Experiment :- 01

AIM:- Android Studio setup for Flutter development with along with Dart SDK.

Step1:InstallingaFlutter.

i.SystemRequirements:

- Assurethatyoursystemmeetstheminimumrequirements.FluttersupportsmacOS,Linux, and Windows.
- OnmacOS,youneedXcodewiththecommand-linetoolsinstalled.
- OnLinux,youneedtohavegit,lib32stdc++6,andotherdependenciesinstalled.

ii. **DownloadFlutter:**

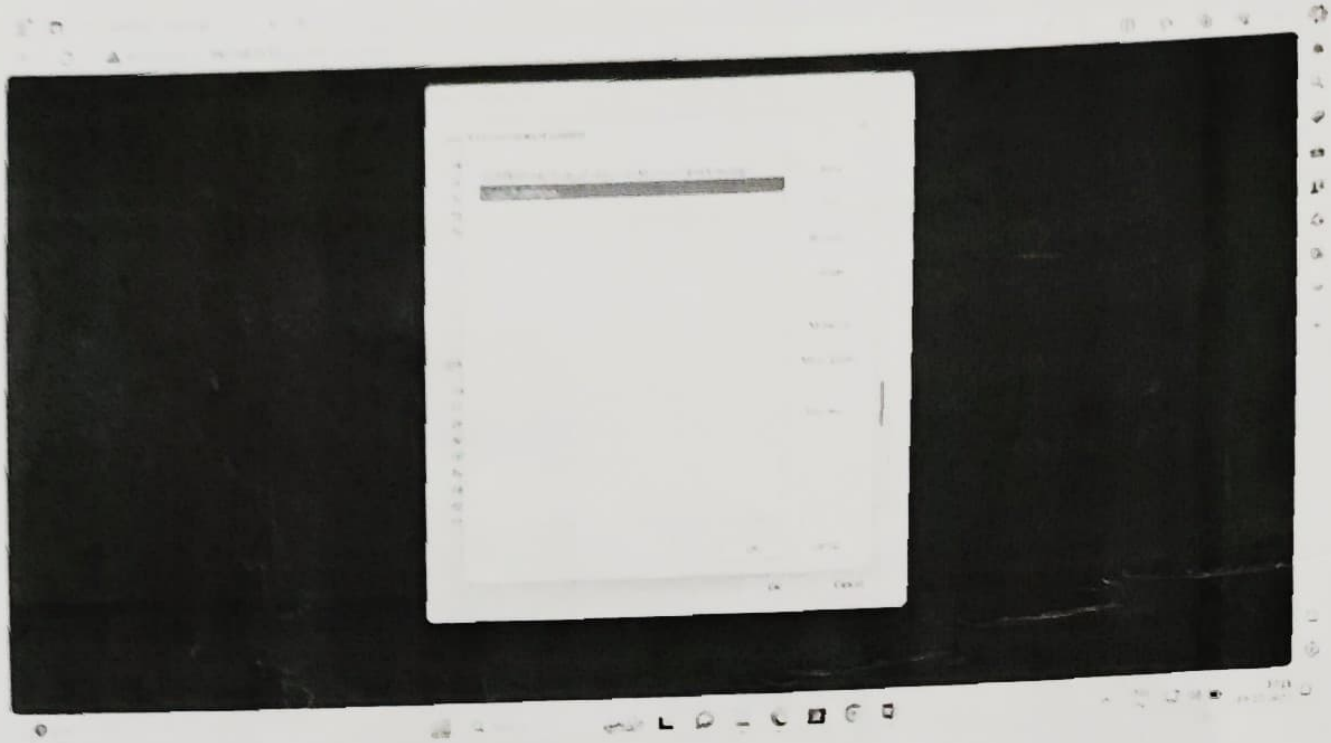
- VisitFlutterWebsiteforInstallationofFlutter->**https://docs.flutter.dev/get-started/install**.

iii. **ExtractFlutter:**

- IfyoudownloadedtheZIPfile,extractittoalocationonyourmachine.**(C:\src\flutter)**.

iv. **SetUpEnvironmentVariables:**

- Addthe**C:\src\flutter\bin**directorytoyoursystem's PATHvariable.



v. **Runflutterdoctor:**

- Openaterminalandrunthefollowingcommand:flutterdoctor
- Thiscommandchecksyourenvironmentanddisplaysareportofanymissingdependencies or issues.

vi. **InstallFlutterDependencies:**

- Followtheinstructionsprovidedbyflutterdoctortoinstallanymissingdependencies.This may include things like Android Studio, Xcode command-line tools, etc.

Step2:InstallingAndroidStudio.

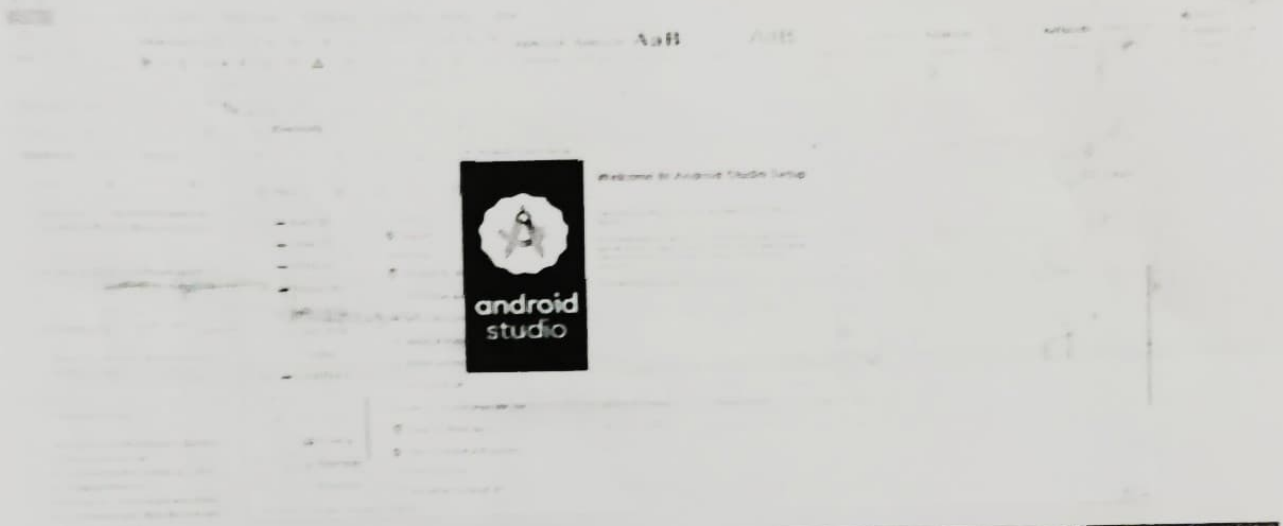
i.DownloadAndroidStudio:

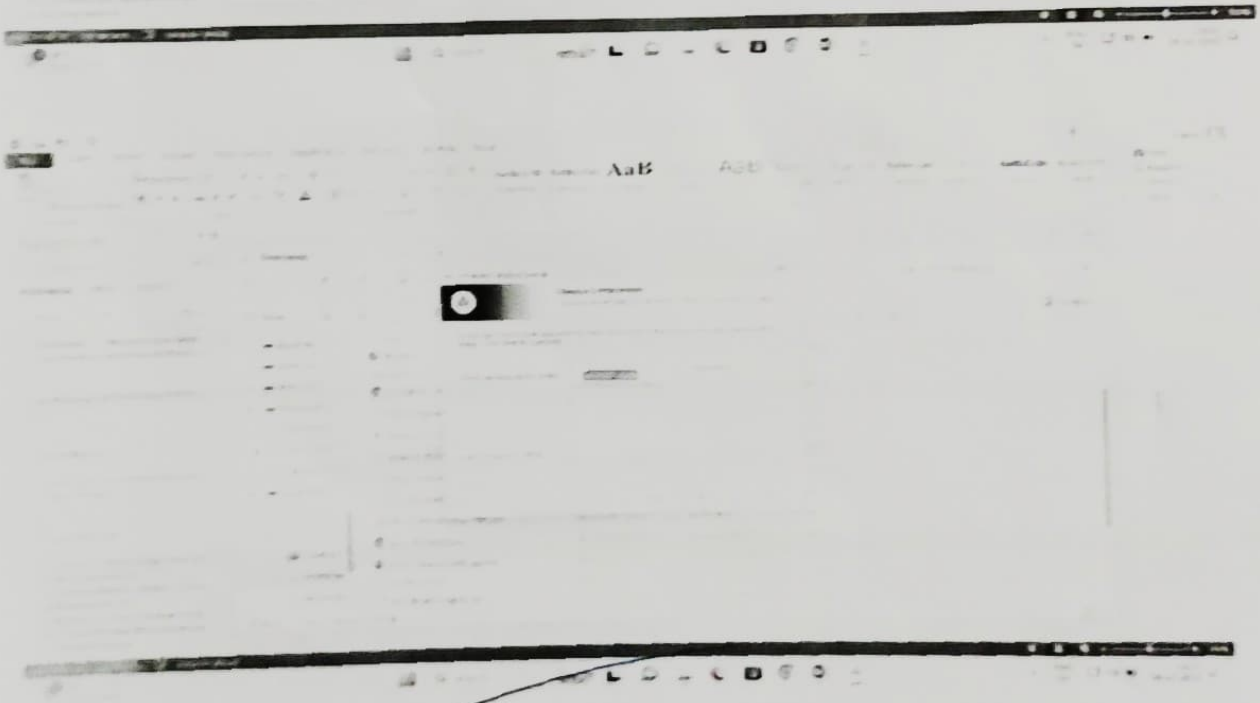
- VisittheAndroidStudiodownloadpage.
- Clickonthe"Download"buttonanddownloadtheWindowsversion.

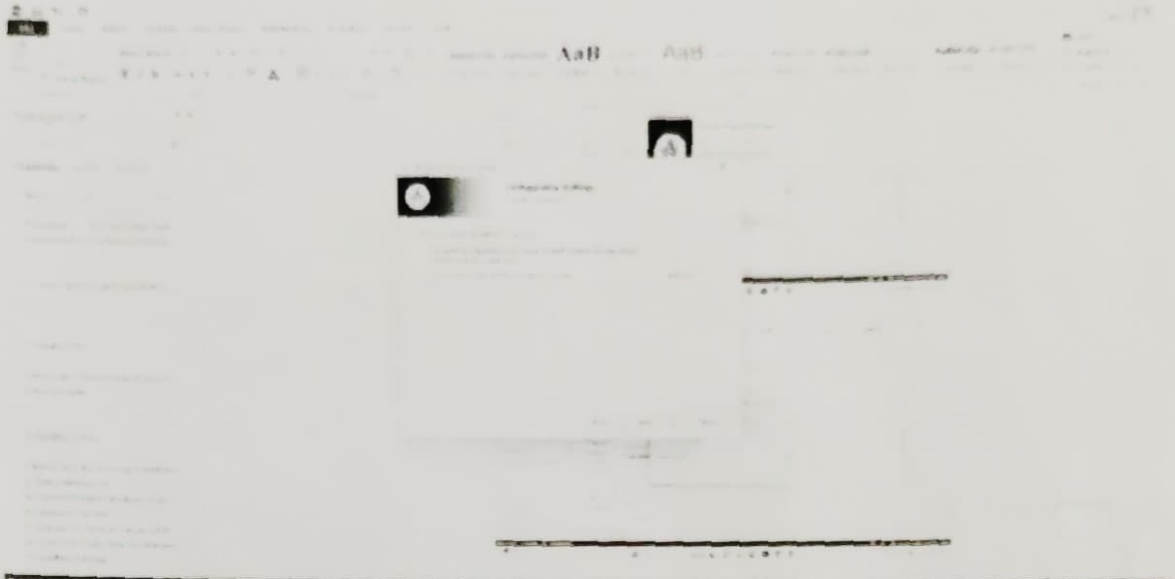
ii. **RuntheInstaller:**

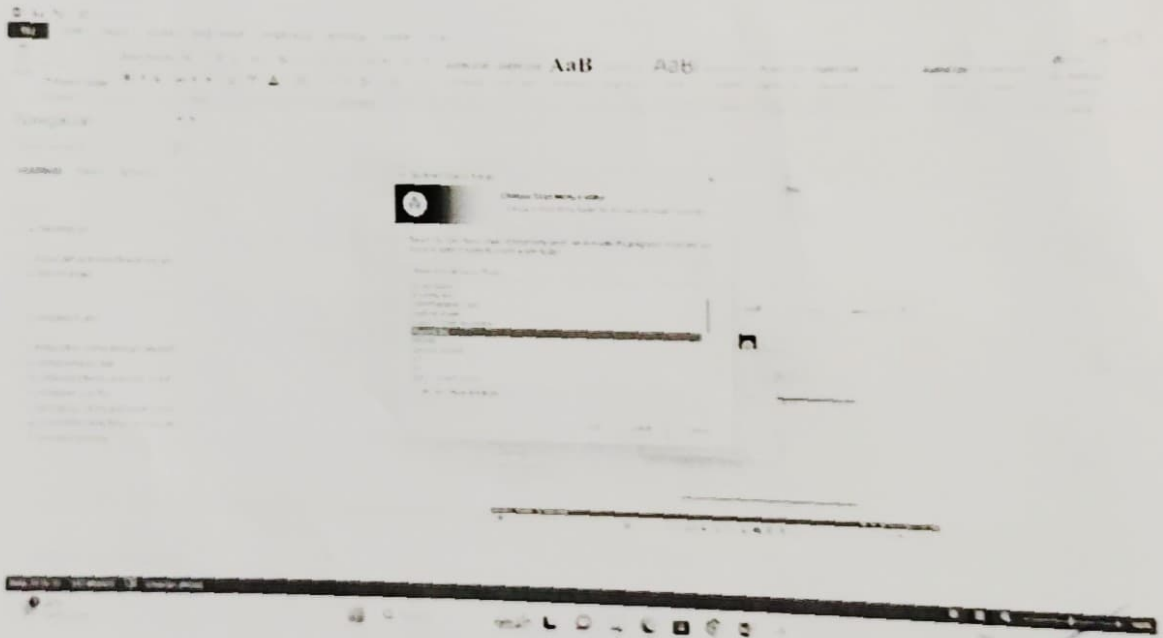
- Oncethedownloadiscomplete,runtheinstallerexecutable(.exe)file.

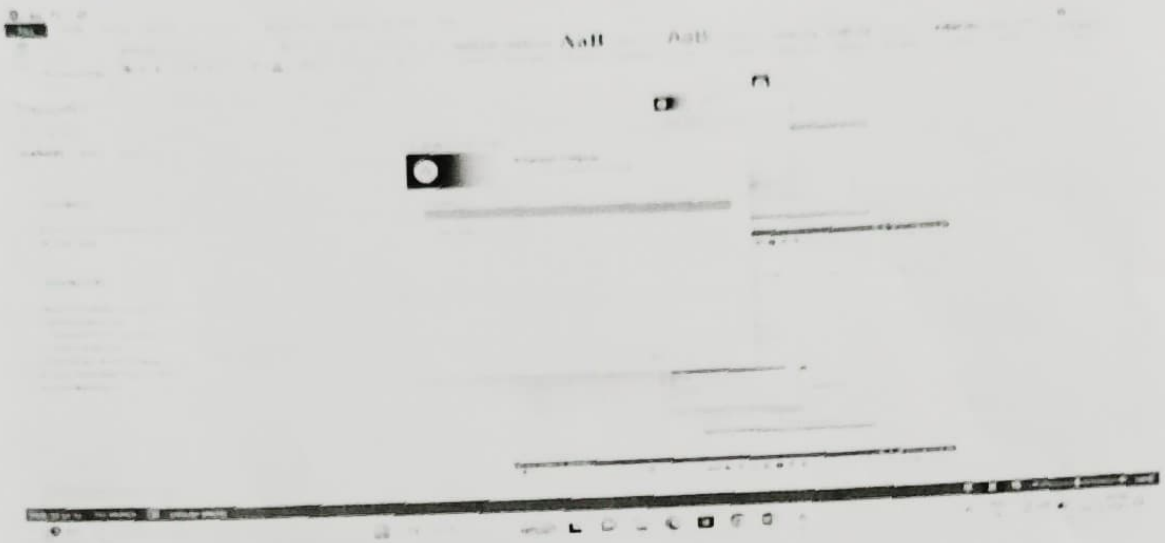
iii. **FollowInstallationWizard:**





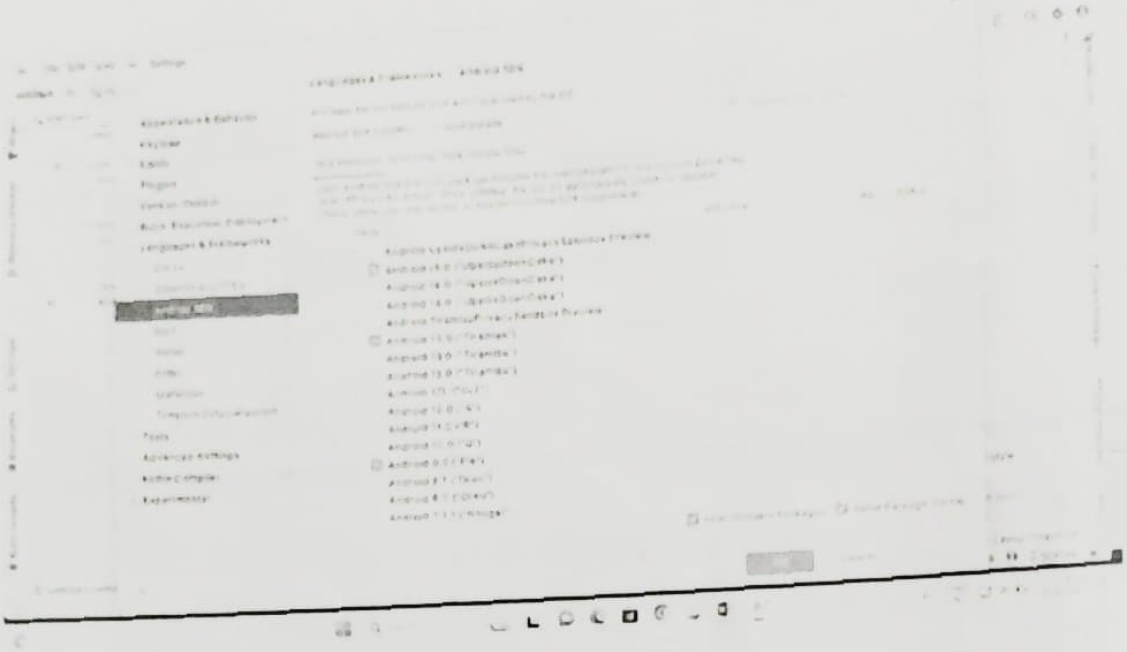




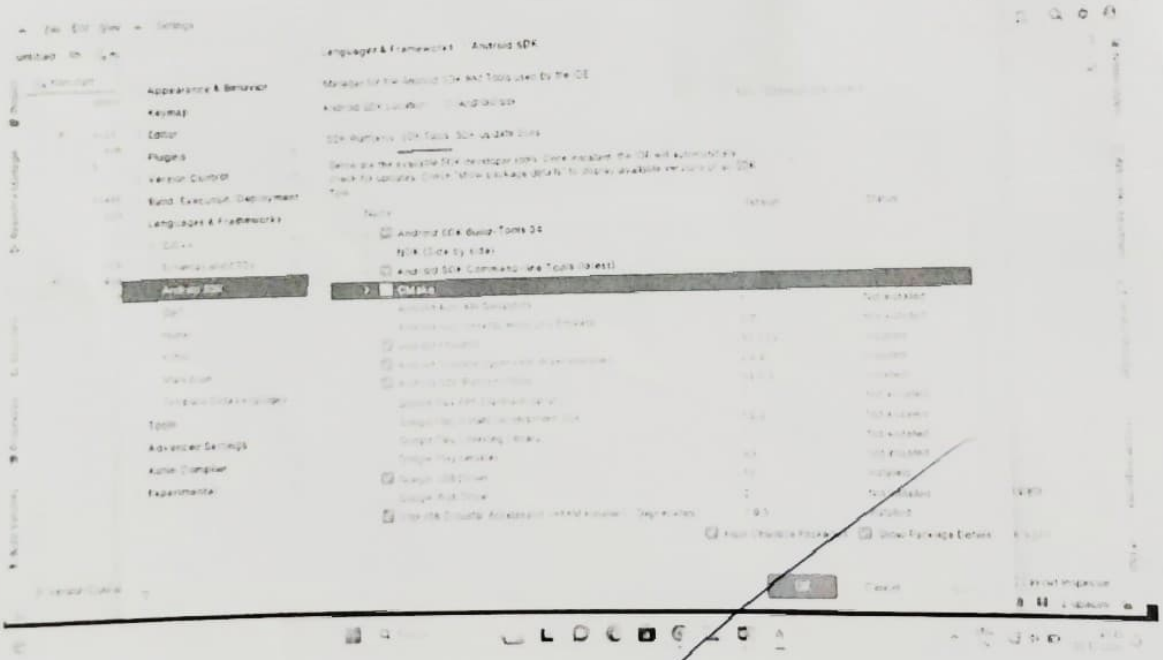


Completing Android Studio Setup
android
studio

- AndroidSDKPlatforms:



- AndroidSDKTools:



- **Step3:Run Following Command for checking Flutter dependencies after installation of android.**

iv. Accept **AndroidLicenses**

- Flutterdoctor--android-licensestodevelopforAndroid,youneedtoacceptheAndroid licenses.
- Runthefollowingcommand:**flutterdoctor--android-licenses**

22/2/25

Experiment 02

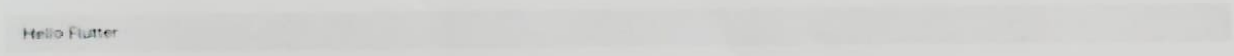
AIM:- Create a Hello Flutter application.

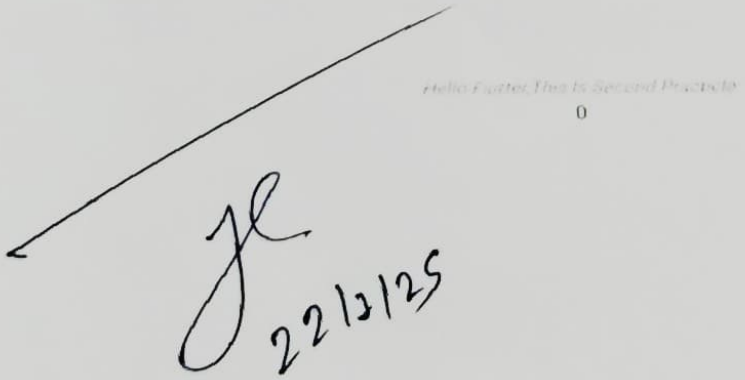
Source :- Mai.dart

Code :-

```
const Text(
'Hello Flutter:',
style: TextStyle(color: Colors.red, fontSize: 25, fontStyle: FontStyle.italic, ),
).Text(
'$_counter',
style: Theme.of(context).textTheme.headlineMedium, ), ],.), ).
```

Output:-





Experiment 03

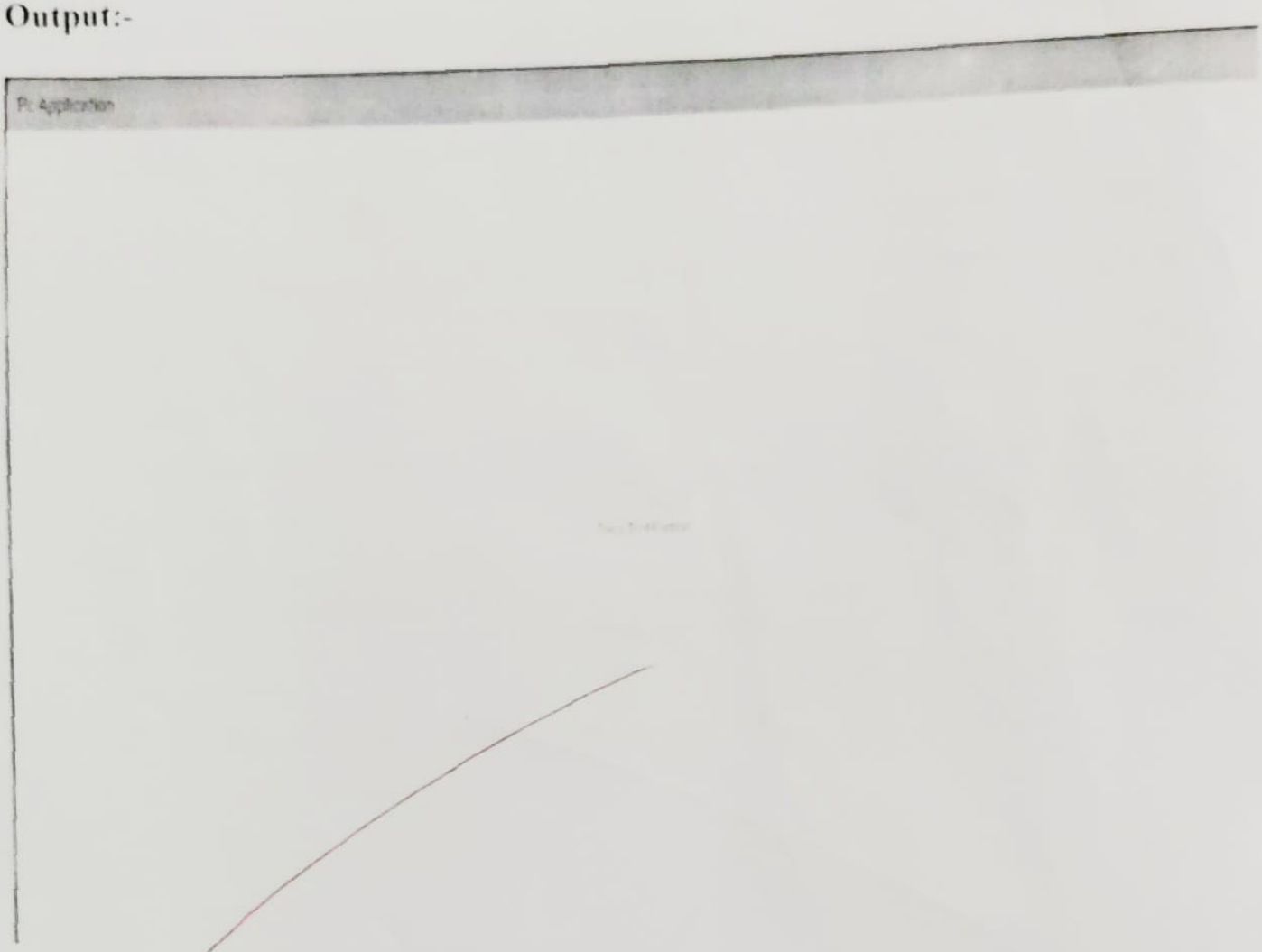
AIM:- Create and application using Flutter Key Widgets.

Source :- Mai.dart

Code :-

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

void main() {
  runApp(
    MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: Text("PC Application"),
          toolbarTextStyle: TextStyle(color: Colors.green, fontWeight: FontWeight.bold, fontSize: 50),
          backgroundColor: Colors.pink, // Set the background color of the AppBar
        ),
        body: Center(
          child: Text(
            'This is Third Practical!',
            textAlign: TextAlign.center,

          ),
        ),
      ),
    ),
  );
```

Output:-



Experiment 04

AIM:- Create and application with Flutter UI Components.

Source :- Mai.dart

Code :-

```
import 'package:flutter/material.dart';
void main() {
 runApp(MaterialApp(
  debugShowCheckedModeBanner: false,
  home: Scaffold(
   appBar: AppBar(
    title: Text(
     "Practical 4",
     style: TextStyle(color: Colors.red),
    ),
   ),
   body: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
     // First Row
     Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
       Container(
```

```
width: 100,
height: 50,
margin: EdgeInsets.all(10),
padding: EdgeInsets.all(10),
decoration: BoxDecoration(
  color: Colors.pink,
  border: Border.all(
    color: Colors.black,
    width: 2.0,
  ),
),
child: Icon(
  Icons.star,
  color: Colors.black,
),
),
Container(
  width: 100,
  height: 50,
  margin: EdgeInsets.all(10),
  padding: EdgeInsets.all(10),
  decoration: BoxDecoration(
    color: Colors.pink,
    border: Border.all(
      color: Colors.black,
```

```
      width: 2.0,
    ),
  ),
  child: Icon(
    Icons.star,
    color: Colors.black,
  ),
),
Container(
  width: 100,
  height: 50,
  margin: EdgeInsets.all(10),
  padding: EdgeInsets.all(10),
  decoration: BoxDecoration(
    color: Colors.pink,
    border: Border.all(
      color: Colors.black,
      width: 2.0,
    ),
  ),
  child: Icon(
    Icons.star,
    color: Colors.black,
  ),
),
```

```
 ],
),
// Second Row
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Container(
      width: 100,
      height: 50,
      margin: EdgeInsets.all(10),
      padding: EdgeInsets.all(10),
      decoration: BoxDecoration(
        color: Colors.pink,
        border: Border.all(
          color: Colors.black,
          width: 2.0,
        ),
      ),
      child: Icon(
        Icons.star,
        color: Colors.black,
      ),
    ),
    Container(
      width: 100,
```

```
height: 50,
margin: EdgeInsets.all(10),
padding: EdgeInsets.all(10),
decoration: BoxDecoration(
  color: Colors.pink,
  border: Border.all(
    color: Colors.black,
    width: 2.0,
  ),
),
child: Icon(
  Icons.star,
  color: Colors.black,
),
),
Container(
  width: 100,
  height: 50,
  margin: EdgeInsets.all(10),
  padding: EdgeInsets.all(10),
  decoration: BoxDecoration(
    color: Colors.pink,
    border: Border.all(
      color: Colors.black,
      width: 2.0,
```

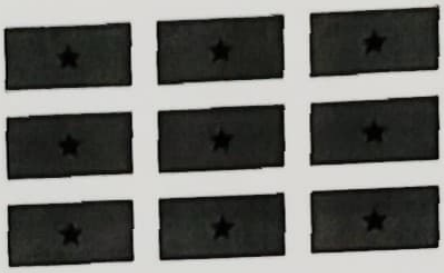
```
      ),
    ),
    child: Icon(
      Icons.star,
      color: Colors.black,
    ),
  ),
],
),
// Third Row
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Container(
      width: 100,
      height: 50,
      margin: EdgeInsets.all(10),
      padding: EdgeInsets.all(10),
      decoration: BoxDecoration(
        color: Colors.pink,
        border: Border.all(
          color: Colors.black,
          width: 2.0,
        ),
      ),
```

```
  child: Icon(
    Icons.star,
    color: Colors.black,
  ),
),
Container(
  width: 100,
  height: 50,
  margin: EdgeInsets.all(10),
  padding: EdgeInsets.all(10),
  decoration: BoxDecoration(
    color: Colors.pink,
    border: Border.all(
      color: Colors.black,
      width: 2.0,
    ),
  ),
  child: Icon(
    Icons.star,
    color: Colors.black,
  ),
),
Container(
  width: 100,
  height: 50,
```

```
margin: EdgeInsets.all(10),
padding: EdgeInsets.all(10),
decoration: BoxDecoration(
  color: Colors.pink,
  border: Border.all(
    color: Colors.black,
    width: 2.0,
  ),
),
child: Icon(
  Icons.star,
  color: Colors.black,
),
),
],
),
],
),
),
));
}
```

Output:-

Practical 4



Experiment 05

AIM:- Create and application with Flutter UI Components.

Source :- Mai.dart

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: const LoginScreen(),
    );
  }
}


class LoginScreen extends StatelessWidget {
```

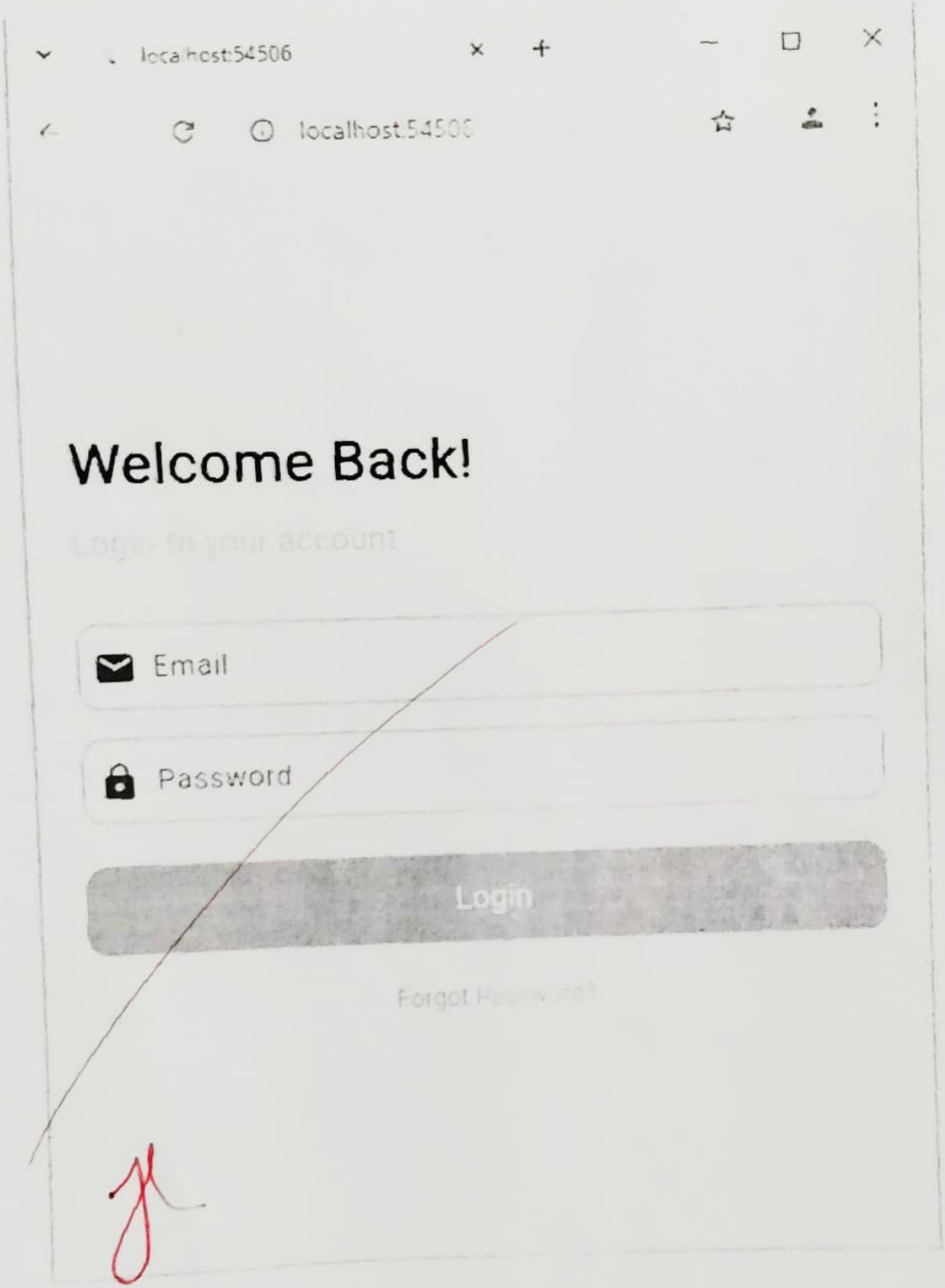
```
const LoginScreen({super.key});

@override
Widget build(BuildContext context) {
 final TextEditingController emailController = TextEditingController();
 final TextEditingController passwordController = TextEditingController();

return Scaffold(
 backgroundColor: Colors.blueGrey[50],
 body: Center(
  child: SingleChildScrollView(
   child: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 24.0),
    child: Column(
     mainAxisAlignment: MainAxisAlignment.center,
     crossAxisAlignment: CrossAxisAlignment.start,
     children: [
      const Text(
       'Welcome Back!',
       style: TextStyle(
        fontSize: 32,
        fontWeight: FontWeight.bold,
       ),
     ),
     const SizedBox(height: 8),
```

```
const Text(
 'Login to your account',
 style: TextStyle(
  fontSize: 18,
  color: Colors.grey,
 ),
),
const SizedBox(height: 32),
TextField(
 controller: emailController,
 decoration: InputDecoration(
  labelText: 'Email',
  border: OutlineInputBorder(
   borderRadius: BorderRadius.circular(12),
  ),
  prefixIcon: const Icon(Icons.email),
 ),
 keyboardType: TextInputType.emailAddress,
),
const SizedBox(height: 16),
TextField(
 controller: passwordController,
 decoration: InputDecoration(
  labelText: 'Password',
  border: OutlineInputBorder(
```

```
    borderRadius: BorderRadius.circular(12),
  ),
  prefixIcon: const Icon(Icons.lock),
),
obscureText: true,
),
const SizedBox(height: 24),
SizedBox(
  width: double.infinity,
  height: 50,
  child: ElevatedButton(
    onPressed: () {
      String email = emailController.text;
      String password = passwordController.text;
      print('Email: $email, Password: $password');
    },
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.blueAccent,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(12),
      ),
    ),
    child: const Text(
      'Login',
      style: TextStyle(fontSize: 18, color: Colors.white),
```

```
    ),
  ),
),
const SizedBox(height: 16),
Center(
  child: TextButton(
    onPressed: () {},
    child: const Text(
      'Forgot Password?',
      style: TextStyle(color: Colors.blueAccent),
    ),
  ),
),
],
),
),
),
),
);
}
}
```

Output :-


localhost:54506
localhost:54506
Welcome Back!
Login to your account
Email
Password
Login
Forgot Password?

Experiment 06

AIM:- Create and application with Flutter UI Components.

Source :- Main.dart

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: const RegisterScreen(),

    );

  }

}

class RegisterScreen extends StatelessWidget {

  const RegisterScreen({super.key});

  @override

  Widget build(BuildContext context) {

    final TextEditingController nameController = TextEditingController();
```

```
final TextEditingController emailController = TextEditingController();
final TextEditingController passwordController = TextEditingController();
final TextEditingController addressController = TextEditingController();
final TextEditingController mobileController = TextEditingController();
return Scaffold(
  backgroundColor: Colors.blueGrey[50],
  body: Center(
    child: SingleChildScrollView(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 24.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            const Text(
              'Create Account',
              style: TextStyle(
                fontSize: 32,
                fontWeight: FontWeight.bold,
              ),
            ),
            const SizedBox(height: 8),
            const Text(
              'Register to get started',
              style: TextStyle(
```

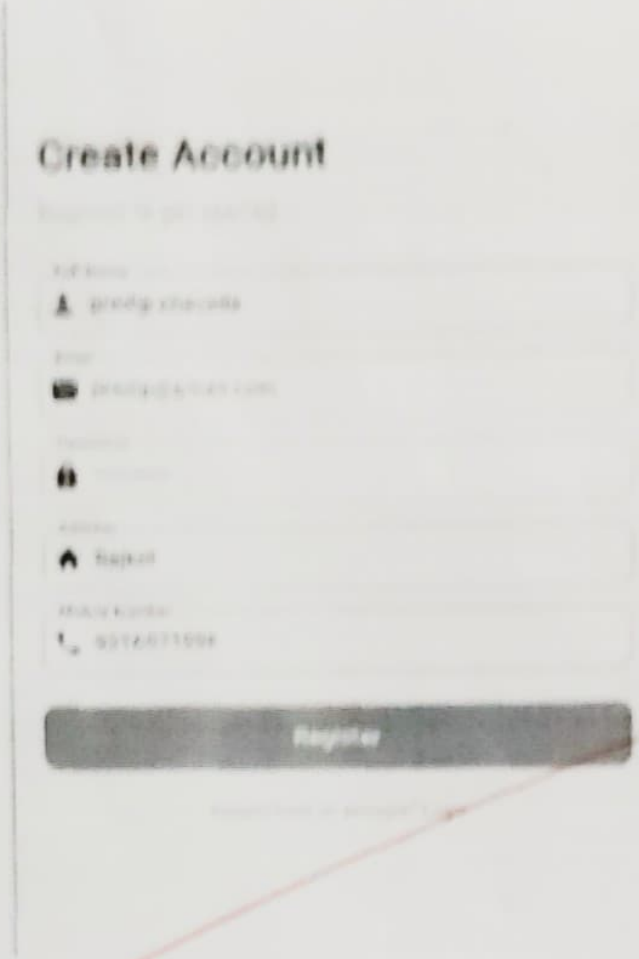
```
fontSize: 18,
color: Colors.grey,
),
),
const SizedBox(height: 32),
TextField(
controller: nameController,
decoration: InputDecoration(
labelText: 'Full Name',
border: OutlineInputBorder(
borderRadius: BorderRadius.circular(12),
),
prefixIcon: const Icon(Icons.person),
),
),
const SizedBox(height: 16),
TextField(
controller: emailController,
decoration: InputDecoration(
labelText: 'Email',
border: OutlineInputBorder(
borderRadius: BorderRadius.circular(12),
),
prefixIcon: const Icon(Icons.email),
),
```

```
 keyboardType: TextInputType.emailAddress,
),
const SizedBox(height: 16),
TextField(
 controller: passwordController,
 decoration: InputDecoration(
  labelText: 'Password',
  border: OutlineInputBorder(
   borderRadius: BorderRadius.circular(12),
  ),
  prefixIcon: const Icon(Icons.lock),
 ),
 obscureText: true,
),
const SizedBox(height: 16),
TextField(
 controller: addressController,
 decoration: InputDecoration(
  labelText: 'Address',
  border: OutlineInputBorder(
   borderRadius: BorderRadius.circular(12),
  ),
  prefixIcon: const Icon(Icons.home),
 ),
),
```

```
const SizedBox(height: 16),
TextField(
 controller: mobileController,
 decoration: InputDecoration(
  labelText: 'Mobile Number',
  border: OutlineInputBorder(
   borderRadius: BorderRadius.circular(12),
  ),
  prefixIcon: const Icon(Icons.phone),
 ),
 keyboardType: TextInputType.phone,
),
const SizedBox(height: 24),
SizedBox(
  width: double.infinity,
  height: 50,
  child: ElevatedButton(
    onPressed: () {
      // Get the text from controllers
      String name = nameController.text;
      String email = emailController.text;
      String password = passwordController.text;
      String address = addressController.text;
      String mobile = mobileController.text;
```

```
// Print the data to the console
print('Name: $name, Email: $email, Password: $password, Address: $address, Mobile: $mobile');
},
style: ElevatedButton.styleFrom(
  backgroundColor: Colors.blueAccent,
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(12),
  ),
),
  child: const Text(
    'Register',
    style: TextStyle(fontSize: 18, color: Colors.white), ),), ),
const SizedBox(height: 16),
Center(
  child: TextButton(
    onPressed: () {},
    child: const Text(
      'Already have an account? Login',
      style: TextStyle(color: Colors.blueAccent), ), ),
  ],
  ),
 ),
),
), );
}
```

Output :-




```
main.dart (Chrome)
dart:sdk/lib/_internal/js_dev_runtime/private/profile.dart 110:39          assertFailed
dart:sdk/lib/_internal/js_dev_runtime/private/profile.dart 110:39          _computeOffsetForInputs
lib/_engine/engine/pointer_binding/event_position_helper.dart 35:16        computeEventOffsetToTarget
lib/_engine/engine/pointer_binding.dart 1008:16                            [_convertEventsToPointerData]
lib/_engine/engine/pointer_binding.dart 1016:9                             <fn>
lib/_engine/engine/pointer_binding.dart 968:7                              <fn>
lib/_engine/engine/pointer_binding.dart 561:9                              loggedHandler
dart:sdk/lib/_internal/js_dev_runtime/patch/js_allow_interop_patch.dart 212:27  _callDartFunctionFast1
Name: pradip chavada, Email: pradip@gmail.com, Password: Pradip@123, Address: Rajkot, Mobile: 9516071998
```

Experiment 07

AIM:- Create and application with Navigation in Flutter.

Source :- RegisterScreen.dart

Code :-

```
const SizedBox(height: 16),
        Center(
          child: TextButton(
            onPressed: () {
              Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => const LoginScreen()),
              );
            },
            child: const Text(
              'Already have an account? Login',
              style: TextStyle(color: Colors.blueAccent),
            ),
```

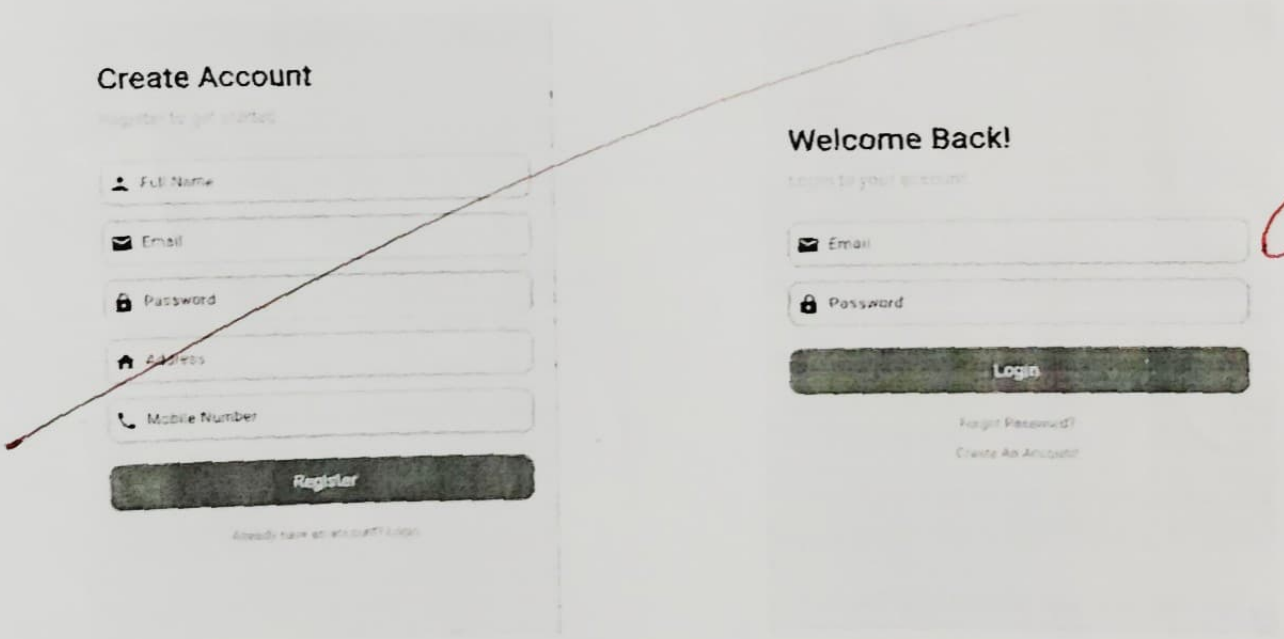
Source :- LoginScreen.dart

Code :-

```
Center(
        child: TextButton(
          onPressed: () {
```

```
Navigator.push(
  context,
  MaterialPageRoute(builder: (context) => const RegisterScreen()),
 );
},
child: const Text(
  'Create An Account!',
  style: TextStyle(color: Colors.blueAccent),
 ),
),
```

Output :-



Experiment 08

AIM:- Create and application with list view in Flutter.

Source :- Main.dart

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: Scaffold(

        appBar: AppBar(

          title: const Text(

            'List View',

            style: TextStyle(color: Colors.cyan),

          ),

          backgroundColor: Colors.black, // Optional: to contrast the cyan text
```

```
),
body: Center(
  child: ListView(
    padding: const EdgeInsets.all(8),
    children: const <Widget>[
      ListTile(
        leading: Icon(Icons.android),
        title: Text('Android'),
        subtitle: Text(
          "App Development",
          style: TextStyle(color: Colors.green),
        ),
      ),
      ListTile(
        leading: Icon(Icons.code), // Used as a replacement for JavaScript
        title: Text('JavaScript'),
        subtitle: Text(
          "Web Development",
          style: TextStyle(color: Colors.yellow),
        ),
      ),
      ListTile(
        leading: Icon(Icons.style), // Used as a replacement for CSS
        title: Text('CSS'),
        subtitle: Text(
```

```
"Web Development",
style: TextStyle(color: Colors.red),
),
),
ListTile(
leading: Icon(Icons.language), // Used as a replacement for HTML
title: Text('HTML'),
subtitle: Text(
"Web Development",
style: TextStyle(color: Colors.blue),
),
),
ListTile(
leading: Icon(Icons.apple), // This is valid for iOS/Swift
title: Text('Swift'),
subtitle: Text(
"iOS Development",
style: TextStyle(color: Colors.grey),
),
),
],
),
),
),
); }}
```

Output :-

Experiment 09

AIM:- Create and application with grid view in Flutter.

Source :- Main.dart

Code :-

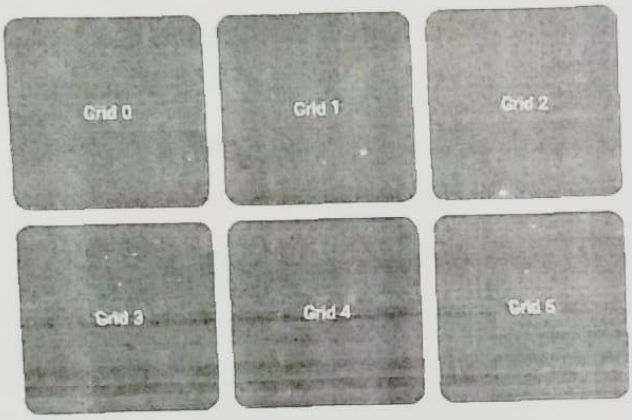
```
import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}


class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      title: 'GridView Example',

      theme: ThemeData(

        primarySwatch: Colors.blue,

      ),

      home: GridViewPage(),

    );

  }

}
```

```
class GridViewPage extends StatelessWidget {
 final List<String> items = List.generate(6, (index) => "Grid $index");


 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
    title: Text("GridView Example"),
   ),
   body: GridView.builder(
    gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
     crossAxisCount: 3, // Number of columns
     crossAxisSpacing: 4.0, // Horizontal spacing
     mainAxisSpacing: 4.0, // Vertical spacing
    ),
    itemCount: items.length,
    itemBuilder: (context, index) {
     return Card(
      color: Colors.blueAccent,
      child: Center(
       child: Text(
        items[index],
        style: TextStyle(color: Colors.white),
       ),
      ),
```

```
    ):
    ],
  ),
),
];
}
```

Output :-

GridView Example



Experiment 10

AIM:- Create and application Crud Operation with SQLite in Flutter.

Source :- Main.dart

Code :-

```
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(
// Remove the debug banner 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> {
```

```
// All journals
List<Map<String, dynamic>> _journals = [];
bool _isLoading = true;
// This function is used to fetch all data from the database void _refreshJournals() async {
final data = await SQLHelper.getItems(); setState(() {
_journals = data;
_isLoading = false;
});
}
@override
void initState() {
super.initState();
_refreshJournals(); // Loading the diary when the app starts
}
final TextEditingController _titleController = TextEditingController();
final TextEditingController _descriptionController = TextEditingController();
// This function will be triggered when the floating button is pressed
// It will also be triggered when you want to update an item void _showForm(int? id) async {
if (id != null) {
// id == null -> create new item
// id != null -> update an existing item final existingJournal =
_journals.firstWhere((element) => element['id'] == id);
_titleController.text = existingJournal['title'];
_descriptionController.text = existingJournal['description'];
}
```

```
showModalBottomSheet( context: context, elevation: 5, isScrollControlled: true, builder: (_) => Container(
padding: EdgeInsets.only( top: 15,
left: 15,
right: 15,
// this will prevent the soft keyboard from covering the text fields 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 {
// Save new journal
if (id == null) { await _addItem();
}
```

```
if (id != null) {
await _updateItem(id);
}
// Clear the text fields
_titleController.text = '';
_descriptionController.text = '';
)
],
),
));
} // Close the bottom sheet Navigator.of(context).pop();
},
child: Text(id == null ? 'Create New' : 'Update'),
// Insert a new journal to the database Future<void> _addItem() async { await SQLHelper.createItem(
_titleController.text, _descriptionController.text);
_refreshJournals();
}
// Update an existing journal Future<void> _updateItem(int id) async { await SQLHelper.updateItem(
id, _titleController.text, _descriptionController.text);
_refreshJournals();
}
// Delete an item
void _deleteItem(int id) async { await SQLHelper.deleteItem(id);
ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text('Successfully deleted a
journal!'),
```

```
});
_refreshJournals();
}
@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']),
```

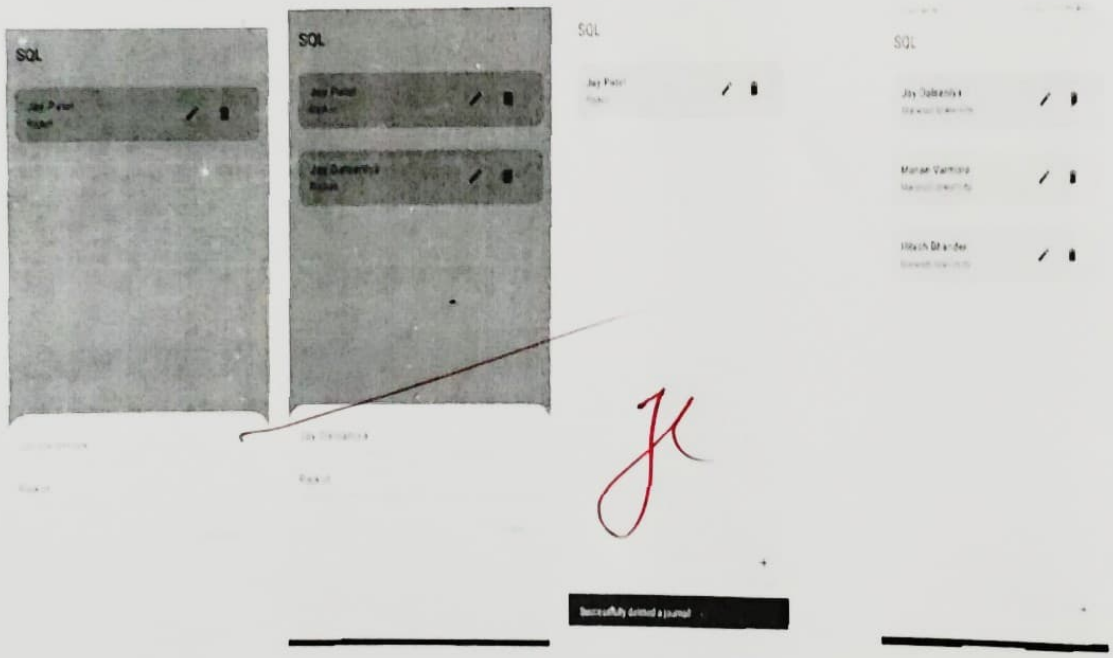
```
),],),)),),

), floatingActionButton: FloatingActionButton( child: const Icon(Icons.add),

onPressed: () => _showForm(null),

),

);

}

}
```

Source :- sqlHelper

```
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
)
"""");
}
// id: the id of a item
// title, description: name and description of your activity
// created_at: the time that the item was created. It will be automatically handled by SQLite
static Future<sql.Database> db() async { return sql.openDatabase(
'dbtech.db', version: 1,
onCreate: (sql.Database database, int version) async { await createTables(database);
},
);
}
// Create new item (journal)
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;
}
// Read all items (journals)
static Future<List<Map<String, dynamic>>> getItems() async { final db = await SQLHelper.db();
return db.query('items', orderBy: "id");
}
// Read a single item by id
// The app doesn't use this method but I put here in case you want to see it 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);
}
// Update an item by id
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;
}
// Delete
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 :-



Experiment 11

AIM:- Create an application Connecting to REST API in Flutter.

Source :- Main.dart

Code :-

```
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(),
);
}
}
```

Source :- api_service.dart

```
import 'dart:convert';
import 'package:http/http.dart' as http;
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'],
);
}
}
class ApiService {
static const String baseUrl = 'https://jsonplaceholder.typicode.com/todos/1';
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');
}
}
}
```

Source :- 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 ${index + 1}:', // Add label here style: TextStyle(
fontWeight: FontWeight.bold, fontSize: 16,
),
),
SizedBox(height: 5), Text(
snapshot.data![index].title, style: TextStyle(
fontWeight: FontWeight.bold, fontSize: 18,
),
),
SizedBox(height: 5), Text(snapshot.data![index].body),
],
),
),
);
},
);
} else if (snapshot.hasError) { return Text("${snapshot.error}"),
}
// By default, show a loading spinner. return CircularProgressIndicator(),
},
),
),
);
}
}
```

Output :-

Posts

Post 10

optio molestias id quia eum

Post 11

et ea vero quia laudantium autem

Post 12

in quibusdam tempore odit est dolorem

Post 13

dolorum ut in voluptas mollitia et saepe quo animi

Posts

Post 36

fuga nam accusamus voluptas reiciendis itaque

Post 37

provident vel ut sit ratione est

Post 38

explicabo et eos deleniti nostrum ab id repellendus

Post 39

eos dolorem iste accusantium est eaque quam

Experiment 12

AIM:- Create and application Parsing JSON data from REST API in Flutter.

Source :- Main.dart

Code :-

```
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(),
);
}
}
```

Source :- api_service.dart

Code :-

```
import 'dart:convert';
```

```
import 'package:http/http.dart' as http;
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'],
);
}
}
class ApiService {
static const String baseUrl = 'https://jsonplaceholder.typicode.com/todos/1';
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');
}
}
}
```

Source :- 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) {
```

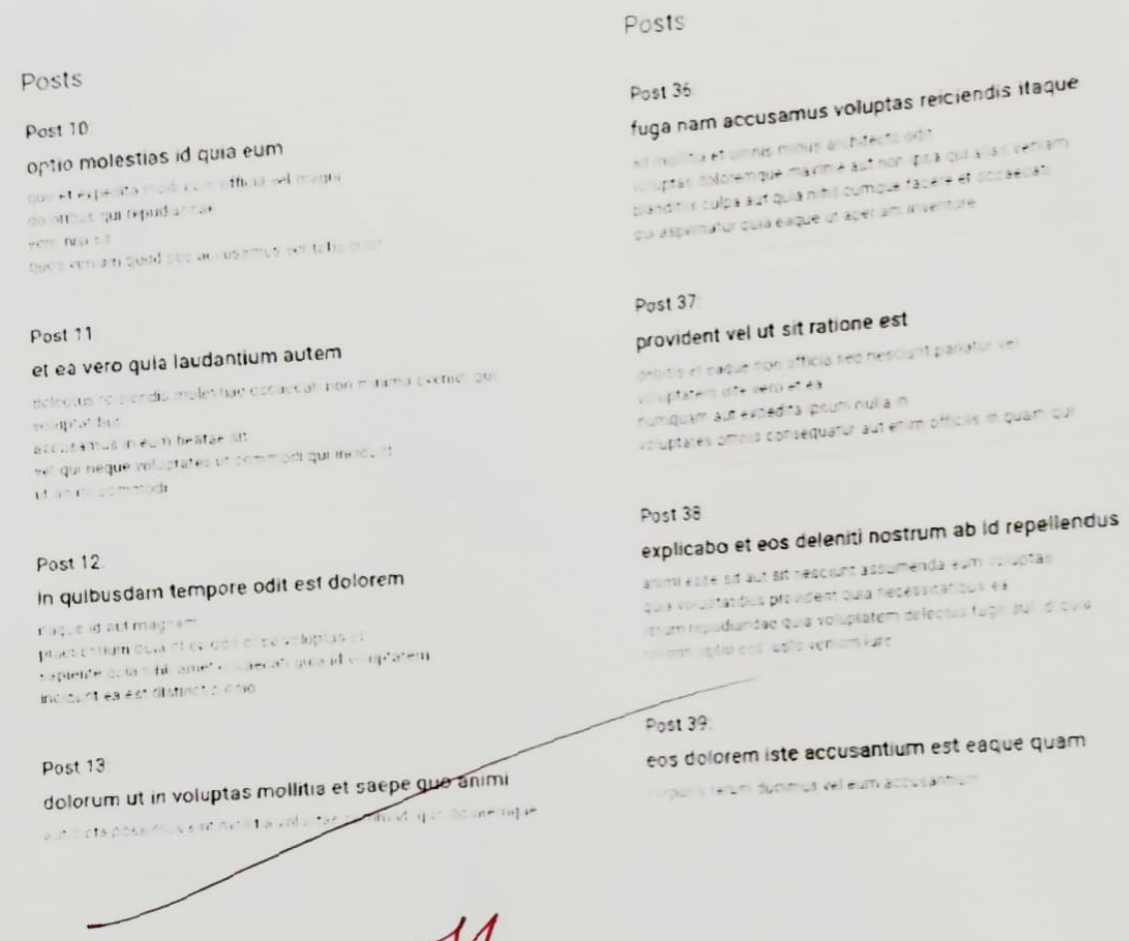
```
return Card( elevation: 3,
margin: EdgeInsets.all(10),
child: Padding(
padding: EdgeInsets.all(10),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text(
'Post ${index + 1}', // Add label here style: TextStyle(
fontWeight: FontWeight.bold, fontSize: 16,
),
),
SizedBox(height: 5), Text(
snapshot.data![index].title, style: TextStyle(
fontWeight: FontWeight.bold, fontSize: 18,
),
),
SizedBox(height: 5), Text(snapshot.data![index].body),
],
),
),
),
),
),
} else if (snapshot.hasError) { return Text("${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'],
);
}
```

}
dev_dependencies: flutter_test:
sdk: flutter http: ^0.13.3

Output :-



Experiment 13

AIM:- Create and application using Hardware Interaction in Flutter.

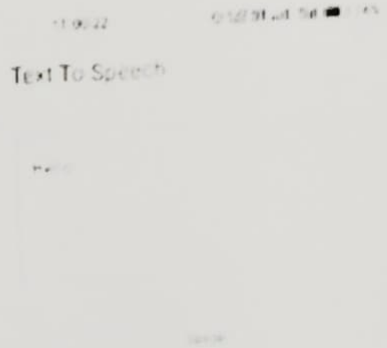
Source :- Main.dart

Code :-

```
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);
}
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 :-