**PRACTICAL:06**

**AIM: Develop an application which shows different types of contents in different views like CardView, ListView, GridView, ScrollView, Fragment, TabbedView, WebView etc**

**Source Code :**

**Main.dart:**

import 'package:flutter/material.dart';

import 'package:webview\_flutter/webview\_flutter.dart';

void main() {

  runApp(

      Views()

  );

}

class Views extends StatefulWidget{

  @override

  State<StatefulWidget> createState() {

    return ViewState();

  }

}

class ViewState extends State<Views>{

  final navigatorKey = GlobalKey<NavigatorState>();

  final List<String> entries = <String>['CardView', 'ListView', 'GridView', 'ScrollView', 'Fragment', 'TabView'];

  int \_selectedDrawerIndex = 0;

  @override

  Widget build(BuildContext context) {

    var drawerOptions = <Widget>[];

    for (var i = 0; i < entries.length; i++) {

      var d = entries[i];

      drawerOptions.add(

          new ListTile(

            title: new Text('$d'),

            selected: i == \_selectedDrawerIndex,

            onTap: () => onSelectItem(i),

          )

      );

    }

    return MaterialApp(

        navigatorKey: navigatorKey,

        debugShowCheckedModeBanner: false,

        home: Scaffold(

          appBar: new AppBar(

            // here we display the title corresponding to the fragment

            // you can instead choose to have a static title

            title: Text("Fragment View"),

          ),

          drawer: new Drawer(

            child: new Column(

              children: <Widget>[

                new UserAccountsDrawerHeader(

                    accountName: new Text("Riya"), accountEmail: Text("Welcome to My views application")),

                new Column(children: drawerOptions)

              ],

            ),

          ),

          body: cardView(),

        )

    );

  }

  onSelectItem(int index) {

    setState(() => \_selectedDrawerIndex = index);

    Navigator.*of*(context).pop(); // close the drawer

  }

  Widget cardView() {

    return Container(

      decoration: new BoxDecoration(

          color: Colors.*lightBlueAccent*,

          borderRadius: BorderRadius.all(Radius.circular(70))

      ),

      child: Column(

        children: <Widget>[

          Text(

            "Card View",

            style: TextStyle(fontSize: 30.0),

          ),

          Card(

              margin: EdgeInsets.symmetric(vertical: 10.0, horizontal: 20.0),

              child: InkWell(

                splashColor: Colors.*pinkAccent*.withAlpha(30),

                onTap: () {

                  navigatorKey.currentState.push(

                    MaterialPageRoute(builder: (context) {

                      return ListV();

                    }),

                  );

                },

                child: ListTile(

                  title: Text("ListView"),

                ),

              )),

          Card(

              margin: EdgeInsets.symmetric(vertical: 10.0, horizontal: 20.0),

              child: InkWell(

                splashColor: Colors.*pinkAccent*.withAlpha(30),

                onTap: () {

                  navigatorKey.currentState.push(

                    MaterialPageRoute(builder: (context) {

                      return GridV();

                    }),

                  );

                },

                child: ListTile(

                  title: Text("Grid and Scroll View"),

                ),

              )),

          Card(

              margin: EdgeInsets.symmetric(vertical: 10.0, horizontal: 20.0),

              child: InkWell(

                splashColor: Colors.*pinkAccent*.withAlpha(30),

                onTap: () {

                },

                child: ListTile(

                  title: Text("Fragment"),

                ),

              )),

          Card(

              margin: EdgeInsets.symmetric(vertical: 10.0, horizontal: 20.0),

              child: InkWell(

                splashColor: Colors.*pinkAccent*.withAlpha(30),

                onTap: () {

                  navigatorKey.currentState.push(

                    MaterialPageRoute(builder: (context) {

                      return TabV();

                    }),

                  );

                },

                child: ListTile(

                  title: Text("TabView"),

                ),

              )),

        ],

      ),

    );

  }

}

class TabV extends StatelessWidget{

  @override

  Widget build(BuildContext context) {

    return DefaultTabController(

        length: 3,

        child: Scaffold(

          appBar: AppBar(

            bottom: TabBar(

              tabs: [

               Tab(icon: Icon(Icons.*directions\_ferry*)),

                Tab(icon: Icon(Icons.*directions\_train*)),

                Tab(icon: Icon(Icons.*directions\_car*)),

              ],

            ),

            title: Text('Tabview'),

          ),

          body: TabBarView(

            children: [

              Icon(Icons.*directions\_ferry*),

              Icon(Icons.*directions\_train*),

              Icon(Icons.*directions\_car*),

            ],

          ),

        )

    );

  }

}

class ListV extends StatelessWidget{

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text("Listview"),

      ),

      body: ListView(

        padding: const EdgeInsets.all(8),

        children: <Widget>[

          Container(

            height: 70,

            color: Colors.*greenAccent*,

            child: const Center(child: Text('Your Orders')),

          ),

          Container(

            height: 70,

            color: Colors.*red*[600],

            child: const Center(child: Text('Your Wish List')),

          ),

          Container(

            height: 70,

            color: Colors.*purpleAccent*[200],

            child: const Center(child: Text('Your Account')),

          ),

          Container(

            height: 70,

            color: Colors.*red*[400],

            child: const Center(child: Text('Program & Features')),

          ),

          Container(

            height: 70,

            color: Colors.*brown*[400],

            child: const Center(child: Text('Settings')),

          ),

        ],

      ),

    );

  }

}

class GridV extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return Scaffold(

        appBar: AppBar(

          title: Text("Grid and Scroll view"),

        ),

        body: Container(

            child: GridView.count(

              primary: false,

              padding: const EdgeInsets.all(20),

              crossAxisSpacing: 10,

              mainAxisSpacing: 10,

              crossAxisCount: 2,

              children: <Widget>[

                Container(

                  padding: const EdgeInsets.all(6),

                  child: const Text("Riya, Welcome to grid and scroll view"),

                  color: Colors.*red*[90],

                ),

                Container(

                  padding: const EdgeInsets.all(6),

                  child: const Text("views list"),

                  color: Colors.*lightBlueAccent*[100],

                ),

                Container(

                  padding: const EdgeInsets.all(6),

                  child: const Text('There are different views'),

                  color: Colors.*lightBlueAccent*[200],

                ),

                Container(

                  padding: const EdgeInsets.all(6),

                  child: const Text('Pantry'),

                  color: Colors.*deepOrange*[300],

                ),

                Container(

                  padding: const EdgeInsets.all(6),

                  child: const Text('Fashion'),

                  color: Colors.*lightBlueAccent*[400],

                ),

                Container(

                  padding: const EdgeInsets.all(6),

                  child: const Text('Appliances'),

                  color: Colors.*brown*[500],

                ),

                Container(

                  padding: const EdgeInsets.all(6),

                  child: const Text('Electronics'),

                  color: Colors.*pink*[600],

                ),

                Container(

                  padding: const EdgeInsets.all(6),

                  child: const Text('Home'),

                  color: Colors.*yellow*[600],

                ),

              ],

            )

        )

    );

  }

}

**Pubsec.yaml**

name: pr7

description: A new Flutter application.

*# The following line prevents the package from being accidentally published to*

*# pub.dev using `pub publish`. This is preferred for private packages.*

publish\_to: 'none' *# Remove this line if you wish to publish to pub.dev*

*# The following defines the version and build number for your application.*

*# A version number is three numbers separated by dots, like 1.2.43*

*# followed by an optional build number separated by a +.*

*# Both the version and the builder number may be overridden in flutter*

*# build by specifying --build-name and --build-number, respectively.*

*# In Android, build-name is used as versionName while build-number used as versionCode.*

*# Read more about Android versioning at https://developer.android.com/studio/publish/versioning*

*# In iOS, build-name is used as CFBundleShortVersionString while build-number used as CFBundleVersion.*

*# Read more about iOS versioning at*

*# https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html*

version: 1.0.0+1

environment:

  sdk: ">=2.7.0 <3.0.0"

dependencies:

  flutter:

    sdk: flutter

*# The following adds the Cupertino Icons font to your application.*

*# Use with the CupertinoIcons class for iOS style icons.*

cupertino\_icons: ^1.0.0

  webview\_flutter: ^1.0.7

dev\_dependencies:

  flutter\_test:

    sdk: flutter

*# For information on the generic Dart part of this file, see the*

*# following page: https://dart.dev/tools/pub/pubspec*

*# The following section is specific to Flutter.*

flutter:

*# The following line ensures that the Material Icons font is*

*# included with your application, so that you can use the icons in*

*# the material Icons class.*

uses-material-design: true

*# To add assets to your application, add an assets section, like this:*

*# assets:*

*#   - images/a\_dot\_burr.jpeg*

*#   - images/a\_dot\_ham.jpeg*

*# An image asset can refer to one or more resolution-specific "variants", see*

*# https://flutter.dev/assets-and-images/#resolution-aware.*

*# For details regarding adding assets from package dependencies, see*

*# https://flutter.dev/assets-and-images/#from-packages*

*# To add custom fonts to your application, add a fonts section here,*

*# in this "flutter" section. Each entry in this list should have a*

*# "family" key with the font family name, and a "fonts" key with a*

*# list giving the asset and other descriptors for the font. For*

*# example:*

*# fonts:*

*#   - family: Schyler*

*#     fonts:*

*#       - asset: fonts/Schyler-Regular.ttf*

*#       - asset: fonts/Schyler-Italic.ttf*

*#         style: italic*

*#   - family: Trajan Pro*

*#     fonts:*

*#       - asset: fonts/TrajanPro.ttf*

*#       - asset: fonts/TrajanPro\_Bold.ttf*

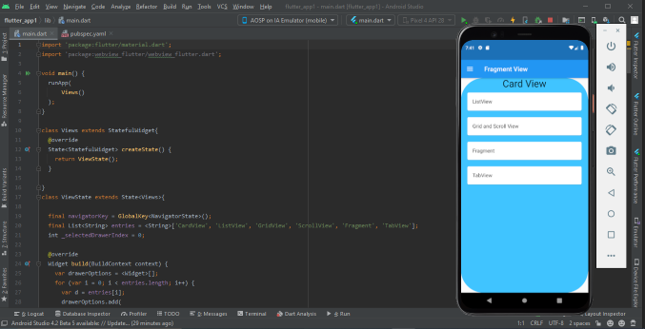
*#         weight: 700*

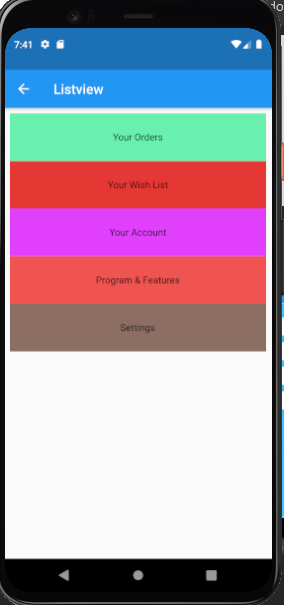
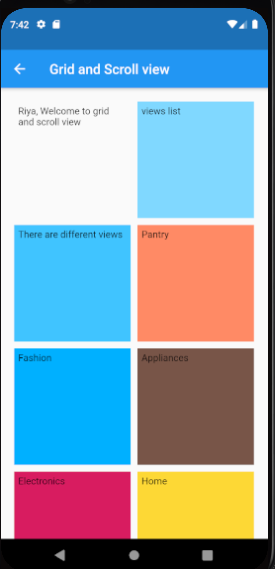
*#*

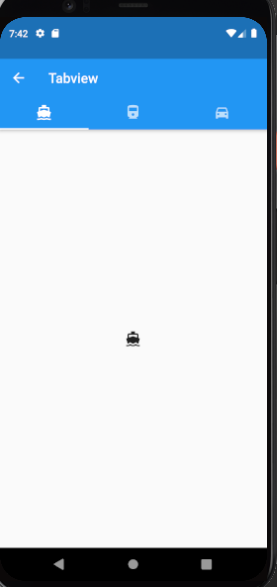
*# For details regarding fonts from package dependencies,*

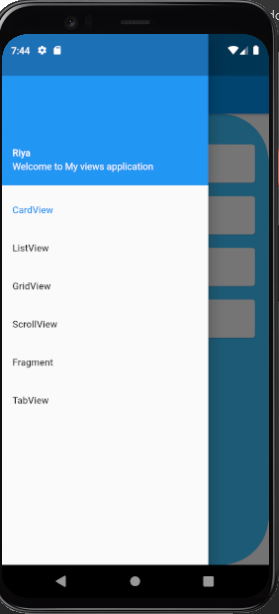
*# see https://flutter.dev/custom-fonts/#from-packages*

Output:





**Learning Outcome:**

Here, in this practical we have learned about how to implement a different view inside a one application using a different color code and different designing.