

Experiment 4

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

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();
  bool _isLoading = false;
  String? _emailError;
  String? _passwordError;

  bool _validateEmail(String email) {
    if (email.isEmpty) {
      setState(() => _emailError = 'Email is required');
      return false;
    }
    if (!email.contains('@') || !email.contains('.')) {
      setState(() => _emailError = 'Please enter a valid email address');
      return false;
    }
    setState(() => _emailError = null);
    return true;
  }

  bool _validatePassword(String password) {
    if (password.isEmpty) {
      setState(() => _passwordError = 'Password is required');
      return false;
    }
    if (password.length < 6) {
      setState(() => _passwordError = 'Password must be at least 6
characters');
      return false;
    }
    setState(() => _passwordError = null);
  }
}
```

```

        return true;
    }

    void _loginUser() {
        final email = _emailController.text.trim();
        final password = _passwordController.text;

        // Validate both fields
        final isEmailValid = _validateEmail(email);
        final isPasswordValid = _validatePassword(password);

        if (!isEmailValid || !isPasswordValid) {
            return;
        }

        setState(() {
            _isLoading = true;
        });

        Future.delayed(Duration(seconds: 2), () {
            setState(() {
                _isLoading = false;
            });
            Navigator.pushNamedAndRemoveUntil(
                context,
                '/home',
                (route) => false,
            );
        });
    }
}

```

```

@override
Widget build(BuildContext context) {
    return Scaffold(
        body: SingleChildScrollView(
            child: Container(
                height: MediaQuery.of(context).size.height,
                decoration: BoxDecoration(
                    gradient: LinearGradient(
                        begin: Alignment.topCenter,

```

```

        end: Alignment.bottomCenter,
        colors: [Colors.blue[400]!, Colors.blue[800]!],
    ),
),
child: SafeArea(
  child: Padding(
    padding: EdgeInsets.all(24.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        IconButton(
          icon: Icon(Icons.arrow_back, color: Colors.white),
          onPressed: () => Navigator.pop(context),
        ),
        SizedBox(height: 32),
        Text(
          'Welcome Back',
          style: TextStyle(
            fontSize: 32,
            fontWeight: FontWeight.bold,
            color: Colors.white,
          ),
        ),
        SizedBox(height: 8),
        Text(
          'Sign in to continue',
          style: TextStyle(
            fontSize: 16,
            color: Colors.white.withOpacity(0.9),
          ),
        ),
        SizedBox(height: 48),
        Container(
          padding: EdgeInsets.all(24),
          decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: BorderRadius.circular(16),
            boxShadow: [
              BoxShadow(
                color: Colors.black.withOpacity(0.1),

```

```

        blurRadius: 10,
        offset: Offset(0, 5),
      ),
    ],
  ),
  child: Column(
    children: [
      TextField(
        controller: _emailController,
        decoration: InputDecoration(
          labelText: 'Email',
          hintText: 'Enter your email',
          prefixIcon: Icon(Icons.email),
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(12),
          ),
          filled: true,
          fillColor: Colors.grey[100],
          errorText: _emailError,
          // Show red color when there's an error
          errorBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(12),
            borderSide: BorderSide(color: Colors.red),
          ),
        ),
        keyboardType: TextInputType.emailAddress,
        onChanged: (value) {
          // Clear error when user starts typing
          if (_emailError != null) {
            setState(() => _emailError = null);
          }
        },
      ),
      SizedBox(height: 16),
      TextField(
        controller: _passwordController,
        decoration: InputDecoration(
          labelText: 'Password',
          hintText: 'Enter your password',
          prefixIcon: Icon(Icons.lock),

```

```
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(12),
        ),
        filled: true,
        fillColor: Colors.grey[100],
        errorText: _passwordError,
        // Show red color when there's an error
        borderSide: BorderSide(color: Colors.red),
      ),
    ),
    obscureText: true,
    onChanged: (value) {
      // Clear error when user starts typing
      if (_passwordError != null) {
        setState(() => _passwordError = null);
      }
    },
  ),
),
),
),
],
),
),
),
),
),
),
);
}


@override
void dispose() {
  _emailController.dispose();
  _passwordController.dispose();
  super.dispose();
}
```



Welcome Back

Sign in to continue

 Email

 Password

Sign In

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