

## MPL EXP 5 Screenshots

### 1. Navigation

Login Page	Signup page
<pre>import 'package:flutter/material.dart'; import 'package:firebase_auth/firebase_auth.dart'; import 'home_page.dart'; import 'signup_page.dart';  class LoginPage extends StatefulWidget {   const LoginPage({super.key});    @override   _LoginPageState createState() =&gt; _LoginPageState(); }  class _LoginPageState extends State&lt;LoginPage&gt; {   final FirebaseAuth _auth = FirebaseAuth.instance;   final TextEditingController emailController =     TextEditingController();   final TextEditingController passwordController =     TextEditingController();   bool isLoading = false;    Future&lt;void&gt; _login() async {     final email = emailController.text.trim();     final password = passwordController.text.trim();      if (email.isEmpty    password.isEmpty) {       _showSnackBar('Please enter both email and password');       return;     }      setState(() =&gt; isLoading = true);      try {       await _auth.signInWithEmailAndPassword(email: email, password: password);       _showSnackBar('Login Successful');        // Navigate to HomePage after successful login       Navigator.pushReplacement(         context,         MaterialPageRoute(builder: (context) =&gt; const HomePage()),       );     } on FirebaseAuthException catch (e) {       _handleFirebaseError(e);     } finally {       setState(() =&gt; isLoading = false);     }   } }</pre>	<pre>import 'package:flutter/material.dart'; import 'package:firebase_auth/firebase_auth.dart'; import 'login_page.dart';  class SignupPage extends StatefulWidget {   const SignupPage({super.key});    @override   _SignupPageState createState() =&gt; _SignupPageState(); }  class _SignupPageState extends State&lt;SignupPage&gt; {   final FirebaseAuth _auth =     FirebaseAuth.instance;   final TextEditingController emailController =     TextEditingController();   final TextEditingController passwordController = TextEditingController();   final TextEditingController confirmPasswordController =     TextEditingController();   bool isLoading = false;    Future&lt;void&gt; _signup() async {     final email = emailController.text.trim();     final password = passwordController.text.trim();     final confirmPassword = confirmPasswordController.text.trim();      if (email.isEmpty    password.isEmpty    confirmPassword.isEmpty) {       _showSnackBar('Please fill all fields');       return;     }      if (password != confirmPassword) {       _showSnackBar('Passwords do not match');       return;     }      setState(() =&gt; isLoading = true);      try {       await</pre>

```

}

void _handleFirebaseError(FirebaseAuthException e)
{
  String message = 'Login failed';
  if (e.code == 'user-not-found') {
    message = 'No account found for this email';
  } else if (e.code == 'wrong-password') {
    message = 'Incorrect password';
  } else if (e.code == 'invalid-email') {
    message = 'Invalid email format';
  }
  _showSnackBar(message);
}

void _showSnackBar(String message) {
  ScaffoldMessenger.of(context).showSnackBar(SnackBar(
    content: Text(message)));
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SingleChildScrollView(
      child: Container(
        padding: const EdgeInsets.all(16.0),
        height: MediaQuery.of(context).size.height,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.blue.shade200,
Colors.purple.shade200],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
      child: Column(
        mainAxisAlignment:
MainAxisAlignment.center,
        children: [
          // App Logo or Icon
          const Icon(
            Icons.lock_person,
            size: 100,
            color: Colors.white,
          ),
          const SizedBox(height: 20),
          const Text(
            'Welcome Back!',
            style: TextStyle(
              fontSize: 28,
              fontWeight: FontWeight.bold,
              color: Colors.white,
            ),
          ),
          const SizedBox(height: 10),

```

```

_auth.createUserWithEmailAndPassword(email:
email, password: password);
_showSnackBar('Signup Successful');

// Navigate to LoginPage after successful
signup
Navigator.pushReplacement(
  context,
  MaterialPageRoute(builder: (context) =>
const LoginPage()),
);
} on FirebaseAuthException catch (e) {
  _handleFirebaseError(e);
} finally {
  setState(() => isLoading = false);
}
}

void _handleFirebaseError(FirebaseAuthException
e) {
  String message = 'Signup failed';
  if (e.code == 'email-already-in-use') {
    message = 'Email is already in use';
  } else if (e.code == 'weak-password') {
    message = 'Password is too weak';
  } else if (e.code == 'invalid-email') {
    message = 'Invalid email format';
  }
  _showSnackBar(message);
}

void _showSnackBar(String message) {
  ScaffoldMessenger.of(context).showSnackBar(Snack
Bar(content: Text(message)));
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SingleChildScrollView(
      child: Container(
        padding: const EdgeInsets.all(16.0),
        height:
MediaQuery.of(context).size.height,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.blue.shade200,
Colors.purple.shade200],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
      child: Column(
        mainAxisAlignment:
MainAxisAlignment.center,

```

```

const Text(
  'Login to continue',
  style: TextStyle(
    fontSize: 16,
    color: Colors.white70,
  ),
),
const SizedBox(height: 40),
// Email Field
TextField(
  controller: emailController,
  decoration: InputDecoration(
    labelText: 'Email',
    labelStyle: const TextStyle(color:
Colors.white),
    filled: true,
    fillColor:
Colors.white.withOpacity(0.1),
    border: OutlineInputBorder(
      borderRadius:
BorderRadius.circular(10),
      borderSide: BorderSide.none,
    ),
    prefixIcon: const Icon(Icons.email,
color: Colors.white),
  ),
  style: const TextStyle(color:
Colors.white),
  keyboardType:
TextInputType.emailAddress,
),
const SizedBox(height: 20),
// Password Field
TextField(
  controller: passwordController,
  obscureText: true,
  decoration: InputDecoration(
    labelText: 'Password',
    labelStyle: const TextStyle(color:
Colors.white),
    filled: true,
    fillColor:
Colors.white.withOpacity(0.1),
    border: OutlineInputBorder(
      borderRadius:
BorderRadius.circular(10),
      borderSide: BorderSide.none,
    ),
    prefixIcon: const Icon(Icons.lock,
color: Colors.white),
  ),
  style: const TextStyle(color:
Colors.white),
),
const SizedBox(height: 30),
// Login Button

```

```

children: [
  // App Logo or Icon
  const Icon(
    Icons.person_add,
    size: 100,
    color: Colors.white,
  ),
  const SizedBox(height: 20),
  const Text(
    'Create an Account',
    style: TextStyle(
      fontSize: 28,
      fontWeight: FontWeight.bold,
      color: Colors.white,
    ),
  ),
  const SizedBox(height: 10),
  const Text(
    'Sign up to get started',
    style: TextStyle(
      fontSize: 16,
      color: Colors.white70,
    ),
  ),
  const SizedBox(height: 40),
  // Email Field
  TextField(
    controller: emailController,
    decoration: InputDecoration(
      labelText: 'Email',
      labelStyle: const
TextStyle(color: Colors.white),
      filled: true,
      fillColor:
Colors.white.withOpacity(0.1),
      border: OutlineInputBorder(
        borderRadius:
BorderRadius.circular(10),
        borderSide: BorderSide.none,
      ),
      prefixIcon: const
Icon(Icons.email, color: Colors.white),
    ),
    style: const TextStyle(color:
Colors.white),
    keyboardType:
TextInputType.emailAddress,
  ),
  const SizedBox(height: 20),
  // Password Field
  TextField(
    controller: passwordController,
    obscureText: true,
    decoration: InputDecoration(
      labelText: 'Password',
      labelStyle: const

```

```

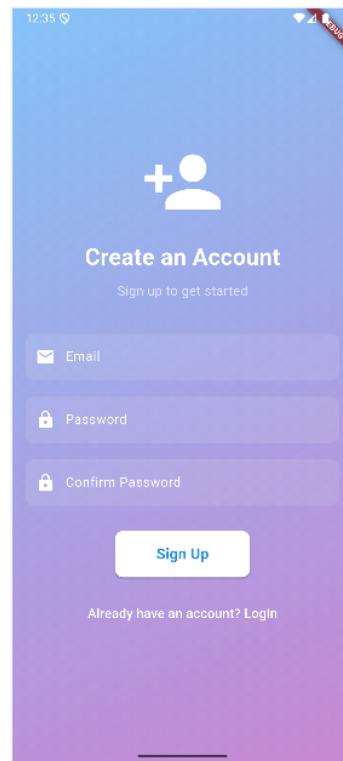
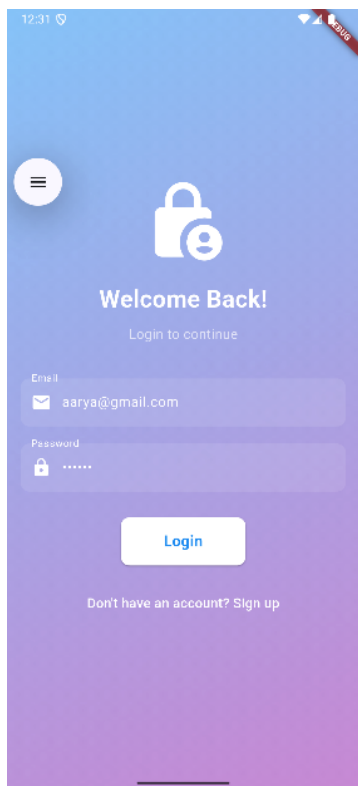
        isLoading
        ? const
CircularProgressIndicator(color: Colors.white)
        : ElevatedButton(
onPressed: _login,
style: ElevatedButton.styleFrom(
    backgroundColor: Colors.white,
padding: const
EdgeInsets.symmetric(
    horizontal: 50, vertical: 15),
shape: RoundedRectangleBorder(
    borderRadius:
BorderRadius.circular(10),
),
),
child: const Text(
    'Login',
style: TextStyle(
    fontSize: 18,
color: Colors.blue,
fontWeight: FontWeight.bold,
),
),
),
const SizedBox(height: 20),
// Signup Link
TextButton(
onPressed: () {
    Navigator.push(
        context,
        MaterialPageRoute(builder:
(context) => const SignupPage()),
    );
},
child: const Text(
    'Don\'t have an account? Sign up',
style: TextStyle(
    color: Colors.white,
fontSize: 16,
),
),
),
),
),
),
),
),
);
}
}

```

```

TextStyle(color: Colors.white),
    filled: true,
    fillColor:
Colors.white.withOpacity(0.1),
    border: OutlineInputBorder(
        borderRadius:
BorderRadius.circular(10),
        borderSide: BorderSide.none,
    ),
    prefixIcon: const
Icon(Icons.lock, color: Colors.white),
),
style: const TextStyle(color:
Colors.white),
),
const SizedBox(height: 20),
// Confirm Password Field
TextField(
    controller:
confirmPasswordController,
    obscureText: true,
    decoration: InputDecoration(
        labelText: 'Confirm Password',
        labelStyle: const
TextStyle(color: Colors.white),
        filled: true,
        fillColor:
Colors.white.withOpacity(0.1),
        border: OutlineInputBorder(
            borderRadius:
BorderRadius.circular(10),
            borderSide: BorderSide.none,
        ),
        prefixIcon: const
Icon(Icons.lock, color: Colors.white),
    ),
    style: const TextStyle(color:
Colors.white),
),
const SizedBox(height: 30),
// Signup Button
isLoading
? const
CircularProgressIndicator(color: Colors.white)
    : ElevatedButton(
onPressed: _signup,
style: ElevatedButton.styleFrom(
    backgroundColor: Colors.white,
padding: const
EdgeInsets.symmetric(
    horizontal: 50, vertical:
15),
shape: RoundedRectangleBorder(
    borderRadius:
BorderRadius.circular(10),
)

```



## 2. Gestures

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

void main() {
  runApp(const HomePage());
}

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

  @override
  State<HomePage> createState() => _MyAppState();
}

class _MyAppState extends State<HomePage> {
  late GoogleMapController mapController;

  final LatLng _center = const LatLng(45.521563, -122.677433);

  void _onMapCreated(GoogleMapController controller) {
    mapController = controller;
  }
}
```

```

}

@override
Widget build(BuildContext context) {
  return MaterialApp(
    theme: ThemeData(
      useMaterial3: true,
      colorSchemeSeed: Colors.green[700],
    ),
    home: Scaffold(
      appBar: AppBar(
        title: const Text('Maps Sample App'),
        elevation: 2,
      ),
      body: GoogleMap(
        onMapCreated: _onMapCreated,
        initialCameraPosition: CameraPosition(
          target: _center,
          zoom: 11.0,
        ),
      ),
    ),
  );
}

```

