

MPL Lab Exp 4

Name: Anurag Gaiwal

Roll No. 17

Div: D15A

Batch: A

Aim:

The aim of this experiment is to create an interactive form using Flutter's form widget. The form will include various form fields such as text fields, dropdown menus, and checkboxes, along with validation logic. Additionally, a button will be added to validate and submit the form data.

Theory:

In Flutter, forms are created using the `Form` widget, which provides a framework for form validation, saving form data, and handling form submission. Here's a brief overview of the key concepts used in this experiment:

Form Widget:

The `Form` widget in Flutter facilitates the creation of interactive user input forms. It manages state, validation, and submission of form data efficiently.

Email Field (TextFormField):

Utilized for email input, `TextFormField` provides validation features ensuring basic email format compliance. The `onSaved` property captures and stores entered email addresses.

Password Field (TextFormField with obscureText):

For secure password input, `TextFormField` with `obscureText` conceals characters. Validation logic ensures password strength, enhancing security and user experience.

By understanding these concepts and implementing them effectively, we can create an interactive form in Flutter that provides a smooth user experience with validation and submission functionality.

Code in main.dart:

```
// main.dart
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:jio_saavn_auth/firebase_options.dart';
import 'package:jio_saavn_auth/models/song.dart';
import 'package:jio_saavn_auth/screens/welcome_screen.dart';
import 'package:provider/provider.dart';
```

```

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  // runApp(
  //   ChangeNotifierProvider(
  //     create: (context) => LikedSongsModel(),
  //     child: MyApp(),
  //   ),
  // );
  runApp(
    MultiProvider(
      providers: [
        ChangeNotifierProvider(create: (context) => LikedSongsModel()),
        // Add other providers if needed
      ],
      child: MyApp(),
    ),
  );
}

```

```

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return ChangeNotifierProvider(
      create: (context) => LikedSongsModel(),
      child: MaterialApp(
        title: 'Spotify',
        home: WelcomePage(), // Use your main screen here
      ),
    );
  }
}

```

```
}
```

Code in signup_screen.dart:

```
import 'dart:developer';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:jio_saavn_auth/screens/phone_auth/home_screem.dart';
import 'login_screen.dart';
```

```
class SignUpScreen extends StatefulWidget {
  const SignUpScreen({Key? key}) : super(key: key);

  @override
  State<SignUpScreen> createState() => _SignUpScreenState();
}
```

```
class _SignUpScreenState extends State<SignUpScreen> {
  TextEditingController emailController = TextEditingController();
  TextEditingController passwordController = TextEditingController();
  TextEditingController cPasswordController = TextEditingController();
```

```
String errorMessage = ""; // Added to manage error message
```

```
void createAccount() async {
  String email = emailController.text.trim();
  String password = passwordController.text.trim();
  String cPassword = cPasswordController.text.trim();

  if (email == "" || password == "" || cPassword == "") {
    setState(() {
      errorMessage = "Please enter all the details!";
    });
  } else if (password != cPassword) {
```

```

setState(() {
  errorMessage = "Passwords do not match!";
});
} else {
  try {
    UserCredential userCredential = await FirebaseAuth.instance
      .createUserWithEmailAndPassword(email: email, password: password);
    if (userCredential.user != null) {
      // User created successfully, navigate to HomeScreen
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => HomeScreen()),
      );
    }
    log("User created!");
  } on FirebaseAuthException catch (ex) {
    log(ex.code.toString());
    setState(() {
      errorMessage = "Error creating account. Please try again.";
    });
  }
}
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: const Color.fromARGB(255, 255, 255, 255),
    body: SafeArea(
      child: SingleChildScrollView(
        child: Column(
          children: [
            Padding(
              padding: const EdgeInsets.all(25.0),

```

```
child: Column(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: [  
    Image.asset(  
      'assets/jiosaavn_name.png',  
      height: 150,  
      width: 200,  
    ),  
    SizedBox(height: 20),  
    Text(  
      'Welcome!, Your awesome music journey starts here!',  
      style: TextStyle(  
        fontSize: 14,  
        fontWeight: FontWeight.normal,  
        color: Color.fromARGB(198, 202, 189, 189),  
      ),  
    ),  
    SizedBox(height: 15),  
    TextField(  
      controller: emailController,  
      style:  
        TextStyle(color: const Color.fromARGB(255, 0, 0, 0)),  
      decoration: InputDecoration(  
        border: OutlineInputBorder(),  
        labelText: "Email",  
        labelStyle: TextStyle(  
          color: const Color.fromARGB(255, 0, 0, 0)),  
        focusedBorder: OutlineInputBorder(  
          borderSide: BorderSide(  
            color: Color.fromARGB(255, 0, 0, 0)),  
          ),  
        ),  
      ),  
    ),  
    SizedBox(  
      height: 15,  
    ),  
  ],  
),
```

```
    height: 15,
  ),
  TextField(
    controller: passwordController,
    style:
      TextStyle(color: const Color.fromARGB(255, 0, 0, 0)),
    decoration: InputDecoration(
      border: OutlineInputBorder(),
      labelText: "Password",
      labelStyle: TextStyle(
        color: const Color.fromARGB(255, 0, 0, 0)),
      focusedBorder: OutlineInputBorder(
        borderSide: BorderSide(
          color: Color.fromARGB(255, 0, 0, 0)),
      ),
    ),
  ),
  SizedBox(
    height: 15,
  ),
  TextField(
    controller: cPasswordController,
    style:
      TextStyle(color: const Color.fromARGB(255, 0, 0, 0)),
    decoration: InputDecoration(
      border: OutlineInputBorder(),
      labelText: "Confirm Password",
      labelStyle: TextStyle(
        color: const Color.fromARGB(255, 0, 0, 0)),
      focusedBorder: OutlineInputBorder(
        borderSide: BorderSide(
          color: Color.fromARGB(255, 0, 0, 0)),
      ),
    ),
  ),
```

```

),
  SizedBox(
    height: 15,
  ),
  CupertinoButton(
    onPressed: () {
      setState(() {
        errorMessage =
          ""; // Clear error message on button press
      });
      createAccount(); // Call createAccount to handle sign-up logic
    },
    color: Color.fromARGB(255, 40, 175, 47),
    child: Text("Sign Up"),
  ),
  Text(
    'Or',
    style: TextStyle(
      fontSize: 16,
      fontWeight: FontWeight.bold,
      color: const Color.fromARGB(255, 0, 0, 0),
    ),
  ),
  CupertinoButton(
    onPressed: () {
      Navigator.push(
        context,
        CupertinoPageRoute(
          builder: (context) => LoginScreen(),
        ),
      );
    },
    child: Text(
      "Already Have an Account?",

```

```
        style: TextStyle(
          color: Color.fromARGB(
            255, 58, 255, 68)), // Set the text color
      ),
    ),
  ],
),
),
Padding(
  padding: const EdgeInsets.all(8.0),
  child: Text(
    errorMessage,
    style: TextStyle(
      fontSize: 14,
      color: Colors.red,
    ),
    textAlign: TextAlign.center,
  ),
),
],
),
),
),
);
}
```




Welcome Back, enjoy more than 50 million songs!

Email

Password

Or

[Create an Account](#)

By continuing, you agree to our terms and privacy policy



Welcome!, Your awesome music journey starts here!

Email

Password

Confirm Password

Or

[Already Have an Account?](#)

