DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATIONDEVELOPMENT (SE-487)

**LAB SESSION 15**

**Range Sliders & List Wheel Scroll View**

**Objective:**

The objective of this lab session is to provide hands-on experience with Flutter widgets RangeSlider and ListWheelScrollView. By the end of this session, you will understand how to implement and customize these widgets in a Flutter application to create interactive and visually appealing UI components.

**Introduction:**

Flutter, Google's UI toolkit, offers a wide range of widgets that enable developers to build natively compiled applications for mobile, web, and desktop from a single codebase. Two useful and interactive widgets in Flutter are RangeSlider and ListWheelScrollView.

**RangeSlider:** This widget allows users to select a range of values along a continuous interval. It is particularly useful in scenarios where users need to filter results within a specific range, such as price filters in shopping apps or selecting a range of dates.

import 'package:flutter/material.dart';

import 'dart:async';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title:'flutter demo',

debugShowCheckedModeBanner: false,

theme:ThemeData(

),

home:const MyHomePage(),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key?key}):super(key:key);

@override

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATIONDEVELOPMENT (SE-487)

State<MyHomePage> createState() => \_MyHomePageState(); }

class \_MyHomePageState extends State<MyHomePage>

{ RangeValues values =RangeValues(0,100);

@override

Widget build(BuildContext context) {

RangeLabels

labels=RangeLabels(values.start.toString(),values.end.toString()) ;

return Scaffold(

appBar: AppBar(title: Text('range slider'),),

body:Center(

child:RangeSlider(

values:values,

labels:labels,

divisions: 10,

min:0,

max:100,

activeColor: Colors.pink,

inactiveColor: Colors.purple,

onChanged:(newValue)

{

values=newValue;

print('${newValue.start},${newValue.end}');

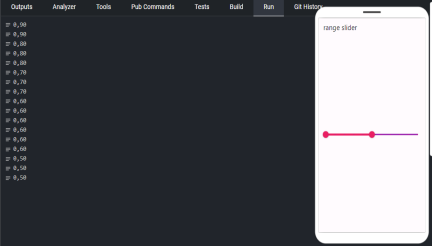
setState(() {

}); }, ) )

);

}

}

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATIONDEVELOPMENT (SE-487) 

**ListWheelScrollView:** This widget displays its children in a scrollable wheel that looks like a 3D carousel. It's useful for creating picker interfaces where users can select from a list of options, such as selecting dates, times, or custom options in a visually appealing manner.

import 'package:flutter/material.dart';

//import IntroPage.dart;

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'flutter demo',

debugShowCheckedModeBanner: false,

theme: ThemeData(

primarySwatch: Colors.lightGreen,

),

home: const MyHomePage(),

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATIONDEVELOPMENT (SE-487)

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key? key}) : super(key: key);

@override

State<MyHomePage> createState() => \_MyHomePageState(); }

class \_MyHomePageState extends State<MyHomePage> {

var arrlist = [

'laiba',

'ali',

'moin',

'maheen',

'maria',

'sara',

'maaz',

'aisha',

'waqas'

];

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Center(child: Text('3D list'))), body: ListWheelScrollView(

itemExtent: 100,

children: arrlist

.map((value) => Padding(

padding: const EdgeInsets.all(8.0),

child: Container(

child: Center(

child: Text('$value', style:

TextStyle(fontSize: 24))),

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(10),

color: Colors.amber),

width: double.infinity,

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATIONDEVELOPMENT (SE-487)

//color: //Colors.blue,

)))

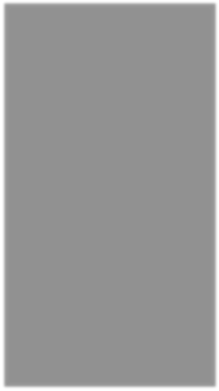
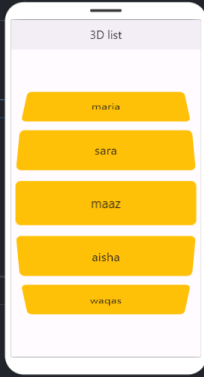
.toList(),

),

);

}

}



DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATIONDEVELOPMENT (SE-487)

**Exercise**

**Question 1: Range Slider**

Objective: Create a RangeSlider that allows users to filter a list of items based on a numerical range. Implement a RangeSlider that filters a list of items with numerical values between 1 and 100. The filtered list should display only the items within the selected range.

Code

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      home: RangeSliderExample(),

    );

  }

}

class RangeSliderExample extends StatefulWidget {

  @override

  \_RangeSliderExampleState createState() => \_RangeSliderExampleState();

}

class \_RangeSliderExampleState extends State<RangeSliderExample> {

  RangeValues \_currentRangeValues = const RangeValues(20, 80);

  List<int> \_items = List.generate(100, (index) => index + 1);

  @override

  Widget build(BuildContext context) {

    List<int> \_filteredItems = \_items.where((item) {

      return item >= \_currentRangeValues.start && item <= \_currentRangeValues.end;

    }).toList();

    return Scaffold(

      appBar: AppBar(

        title: Text('Range Slider Example'),

      ),

      body: Column(

        children: [

          RangeSlider(

            values: \_currentRangeValues,

            min: 1,

            max: 100,

            divisions: 99,

            labels: RangeLabels(

              \_currentRangeValues.start.round().toString(),

              \_currentRangeValues.end.round().toString(),

            ),

            onChanged: (RangeValues values) {

              setState(() {

                \_currentRangeValues = values;

              });

            },

          ),

          Expanded(

            child: ListView.builder(

              itemCount: \_filteredItems.length,

              itemBuilder: (context, index) {

                return ListTile(

                  title: Text(\_filteredItems[index].toString()),

                );

              },

            ),

          ),

        ],

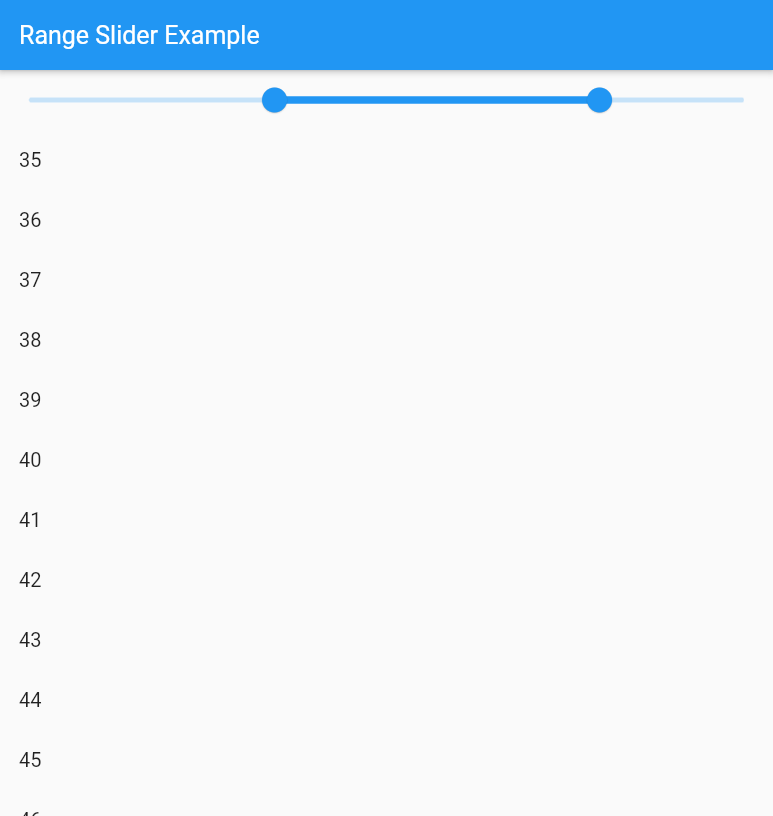
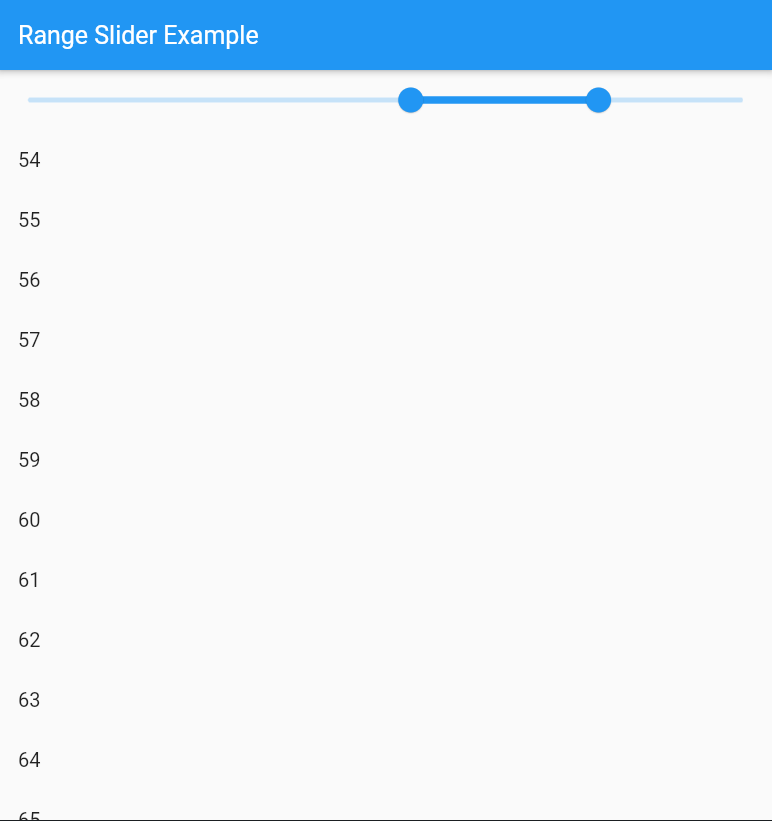
      ),

    );

  }

}

Output



**Question 2: List Wheel Scroll View**

Objective: Create a custom picker using ListWheelScrollView. Implement a custom picker that allows users to select a color from a list of predefined colors. Display the selected color in a box below the ListWheelScrollView.