Practical no:2:-

**Aim:- hello world pratical**

**Code:-**

Practical no 3:-

**Aim:- To study and create an android app that demonstrate user interface using different type of layout create an android app that demonstrate layouts.**

**main.dart**

**CODE:**

import 'package:flutter/material.dart';  
  
import 'login\_screen.dart';  
void main() {  
 runApp(MyApp());  
}  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title:'Flutter Login',  
 theme: ThemeData(  
 primarySwatch: Colors.*blue*,  
 ),  
 home: LoginScreen(),  
 );  
 }  
}

login\_screen.dart

CODE:

import 'package:flutter/material.dart';  
  
class LoginScreen extends StatefulWidget {  
 @override  
 \_LoginScreenState createState() => \_LoginScreenState();  
}  
  
class \_LoginScreenState extends State<LoginScreen> {  
 final \_usernameController = TextEditingController();  
 final \_passwordController = TextEditingController();  
  
 void \_login() {  
 String username = \_usernameController.text;  
 String password = \_passwordController.text;  
  
 if (username.isEmpty || password.isEmpty) {  
 *// Show an error message if fields are empty* ScaffoldMessenger.*of*(context).showSnackBar(  
 SnackBar(content: Text('Please enter both username and password')),  
 );  
 } else {  
 *// Proceed with login (you can add more logic here)* print('Username: $username, Password: $password');  
 }  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Login'),  
 ),  
 body: Padding(  
 padding: EdgeInsets.all(16.0),  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 crossAxisAlignment: CrossAxisAlignment.stretch,  
 children: <Widget>[  
 Text(  
 'Login to Your Account',  
 textAlign: TextAlign.center,  
 style: TextStyle(fontSize: 24.0, fontWeight: FontWeight.*bold*),  
 ),  
 SizedBox(height: 20.0),  
  
 *// Username Text Field* TextField(  
 controller: \_usernameController,  
 decoration: InputDecoration(  
 labelText: 'Username',  
 border: OutlineInputBorder(),  
 ),  
 ),  
 SizedBox(height: 10.0),  
  
 *// Password Text Field* TextField(  
 controller: \_passwordController,  
 obscureText: true,  
 decoration: InputDecoration(  
 labelText: 'Password',  
 border: OutlineInputBorder(),  
 ),  
 ),  
 SizedBox(height: 20.0),  
  
 *// Login Button* ElevatedButton(  
 onPressed: \_login,  
 child: Text('Login'),  
 style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.symmetric(vertical: 14.0),  
 textStyle: TextStyle(fontSize: 18.0),  
 ),  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
}

practical no:4

**Aim: the create an android app use of keyboard inputs.**

**A)Textview**

**B)EditText**

**C) Auto Complete Textview**

**Code:**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp( MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 debugShowCheckedModeBanner:false,  
 title: 'Flutter Text Widgets Demo',  
 theme: ThemeData(  
 primarySwatch:Colors.*blue*,  
 ),  
 home: MyHomePage(),  
 );  
 }  
}  
  
class MyHomePage extends StatefulWidget {  
 @override  
 \_MyHomePageState createState() => \_MyHomePageState();  
}  
  
class \_MyHomePageState extends State<MyHomePage> {  
 final TextEditingController \_controller=TextEditingController();  
 final List<String> \_suggestions=['Flutter','Dart','Java','Kotlin','Javascript'];  
 String \_selectedItem='';  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Text Widgets Demo'),  
 ),  
 body: Padding(  
 padding:const EdgeInsets.all(16.0),  
 child: ListView(  
 children:[  
 Text(  
 'This is a Textview(Text widget in Flutter)',  
 style:TextStyle(fontSize:24,fontWeight:FontWeight.*bold*),  
 ),  
 SizedBox(height:20),  
 TextField(  
 controller:\_controller,  
 decoration:InputDecoration(  
 labelText:'Enter text',  
 border:OutlineInputBorder(),  
 ),  
 ),  
 SizedBox(height:20),  
 ElevatedButton(  
 onPressed:(){  
 showDialog(  
 context:context,  
 builder:(context)=>AlertDialog(  
 title:Text('Entered Text'),  
 content:Text(\_controller.text),  
 actions:[  
 TextButton(  
 onPressed:(){  
 Navigator.*pop*(context);  
 },  
 child:Text('OK'),  
 ),  
 ],  
 ),  
 );  
 },  
 child:Text('Show Entered Text'),  
 ),  
 SizedBox(height:20),  
 Autocomplete<String>(  
 optionsBuilder:(TextEditingValue textEditingValue){  
 if(textEditingValue.text.isEmpty){  
 return const Iterable<String>.empty();  
 }  
 return \_suggestions.where((suggestion)=>suggestion.toLowerCase().contains(textEditingValue.text.toLowerCase()));  
 },  
 onSelected:(String selection){  
 setState((){  
 \_selectedItem=selection;  
 });  
 },  
 ),  
 SizedBox(height:20),  
  
  
 Text(  
 'You selected:$\_selectedItem',  
 style: TextStyle(fontSize:20,color:Colors.*blue*),  
 ),  
 ],  
 ),  
 ), *// This trailing comma makes auto-formatting nicer for build methods.* );  
 }  
}

**pratical no :5**

**Aim:-To study and create an android app that demonstrate the use of following Buttons elements.**

1. **Button**
2. **ToggleButton**
3. **Radio Button and RadioGroup**
4. **Checkbox**

**CODE:-**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Flutter Button with Image Demo',  
 theme: ThemeData(  
 primarySwatch: Colors.*blue*,  
 ),  
 home: MyHomePage(),  
 );  
 }  
}  
  
class MyHomePage extends StatefulWidget {  
 @override  
 \_MyHomePageState createState() => \_MyHomePageState();  
}  
  
class \_MyHomePageState extends State<MyHomePage> {  
 bool \_checkboxValue = false;  
 bool \_toggleButtonValue = false;  
 int \_radioValue = 0;  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Flutter Button with Image Demo'),  
 ),  
 body: Padding(  
 padding: const EdgeInsets.all(16.0),  
 child: ListView(  
 children: <Widget>[  
  
 Column(  
 children: <Widget>[  
 Text('Choose a number:'),  
 ListTile(  
 title: Text('Option 1'),  
 leading: Radio<int>(  
 value: 0,  
 groupValue: \_radioValue,  
 onChanged: (int? value) {  
 setState(() {  
 \_radioValue = value!;  
 });  
 },  
 ),  
 ),  
 ListTile(  
 title: Text('Option 2'),  
 leading: Radio<int>(  
 value: 1,  
 groupValue: \_radioValue,  
 onChanged: (int? value) {  
 setState(() {  
 \_radioValue = value!;  
 });  
 },  
 ),  
 ),  
 ],  
 ),  
 SizedBox(height: 20),  
  
 *// Checkbox* CheckboxListTile(  
 title: Text('Accept Terms and Conditions'),  
 value: \_checkboxValue,  
 onChanged: (bool? value) {  
 setState(() {  
 \_checkboxValue = value!;  
 });  
 },  
 ),  
 SizedBox(height: 20),  
  
 *// ToggleButton* SwitchListTile(  
 title: Text('Enable Notifications'),  
 value: \_toggleButtonValue,  
 onChanged: (bool value) {  
 setState(() {  
 \_toggleButtonValue = value;  
 });  
 },  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
}

pratical no :6

**Aim: To study and create an android app that demonstrates the use of following Input Controls:**

**A. ProgressBar**

**Code:**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Progress Bar App',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: ProgressBarPage(),

);

}

}

class ProgressBarPage extends StatefulWidget {

@override

\_ProgressBarPageState createState() =>

\_ProgressBarPageState();

}

class \_ProgressBarPageState extends

State<ProgressBarPage> {

double \_progressValue = 0.0;

bool \_isLoading = false;

// Function to simulate progress

void \_startProgress() {

setState(() {

\_isLoading = true;

\_progressValue = 0.0; // Reset progress

});

// Simulating a loading process (incrementing progress)

for (int i = 1; i <= 10; i++) {

Future.delayed(Duration(seconds: i), () {

setState(() {

\_progressValue = i / 10; // Update progress

});

if (i == 10) {

setState(() {

\_isLoading = false; // Loading finished

});

}

});

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Progress Bar Demo'),

),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

mainAxisAlignment:

MainAxisAlignment.center,

children: <Widget>[

// Linear Progress Indicator

Text('Linear Progress Bar'),

SizedBox(height: 20),

\_isLoading

? LinearProgressIndicator(

value: \_progressValue,

minHeight: 10,

)

: ElevatedButton(

onPressed: \_startProgress,

child: Text('Start Progress'),

),

SizedBox(height: 40),

// Circular Progress Indicator

Text('Circular Progress Bar'),

SizedBox(height: 20),

\_isLoading

? CircularProgressIndicator(

value: \_progressValue,

)

: ElevatedButton(

onPressed: \_startProgress,

child: Text('Start Progress'),

),

],

),

),

);

}

}

**B. Spinner**

**Code :**

import 'package:flutter/material.dart';  
void main() {  
  runApp(MyApp());  
}  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Flutter Dropdown Example',  
      theme: ThemeData(  
        primarySwatch: Colors.*blue*,  
      ),  
      home: MyHomePage(),  
    );  
  }  
}  
  class MyHomePage extends StatefulWidget {  
  @override  
  \_MyHomePageState createState() => \_MyHomePageState();  
  }  
class \_MyHomePageState extends State<MyHomePage> {  
  *// List of items in the dropdown* List<String> \_dropdownItems = ['Apple', 'Banana', 'Orange', 'Grapes'];  
  *// Currently selected value* String? \_selectedItem;  
  @override  
  void initState() {  
    super.initState();  
    *// Initialize the first item as selected* \_selectedItem = \_dropdownItems[0];  
  }  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Flutter Dropdown Example'),  
      ),  
      body: Center(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: <Widget>[  
            *// Dropdown Button with larger items* DropdownButton<String>(  
              value: \_selectedItem,  
              onChanged: (String? newValue) {  
                setState(() {  
                  \_selectedItem = newValue;  
                });  
              },  
              items: \_dropdownItems.map<DropdownMenuItem<String>>((  
                  String value) {  
                return DropdownMenuItem<String>(  
                  value: value,  
                  child: Padding(  
                    padding: const EdgeInsets.symmetric(  
                        vertical: 12.0, horizontal: 10.0),  
                    child: Text(  
                      value,  
                      style: TextStyle(fontSize: 24), *// Increased font size* ),  
                  ),  
                );  
              }).toList(),  
              iconSize: 40,  
              *// Increase the dropdown icon size* style: TextStyle(fontSize: 24),  
              *// Font size for the selected item* dropdownColor: Colors.*blue*[50],  
              *// Optional: background color for the dropdown* isExpanded: true, *// Make the dropdown take up the full width* ),  
            SizedBox(height: 20),  
            *// Display selected value* Text(  
              'Selected Item: $\_selectedItem',  
              style: TextStyle(fontSize: 20),  
            ),  
          ],  
        ),  
      ),  
    );  
  }  
}

**practical no:-7**

**Explore different phases of Activity Lifecycle. Create an android app that demonstrates Activity Lifecycle and Instance State.**

**Code:**-

import 'package:flutter/material.dart';  
import 'package:flutter/widgets.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Activity Lifecycle Demo',  
 theme: ThemeData(  
 primarySwatch: Colors.*blue*,  
 ),  
 home: ActivityLifecycleDemo(),  
 );  
 }  
}  
class ActivityLifecycleDemo extends StatefulWidget {  
 @override  
 \_ActivityLifecycleDemoState createState() => \_ActivityLifecycleDemoState();  
}  
class \_ActivityLifecycleDemoState extends State<ActivityLifecycleDemo> with WidgetsBindingObserver {  
 String \_lifecycleState = '';  
 @override  
 void initState() {  
 super.initState();  
 WidgetsBinding.*instance*.addObserver(this);  
 \_updateLifecycleState('initState called');  
 }  
 @override  
 void didChangeAppLifecycleState(AppLifecycleState state) {  
 switch (state) {  
 case AppLifecycleState.paused:  
 \_updateLifecycleState('App is paused');  
 break;  
 case AppLifecycleState.resumed:  
 \_updateLifecycleState('App is resumed');  
 break;  
 case AppLifecycleState.inactive:  
 \_updateLifecycleState('App is inactive');  
 break;  
 case AppLifecycleState.detached:  
 \_updateLifecycleState('App is detached');  
 break;  
 case AppLifecycleState.hidden:  
 \_updateLifecycleState('App is hidden');  
 break;  
 }  
 }  
 @override  
 void dispose() {  
 WidgetsBinding.*instance*.removeObserver(this);  
 super.dispose();  
 }  
 void \_updateLifecycleState(String message) {  
 setState(() {  
 \_lifecycleState = message;  
 });  
 }  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Activity Lifecycle Demo'),  
 ),  
 body: Center(  
 child: Text(  
 \_lifecycleState,  
 style: TextStyle(fontSize: 24),  
 ),  
 ),  
 );  
 }  
}

**pratical no:-8**

**:- To study and create an android app that demonstrates the use of Alerts and Pickers**

**Code1(TimePicker):-**

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(  
 title: 'Time Picker Demo',  
 theme: ThemeData(  
 primarySwatch: Colors.*blue*,  
 ),  
 home: const MyHomePage(),  
 );  
 }  
}  
class MyHomePage extends StatefulWidget {  
 const MyHomePage({super.key});  
 @override  
 \_MyHomePageState createState() => \_MyHomePageState();  
}  
class \_MyHomePageState extends State<MyHomePage> {  
 TimeOfDay \_selectedTime = TimeOfDay.now();  
  
 Future<void> \_pickTime() async {  
 TimeOfDay? pickedTime = await showTimePicker(  
 context: context,  
 initialTime: \_selectedTime,  
 );  
  
 if (pickedTime !=null && pickedTime != \_selectedTime) {  
 setState(() {  
 \_selectedTime = pickedTime;  
 });  
 }  
 }  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: const Text('Time Picker Demo'),  
 ),  
 body: Padding(  
 padding: const EdgeInsets.all(16.0),  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Text(  
 'Selected Time:${\_selectedTime.format(context)}',  
 style: const TextStyle(fontSize: 18),  
 ),  
 const SizedBox(height: 20),  
 ElevatedButton(  
 onPressed: \_pickTime,  
 child: const Text('Pick a Time'),  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
}

**Code2(DatePicker):**-

import 'package:flutter/material.dart';  
import 'package:intl/intl.dart'; *// For date*void main() {  
 runApp(MyApp());  
}  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Simple Date Picker',  
 theme: ThemeData(  
 primarySwatch: Colors.*blue*,  
 ),  
 home: DatePickerScreen(),  
 );  
 }  
}  
class DatePickerScreen extends StatefulWidget {  
 @override  
 \_DatePickerScreenState createState() =>  
 \_DatePickerScreenState();  
}  
class \_DatePickerScreenState extends  
State<DatePickerScreen> {  
*// Store the selected date* DateTime? \_selectedDate;  
*// Function to show the date picker dialog* Future<void> \_pickDate() async {  
 DateTime initialDate = DateTime.now(); *//* DateTime firstDate = DateTime(2000); *// The* DateTime lastDate = DateTime(2101); *// The  
  
// Show the date picker* final DateTime? picked = await showDatePicker(  
 context: context,  
 initialDate: \_selectedDate ?? initialDate,  
 firstDate: firstDate,  
 lastDate: lastDate,  
 );  
*// Update the selected date if the user picked* if (picked != null && picked != \_selectedDate)  
 {  
 setState(() {  
 \_selectedDate = picked;  
 });  
 }  
 }  
 @override  
 Widget build(BuildContext context) {  
*// Format the selected date (if any) using* String formattedDate = \_selectedDate != null  
  
 ? DateFormat('yyyy-MM-dd').format(\_selectedDate!)  
  
 : 'No date selected';  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Simple Date Picker'),  
 ),  
 body: Center(  
 child: Column(  
 mainAxisAlignment:  
 MainAxisAlignment.center,  
 children: <Widget>[  
 Text(  
  
 'Selected Date: \n$formattedDate',  
 style: TextStyle(fontSize: 24),  
 textAlign: TextAlign.center,  
 ),  
 SizedBox(height: 20),  
 ElevatedButton(  
 onPressed: \_pickDate, *// Call* child: Text('Pick a Date'),  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
}

**practical no:-9**

**Explore different types of Menu in android. Create an android app that demonstrates the use of an Menu.**

**Code:**-

import 'package:flutter/material.dart';  
  
void main() {  
 runApp( MyApp());  
}  
class MyApp extends StatelessWidget {  
 *// This widget is the root of your application.* @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Popup Menu Demo',  
 theme: ThemeData(  
 primarySwatch: Colors.*blue*,  
 ),  
 home: MyHomePage(),  
 );  
 }  
}  
class MyHomePage extends StatefulWidget {  
 @override  
 \_MyHomePageState createState() => \_MyHomePageState();  
}  
class \_MyHomePageState extends State<MyHomePage> {  
 void \_onMenuItemSelected(String value) {  
 switch (value) {  
 case 'Item 1':  
 ScaffoldMessenger.*of*(context).showSnackBar(  
 SnackBar(content: Text('Item 1 Selected')),  
 );  
 break;  
 case 'Item 2':  
 ScaffoldMessenger.*of*(context).showSnackBar(  
 SnackBar(content: Text('Item 2 Selected')),  
 );  
 break;  
 case 'Item 3':  
 ScaffoldMessenger.*of*(context).showSnackBar(  
 SnackBar(content: Text('Item 3 Selected')),  
 );  
 break;  
 case 'Exit':  
 ScaffoldMessenger.*of*(context).showSnackBar(  
 SnackBar(content: Text('Exit Selected')),  
 );  
 break;  
 default:  
 print('Unknown option selected');  
 }  
 }  
  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Popup Menu Demo'),  
 actions: <Widget>[  
 PopupMenuButton<String>(  
 onSelected: \_onMenuItemSelected,  
 itemBuilder: (BuildContext context) {  
 return [  
 PopupMenuItem<String>(  
 value: 'Item 1',  
 child: Text('Item 1'),  
 ),  
 PopupMenuItem<String>(  
 value: 'Item 2',  
 child: Text('Item 2'),  
 ),  
 PopupMenuItem<String>(  
 value: 'Item 3',  
 child: Text('Item 3'),  
 ),  
 PopupMenuItem<String>(  
 value: 'Exit',  
 child: Text('Exit'),  
 ),  
 ];  
 },  
 ),  
 ],  
 ),  
 body: Center(  
 child: Text('Select a menu item from the top-right corner'),  
 ),  
 );  
 }  
}

**practical no:-10**

**AIM :**

**A.** To study Screen Navigation. Create an android app that demonstrates Screen Navigation Using the App Bar and Tabs.

**B.** Explore the concept of Drawables , Themes and Style. Create an android to demonstrate Style.

A. Demonstrates Screen Navigation

**Code:-**

**i**mport 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Tab Navigation Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: MyHomePage(),

);

}

}

class MyHomePage extends StatelessWidget {

@override

Widget build(BuildContext context) {

return DefaultTabController(

length: 3, // Number of tabs

child: Scaffold(

appBar: AppBar(

title: Text('Tab Navigation Demo'),

bottom: TabBar(

tabs: [

Tab(text: 'Home'),

Tab(text: 'Profile'),

Tab(text: 'Settings'),

],

),

),

body: TabBarView(

children: [

HomeTab(),

ProfileTab(),

SettingsTab(),

],

),

),

);

}

}

class HomeTab extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Center(

child: Text(

'Home Screen',

style: TextStyle(fontSize: 24),

),

);

}

}

class ProfileTab extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Center(

child: Text(

'Profile Screen',

style: TextStyle(fontSize: 24),

),

);

}

}

class SettingsTab extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Center(

child: Text(

'Settings Screen',

style: TextStyle(fontSize: 24),

),

);

}

}

B. Drawables

**Code:-**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Styles Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: HomeScreen(),

);

}

}

class HomeScreen extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Flutter Styles Demo'),

),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: ListView(

children: [

// Text Styles with Bold, Italic, and Underline

Text(

'Bold, Italic, and Underlined Text',

style: TextStyle(

fontSize: 24,

fontWeight: FontWeight.bold, // Bold style

fontStyle: FontStyle.italic, // Italic style

decoration: TextDecoration.underline, // Underline style

color: Colors.blue, // Text color

),

),

SizedBox(height: 20),

// Container Decoration with Border, Shadow, and Color

Container(

height: 100,

decoration: BoxDecoration(

color: Colors.amber,

borderRadius: BorderRadius.circular(12),

border: Border.all(

color: Colors.black,

width: 2,

),

boxShadow: [

BoxShadow(

color: Colors.black26,

offset: Offset(4, 4),

blurRadius: 8,

),

],

),

child: Center(

child: Text(

'Styled Container',

style: TextStyle(

fontSize: 20,

fontWeight: FontWeight.bold,

color: Colors.white,

),

),

),

),

SizedBox(height: 20),

// Button Style

ElevatedButton(

onPressed: () {

// Button action

},

style: ElevatedButton.styleFrom(

backgroundColor: Colors.blue, // background color

foregroundColor: Colors.white, // text color

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(8),

),

padding: EdgeInsets.symmetric(horizontal: 20, vertical: 12),

),

child: Text('Styled Button'),

),

SizedBox(height: 20),

// Icon Style

Icon(

Icons.star,

size: 50,

color: Colors.orange,

),

SizedBox(height: 20),

// TextField Styling

TextField(

decoration: InputDecoration(

labelText: 'Enter Text',

labelStyle: TextStyle(color: Colors.blue),

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

),

focusedBorder: OutlineInputBorder(

borderRadius: BorderRadius.circular(12),

borderSide: BorderSide(color: Colors.blue),

),

),

),

],

),

),

)

}

}

**practical no:-11**

**Aim: To study and create an android app to demonstrate use of Animation.**

**CODE:-**

import 'package:flutter/material.dart';  
void main() {  
 runApp(const FadeAppTest());  
}  
class FadeAppTest extends StatelessWidget {  
 const FadeAppTest({super.key});  
*// This widget is the root of your application.* @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Fade Demo',  
 theme: ThemeData(  
 colorScheme: ColorScheme.fromSeed(seedColor: Colors.*deepPurple*),  
 ),  
 home: const MyFadeTest(title: 'Fade Demo'),  
 );  
 }  
}  
class MyFadeTest extends StatefulWidget {  
 const MyFadeTest({super.key, required this.title});  
 final String title;  
 @override  
 State<MyFadeTest> createState() => \_MyFadeTest();  
}  
class \_MyFadeTest extends State<MyFadeTest> with TickerProviderStateMixin {  
 late AnimationController controller;  
 late CurvedAnimation curve;  
 @override  
 void initState() {  
 super.initState();  
 controller = AnimationController(  
 duration: const Duration(milliseconds: 2000),  
 vsync: this,  
 );  
 curve = CurvedAnimation(parent: controller, curve: Curves.*easeIn*);  
 }  
 @override  
  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(title: Text(widget.title)),  
 body: Center(  
 child: FadeTransition(  
 opacity: curve,  
 child: const FlutterLogo(size: 305),  
 ),  
 ),  
 floatingActionButton: FloatingActionButton(  
 tooltip: 'Fade',  
 onPressed: () {  
 controller.forward();  
 },  
 child: const Icon(Icons.*brush*),  
 ),  
 );  
 }  
}

**practical no:-12**

**Aim:-To study firebase and Admob. Create an android app to develop calculator**

**Code:-**

import 'package:flutter/material.dart';  
import 'package:math\_expressions/math\_expressions.dart';  
  
void main() {  
 runApp(const CalculatorApp());  
}  
  
class CalculatorApp extends StatelessWidget {  
 const CalculatorApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Calculator',  
 theme: ThemeData.dark(),  
 home: const CalculatorScreen(),  
 debugShowCheckedModeBanner: false,  
 );  
 }  
}  
  
class CalculatorScreen extends StatefulWidget {  
 const CalculatorScreen({super.key});  
  
 @override  
 \_CalculatorScreenState createState() => \_CalculatorScreenState();  
}  
  
class \_CalculatorScreenState extends State<CalculatorScreen> {  
 String \_expression = '';  
 String \_result = '';  
  
 void \_onButtonPressed(String value) {  
 setState(() {  
 if (value == 'C') {  
 \_expression = '';  
 \_result = '';  
 } else if (value == '⌫') {  
 \_expression = \_expression.isNotEmpty ? \_expression.substring(0, \_expression.length - 1) : '';  
 } else if (value == '=') {  
 \_calculateResult();  
 } else {  
 \_expression += value;  
 }  
 });  
 }  
  
 void \_calculateResult() {  
 try {  
 Parser p = Parser();  
 Expression exp = p.parse(\_expression.replaceAll('×', '\*').replaceAll('÷', '/'));  
 ContextModel cm = ContextModel();  
 double eval = exp.evaluate(EvaluationType.REAL, cm);  
 \_result = eval.toString();  
 } catch (e) {  
 \_result = 'Error';  
 }  
 }  
  
 Widget \_buildButton(String text, {Color? color}) {  
 return Expanded(  
 child: ElevatedButton(  
 onPressed: () => \_onButtonPressed(text),  
 style: ElevatedButton.*styleFrom*(  
 padding: const EdgeInsets.all(20),  
 backgroundColor: color ?? Colors.*grey*[800],  
 ),  
 child: Text(text, style: const TextStyle(fontSize: 24)),  
 ),  
 );  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(title: const Text('Calculator')),  
 body: Column(  
 children: [  
 Expanded(  
 child: Container(  
 alignment: Alignment.*bottomRight*,  
 padding: const EdgeInsets.all(20),  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.end,  
 crossAxisAlignment: CrossAxisAlignment.end,  
 children: [  
 Text(\_expression, style: const TextStyle(fontSize: 32)),  
 const SizedBox(height: 10),  
 Text(\_result, style: const TextStyle(fontSize: 40, fontWeight: FontWeight.*bold*)),  
 ],  
 ),  
 ),  
 ),  
 Column(  
 children: [  
 Row(children: [\_buildButton('C', color: Colors.*red*), \_buildButton('⌫'), \_buildButton('%'), \_buildButton('÷')]),  
 Row(children: [\_buildButton('7'), \_buildButton('8'), \_buildButton('9'), \_buildButton('×')]),  
 Row(children: [\_buildButton('4'), \_buildButton('5'), \_buildButton('6'), \_buildButton('-')]),  
 Row(children: [\_buildButton('1'), \_buildButton('2'), \_buildButton('3'), \_buildButton('+')]),  
 Row(children: [\_buildButton('0'), \_buildButton('.'), \_buildButton('=', color: Colors.*green*)]),  
 ],  
 ),  
 ],  
 ),  
 );  
 }  
}