DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATIONDEVELOPMENT (SE-487)

**LAB SESSION 17**

**Shared Preferences in Flutter**

**Objective**

The objective of this lab is to understand and apply the concept of shared preferences in Flutter. By the end of this lab, you should be able to save and retrieve simple data types using shared preferences and implement persistent storage in a Flutter application using shared preferences.

**Introduction**

Shared preferences in Flutter are used to store simple data (such as strings, integers, and Booleans) in a key-value pair format. This data persists across app launches and is useful for storing user preferences and settings.

**Theory**

Shared preferences provide a simple way to save data on a device. It is commonly used for storing small amounts of data, such as user preferences, settings, and application state. The shared\_preferences plugin allows you to read and write key-value pairs in persistent storage. The first step is to go to pub.dev and search for shared preferences package .retrieve the latest package and add it in the pubsec.yaml file. Run pub.get and import the package .after this you are able to access the shared preferences package functions.

import 'package:flutter/material.dart';

import 'package:shared\_preferences/shared\_preferences.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

// title: 'To-Do App',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: HomePage(),

);

}

}

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATIONDEVELOPMENT (SE-487)

class HomePage extends StatefulWidget {

@override

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

}

class \_HomePageState extends State<HomePage> {

@override

var namecontroller = TextEditingController();

var namevalue = "no value";

void initState() {

super.initState();

getvalue();

}

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Shared Preferences'),

),

body: Container(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

TextField(

controller: namecontroller,

),

ElevatedButton(

onPressed: () async {

var name = namecontroller.text.toString();

var prefs=await

SharedPreferences.getInstance();

prefs.setString("name", name);

},

child: Text('save')),

SizedBox(height: 10),

Text(namevalue)

],

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATIONDEVELOPMENT (SE-487)

)));

}

void getvalue() async {

var prefs = await SharedPreferences.getInstance();

var getname = prefs.getString("name");

//namevalue = getname!=null? getname : "no value saved"; namevalue = getname ?? "no value saved";

setState(() {});

}

}

**Conclusion**

Shared preferences provide a simple and efficient way to persist data in Flutter applications. By using shared preferences, you can store user preferences and other small pieces of data that need to be retained across app launches. This lab demonstrated how to save and retrieve data using shared preferences in Flutter.

DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATIONDEVELOPMENT (SE-487)

**Exercise:**

1. Modify the above example to save and retrieve an integer value.

Code

import 'package:flutter/material.dart';

import 'package:shared\_preferences/shared\_preferences.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      theme: ThemeData(

        primarySwatch: Colors.blue,

      ),

      home: HomePage(),

    );

  }

}

class HomePage extends StatefulWidget {

  @override

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

}

class \_HomePageState extends State<HomePage> {

  var nameController = TextEditingController();

  var intController = TextEditingController();

  var nameValue = "no value";

  var intValue = "no value";

  @override

  void initState() {

    super.initState();

    getValues();

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text('Shared Preferences'),

      ),

      body: Container(

        child: Column(

          mainAxisAlignment: MainAxisAlignment.center,

          children: [

            TextField(

              controller: nameController,

              decoration: InputDecoration(labelText: 'Enter Name'),

            ),

            SizedBox(height: 10),

            TextField(

              controller: intController,

              decoration: InputDecoration(labelText: 'Enter Integer'),

              keyboardType: TextInputType.number,

            ),

            SizedBox(height: 10),

            ElevatedButton(

              onPressed: () async {

                var name = nameController.text.toString();

                var intValue = int.parse(intController.text.toString());

                var prefs = await SharedPreferences.getInstance();

                prefs.setString("name", name);

                prefs.setInt("intValue", intValue);

              },

              child: Text('Save'),

            ),

            SizedBox(height: 10),

            Text('Name: $nameValue'),

            Text('Integer: $intValue'),

          ],

        ),

      ),

    );

  }

  void getValues() async {

    var prefs = await SharedPreferences.getInstance();

    var getName = prefs.getString("name");

    var getInt = prefs.getInt("intValue");

    setState(() {

      nameValue = getName ?? "no value saved";

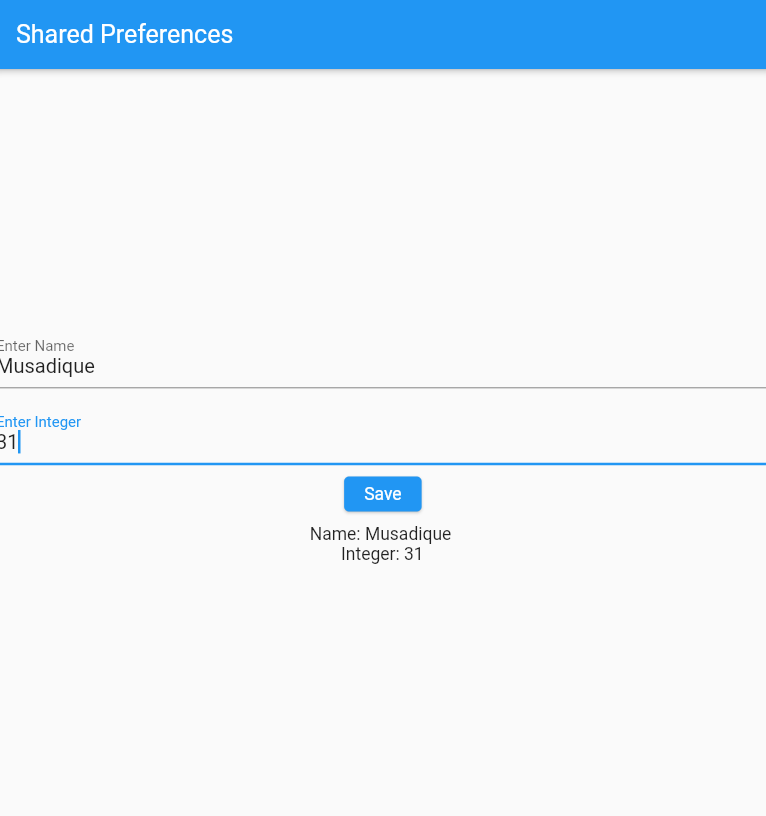
      intValue = getInt?.toString() ?? "no value saved";

    });

  }

}

Output



2. Implement a login screen where the user’s login credentials are saved using shared preferences and auto-filled the next time the app is opened.

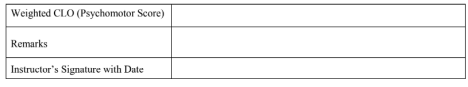
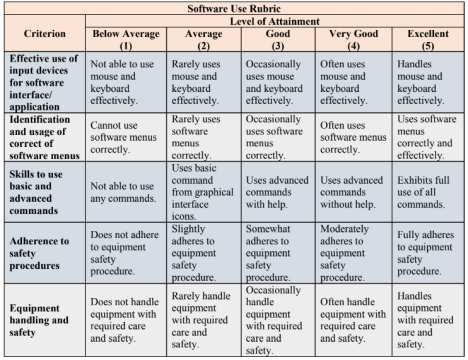
DEPARTMENT OF SOFTWARE ENGINEERING MOBILE APPLICATIONDEVELOPMENT (SE-487) 

**NED University of Engineering & Technology**

**Department of Software Engineering**

**Course Code and Title: MOBILE APPLICATION DEVELOPMENT (SE-487)**

**Laboratory Session No. 17 Date:**

****