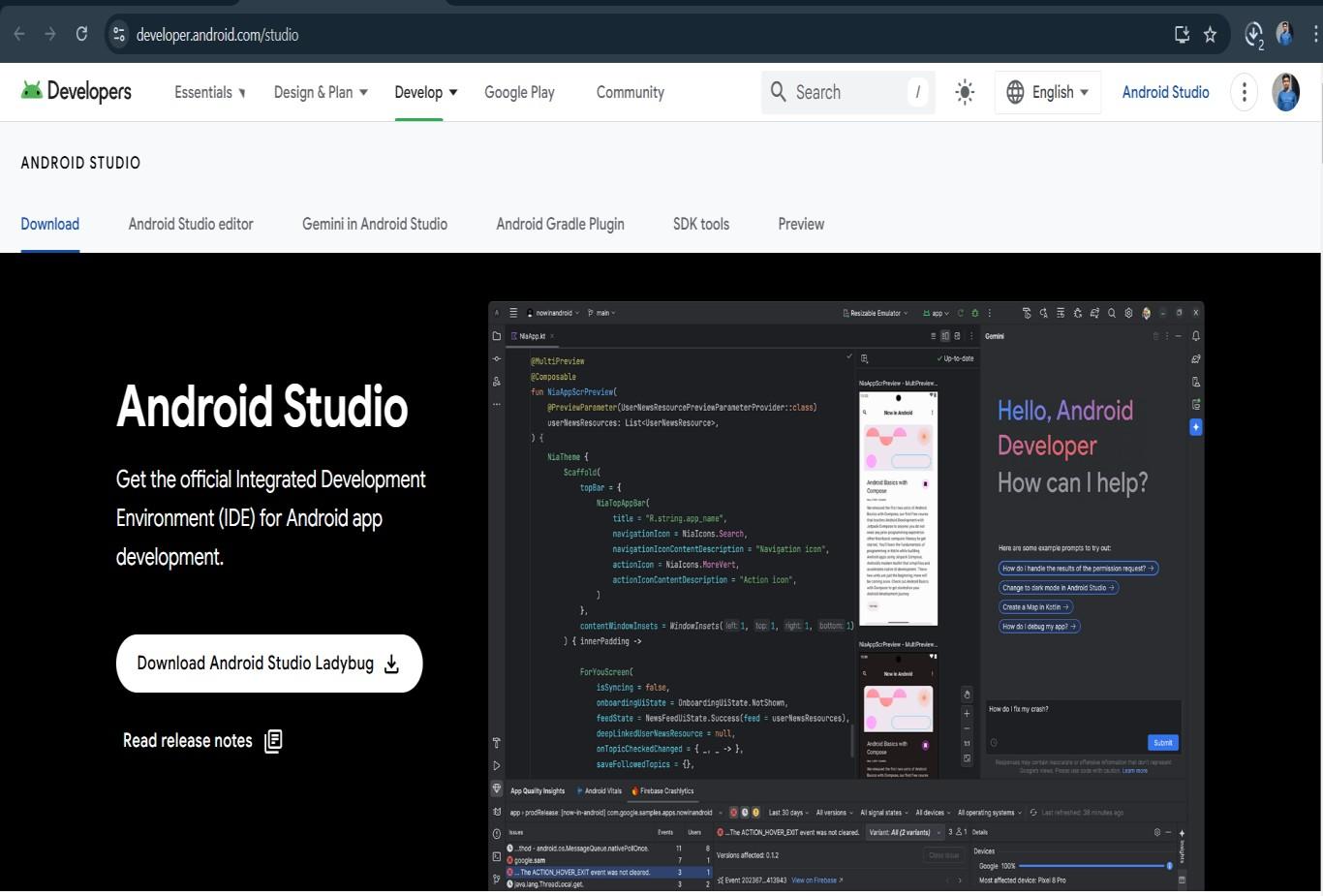
**Practical 1 : Android Studio setup for Flutter development along with**

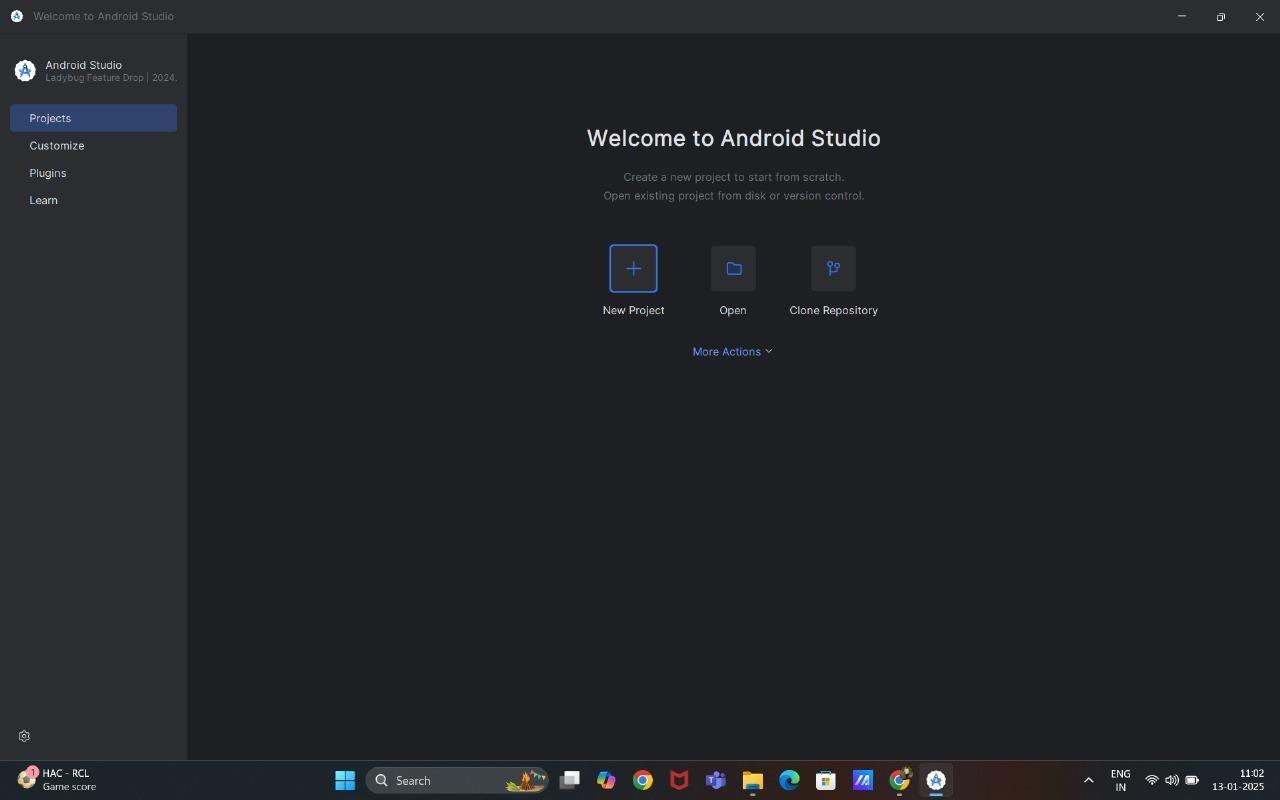
**Dart SDK.**

**Step 01 :** Go to Android from <https://developer.android.com/studio> and download Android studio. And also install it.







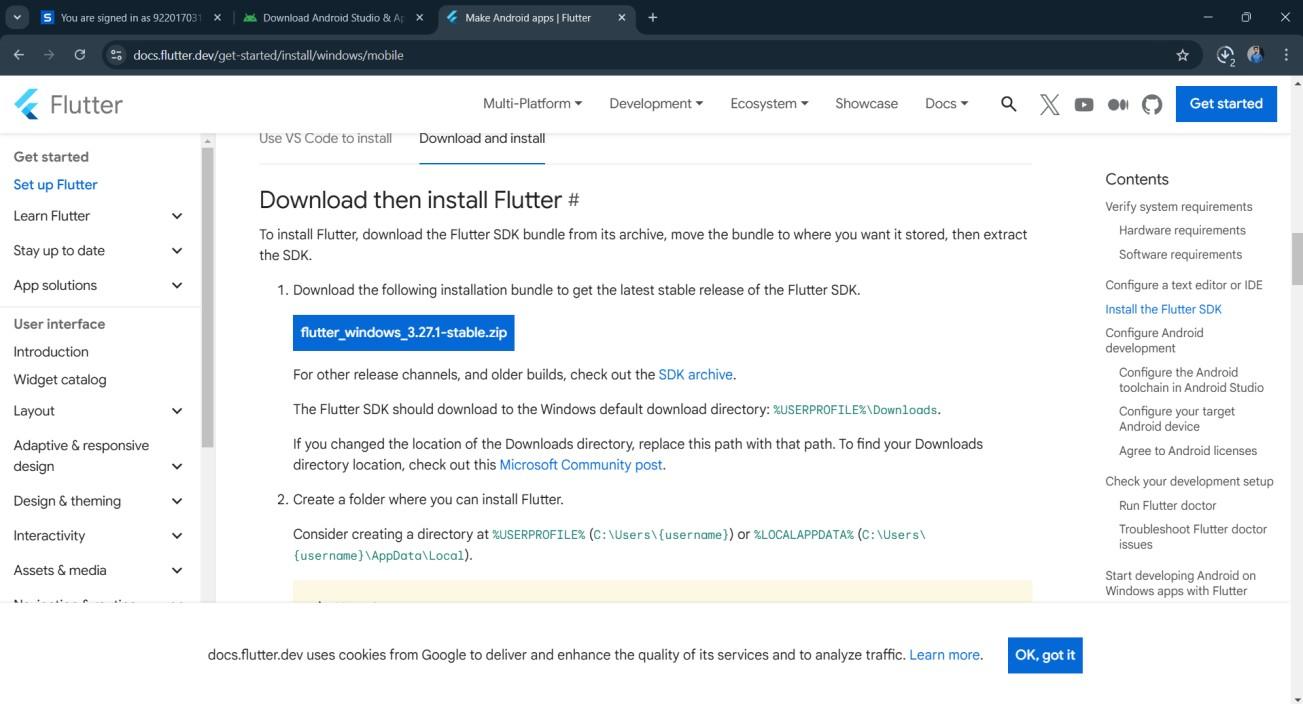


**Step 03 :** Install Flutter and Dart plugins.

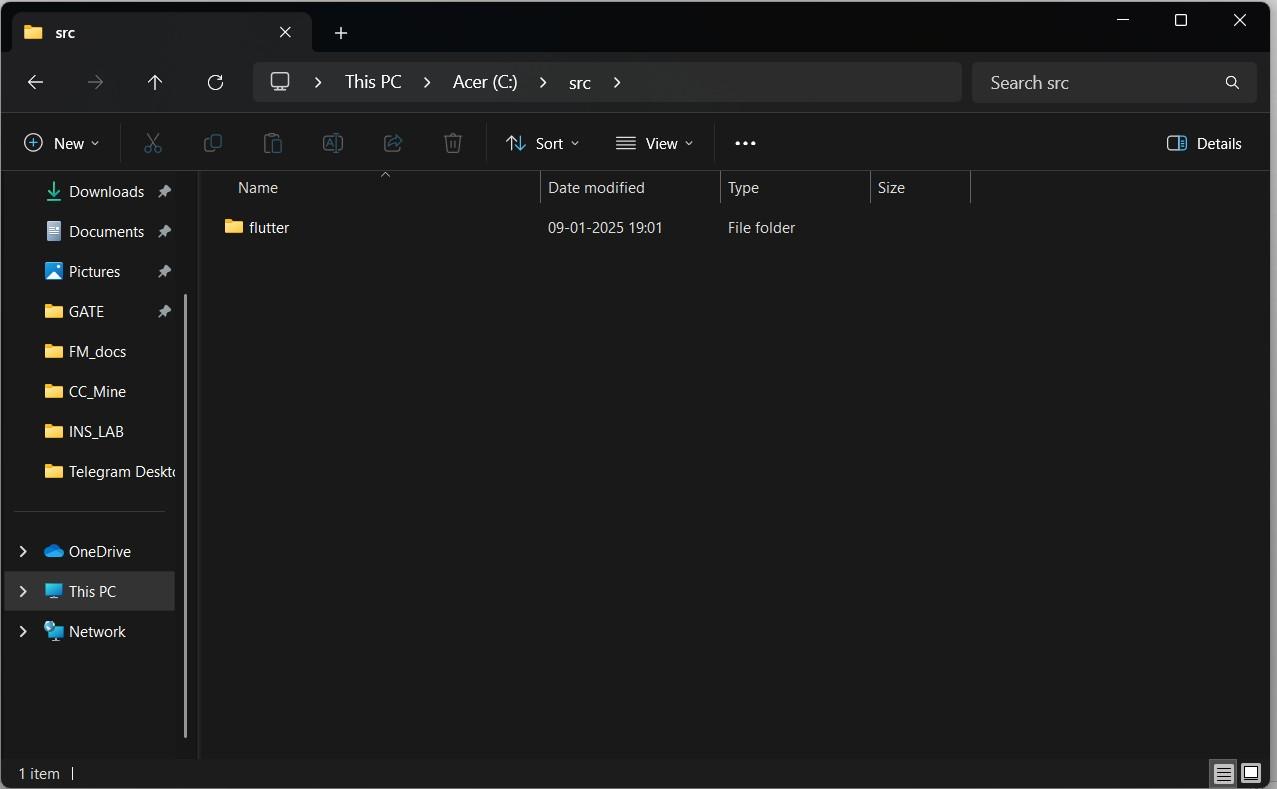




**Step 04 :** Download Flutter SDK from <https://docs.flutter.dev/get-started/install/windows/mobile>



**Step 05 :** Extract SDK zip file. create a new folder named src in C drive. Paste flutter folder from extracted SDk file to src folder.



**Step 05 :** Set environment variables. Follow below steps.

1. Click on Environment variable button.



1. Click on path and press edit button



1. Click on new and paste Flutter SDK path



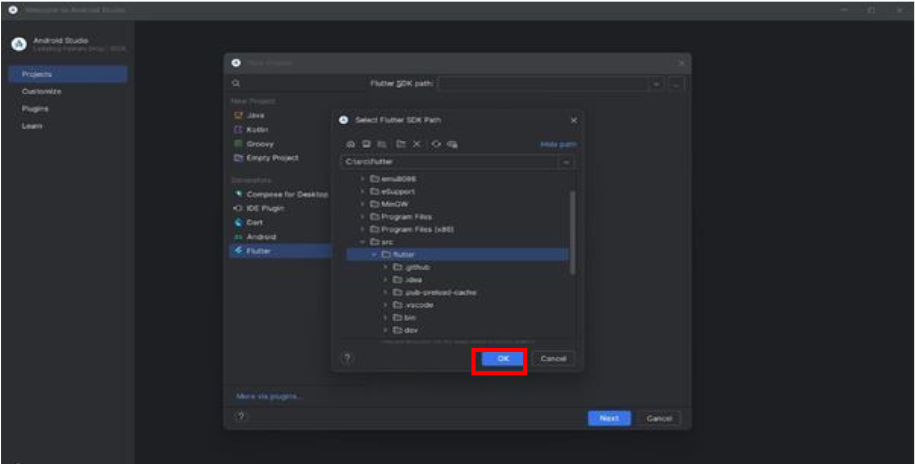
1. Do same for System variables too.





**Practical 2 : Create a “Hello Flutter” application.**

**Step 01 :** Create a new Flutter Project**.** Select Flutter SDK path.



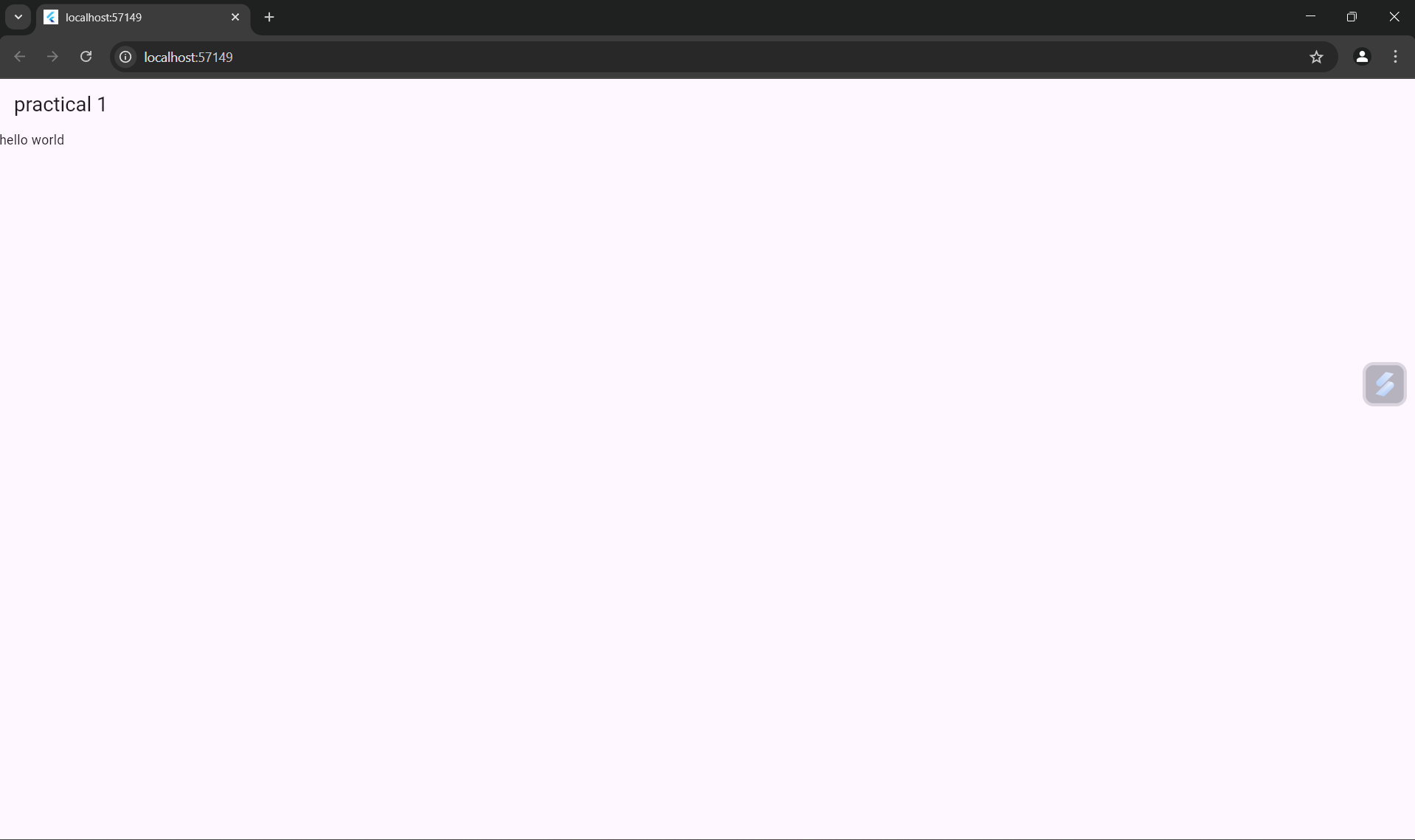
**Step 02 :**Select appropriate fields.



**Code :**

import 'package:flutter/material.dart';  
  
void main(){  
 runApp(MaterialApp  
 (debugShowCheckedModeBanner: false,home:   
 Scaffold(appBar:   
 AppBar(title:   
 Text("practical 1"),),body:  
 Text("hello world",),),));  
}

**Output 2 :**



**Practical 3(A) : Create and application using Flutter Key Widgets.**

**Code :**

import 'package:flutter/material.dart';

void main() {

runApp(MaterialApp(

debugShowCheckedModeBanner: false, // Disable the debug banner

home: Scaffold(

appBar: AppBar(

title: Text("practical 3(a)"),

),

body: Center(

child: Text(

"Hello World",

style: TextStyle(

color: Colors.pink,

fontSize: 86,

fontWeight: FontWeight.bold,

),

),

),

),

));

}

**Output 3 (A):**



**Practical 3(B): Create and application using Flutter Key Widgets.**

**code:**

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

home: Scaffold(

appBar: AppBar(

title: Text("Practical 3.1"),

backgroundColor: Colors.deepPurple,

elevation: 5,

),

body: Center(

child: Padding(

padding: EdgeInsets.all(10.0),

child: Container(

height: 80,

width: 80,

decoration: BoxDecoration(

color: Colors.blue,

borderRadius: BorderRadius.circular(15),

boxShadow: [

BoxShadow(

color: Colors.black26,

blurRadius: 10,

spreadRadius: 2,

offset: Offset(4, 4),

),

],

),

child: Icon(Icons.star, color: Colors.white, size: 40),

),

),

),

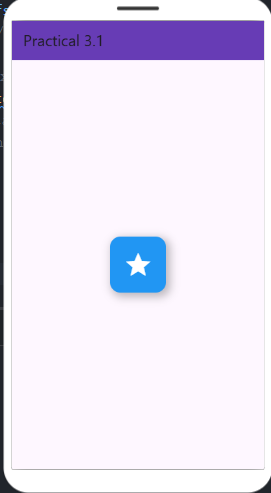
),

);

}

}

**Output 3(B) :**

****

**Practical 4 : Create a “Hello Flutter” application.**

**code:**

import 'package:flutter/material.dart';

void main() {

runApp(MaterialApp(

home: Scaffold(

appBar: AppBar(

title: Text("practial\_4"),

),

body: Row(

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

crossAxisAlignment: CrossAxisAlignment.center,

mainAxisSize: MainAxisSize.max,

textDirection: TextDirection.ltr,

children: [

Text(

'dhruv',

style: TextStyle(

color: Colors.grey,

fontSize: 24, // Apply font size here

fontWeight: FontWeight.bold, // Optional: Apply font weight

),

),

Text(

'dhruv',

style: TextStyle(

fontSize: 24, // Apply font size here

fontWeight: FontWeight.bold, // Optional: Apply font weight

),

),

],

),

),

));

}

**Output 4 :**

****

**Practical 5 : Create and application with Flutter UI Components.**

**code:**

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(

debugShowCheckedModeBanner: false,

theme: ThemeData(

primaryColor: Colors.blue,

scaffoldBackgroundColor: Colors.grey[200],

inputDecorationTheme: InputDecorationTheme(

filled: true,

fillColor: Colors.white,

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(12.0),

borderSide: BorderSide.none,

),

),

),

home: LoginScreen(),

);

}

}

class LoginScreen extends StatelessWidget {

final TextEditingController emailController = TextEditingController();

final TextEditingController passwordController = TextEditingController();

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: Padding(

padding: const EdgeInsets.all(20.0),

child: Column(

mainAxisSize: MainAxisSize.min,

children: [

const Text(

'Login',

style: TextStyle(

fontSize: 32,

fontWeight: FontWeight.bold,

color: Colors.blueAccent,

),

),

const SizedBox(height: 20),

TextField(

controller: emailController,

decoration: const InputDecoration(

labelText: 'Email',

prefixIcon: Icon(Icons.email, color: Colors.blueAccent),

),

keyboardType: TextInputType.emailAddress,

),

const SizedBox(height: 10),

TextField(

controller: passwordController,

decoration: const InputDecoration(

labelText: 'Password',

prefixIcon: Icon(Icons.lock, color: Colors.blueAccent),

),

obscureText: true,

),

const SizedBox(height: 20),

ElevatedButton(

onPressed: () {

String email = emailController.text;

String password = passwordController.text;

// Add login logic here

print('Email: $email, Password: $password');

},

style: ElevatedButton.styleFrom(

minimumSize: const Size(double.infinity, 50),

backgroundColor: Colors.blueAccent,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12.0),

),

),

child: const Text(

'Login',

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

),

),

],

),

),

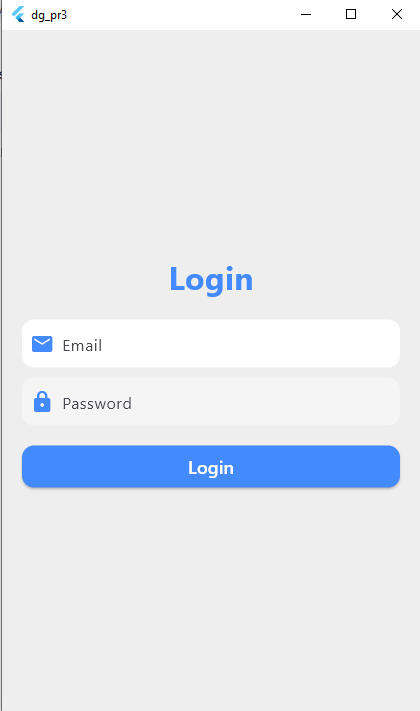
),

);

}

}

**out put:**

****

**Practical 6 : Create and application with Flutter UI Components for.**

**code:**

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(

debugShowCheckedModeBanner: false,

home: RegisterScreen(),

);

}

}

class RegisterScreen extends StatelessWidget {

final TextEditingController nameController = TextEditingController();

final TextEditingController emailController = TextEditingController();

final TextEditingController passwordController = TextEditingController();

final TextEditingController confirmPasswordController = TextEditingController();

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: Padding(

padding: const EdgeInsets.all(20.0),

child: Column(

mainAxisSize: MainAxisSize.min,

children: [

const Text(

'Register',

style: TextStyle(fontSize: 28, fontWeight: FontWeight.bold),

),

const SizedBox(height: 20),

TextField(

controller: nameController,

decoration: const InputDecoration(

labelText: 'Name',

border: OutlineInputBorder(),

prefixIcon: Icon(Icons.person),

),

),

const SizedBox(height: 10),

TextField(

controller: emailController,

decoration: const InputDecoration(

labelText: 'Email',

border: OutlineInputBorder(),

prefixIcon: Icon(Icons.email),

),

keyboardType: TextInputType.emailAddress,

),

const SizedBox(height: 10),

TextField(

controller: passwordController,

decoration: const InputDecoration(

labelText: 'Password',

border: OutlineInputBorder(),

prefixIcon: Icon(Icons.lock),

),

obscureText: true,

),

const SizedBox(height: 10),

TextField(

controller: confirmPasswordController,

decoration: const InputDecoration(

labelText: 'Confirm Password',

border: OutlineInputBorder(),

prefixIcon: Icon(Icons.lock\_outline),

),

obscureText: true,

),

const SizedBox(height: 20),

ElevatedButton(

onPressed: () {

String name = nameController.text.trim();

String email = emailController.text.trim();

String password = passwordController.text.trim();

String confirmPassword = confirmPasswordController.text.trim();

if (name.isEmpty || email.isEmpty || password.isEmpty || confirmPassword.isEmpty) {

print('All fields are required');

} else if (!RegExp(r'^[^@\s]+@[^@\s]+\.[^@\s]+\$').hasMatch(email)) {

print('Enter a valid email');

} else if (password.length < 6) {

print('Password must be at least 6 characters long');

} else if (password != confirmPassword) {

print('Passwords do not match');

} else {

print('Name: $name, Email: $email, Password: $password');

}

},

style: ElevatedButton.styleFrom(

minimumSize: const Size(double.infinity, 50),

),

child: const Text('Register'),

),

],

),

),

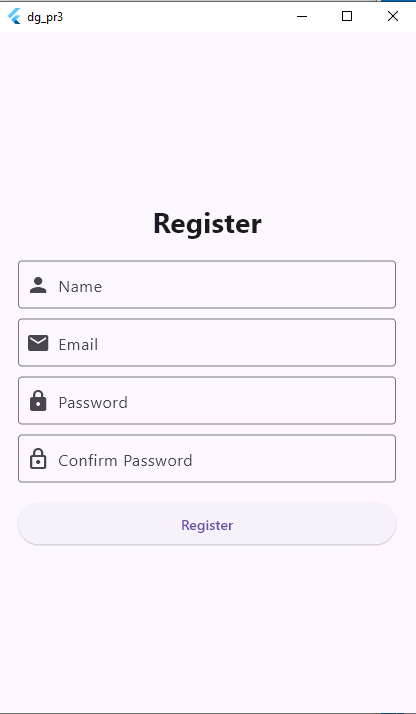
),

);

}

}

**output :**

****

**Practical 8 : Create and application with list view in Flutter.**

**Code :**

import 'package:flutter/material.dart';

void main() {

runApp(MaterialApp(

debugShowCheckedModeBanner: false,

home: Scaffold(

appBar: AppBar(

title: Text("Practical 8 "),

),

body: ListView(

children: [

ListTile(

leading: Icon(Icons.phone), // Mobile icon

title: Text("Mobile"),

),

ListTile(

leading: Icon(Icons.account\_balance), // Bank icon

title: Text("Bank"),

),

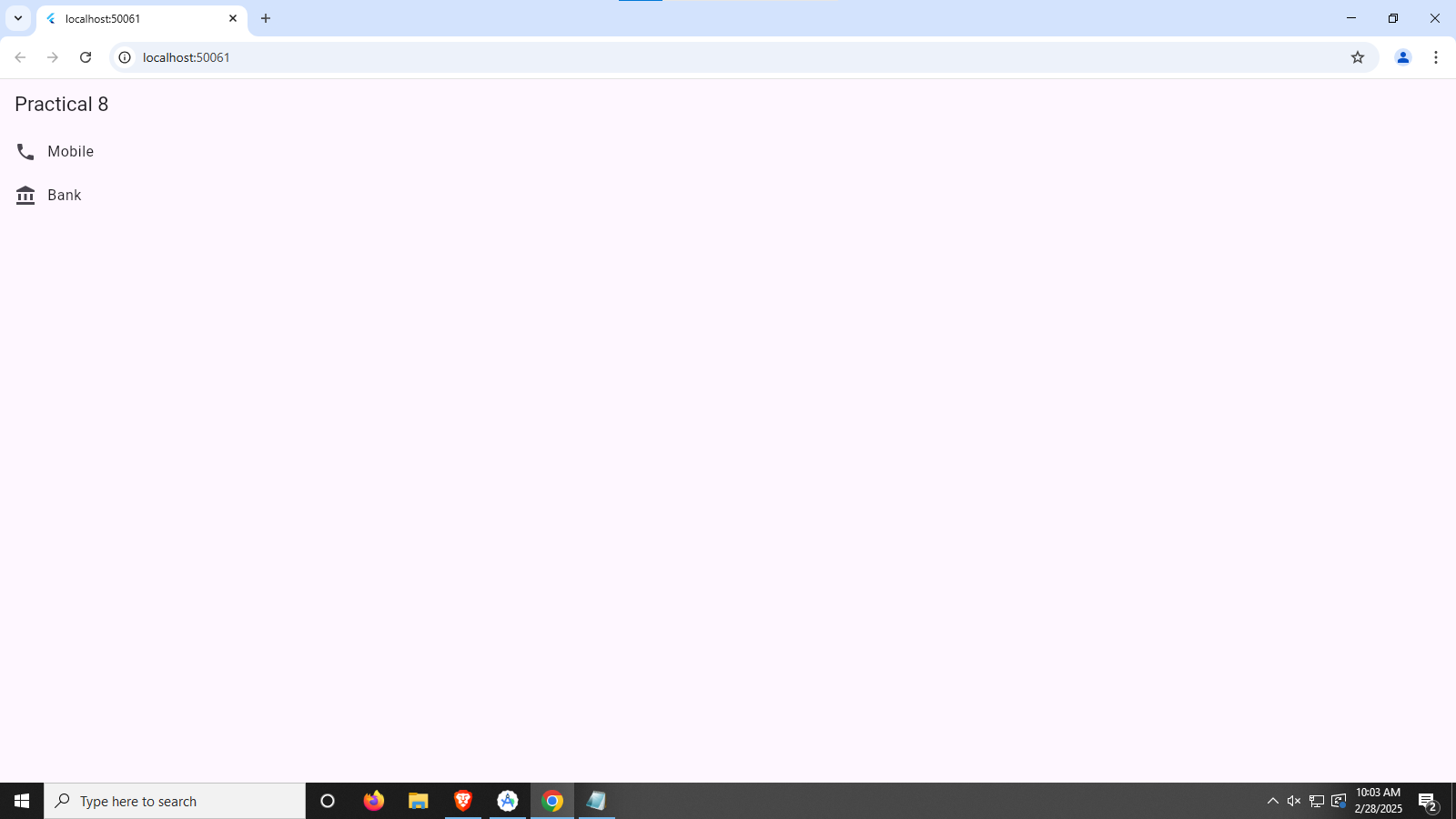
],

),

),

)); }

**out put:**

****

**Practical 9 : Create and application with grid view in Flutter.**

**Code :**

**import 'package:flutter/material.dart';**

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

home: Scaffold(

appBar: AppBar(title: Text('Practical 9')),

body: GridView.count(

crossAxisCount: 2, // Number of columns

crossAxisSpacing: 10,

mainAxisSpacing: 10,

padding: EdgeInsets.all(10),

children: List.generate(6, (index) {

return Card(

color: Colors.blueAccent,

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Icon(Icons.favorite, size: 40, color: Colors.white),

SizedBox(height: 10),

Text('Item ${index + 1}',

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

],

),

);

}),

),

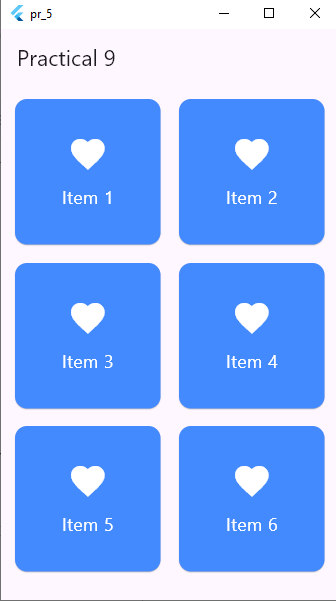
),

);

}

}

**out put:**

****

**Practical 10 : Create and application Crud Operation with SQLite in Flutter.**

**Code :**

**main.datr:**

import 'package:flutter/material.dart';

import 'package:resetapi/sqlHelper.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

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

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

title: 'SQLITE',

theme: ThemeData(

primarySwatch: Colors.orange,

),

home: const HomePage(),

);

}

}

class HomePage extends StatefulWidget {

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

@override

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

}

class \_HomePageState extends State<HomePage> {

List<Map<String, dynamic>> \_journals = [];

bool \_isLoading = true;

final TextEditingController \_titleController = TextEditingController();

final TextEditingController \_descriptionController = TextEditingController();

@override

void initState() {

super.initState();

\_refreshJournals();

}

void \_refreshJournals() async {

final data = await SQLHelper.getItems();

setState(() {

\_journals = data;

\_isLoading = false;

});

}

Future<void> \_addItem() async {

await SQLHelper.createItem(\_titleController.text, \_descriptionController.text);

\_refreshJournals();

}

Future<void> \_updateItem(int id) async {

await SQLHelper.updateItem(id, \_titleController.text, \_descriptionController.text);

\_refreshJournals();

}

void \_deleteItem(int id) async {

await SQLHelper.deleteItem(id);

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(content: Text('Successfully deleted a journal!')),

);

\_refreshJournals();

}

void \_showForm(int? id) async {

if (id != null) {

final existingJournal = \_journals.firstWhere((element) => element['id'] == id);

\_titleController.text = existingJournal['title'];

\_descriptionController.text = existingJournal['description'];

} else {

\_titleController.clear();

\_descriptionController.clear();

}

showModalBottomSheet(

context: context,

elevation: 5,

isScrollControlled: true,

builder: (\_) => Container(

padding: EdgeInsets.only(

top: 15,

left: 15,

right: 15,

bottom: MediaQuery.of(context).viewInsets.bottom + 120,

),

child: Column(

mainAxisSize: MainAxisSize.min,

crossAxisAlignment: CrossAxisAlignment.end,

children: [

TextField(

controller: \_titleController,

decoration: const InputDecoration(hintText: 'Title'),

),

const SizedBox(height: 10),

TextField(

controller: \_descriptionController,

decoration: const InputDecoration(hintText: 'Description'),

),

const SizedBox(height: 20),

ElevatedButton(

onPressed: () async {

if (id == null) {

await \_addItem();

} else {

await \_updateItem(id);

}

\_titleController.clear();

\_descriptionController.clear();

Navigator.of(context).pop();

},

child: Text(id == null ? 'Create New' : 'Update'),

),

],

),

),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: const Text('SQL')),

body: \_isLoading

? const Center(child: CircularProgressIndicator())

: ListView.builder(

itemCount: \_journals.length,

itemBuilder: (context, index) => Card(

color: Colors.orange[200],

margin: const EdgeInsets.all(15),

child: ListTile(

title: Text(\_journals[index]['title']),

subtitle: Text(\_journals[index]['description']),

trailing: SizedBox(

width: 100,

child: Row(

children: [

IconButton(

icon: const Icon(Icons.edit),

onPressed: () => \_showForm(\_journals[index]['id']),

),

IconButton(

icon: const Icon(Icons.delete),

onPressed: () => \_deleteItem(\_journals[index]['id']),

),

],

),

),

),

),

),

floatingActionButton: FloatingActionButton(

child: const Icon(Icons.add),

onPressed: () => \_showForm(null),

),

);

}

}

**sqlHelper.dart:**

import 'package:flutter/foundation.dart';

import 'package:sqflite/sqflite.dart' as sql;

class SQLHelper {

static Future<void> createTables(sql.Database database) async {

await database.execute("""

CREATE TABLE items(

id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,

title TEXT,

description TEXT,

createdAt TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP

)

""");

}

static Future<sql.Database> db() async {

return sql.openDatabase(

'dbtech.db',

version: 1,

onCreate: (sql.Database database, int version) async {

await createTables(database);

},

);

}

static Future<int> createItem(String title, String? description) async {

final db = await SQLHelper.db();

final data = {'title': title, 'description': description};

final id = await db.insert('items', data,

conflictAlgorithm: sql.ConflictAlgorithm.replace);

return id;

}

static Future<List<Map<String, dynamic>>> getItems() async {

final db = await SQLHelper.db();

return db.query('items', orderBy: "id");

}

static Future<List<Map<String, dynamic>>> getItem(int id) async {

final db = await SQLHelper.db();

return db.query('items', where: "id = ?", whereArgs: [id], limit: 1);

}

static Future<int> updateItem(int id, String title, String? description) async {

final db = await SQLHelper.db();

final data = {

'title': title,

'description': description,

'createdAt': DateTime.now().toString()

};

final result = await db.update('items', data, where: "id = ?", whereArgs: [id]);

return result;

}

static Future<void> deleteItem(int id) async {

final db = await SQLHelper.db();

try {

await db.delete("items", where: "id = ?", whereArgs: [id]);

} catch (err) {

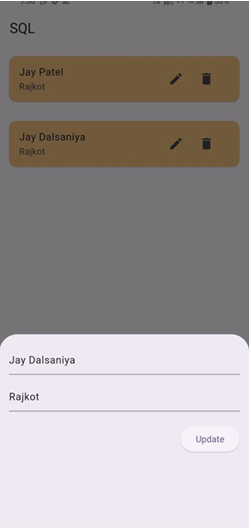
debugPrint("Something went wrong when deleting an item: $err");

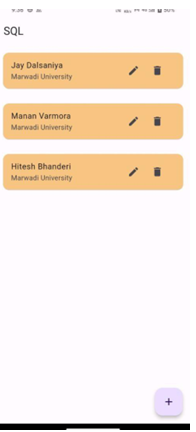
}

}

}

**output:**

****

****

**Practical 11: Create and application Connecting to REST API in Flutter.**

**main.dart:**

import 'package:flutter/material.dart';

import 'package:resetapi/data\_screen.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

title: 'Flutter REST API Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: DataScreen(),

);

}

}

**api\_service.dart:**

import 'dart:convert';

import 'package:http/http.dart' as http;

// Model class for Post

class Post {

final int userId;

final int id;

final String title;

final String body;

Post({

required this.userId,

required this.id,

required this.title,

required this.body,

});

// Factory method to create a Post from JSON

factory Post.fromJson(Map<String, dynamic> json) {

return Post(

userId: json['userId'],

id: json['id'],

title: json['title'],

body: json['body'],

);

}

}

// Service class to fetch data from API

class ApiService {

static const String baseUrl = 'https://jsonplaceholder.typicode.com';

static Future<List<Post>> fetchPosts() async {

final response = await http.get(Uri.parse('$baseUrl/posts'));

if (response.statusCode == 200) {

List<dynamic> jsonResponse = json.decode(response.body);

return jsonResponse.map((post) => Post.fromJson(post)).toList();

} else {

throw Exception('Failed to load posts');

}

}

}

**data\_screen.dart:**

import 'package:flutter/material.dart';

import 'package:resetapi/api\_service.dart';

class DataScreen extends StatefulWidget {

@override

\_DataScreenState createState() => \_DataScreenState();

}

class \_DataScreenState extends State<DataScreen> {

late Future<List<Post>> posts;

@override

void initState() {

super.initState();

posts = ApiService.fetchPosts();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Posts'),

),

body: Center(

child: FutureBuilder<List<Post>>(

future: posts,

builder: (context, snapshot) {

if (snapshot.hasData) {

return ListView.builder(

itemCount: snapshot.data!.length,

itemBuilder: (context, index) {

return Card(

elevation: 3,

margin: EdgeInsets.all(10),

child: Padding(

padding: EdgeInsets.all(10),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(

'Post ${index + 1}:',

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 16,

),

),

SizedBox(height: 5),

Text(

snapshot.data![index].title,

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 18,

),

),

SizedBox(height: 5),

Text(snapshot.data![index].body),

],

),

),

);

},

);

} else if (snapshot.hasError) {

return Text("${snapshot.error}");

}

// By default, show a loading spinner

return CircularProgressIndicator();

},

),

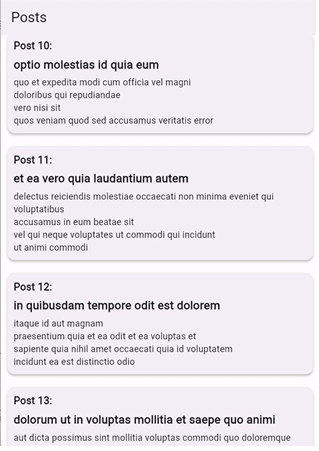
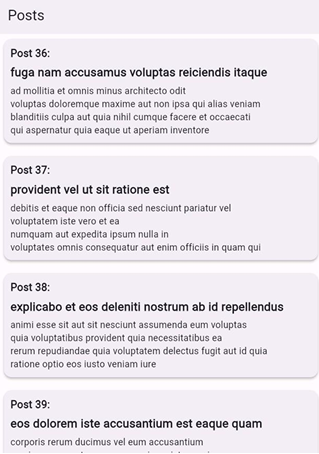
),

);

}

}

**Output:**

** **

**Practical 12: Create and application Parsing JSON data from REST API in Flutter.**

**main.dart:**

import 'package:flutter/material.dart';

import 'package:resetapi/data\_screen.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

title: 'Flutter REST API Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: DataScreen(),

);

}

}

**api\_service.dart:**

import 'dart:convert';

import 'package:http/http.dart' as http;

// Model class representing a Post

class Post {

final int userId;

final int id;

final String title;

final String body;

Post({

required this.userId,

required this.id,

required this.title,

required this.body,

});

// Factory constructor to create Post object from JSON

factory Post.fromJson(Map<String, dynamic> json) {

return Post(

userId: json['userId'],

id: json['id'],

title: json['title'],

body: json['body'],

);

}

}

// API Service class to handle API calls

class ApiService {

static const String baseUrl = 'https://jsonplaceholder.typicode.com';

static Future<List<Post>> fetchPosts() async {

final response = await http.get(Uri.parse('$baseUrl/posts'));

if (response.statusCode == 200) {

List<dynamic> jsonResponse = json.decode(response.body);

return jsonResponse.map((post) => Post.fromJson(post)).toList();

} else {

throw Exception('Failed to load posts');

}

}

}

**data\_screen.dart:**

import 'package:flutter/material.dart';

import 'package:resetapi/api\_service.dart';

class DataScreen extends StatefulWidget {

@override

\_DataScreenState createState() => \_DataScreenState();

}

class \_DataScreenState extends State<DataScreen> {

late Future<List<Post>> posts;

@override

void initState() {

super.initState();

posts = ApiService.fetchPosts();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Posts'),

),

body: Center(

child: FutureBuilder<List<Post>>(

future: posts,

builder: (context, snapshot) {

if (snapshot.hasData) {

return ListView.builder(

itemCount: snapshot.data!.length,

itemBuilder: (context, index) {

return Card(

elevation: 3,

margin: EdgeInsets.all(10),

child: Padding(

padding: EdgeInsets.all(10),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(

'Post ${index + 1}:',

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 16,

),

),

SizedBox(height: 5),

Text(

snapshot.data![index].title,

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 18,

),

),

SizedBox(height: 5),

Text(snapshot.data![index].body),

],

),

),

);

},

);

} else if (snapshot.hasError) {

return Text("${snapshot.error}");

}

// By default, show a loading spinner

return CircularProgressIndicator();

},

),

),

);

}

}

**post\_model.dart:**

class Post {

final int userId;

final int id;

final String title;

final String body;

Post({

required this.userId,

required this.id,

required this.title,

required this.body,

});

factory Post.fromJson(Map<String, dynamic> json) {

return Post(

userId: json['userId'],

id: json['id'],

title: json['title'],

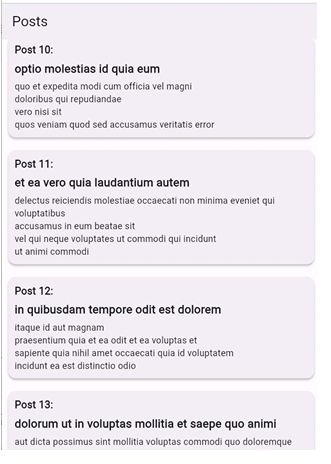
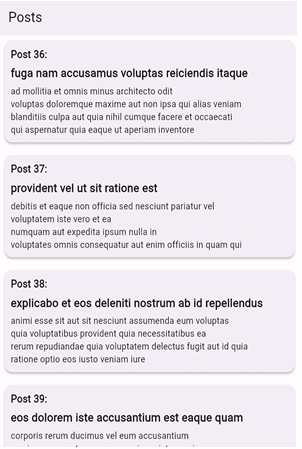
body: json['body'],

);

}

}

**output:**

** **

**Practical 13: Create and application using Hardware Interaction in Flutter.**

**main.dart:**

import 'package:flutter/material.dart';

import 'home\_screen.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

title: "Text To Speech",

theme: ThemeData(

primarySwatch: Colors.indigo,

),

home: HomeScreen(),

);

}

}

**homescreen.dart:**

import 'dart:async';

import 'package:flutter/material.dart';

import 'package:flutter\_tts/flutter\_tts.dart';

class HomeScreen extends StatefulWidget {

const HomeScreen({super.key});

@override

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

}

class \_HomeScreenState extends State<HomeScreen> {

final FlutterTts flutterTts = FlutterTts();

final TextEditingController textController = TextEditingController();

@override

void dispose() {

textController.dispose();

super.dispose();

}

Future<void> speak(String text) async {

await flutterTts.setLanguage('en-US');

await flutterTts.setPitch(1.0);

await flutterTts.setSpeechRate(0.5);

await flutterTts.speak(text);

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text("Text To Speech"),

),

body: Padding(

padding: EdgeInsets.all(20),

child: Column(

crossAxisAlignment: CrossAxisAlignment.stretch,

children: [

TextField(

controller: textController,

decoration: InputDecoration(

hintText: 'Enter Text',

border: OutlineInputBorder(),

),

maxLines: 4,

),

SizedBox(height: 30),

ElevatedButton(

onPressed: () {

speak(textController.text);

},

child: Text('Speak'),

),

],

),

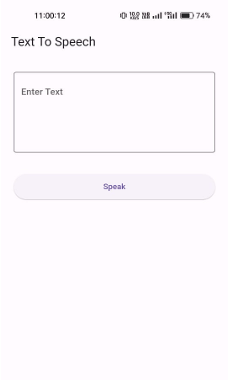
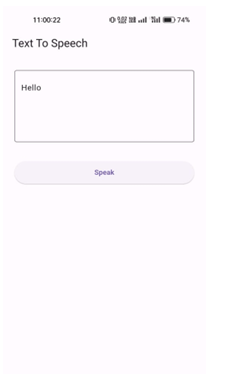
),

);

}

}

**output:**

** **