

Experiment No. 4

Mohit Patil -D15A-36

AIM : To create an interactive Form using form widget.

Widgets :

TextField: For user input (e.g., username, password).

FlatButton or ElevatedButton: For submitting login credentials. Text: To display static text like "Welcome" or error messages.

Column: To arrange multiple widgets vertically (e.g., input fields and buttons).

Row: For horizontally aligning widgets (e.g., forgot password link and other actions).

Padding: To add space around widgets.

Icon: To show icons in the input fields or buttons.

Container: For custom styling of sections, buttons, or inputs. Form: To handle validation of login inputs.

GestureDetector: For detecting taps (e.g., to navigate or submit).

ThemeData: To define the overall visual properties of your app (colors, fonts, etc.).

MaterialApp: To wrap your app with a theme and other configurations.

Code :

- Register page code:

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:google_sign_in/google_sign_in.dart';
import 'login.dart';

class RegisterPage extends StatefulWidget {
  const RegisterPage({super.key});

  @override
  _RegisterPageState createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;

  void _registerUser() async {
    if (_formKey.currentState!.validate()) {
      try {

```

```

        await _auth.createUserWithEmailAndPassword(
          email: _emailController.text.trim(),
          password: _passwordController.text.trim(),
        );
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text('Registration Successful!')),
        );
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(builder: (context) => const LoginPage()),
        );
      } catch (e) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text('Registration Failed: $e')),
        );
      }
    }
  }
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      color: Colors.black,
      child: Center(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Card(
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(20),
            ),
            elevation: 10,
            color: Colors.white,
            child: Padding(
              padding: const EdgeInsets.all(30.0),
              child: Form(
                key: _formKey,
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.min,
                  crossAxisAlignment: CrossAxisAlignment.stretch,
                  children: [
                    Column(
                      children: [
                        Image.asset(
                          'assets/c1.png',
                          height: 150,
                        ),
                        const SizedBox(height: 10),
                        Text(

```

```

        'CarConnect',
        style: GoogleFonts.alegreyaSans(
          fontSize: 40,
          fontWeight: FontWeight.bold,
          color: Colors.black,
        ),
      ),
    ],
  ),
  const SizedBox(height: 20),
  TextFormField(
    controller: _emailController,
    decoration: InputDecoration(
      labelText: 'Email',
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(12),
      ),
      prefixIcon:
        const Icon(Icons.email, color: Colors.black),
    ),
    validator: (value) {
      if (value == null ||
        value.isEmpty ||
        !value.contains('@')) {
        return 'Enter a valid email';
      }
      return null;
    },
  ),
  const SizedBox(height: 15),
  TextFormField(
    controller: _passwordController,
    obscureText: true,
    decoration: InputDecoration(
      labelText: 'Password',
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(12),
      ),
      prefixIcon:
        const Icon(Icons.lock, color: Colors.black),
    ),
    validator: (value) {
      if (value == null || value.length < 6) {
        return 'Password must be at least 6 characters';
      }
      return null;
    },
  ),
  const SizedBox(height: 20),

```

```

ElevatedButton(
  style: ElevatedButton.styleFrom(
    padding: const EdgeInsets.symmetric(vertical: 14),
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(12),
    ),
    backgroundColor: Colors.black,
  ),
  onPressed: _registerUser,
  child: const Text(
    'Register',
    style: TextStyle(fontSize: 20, color: Colors.white),
  ),
),
const SizedBox(height: 15),
TextButton(
  onPressed: () {
    // Simulate successful registration, then navigate to
login page
    Navigator.pushReplacementNamed(context, '/login');
  },
  child: const Text(
    'Already have an account? Login',
    style: TextStyle(color: Colors.black),
  ),
),
],
),
),
),
),
),
),
),
),
);
}
}

```

- Login page code:

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:google_sign_in/google_sign_in.dart';

class LoginPage extends StatefulWidget {
  const LoginPage({super.key});

  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final FirebaseAuth _auth = FirebaseAuth.instance;

  // Use the renderButton method for Google Sign-In on web
  void _signInWithGoogle() async {
    final GoogleSignIn googleSignIn = GoogleSignIn();

    try {
      GoogleSignInAccount? googleUser = await googleSignIn.signIn();
      if (googleUser == null) return;

      final GoogleSignInAuthentication googleAuth =
        await googleUser.authentication;

      final AuthCredential credential = GoogleAuthProvider.credential(
        accessToken: googleAuth.accessToken,
        idToken: googleAuth.idToken,
      );

      await _auth.signInWithCredential(credential);
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Google Sign-In Successful!')),
      );
    } catch (e) {
      print("Google Sign-In Error: $e");
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Google Sign-In Failed: $e')),
      );
    }
  }

  void _signInWithEmailAndPassword() async {
```

```

if (_formKey.currentState!.validate()) {
  try {
    await _auth.signInWithEmailAndPassword(
      email: _emailController.text.trim(),
      password: _passwordController.text.trim(),
    );
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Login Successful!')),
    );

    // Redirect to Home Page after login
    Navigator.pushReplacementNamed(context, '/home');
  } catch (e) {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text('Login Failed: $e')),
    );
  }
}
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      color: Colors.black,
      child: Center(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Card(
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(20),
            ),
            elevation: 10,
            color: Colors.white,
            child: Padding(
              padding: const EdgeInsets.all(30.0),
              child: Form(
                key: _formKey,
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.min,
                  crossAxisAlignment: CrossAxisAlignment.stretch,
                  children: [
                    Column(
                      children: [
                        Image.asset(
                          'assets/c1.png', // Ensure image is in assets folder
                          height: 150,
                        ),
                        const SizedBox(height: 10),

```

```

        Text(
          'CarConnect',
          style: GoogleFonts.alegreyaSans(
            fontSize: 40,
            fontWeight: FontWeight.bold,
            color: Colors.black,
          ),
        ),
      ],
    ),
    const SizedBox(height: 20),
    TextFormField(
      controller: _emailController,
      decoration: InputDecoration(
        labelText: 'Email',
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(12),
        ),
      ),
      prefixIcon:
        const Icon(Icons.email, color: Colors.black),
    ),
    validator: (value) {
      if (value == null ||
        value.isEmpty ||
        !value.contains('@')) {
        return 'Enter a valid email';
      }
      return null;
    },
  ),
  const SizedBox(height: 15),
  TextFormField(
    controller: _passwordController,
    obscureText: true,
    decoration: InputDecoration(
      labelText: 'Password',
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(12),
      ),
    ),
    prefixIcon:
      const Icon(Icons.lock, color: Colors.black),
  ),
  validator: (value) {
    if (value == null || value.length < 6) {
      return 'Password must be at least 6 characters';
    }
    return null;
  },
),

```

```

const SizedBox(height: 20),
ElevatedButton(
  style: ElevatedButton.styleFrom(
    padding: const EdgeInsets.symmetric(vertical: 14),
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(12),
    ),
    backgroundColor: Colors.black,
  ),
  onPressed: _signInWithEmailAndPassword,
  child: const Text(
    'Login',
    style: TextStyle(fontSize: 20, color: Colors.white),
  ),
),
const SizedBox(height: 15),
ElevatedButton(
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.white,
    padding: const EdgeInsets.symmetric(vertical: 14),
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(12),
      side: const BorderSide(color: Colors.black),
    ),
  ),
  onPressed: _signInWithGoogle,
  child: const Text(
    'Sign in with Google',
    style: TextStyle(fontSize: 20, color: Colors.black),
  ),
),
),
],
),
),
),
),
),
),
),
),
);
}
}

```


- **Main.dart file code:**

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:proj1/screens/login.dart';
import 'package:proj1/screens/RegisterPage.dart';
import 'package:proj1/screens/home.dart'; // Import Home Page
import 'firebase_options.dart';

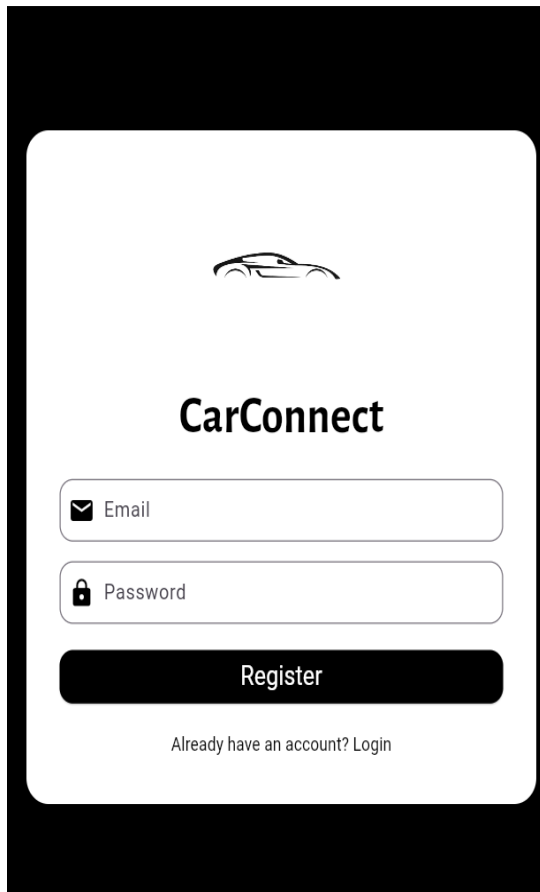
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});


  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      initialRoute: '/register',
      routes: {
        '/register': (context) => const RegisterPage(),
        '/login': (context) => const LoginPage(),
        '/home': (context) => const HomePage(), // Add Home Page Route
      },
    );
  }
}
```

Screen Shots :


1.Register page




The register page features a white rounded rectangle on a black background. At the top is a car icon. Below it is the text 'CarConnect'. There are two input fields: 'Email' with an envelope icon and 'Password' with a lock icon. A black 'Register' button is below the fields. At the bottom, a link says 'Already have an account? Login'.



CarConnect

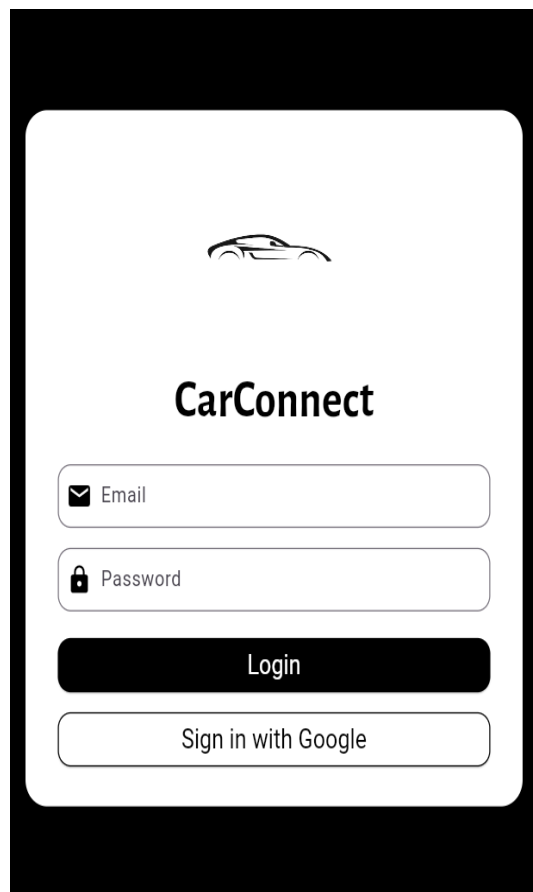
 Email

 Password


Register

[Already have an account? Login](#)


2.Login page




The login page features a white rounded rectangle on a black background. At the top is a car icon. Below it is the text 'CarConnect'. There are two input fields: 'Email' with an envelope icon and 'Password' with a lock icon. A black 'Login' button is below the fields. At the bottom, a link says 'Sign in with Google'.



CarConnect

 Email

 Password

Login

[Sign in with Google](#)