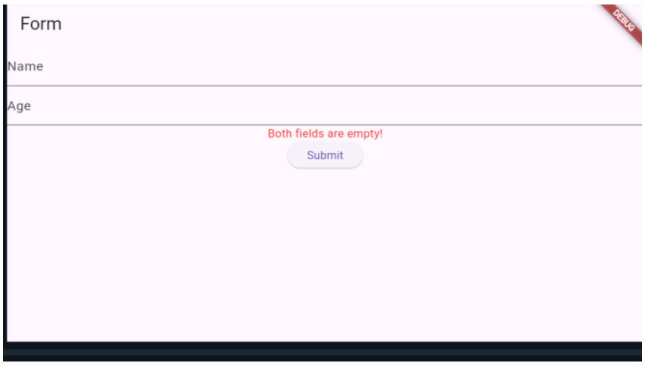
Name
charvi

Age
67

```
name: charvi
age: 67
```

7 b) Implement form validation and error handling.

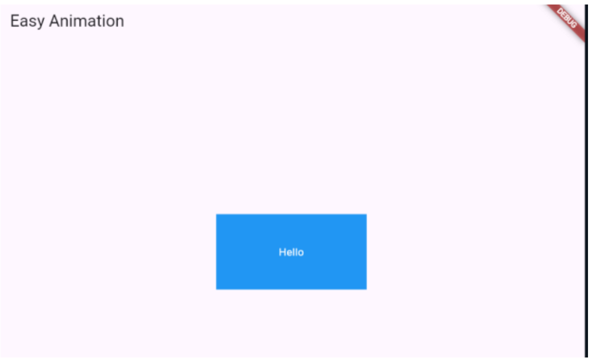
```
import 'package:flutter/material.dart';


void main() => runApp(MaterialApp(home: MyForm()));


class MyForm extends StatefulWidget {
  @override
  State<MyForm> createState() => _MyFormState();
}


class _MyFormState extends State<MyForm> {
  final name = TextEditingController();
  final age = TextEditingController();
  String error = "";


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Form")),
      body: Column(children: [
        TextField(controller: name, decoration: InputDecoration(labelText: "Name")),
    TextField(controller: age, decoration: InputDecoration(labelText: "Age")),
    Text(error, style: TextStyle(color: Colors.red)),
    ElevatedButton(
     child: Text("Submit"),
     onPressed: () {
      setState(() {
       if (name.text.isEmpty && age.text.isEmpty) {
        error = "Both fields are empty!";
       } else if (int.tryParse(name.text) != null) {
        error = "Name cannot have numbers!";
       } else if (int.tryParse(age.text) == null) {
        error = "Age must be numbers!";
       } else {
        print("Name: ${name.text}, Age: ${age.text}");
       }
      });
     },
    )
   ]),
  );
 }
}
```



Both fields are empty!

8 a) Add animations to UI elements using Flutter's animation framework.

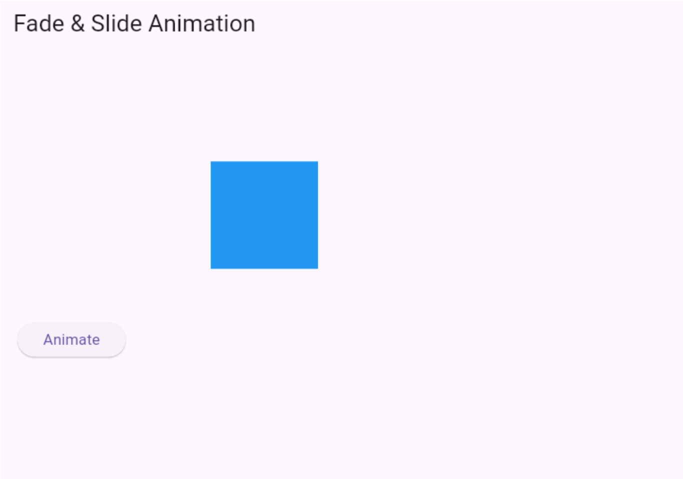
```
import 'package:flutter/material.dart';

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

class MyApp extends StatefulWidget {
  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  double width = 100;

  @override
  void initState() {
    super.initState();
    // Animate after build
    Future.delayed(Duration.zero, () {
      setState(() {
        width = 200; // final width
      });
    });
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text("Easy Animation")),
        body: Center(
          child: AnimatedContainer(
      duration: Duration(seconds: 2),
      width: width,
      height: 100,
      color: Colors.blue,
      child: Center(
       child: Text("Hello", style: TextStyle(color: Colors.white)),
      ),
     ),
    ),
   ),
  );
 }
}
```



8 b) Experiment with different types of animations (fade, slide)

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


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


class MyApp extends StatefulWidget {
 @override
 State<MyApp> createState() => _MyAppState();
}
```

```
class _MyAppState extends State<MyApp> {
 double opacity = 0.0;
 double left = 0.0;

 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   home: Scaffold(
    appBar: AppBar(title: Text("Fade & Slide Animation")),
    body: Stack(
     children: [
      AnimatedPositioned(
       duration: Duration(seconds: 1),
       left: left,
       top: 100,
       child: AnimatedOpacity(
        duration: Duration(seconds: 1),
        opacity: opacity,
        child: Container(width: 100, height: 100, color: Colors.blue),
       ),
      ),
      Positioned(
       top: 250,
       left: 20,
       child: ElevatedButton(
        child: Text("Animate"),
        onPressed: () => setState(() {
         opacity = opacity == 0 ? 1 : 0;
         left = left == 0 ? 200 : 0;
        }),
              ),
            )
          ],
        ),
      ),
    );
  }
}
```


Fade & Slide Animation
Animate