

## MPL Practical 04

**Aim:** To create an interactive Form using a form widget.

**Theory:**

Forms are a fundamental part of mobile applications, enabling users to input and submit data. In Flutter, the Form widget provides a structured way to handle user inputs along with validation. The TextFormField widget is commonly used within a form to accept inputs such as email and password. Validation ensures that user credentials follow the required format before submission.

**Implementation in MedLink App**

In this experiment, we implemented a Login and Signup form for the MedLink app. The app:

- Redirects users to the Login page on first launch.
- Validates user credentials using the email\_validator package.
- Ensures passwords contain at least one uppercase letter, one lowercase letter, one number, and one special character.
- Allows users to switch between Login and Signup screens.
- Navigates to the Home screen after successful login.

---

**lib/screens/login\_screen.dart**

```
import 'package:flutter/material.dart';
import 'package:email_validator/email_validator.dart';
import 'signup_screen.dart';
import 'home_screen.dart';

class LoginScreen extends StatefulWidget {
  @override
  _LoginScreenState createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  void _login() {
    if (_formKey.currentState!.validate()) {
      // Simulating authentication (you can replace it with actual authentication logic)
      Navigator.pushReplacement(
        context,
```

```
MaterialPageRoute(builder: (context) => HomeScreen()),  
  );  
}  
}
```

@override

```
Widget build(BuildContext context) {  
  return Scaffold(  
    backgroundColor: Theme.of(context).colorScheme.background,  
    body: Center(  
      child: Padding(  
        padding: EdgeInsets.all(20),  
        child: Form(  
          key: _formKey,  
          child: Column(  
            mainAxisAlignment: MainAxisAlignment.center,  
            children: [  
              Text(  
                'MedLink Login',  
                style: TextStyle(  
                  fontFamily: 'Poppins',  
                  fontSize: 24,  
                  fontWeight: FontWeight.bold,  
                  color: Theme.of(context).colorScheme.primary,  
                ),  
              ),  
              SizedBox(height: 20),  
              TextFormField(  
                controller: emailController,  
                keyboardType: TextInputType.emailAddress,  
                decoration: InputDecoration(labelText: 'Email'),  
                validator: (value) {  
                  if (value == null || value.isEmpty) {  
                    return 'Enter your email';  
                  } else if (!EmailValidator.validate(value)) {  
                    return 'Enter a valid email';  
                  }  
                },  
                return null;  
              ),  
            ],  
          ),  
        ),  
      ),  
    ),  
  );  
}
```

```

        SizedBox(height: 10),
        TextFormField(
          controller: passwordController,
          obscureText: true,
          decoration: InputDecoration(labelText: 'Password'),
          validator: (value) {
            if (value == null || value.isEmpty) {
              return 'Enter your password';
            } else if
(!RegExp(r'^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[!@#\$&*~]).{6,}$').hasMatch(value)) {
              return 'Password must have 1 uppercase, 1 lowercase, 1 number, and 1 special
character';
            }
            return null;
          },
        ),
        SizedBox(height: 20),
        ElevatedButton(
          onPressed: _login,
          child: Text('Login'),
        ),
        SizedBox(height: 10),
        TextButton(
          onPressed: () {
            Navigator.push(context, MaterialPageRoute(builder: (context) => SignupScreen()));
          },
          child: Text('Don\'t have an account? Sign up'),
        ),
      ],
    ),
  ),
),
),
),
);
}
}

```

### **lib/screens/signup\_screen.dart**

```

import 'package:flutter/material.dart';
import 'package:email_validator/email_validator.dart';

```

```
import 'login_screen.dart';

class SignupScreen extends StatefulWidget {
  @override
  _SignupScreenState createState() => _SignupScreenState();
}

class _SignupScreenState extends State<SignupScreen> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  final TextEditingController confirmPasswordController = TextEditingController();

  void _signup() {
    if (_formKey.currentState!.validate()) {
      // Simulating user registration (replace with actual logic)
      Navigator.pushReplacement(context, MaterialPageRoute(builder: (context) =>
LoginScreen()));
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Theme.of(context).colorScheme.background,
      body: Center(
        child: Padding(
          padding: EdgeInsets.all(20),
          child: Form(
            key: _formKey,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Text(
                  'Create Account',
                  style: TextStyle(
                    fontFamily: 'Poppins',
                    fontSize: 24,
                    fontWeight: FontWeight.bold,
                    color: Theme.of(context).colorScheme.primary,
```

```

    ),
    ),
    SizedBox(height: 20),
    TextFormField(
      controller: emailController,
      keyboardType: TextInputType.emailAddress,
      decoration: InputDecoration(labelText: 'Email'),
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Enter your email';
        } else if (!EmailValidator.validate(value)) {
          return 'Enter a valid email';
        }
        return null;
      },
    ),
    SizedBox(height: 10),
    TextFormField(
      controller: passwordController,
      obscureText: true,
      decoration: InputDecoration(labelText: 'Password'),
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Enter a password';
        } else if
(!RegExp(r'^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[!@#\$&*~]).{6,}$').hasMatch(value)) {
          return 'Password must have 1 uppercase, 1 lowercase, 1 number, and 1 special
character';
        }
        return null;
      },
    ),
    SizedBox(height: 10),
    TextFormField(
      controller: confirmPasswordController,
      obscureText: true,
      decoration: InputDecoration(labelText: 'Confirm Password'),
      validator: (value) {
        if (value != passwordController.text) {
          return 'Passwords do not match';
        }
      },
    ),

```

```
    }
    return null;
  },
),
 SizedBox(height: 20),
 ElevatedButton(
   onPressed: _signup,
   child: Text('Sign Up'),
 ),
 SizedBox(height: 10),
 TextButton(
   onPressed: () {
     Navigator.push(context, MaterialPageRoute(builder: (context) => LoginScreen()));
   },
   child: Text('Already have an account? Login'),
 ),
 ],
),
),
),
),
);
}
```

## MedLink Login

Email

HARSH

Enter a valid email

Password

...

Password must have 1 uppercase, 1 lowercase, 1 number, and 1 special c...

Login

[Don't have an account? Sign up](#)

**Create Account**

Email  
HARSH

Enter a valid email

Password  
...

Password must have 1 uppercase, 1 lowercase, 1 number, and 1 special c...

Confirm Password  
.....

Passwords do not match

[Sign Up](#)

[Already have an account? Login](#)

## Conclusion

In this experiment, we successfully implemented a Login and Signup form with proper validation using the Form widget in Flutter. Initially, we faced issues with email validation and password constraints, but we resolved them by integrating the email\_validator package and using regular expressions for password validation, ensuring a smooth authentication process.