***Mobile Application Development Lab***

***CSL-341***

***Lab Journal***



**Student Name: Muhammad Hammad.**

**Enrollment No: 01-135211-102**

**Class and Section: BS-IT 6(A).**

**Department of Computer Science**

**BAHRIA UNIVERSITY ISLAMABAD**

# Lab 11 – Persistence Storage

## Objectives:

Saving data using shared preferences and SQLite database of Flutter application using visual studio code

## Tools Used:

VS Code

### Submission Date:

## Evaluation Signatures of Lab Instructor

# Tasks

1. Create an application with the following requirements:
   1. **First Screen**:

Your application has a splash screen with a logo in the center. Your logo flashes for 2 seconds and goes to the next screen.

* 1. **Second Screen**:

Your second screen has a user registration form with a username, email, password, confirm password text field or text form field and a registration button at the end.

When a user clicks on the registration button, validate this form (every field is required) and save the data in the SQLite database.

When your form validation is successful, go to the next screen.

* 1. **Third Screen:**

Your third screen is a login page of your application with a username/email, password text field or text form field and a login button at the end.

When a user clicks on the login button, your application shows login successful message if the information entered is correct and shows an unsuccessful message if the information entered is incorrect.

* 1. **Fourth Screen:**

If credentials are correct, display all the information of current login user in the fourth screen.

**Code:**

import 'package:flutter/material.dart';

import 'package:flutter/widgets.dart';

void main() {

  runApp(const MyApp());

}

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  @override

  Widget build(BuildContext context) {

    return const MaterialApp(

      home: SplashScreen(),

    );

  }

}

class SplashScreen extends StatefulWidget {

  const SplashScreen({super.key});

  @override

  State<SplashScreen> createState() => \_SplashScreenState();

}

class \_SplashScreenState extends State<SplashScreen>

    with SingleTickerProviderStateMixin

{

  late AnimationController \_controller;

  late Animation<double> \_animation;

  @override

  void initState() {

    super

        .initState(); //super class functionalities will also be intialized in there.

    \_controller =

        AnimationController(duration: const Duration(seconds: 2), vsync: this);

    \_animation = Tween<double>(begin: 0.0, end: 1.0).animate(\_controller);

    \_controller.forward();

    \_controller.addStatusListener((status) {

      if (status == AnimationStatus.completed) {

        Navigator.pushReplacement(context,

            MaterialPageRoute(builder: (context) => SecondScreen()));

      }

    });

  }

  @override

  void dispose() {

    \_controller.dispose();

    super.dispose();

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text("First Screen"),

        backgroundColor: Colors.blueAccent,

      ),

        body: FadeTransition(

          opacity: \_animation,

          child:  Column(

            mainAxisAlignment: MainAxisAlignment.center,

            children: [

              Image.asset("assets/flutterlogo.jpeg",

                height: MediaQuery.of(context).size.height \* 1.0,

                width: 1200,

                ),

                 SecondScreen(),

            ],

          )

        ),

      );

 }

}

class SecondScreen extends StatelessWidget

{

   SecondScreen({super.key});

    final \_formkey=GlobalKey<FormState>();

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Scaffold(

         appBar: AppBar(

        title: const Text("First Screen"),

        backgroundColor: Colors.blueAccent,

      ),

        body: Center(

          child: Form(

            key: \_formkey,

            child: Padding(

              padding: const EdgeInsets.all(15.0),

              child: Column(

                crossAxisAlignment: CrossAxisAlignment.center,

                children: [

                  const Padding(

                    padding: EdgeInsets.all(20)),

                  TextFormField(

                    decoration: const InputDecoration(

                      border: OutlineInputBorder(),

                      labelText: 'Enter a username',

                    ),

                    validator: (value){

                      if(value==null ||value.isEmpty)

                      {

                        return 'Enter some text';

                      }

                      return null;

                    }

                  ),

                  const SizedBox(

                    height: 10,

                  ),

                  TextFormField(

                    decoration: const InputDecoration(

                      border: OutlineInputBorder(),

                      labelText: 'Enter email here',

                    ),

                    validator: (value){

                      if(value==null ||value.isEmpty)

                      {

                        return 'Please enter your email';

                      }

                      return null;

                    }

                  ),

                  const SizedBox(

                    height: 10,

                  ),

                  TextFormField(

                    decoration: const InputDecoration(

                      border: OutlineInputBorder(),

                      labelText: 'Enter Your Password',

                    ),

                    validator: (value){

                      if(value==null ||value.isEmpty)

                      {

                        return 'Enter your password Here';

                      }

                      return null;

                    }

                  ),

                  const SizedBox(

                    height: 10,

                  ),

                  TextFormField(

                    decoration: const InputDecoration(

                      border: OutlineInputBorder(),

                      labelText: 'Confirm your password',

                    ),

                    validator: (value){

                      if(value==null ||value.isEmpty)

                      {

                        return 'Confirm your password';

                      }

                      return null;

                    }

                  ),

                  const SizedBox(

                    height: 10,

                  ),

                  ElevatedButton(

                    onPressed:()

                    {

                      if(\_formkey.currentState!.validate())

                    {

                      Navigator.push(

                        context,

                        MaterialPageRoute(builder: (context)=> const ThirdScreen(),

                        ),

                      );

                    }

                    },

                  child: const Text("Submit"),),

                ],

              ),

            ),

            ),

          ),

        ),

      );

  }

}

class ThirdScreen extends StatefulWidget {

   const ThirdScreen({super.key});

  @override

  State<ThirdScreen> createState() => \_ThirdScreenState(); // Fix generic type parameter

}

class \_ThirdScreenState extends State<ThirdScreen>

{

  final \_username=  TextEditingController();

  final \_password=  TextEditingController();

  String message='';

  void login()

  {

    String username=\_username.text;

    String password=\_password.text;

    if(username=='Admin' && password=='123')

    {

      setState(()

      {

        message='Login Successfull';

      });

    }

    else

    {

      setState(()

      {

        message='Incorrect details enter correct ones';

      });

    }

  }

  final \_formkey=GlobalKey<FormState>();

  @override

  Widget build(BuildContext context)

  {

    return Scaffold(

      appBar: AppBar(

        title: const Text("Third Screen",

        style: TextStyle(fontWeight: FontWeight.bold),

        ),

        backgroundColor: Colors.blueAccent,

      ),

      body:  Form(

        key: \_formkey,

        child: Padding(

          padding: const EdgeInsets.all(15.0),

          child: Column(

            children: [

              TextFormField(

                    decoration: const InputDecoration(

                      border: OutlineInputBorder(),

                      labelText: 'Enter your Email/UserName',

                    ),

                    controller: \_username,

                    validator: (value){

                      if(value==null ||value.isEmpty)

                      {

                        return 'Enter Your Email or Password!';

                      }

                      return null;

                    },

                  ),

                  const SizedBox(

                    height: 10,

                  ),

                  TextFormField(

                    decoration: const InputDecoration(

                      border: OutlineInputBorder(),

                      labelText: 'Enter your Password',

                    ),

                    controller: \_password,

                    validator: (value){

                      if(value==null ||value.isEmpty)

                      {

                        return 'Enter Your Password!';

                      }

                      return null;

                    },

                  ),

                  const SizedBox(

                    height: 10,

                  ),

                  ElevatedButton(

                    onPressed: (){

                      Navigator.push(

                        context,

                        MaterialPageRoute(builder: (context)=>  Fourthscreen(\_username.text,\_password.text),

                        ),

                      );

                    },

                  child: const Text('Login',style: TextStyle(fontSize: 30, fontWeight: FontWeight.bold),),

           ),

           const SizedBox(

                    height: 10,

                  ),

                  Text(message),

            ],

            ),

        ),

          ),

    );

  }

}

class Fourthscreen extends StatelessWidget

{

  final String username;

  final String password;

  const  Fourthscreen(this.password, this.username,{super.key});

  @override

  Widget build(BuildContext context)

  {

    return Scaffold(

      appBar: AppBar(

        title: const Text('Fourth Screen',

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

        backgroundColor: Colors.blueAccent,

      ),

      body:   Center(

        child: Padding(padding: const EdgeInsets.all(30),

        child:  Column(

          crossAxisAlignment: CrossAxisAlignment.center,

          children: [

            Row(

              children: [

                Text("Username is : $password",style: const TextStyle(fontSize: 30, fontWeight: FontWeight.bold),),

                const SizedBox(

                  height: 25,

                ),

                Text("Password is  : $username", style: const TextStyle(fontSize: 30, fontWeight: FontWeight.bold),)

              ],

            )

          ],

        ),

        ),

      ),

    );

  }

}

**Output:**

**First Screen:**

A screenshot of a computer

Description automatically generated

**Second Screen:**

A screenshot of a computer

Description automatically generated

**Third Screen:**

A screenshot of a computer

Description automatically generated

**Fourth screen:**

A screenshot of a computer

Description automatically generated