**Task- 1**

**Aim: Create an app as shown in below image. (OTP)**

**Program:**

import 'dart:math';

import 'package:flutter/material.dart';

void main() {

runApp(

const MaterialApp(

debugShowCheckedModeBanner: false,

home: otpGenerator(),

),

);

}

class otpGenerator extends StatefulWidget {

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

@override

State<otpGenerator> createState() => \_otpGeneratorState();

}

class \_otpGeneratorState extends State<otpGenerator> {

TextEditingController SizeController = TextEditingController();

String num = '';

String i1 = '';

String i2 = '';

String i3 = '';

String i4 = '';

String i5 = '';

String i6 = '';

String n = '';

Random r1 = Random();

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Color(0xff15172b),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

const Text(

"OTP Generator",

style: TextStyle(color: Color(0xfff6db87), fontSize: 40),

),

Container(

width: MediaQuery.of(context).size.width \* 0.7,

child: TextField(

keyboardType: TextInputType.number,

controller: SizeController,

style: const TextStyle(

color: Color(0xffffe8b8),

fontSize: 25,

),

decoration: const InputDecoration(

hintText: 'Enter OTP Length',

hintStyle: TextStyle(color: Color(0xfffcf6ba), fontSize: 15),

enabledBorder: UnderlineInputBorder(

borderSide: BorderSide(

color: Color(0xfffcf6ba),

),

),

),

),

),

InkWell(

onTap: () {

setState(() {

num = SizeController.text;

n = num;

if (int.parse(n) <= 6) {

if (int.parse(n) == 1) {

for (int i = 0; i < int.parse(n); i++) {

i1 = r1.nextInt(5).toString();

}

} else if (int.parse(n) == 2) {

for (int i = 0; i < int.parse(n); i++) {

i1 = r1.nextInt(9).toString();

i2 = r1.nextInt(9).toString();

}

} else if (int.parse(n) == 3) {

for (int i = 0; i < int.parse(n); i++) {

i1 = r1.nextInt(9).toString();

i2 = r1.nextInt(9).toString();

i3 = r1.nextInt(9).toString();

}

} else if (int.parse(n) == 4) {

for (int i = 0; i < int.parse(n); i++) {

i1 = r1.nextInt(9).toString();

i2 = r1.nextInt(9).toString();

i3 = r1.nextInt(9).toString();

i4 = r1.nextInt(9).toString();

}

} else if (int.parse(n) == 5) {

for (int i = 0; i < int.parse(n); i++) {

i1 = r1.nextInt(9).toString();

i2 = r1.nextInt(9).toString();

i3 = r1.nextInt(9).toString();

i4 = r1.nextInt(9).toString();

i5 = r1.nextInt(9).toString();

}

} else if (int.parse(n) == 6) {

for (int i = 0; i < int.parse(n); i++) {

i1 = r1.nextInt(9).toString();

i2 = r1.nextInt(9).toString();

i3 = r1.nextInt(9).toString();

i4 = r1.nextInt(9).toString();

i5 = r1.nextInt(9).toString();

i6 = r1.nextInt(9).toString();

}

}

}

});

},

child: Container(

height: 60,

width: 270,

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(20),

gradient: const LinearGradient(

begin: Alignment.topRight,

end: Alignment.bottomLeft,

colors: [

Color(0xfff6db87),

Color(0xfff6db87),

Color(0xffffe8b8),

],

),

boxShadow: [

BoxShadow(

color: Color(0xfffcf6ba).withOpacity(0.6),

spreadRadius: 2,

blurRadius: 5,

offset: Offset(0, 2),

),

]),

child: const Center(

child: Text(

"Generator OTP",

style: TextStyle(fontSize: 25, fontWeight: FontWeight.bold),

),

),

),

),

Container(

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: [

OTPP(i1),

OTPP(i2),

OTPP(i3),

OTPP(i4),

OTPP(i5),

OTPP(i6),

],

),

alignment: Alignment.center,

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(20),

gradient: const LinearGradient(

begin: Alignment.topRight,

end: Alignment.bottomLeft,

colors: [

Color(0xfff6db87),

Color(0xfff6db87),

Color(0xffffe8b8),

]),

),

height: MediaQuery.of(context).size.height \* 0.090,

width: MediaQuery.of(context).size.width \* 0.80,

),

InkWell(

onTap: () {

setState(() {

i1 = "";

i2 = "";

i3 = "";

i4 = "";

i5 = "";

i6 = "";

SizeController.clear();

});

},

child: Container(

child: const Text(

"Reset",

style: TextStyle(

color: Color(0xff15412b),

fontSize: 25,

fontWeight: FontWeight.bold,

),

),

alignment: Alignment.center,

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(15),

gradient: const LinearGradient(

begin: Alignment.topRight,

end: Alignment.bottomLeft,

colors: [

Color(0xfff6db87),

Color(0xfff6db87),

Color(0xffffe8b8),

]),

boxShadow: [

BoxShadow(

color: Colors.grey.withOpacity(0.6),

offset: Offset(0, 2),

spreadRadius: 2,

blurRadius: 5,

),

]),

height: MediaQuery.of(context).size.height \* 0.070,

width: MediaQuery.of(context).size.width \* 0.35,

),

),

],

),

),

);

}

Widget OTPP(String num) {

return Text(

num,

style: const TextStyle(

fontSize: 30,

fontWeight: FontWeight.bold,

color: Color(0xff15172b),

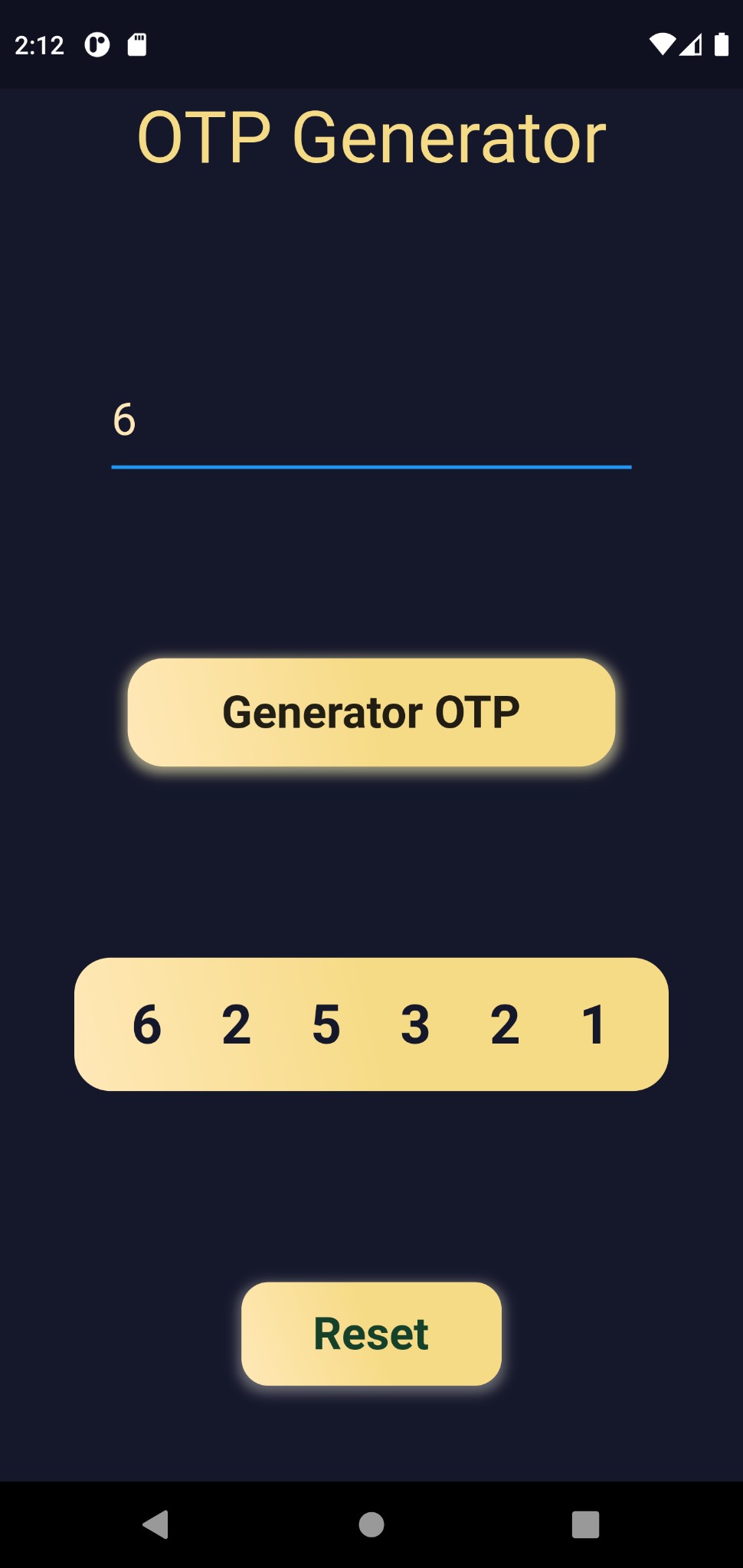
),

);

}

}

**Output:**

****

**Task- 2**

**Aim: Create an app as shown in below image. (COLOR palette)**

**Program:**

import 'dart:math';

import 'package:flutter/material.dart';

void main() {

runApp(

const MaterialApp(

debugShowCheckedModeBanner: false,

home: colorPallet(),

),

);

}

class colorPallet extends StatefulWidget {

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

@override

\_colorPalletState createState() => \_colorPalletState();

}

class \_colorPalletState extends State<colorPallet> {

@override

Random r1 = Random();

List colors = [

Colors.black,

Colors.orange,

Colors.amber,

Colors.indigo,

Colors.red,

Colors.grey,

Colors.blue,

Colors.purple,

Colors.green,

Colors.yellow,

Colors.pink,

Colors.black12,

Colors.teal,

Colors.cyan,

Colors.deepOrange,

Colors.blueGrey,

Colors.yellowAccent,

];

Widget build(BuildContext context) {

return Scaffold(

body: Stack(

children: [

Container(

height: MediaQuery.of(context).size.height,

width: MediaQuery.of(context).size.width,

decoration: const BoxDecoration(

gradient: LinearGradient(

begin: Alignment.topCenter,

end: Alignment.bottomCenter,

colors: [

Color(0xFFFFFFFF),

Color(0xFFD8E6F4),

])),

),

Container(

alignment: Alignment.center,

child: Column(

crossAxisAlignment: CrossAxisAlignment.center,

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: [

const Text(

"Color Palette\nGenerator",

textAlign: TextAlign.center,

style: TextStyle(

color: Color(0xFF3BB3F9),

fontSize: 30,

fontWeight: FontWeight.w700),

),

Column(

children: List.generate(

6,

(index) => Container(

height: MediaQuery.of(context).size.height \* 0.09,

width: MediaQuery.of(context).size.height \* 0.15,

decoration: BoxDecoration(

color: colors[r1.nextInt(16)],

borderRadius: radiusFinder(index),

),

)),

),

OutlinedButton(

onPressed: () {

setState(() {});

},

child: const Text(

"Generate",

style: TextStyle(

color: Color(0xFF3BB3F9),

fontSize: 30,

fontWeight: FontWeight.w700),

),

style: OutlinedButton.styleFrom(

side: const BorderSide(

color: Color(0xFF3BB3F9),

width: 3,

),

padding: const EdgeInsets.symmetric(

vertical: 10, horizontal: 60),

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(15)),

),

),

],

),

),

],

),

);

}

radiusFinder(int n) {

if (n == 0) {

return const BorderRadius.only(

topRight: Radius.circular(20), topLeft: Radius.circular(20));

} else if (n == 5) {

return const BorderRadius.only(

bottomRight: Radius.circular(20), bottomLeft: Radius.circular(20));

} else {

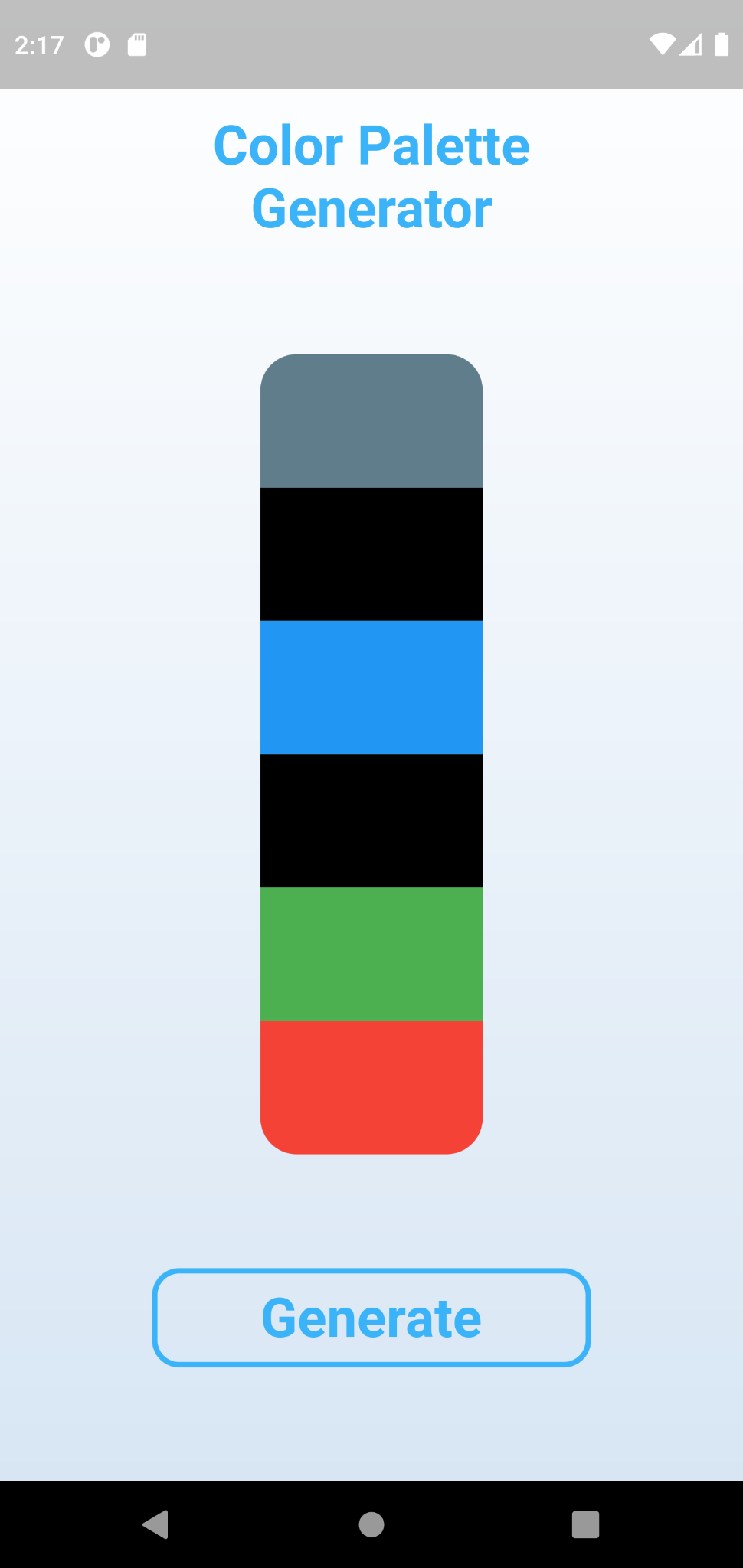
BorderRadius.circular(0);

}

}

}

**Output:**

****