**Program 1,2,3**

import 'package:flutter/material.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

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

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: const HomeScreen(),

);

}

}

class HomeScreen extends StatelessWidget {

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

@override

Widget build(BuildContext context) {

return Scaffold(

body: Image.network(

"https://images.unsplash.com/photo-1565299624946-b28f40a0ae38?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=481&q=80",

),

floatingActionButton: FloatingActionButton(

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(builder: (\_) => const MenuScreen()),

);

},

child: const Icon(Icons.add),

),

);

}

}

class MenuScreen extends StatelessWidget {

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

static const \_foodList = [

"https://images.unsplash.com/photo-1565299624946-b28f40a0ae38?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=481&q=80",

"https://images.unsplash.com/photo-1565299624946-b28f40a0ae38?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=481&q=80",

"https://images.unsplash.com/photo-1565299624946-b28f40a0ae38?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=481&q=80",

"https://images.unsplash.com/photo-1565299624946-b28f40a0ae38?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=481&q=80",

"https://images.unsplash.com/photo-1565299624946-b28f40a0ae38?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=481&q=80",

"https://images.unsplash.com/photo-1565299624946-b28f40a0ae38?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=481&q=80",

"https://images.unsplash.com/photo-1565299624946-b28f40a0ae38?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=481&q=80",

];

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(),

body: ListView.builder(

itemCount: \_foodList.length,

itemBuilder: (context, index) {

return ListTile(

leading: Image.network(\_foodList[index]),

title: Text("Food $index"),

subtitle: Text("Food sub $index"),

);

},

),

);

}

}

**Program 7,8**

import 'package:flutter/material.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

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

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: const HomeScreen(),

);

}

}

class HomeScreen extends StatefulWidget {

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

@override

State<HomeScreen> createState() => \_HomeScreenState();

}

class \_HomeScreenState extends State<HomeScreen> {

final \_nameController = TextEditingController();

final \_emailController = TextEditingController();

final \_passwordController = TextEditingController();

final \_formKey = GlobalKey<FormState>();

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text("Login"),

),

body: Padding(

padding: const EdgeInsets.all(8.0),

child: Form(

key: \_formKey,

child: Column(

children: [

TextFormField(

controller: \_nameController,

validator: (value) {

if (value == null) return "Enter name";

if (value.contains(RegExp("[0-9]"))) {

return "Username cannot have a number";

}

return null;

},

autovalidateMode: AutovalidateMode.onUserInteraction,

decoration: const InputDecoration(

labelText: "Username",

hintText: "darshan",

),

),

TextFormField(

controller: \_emailController,

validator: (value) {

if (value == null) return "Enter name";

if (!RegExp(

r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'\*+-/=?^\_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z]+")

.hasMatch(value)) {

return "Email not in proper format";

}

return null;

},

autovalidateMode: AutovalidateMode.onUserInteraction,

decoration: const InputDecoration(

labelText: "Email",

hintText: "darshanrander@gmail.com",

),

),

TextFormField(

controller: \_passwordController,

validator: (value) {

if (value == null) return "Enter name";

if (value.length < 6) {

return "Password should be more than 6 chars";

}

return null;

},

autovalidateMode: AutovalidateMode.onUserInteraction,

obscureText: true,

decoration: const InputDecoration(

labelText: "Password",

hintText: "\*\*\*\*",

),

),

const SizedBox(height: 20),

ElevatedButton(

onPressed: () {

if (!\_formKey.currentState!.validate()) return;

Navigator.push(

context,

MaterialPageRoute(

builder: (\_) => InfoScreen(

name: \_nameController.text,

email: \_emailController.text,

),

),

);

},

child: const Text("Login"),

),

],

),

),

),

);

}

}

class InfoScreen extends StatelessWidget {

const InfoScreen({

Key? key,

required this.name,

required this.email,

}) : super(key: key);

final String name, email;

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text("Info"),

),

body: Padding(

padding: const EdgeInsets.all(8.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [Text("Name: $name"), Text("Email: $email")],

),

),

);

}

}

**Program 9**

import 'package:number\_to\_words/number\_to\_words.dart';

import 'dart:io';

void main(List<String> arguments) {

stdout.write("What is your name?\r\n");

int? x = int.parse(stdin.readLineSync()!);

print(NumberToWord().convert('en-in', x));

print("Divisors are");

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

if (x % i == 0) {

print(i);

}

}

print("To check wheter no is prime or not");

int flag = 0;

for (int j = 2; j < x; j += 1) {

if (x % j == 0) {

flag = 1;

}

}

if (flag == 1) {

print("Not prime");

} else {

print(" prime");

}

}

**Program 11, 12, 13**

import 'dart:io';

class Employee {

late int id;

late String fname;

late String lname;

late int age;

late int salary;

Employee({

required this.id,

required this.fname,

required this.lname,

required this.age,

required this.salary,

});

Map<String, dynamic> toJson() => <String, dynamic>{

'id': id,

'fname': fname,

'lname': lname,

'age': age,

'salary': salary,

};

}

class Company {

late List<Employee> \_employees;

Company() {

\_employees = <Employee>[];

}

bool createEmployee(String fname, String lname, int age, int salary) {

try {

\_employees.add(

Employee(

fname: fname,

lname: lname,

age: age,

salary: salary,

id: \_employees.length,

),

);

return true;

} catch (e) {

return false;

}

}

Map<String, dynamic> getEmployee(int index) {

return \_employees[index].toJson();

}

bool deleteEmployee(int index) {

try {

\_employees.removeAt(index);

return true;

} catch (e) {

return false;

}

}

bool updateEmployee(int index, int salary) {

try {

Employee k = \_employees[index];

k.salary = salary;

return true;

} catch (e) {

return false;

}

}

}

class CompanyV2 {

late Map<int, Employee> \_employees;

late int count;

CompanyV2() {

\_employees = <int, Employee>{};

count = 0;

}

int? createEmployee(String fname, String lname, int age, int salary) {

try {

\_employees[count] = Employee(

fname: fname,

lname: lname,

age: age,

salary: salary,

id: \_employees.length,

);

count++;

return count - 1;

} catch (e) {

return null;

}

}

Map<String, dynamic>? getEmployee(int index) {

Employee? k = \_employees[index];

if (k == null) {

return null;

} else {

return k.toJson();

}

}

bool deleteEmployee(int index) {

try {

Employee? k = \_employees.remove(index);

if (k != null) {

return true;

} else {

return false;

}

} catch (e) {

return false;

}

}

bool updateEmployee(int index, int salary) {

try {

Employee? k = (\_employees[index]);

if (k != null) {

k.salary = salary;

return true;

}

return false;

} catch (e) {

return false;

}

}

}

void main() {

CompanyV2 company = CompanyV2();

int choice = 1;

while (choice != 0) {

print("---Company---");

print("1. Add employee");

print("2. remove Employee");

print("3. update employee");

print("4. get employee");

print("Enter your choice: ");

choice = int.parse(stdin.readLineSync()!);

switch (choice) {

case 1:

print("Enter the employee details: ");

print("Enter the first name: ");

String fname = stdin.readLineSync()!;

print("Enter the last name: ");

String lname = stdin.readLineSync()!;

print("Enter the age: ");

int age = int.parse(stdin.readLineSync()!);

print("Enter the salary: ");

int salary = int.parse(stdin.readLineSync()!);

int? id = company.createEmployee(fname, lname, age, salary);

if (id != null) {

print("Employee added successfully");

print("Your id is: $id");

} else {

print("Employee not added");

}

break;

case 2:

print("Enter the employee id: ");

int id = int.parse(stdin.readLineSync()!);

if (company.deleteEmployee(id)) {

print("Employee deleted successfully");

} else {

print("Employee not deleted");

}

break;

case 3:

print("Enter the employee id: ");

int id = int.parse(stdin.readLineSync()!);

print("Enter the salary: ");

int salary = int.parse(stdin.readLineSync()!);

if (company.updateEmployee(id, salary)) {

print("Employee added successfully");

} else {

print("Employee not added");

}

break;

case 4:

print("Enter the employee id: ");

int id = int.parse(stdin.readLineSync()!);

Map<String, dynamic>? employee = company.getEmployee(id);

if (employee != null) {

print("Employee details: ");

print("First name: ${employee['fname']}");

print("Last name: ${employee['lname']}");

print("Age: ${employee['age']}");

print("Salary: ${employee['salary']}");

print(employee);

} else {

print("Employee not found");

}

break;

case 0:

print("Exiting...");

break;

default:

print("Invalid choice");

}

}}

**Program 4**

**One.dart**

import 'package:flutter/material.dart';

import 'button.dart';

import 'package:math\_expressions/math\_expressions.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

home: HomePage(),

); // MaterialApp

}

}

class HomePage extends StatefulWidget {

@override

\_HomePageState createState() => \_HomePageState();

}

class \_HomePageState extends State<HomePage> {

var userInput = '';

var answer = '';

// Array of button

final List<String> buttons = [

'C',

'+/-',

'%',

'DEL',

'7',

'8',

'9',

'/',

'4',

'5',

'6',

'x',

'1',

'2',

'3',

'-',

'0',

'.',

'=',

'+',

];

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: new AppBar(

title: new Text("Calculator"),

), //AppBar

backgroundColor: Colors.white38,

body: Column(

children: <Widget>[

Expanded(

child: Container(

child: Column(mainAxisAlignment: MainAxisAlignment.spaceEvenly, children: <Widget>[

Container(

padding: EdgeInsets.all(20),

alignment: Alignment.centerRight,

child: Text(

userInput,

style: TextStyle(fontSize: 18, color: Colors.white),

),

),

Container(

padding: EdgeInsets.all(15),

alignment: Alignment.centerRight,

child: Text(

answer,

style: TextStyle(fontSize: 30, color: Colors.white, fontWeight: FontWeight.bold),

),

)

]),

),

),

Expanded(

flex: 3,

child: Container(

child: GridView.builder(

itemCount: buttons.length,

gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 4),

itemBuilder: (BuildContext context, int index) {

// Clear Button

if (index == 0) {

return MyButton(

buttontapped: () {

setState(() {

userInput = '';

answer = '0';

});

},

buttonText: buttons[index],

color: Colors.blue[50],

textColor: Colors.black,

);

}

// +/- button

else if (index == 1) {

return MyButton(

buttonText: buttons[index],

color: Colors.blue[50],

textColor: Colors.black,

);

}

// % Button

else if (index == 2) {

return MyButton(

buttontapped: () {

setState(() {

userInput += buttons[index];

});

},

buttonText: buttons[index],

color: Colors.blue[50],

textColor: Colors.black,

);

}

// Delete Button

else if (index == 3) {

return MyButton(

buttontapped: () {

setState(() {

userInput = userInput.substring(0, userInput.length - 1);

});

},

buttonText: buttons[index],

color: Colors.blue[50],

textColor: Colors.black,

);

}

// Equal\_to Button

else if (index == 18) {

return MyButton(

buttontapped: () {

setState(() {

equalPressed();

});

},

buttonText: buttons[index],

color: Colors.orange[700],

textColor: Colors.white,

);

}

// other buttons

else {

return MyButton(

buttontapped: () {

setState(() {

userInput += buttons[index];

});

},

buttonText: buttons[index],

color: isOperator(buttons[index]) ? Colors.blueAccent : Colors.white,

textColor: isOperator(buttons[index]) ? Colors.white : Colors.black,

);

}

}), // GridView.builder

),

),

],

),

);

}

bool isOperator(String x) {

if (x == '/' || x == 'x' || x == '-' || x == '+' || x == '=') {

return true;

}

return false;

}

// function to calculate the input operation

void equalPressed() {

String finaluserinput = userInput;

finaluserinput = userInput.replaceAll('x', '\*');

Parser p = Parser();

Expression exp = p.parse(finaluserinput);

ContextModel cm = ContextModel();

double eval = exp.evaluate(EvaluationType.REAL, cm);

answer = eval.toString();

}

}

**Two.dart**

import 'package:flutter/material.dart';

// creating Stateless Widget for buttons

class MyButton extends StatelessWidget {

// declaring variables

final color;

final textColor;

final buttonText;

final buttontapped;

//Constructor

MyButton({this.color, this.textColor, this.buttonText, this.buttontapped});

@override

Widget build(BuildContext context) {

return GestureDetector(

onTap: buttontapped,

child: Padding(

padding: const EdgeInsets.all(0.2),

child: ClipRRect(

// borderRadius: BorderRadius.circular(25),

child: Container(

color: color,

child: Center(

child: Text(

buttonText,

style: TextStyle(

color: textColor,

fontSize: 25,

fontWeight: FontWeight.bold,

),

),

),

),

),

),

);

}

}

**Program 6**

import 'package:flutter/material.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

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

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: const HomeScreen(),

);

}

}

class HomeScreen extends StatelessWidget {

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

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text("Gestures"),

),

body: Center(

child: GestureDetector(

onDoubleTap: () {

\_showGestureType(context, "onDoubleTap");

},

onLongPress: () {

\_showGestureType(context, "onLongPress");

},

onTap: () {

\_showGestureType(context, "onTap");

},

onVerticalDragEnd: (details) {

\_showGestureType(context, "onVerticalDragEnd");

},

child: Container(

height: 300,

width: 300,

color: Colors.amber,

child: const Center(child: Text("Gestures here...")),

),

),

),

);

}

Future \_showGestureType(BuildContext context, String gestureType) {

return showDialog(

context: context,

builder: (context) {

return AlertDialog(

title: Text(gestureType),

actions: [

ElevatedButton(

onPressed: () {

Navigator.pop(context);

},

child: const Text("OK"),

),

],

);

},

);

}

}

**Program 5**

import 'package:flutter/material.dart';

class Convertor extends StatefulWidget {

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

\_ConvertorState createState() => \_ConvertorState();

}

class \_ConvertorState extends State<Convertor> {

@override

void dowork() {

String x = c.text;

double f1 = double.parse(x) \* 1.8 + 32;

f.text = "$f1";

}

final c = TextEditingController();

final f = TextEditingController();

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Convertor'),

),

body: Column(

children: <Widget>[

TextField(

controller: c,

decoration: InputDecoration(

labelText: 'Celcius',

)),

SizedBox(

width: 60.0,

height: 60.0,

),

TextField(

controller: f,

decoration: InputDecoration(

labelText: 'Farhenit',

)),

SizedBox(

width: 60.0,

height: 60.0,

),

RaisedButton(

onPressed: dowork,

child: Text('Click'),

),

SizedBox(

width: 60.0,

height: 60.0,

),

],

));

}

}