Q) Slider widget to change the value of setstate.

import 'package:flutter/material.dart';

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

class SliderApp extends StatelessWidget {

const SliderApp({super.key});

@override

Widget build(BuildContext context) {

return const MaterialApp(

home: SliderExample(),

);

}

}

class SliderExample extends StatefulWidget {

const SliderExample({super.key});

@override

State<SliderExample> createState() => \_SliderExampleState();

}

class \_SliderExampleState extends State<SliderExample> {

double \_currentSliderValue = 20;

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: const Text('Slider')),

body: Slider(

value: \_currentSliderValue,

max: 100,

divisions: 5,

label: \_currentSliderValue.round().toString(),

onChanged: (double value) {

setState(() {

\_currentSliderValue = value;

});

},

),

);

}

**Q ) To Implement TapBar**

import 'package:flutter/material.dart';

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

class TabBarApp extends StatelessWidget {

const TabBarApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

theme: ThemeData(useMaterial3: true),

home: const TabBarExample(),

);

}

}

class TabBarExample extends StatelessWidget {

const TabBarExample({super.key});

@override

Widget build(BuildContext context) {

return DefaultTabController(

initialIndex: 1,

length: 3,

child: Scaffold(

appBar: AppBar(

title: const Text('TabBar Sample'),

bottom: const TabBar(

tabs: <Widget>[

Tab(

icon: Icon(Icons.cloud\_outlined),

),

Tab(

icon: Icon(Icons.beach\_access\_sharp),

),

Tab(

icon: Icon(Icons.brightness\_5\_sharp),

),

],

),

),

body: const TabBarView(

children: <Widget>[

Center(

child: Text("It's cloudy here"),

),

Center(

child: Text("It's rainy here"),

),

Center(

child: Text("It's sunny here"),

),

],

),

),

);

}

}

**Q)Implement navigation with named routers**

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(

debugShowCheckedModeBanner: false,

initialRoute: '/',

routes: {

'/': (context) => const HomePage(),

'/screen1': (context) => const Screen1(),

'/screen2': (context) => const Screen2(),

},

);

}

}

class HomePage extends StatelessWidget {

const HomePage({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: const Text('Home Page')),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

ElevatedButton(

onPressed: () => Navigator.pushNamed(context, '/screen1'),

child: const Text('Go to Screen 1'),

),

const SizedBox(height: 16),

ElevatedButton(

onPressed: () => Navigator.pushNamed(context, '/screen2'),

child: const Text('Go to Screen 2'),

),

],

),

),

);

}

}

class Screen1 extends StatelessWidget {

const Screen1({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: const Text('Screen 1')),

body: Center(

child: ElevatedButton(

onPressed: () => Navigator.pop(context),

child: const Text('Back to Home'),

),

),

);

}

}

class Screen2 extends StatelessWidget {

const Screen2({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: const Text('Screen 2')),

body: Center(

child: ElevatedButton(

onPressed: () => Navigator.pop(context),

child: const Text('Back to Home'),

),

),

);

}

}

**Q)**) **Set up navigation between different screens using navigator.**

**Create three files**

main.dart

import 'package:flutter/material.dart';

import 'package:flutter\_application\_1/first.dart';

void main() {

runApp(const MainApp());

}

class MainApp extends StatelessWidget {

const MainApp({super.key});

@override

Widget build(BuildContext context) {

return const MaterialApp(

debugShowCheckedModeBanner: false,

home: FirstPage(),

);

}

}

first.dart

import 'package:flutter/material.dart';

import 'package:flutter\_application\_1/second.dart';

class FirstPage extends StatelessWidget {

const FirstPage({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text("First Page")),

body: Center(

child: Container(

width: double.infinity,

child: ElevatedButton(

child: Text("Click here to navigate"),

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => SecondPage(),

),

);

},

),

),

),

);

}

}

second.dart

import 'package:flutter/material.dart';

class SecondPage extends StatelessWidget {

const SecondPage({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text("Second Page"),

),

body: const Center(

child: Text(

"I am in second Page",

style: TextStyle(

color: Colors.purple,

fontSize: 20,

),

),

),

);

}

}

**Q) Learn about stateful and stateless widgets.**

**1. 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: Text('Hello, Stateless Widget!'),

),

),

);

}

}

**2. Stateful Widget**

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

home: CounterWidget(),

);

}

}

class CounterWidget extends StatefulWidget {

@override

\_CounterWidgetState createState() => \_CounterWidgetState();

}

class \_CounterWidgetState extends State<CounterWidget> {

int \_counter = 0;

void \_incrementCounter() {

setState(() {

\_counter++;

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Stateful Widget Example')),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Text('You have pressed the button this many times:'),

Text(

'$\_counter',

style: Theme.of(context).textTheme.headlineMedium,

),

],

),

),

floatingActionButton: FloatingActionButton(

onPressed: \_incrementCounter,

tooltip: 'Increment',

child: Icon(Icons.add),

),

);

}

}

**Q)Experiment with different types of animations (fade, slide, etc.).**

import 'package:flutter/material.dart';

import 'dart:math';

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

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

home: AnimationDemo(),

);

}

}

class AnimationDemo extends StatefulWidget {

@override

\_AnimationDemoState createState() => \_AnimationDemoState();

}

class \_AnimationDemoState extends State<AnimationDemo> with SingleTickerProviderStateMixin {

late AnimationController \_controller;

late Animation<Offset> \_slideAnimation;

late Animation<double> \_rotationAnimation;

bool \_visible = true;

bool \_isLarge = false;

@override

void initState() {

super.initState();

\_controller = AnimationController(

duration: Duration(seconds: 2),

vsync: this,

);

\_slideAnimation = Tween<Offset>(begin: Offset(0.0, 1.0), end: Offset(0.0, 0.0)).animate(\_controller);

\_rotationAnimation = Tween<double>(begin: 0, end: 2 \* pi).animate(\_controller);

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Combined Animation Demo')),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

ElevatedButton(

onPressed: () {

setState(() {

\_visible = !\_visible;

});

},

child: Text('Fade Animation'),

),

SizedBox(width: 10),

ElevatedButton(

onPressed: () {

\_controller.forward(from: 0.0);

},

child: Text('Slide Animation'),

),

SizedBox(width: 10),

ElevatedButton(

onPressed: () {

setState(() {

\_isLarge = !\_isLarge;

});

},

child: Text('Scale Animation'),

),

],

),

SizedBox(height: 20),

AnimatedOpacity(

opacity: \_visible ? 1.0 : 0.0,

duration: Duration(seconds: 2),

child: Container(

width: 100,

height: 100,

color: Colors.blue,

child: Center(child: Text('Fade In/Out', style: TextStyle(color: Colors.white))),

),

),

SizedBox(height: 20),

SlideTransition(

position: \_slideAnimation,

child: Container(

width: 100,

height: 100,

color: Colors.green,

child: Center(child: Text('Slide Up', style: TextStyle(color: Colors.white))),

),

),

SizedBox(height: 20),

AnimatedScale(

scale: \_isLarge ? 1.5 : 1.0,

duration: Duration(seconds: 1),

child: Container(

width: 100,

height: 100,

color: Colors.red,

child: Center(child: Text('Scale Up/Down', style: TextStyle(color: Colors.white))),

),

),

SizedBox(height: 20),

AnimatedBuilder(

animation: \_controller,

builder: (context, child) {

return Transform.rotate(

angle: \_rotationAnimation.value,

child: Container(

width: 200,

height: 200,

color: Colors.purple,

child: Center(child: Text('Rotate', style: TextStyle(color: Colors.white))),

),

);

},

),

],

),

),

);

}

@override

void dispose() {

\_controller.dispose();

super.dispose();

}

}

**Q) Switch classes**

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('Switch Sample')),

body: const Center(

child: SwitchExample(),

),

),

);

}

}

class SwitchExample extends StatefulWidget {

const SwitchExample({super.key});

@override

State<SwitchExample> createState() => \_SwitchExampleState();

}

class \_SwitchExampleState extends State<SwitchExample> {

bool light = true;

@override

Widget build(BuildContext context) {

return Switch(

// This bool value toggles the switch.

value: light,

activeColor: Colors.red,

onChanged: (bool value) {

// This is called when the user toggles the switch.

setState(() {

light = value;

});

},

);

}

}