5 a) Learn about stateful and stateless widgets

5.a)1.

```
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> {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        body: Column(children:[Text('data1'),Text('data2'),Text('data3')],),
      ),
    );
  }
}
```

5.a) 2

```
import 'package:flutter/material.dart';
void main() => runApp(MyApp());
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        body: Text("hello"),
      ),
    );}}
```

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

```
import 'package:flutter/material.dart';
void main() => runApp(MyApp());

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

class _MyAppState extends State<MyApp> {
  String value = "Test";
  void clickMe() {
    setState(() {
      value = "Test done";
    });
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('Simple Flutter App')),
        body: Center(child: Text(value)),
        floatingActionButton: FloatingActionButton(
          onPressed: clickMe,
          child: Icon(Icons.add),
        ),
      ),
    );
  }
}
```

6 a) Create custom widgets for specific UI elements.

6 b) Apply styling using themes and custom styles.

```
import 'package:flutter/material.dart';
void main() => runApp(MyApp());

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

class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Custom Widget Example")),
      body: Center(
        child: IconTextButton(
          icon: Icons.thumb_up,
          text: 'Like',
          onPressed: () {
            print('Button Pressed!');
          },
        ),
      ),
    );
  }
}

class IconTextButton extends StatelessWidget {
  IconData icon;
  String text;
  VoidCallback onPressed;

  IconTextButton({
    required this.icon,
    required this.text,
    required this.onPressed,
  });

  @override
  Widget build(BuildContext context) {
    return ElevatedButton.icon(
      onPressed: onPressed,
      icon: Icon(icon),
      label: Text(text),
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.pink,      // add the highlighted part for 6.b question
        foregroundColor: Colors.black,
      ),
    );
  }
}
```

7 a) Design a form with various input fields.

7 b) Implement form validation and error handling.

```
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('Simple Form')),
    body: SimpleForm(),
   ),
  );
 }
}
class SimpleForm extends StatefulWidget {
 @override
 _SimpleFormState createState() => _SimpleFormState();
}
class _SimpleFormState extends State<SimpleForm> {
 final _formKey = GlobalKey<FormState>(); // added for validation
 final nameController = TextEditingController();
 final ageController = TextEditingController();
 @override
 Widget build(BuildContext context) {
  return Padding(
   padding: EdgeInsets.all(16),
   child: Form(
    key: _formKey,
    child: Column(
     children: [
            TextFormField(
              controller: nameController,
              decoration: InputDecoration(labelText: 'Name'),
              validator: (value) =>value!.isEmpty ? "Please enter your name" : null,
            ),
            TextFormField(
              controller: ageController,
              decoration: InputDecoration(labelText: 'Age'),
              validator: (value) {
                if (value!.isEmpty) return "Please enter age";
                final age = int.tryParse(value);
                return (age == null || age <= 0)? "Enter a valid age": null;
              },
            ),
            ElevatedButton(
              onPressed: () {
                if (_formKey.currentState!.validate()) {
                  print("Name: ${nameController.text}, Age: ${ageController.text}");
                }
              },
              child: Text('Submit'),
            ),
          ],
        ),
      ),
    );
  }
}
```