

## **EXPERIMENT NO: - 04**

**Name:-** Laksh Sodhai

**Class:-** D15A

**Roll:No: -** 57

**AIM: -** To create an interactive Form using form widget.

---

### **Theory: -**

A form in Flutter is a structured container that collects user input through various fields like text fields, dropdowns, checkboxes, and buttons. It plays a crucial role in applications that require user data entry, such as login pages, registration forms, and feedback submissions. Flutter provides the Form widget, which works alongside TextFormField and other input elements to manage validation, state handling, and error messages efficiently. By using form validation techniques, developers can ensure data accuracy and enhance user experience.

When you create a form, it is necessary to provide the GlobalKey. This key uniquely identifies the form and allows you to do any validation in the form fields. The form widget uses child widget TextFormField to provide the users to enter the text field. This widget renders a material design text field and also allows us to display validation errors when they occur.

### **Creation of a Form**

- While creating a form in Flutter, the **Form widget** is essential as it acts as a container for grouping multiple form fields and managing validation.
- A **GlobalKey<FormState>** is required to uniquely identify the form and enable validation or data retrieval from the form fields.
- The **TextFormField widget** is used to provide input fields where users can enter data such as names, phone numbers, or email addresses.
- To enhance the appearance and usability of input fields, **InputDecoration** is used, allowing customization of labels, icons, borders, and hint text.
- Validation plays a crucial role in forms, and the **validator property** within **TextFormField** ensures user input meets specific criteria before submission.

- Different types of input require appropriate **keyboard types**, such as `TextInputType.number` for numeric fields or `TextInputType.emailAddress` for email fields.
- Proper **state management** is needed to store and retrieve user input, ensuring the form data is processed correctly.
- A **submit button** is necessary to trigger form validation and submit the collected data for further processing.

### Some Properties of Form Widget

- **key:** A `GlobalKey` that uniquely identifies the `Form`. You can use this key to interact with the form, such as validating, resetting, or saving its state.
- **child:** The child widget that contains the form fields. Typically, this is a `Column`, `ListView`, or another widget that allows you to arrange the form fields vertically.
- **autovalidateMode:** An enum that specifies when the form should automatically validate its fields.

### Some Methods of Form Widget

- **validate():** This method is used to trigger the validation of all the form fields within the `Form`. It returns `true` if all fields are valid, otherwise `false`. You can use it to check the overall validity of the form before submitting it.
- **save():** This method is used to save the current values of all form fields. It invokes the `onSaved` callback for each field. Typically, this method is called after validation succeeds.
- **reset():** Resets the form to its initial state, clearing any user-entered data.
- **currentState:** A getter that returns the current `FormState` associated with the `Form`.

**Code:**

**login\_screen.dart**

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

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

  @override
  State<LoginScreen> createState() =>
    _LoginScreenState();
}

class _LoginScreenState extends
State<LoginScreen> {
  final _formKey = GlobalKey<FormState>();
  final _emailController =
TextEditingController();
  final _passwordController =
TextEditingController();
  bool _isPasswordVisible = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: const BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [
              Color(0xFF1A237E),
              Color(0xFF0D47A1),
              Color(0xFF01579B),
            ],
          ),
        ),
        child: Stack(
          children: [
            Positioned(
              top: -100,
              right: -100,
              child: Container(
                width: 200,
                height: 200,
                decoration: BoxDecoration(
                  shape: BoxShape.circle,
                  color: Colors.white.withOpacity(0.1),
```

```
                ),
              ),
            ),
            Positioned(
              bottom: -50,
              left: -50,
              child: Container(
                width: 150,
                height: 150,
                decoration: BoxDecoration(
                  shape: BoxShape.circle,
                  color: Colors.white.withOpacity(0.1),
                ),
              ),
            ),
          ],
        ),
        SingleChildScrollView(
          child: Padding(
            padding: const EdgeInsets.all(20.0),
            child: Form(
              key: _formKey,
              child: Column(
                mainAxisAlignment:
MainAxisAlignment.center,
                children: [
                  const SizedBox(height: 80),
                  Container(
                    padding: const EdgeInsets.all(20),
                    decoration: BoxDecoration(
                      borderRadius:
BorderRadius.circular(30),
                      boxShadow: [
                        BoxShadow(
                          color:
Colors.blue.withOpacity(0.2),
                          blurRadius: 20,
                          spreadRadius: 5,
                        ),
                      ],
                    ),
                    child: Column(
                      children: [
                        Container(
                          padding: const
EdgeInsets.all(16),
                          decoration: BoxDecoration(
                            color: Colors.white,
                            shape: BoxShape.circle,
                            boxShadow: [
                              BoxShadow(
                                color:
Colors.blue.withOpacity(0.5),
                                blurRadius: 20,
```

```

        spreadRadius: 5,
      ),
    ],
  ),
  child: const Icon(
    Icons.shield,
    size: 80,
    color: Colors.blue,
  ),
),
const SizedBox(height: 20),
ShaderMask(
  shaderCallback: (bounds) =>
const LinearGradient(
  colors: [Colors.white,
Colors.lightBlueAccent],
).createShader(bounds),
child: const Text(
'SecureShield VPN',
style: TextStyle(
  fontSize: 32,
fontWeight:
FontWeight.bold,
color: Colors.white,
),
),
),
const Text(
'Secure • Fast • Reliable',
style: TextStyle(
  color: Colors.white70,
  fontSize: 16,
),
),
),
const SizedBox(height: 40),
Container(
  decoration: BoxDecoration(
    color: Colors.white10,
borderRadius:
BorderRadius.circular(15),
border: Border.all(color:
Colors.white24),
),
child: TextFormField(
  controller: _emailController,
  style: const TextStyle(color:
Colors.white),
  decoration: InputDecoration(
    labelText: 'Email',

```

```

    labelStyle: const
TextStyle(color: Colors.white70),
border: OutlineInputBorder(
  borderRadius:
BorderRadius.circular(15),
borderSide: BorderSide.none,
),
prefixIcon: const
Icon(Icons.email, color: Colors.white70),
),
validator: (value) {
  if (value?.isEmpty ?? true) {
    return 'Please enter email';
  }
  return null;
},
),
const SizedBox(height: 20),
Container(
  decoration: BoxDecoration(
    color: Colors.white10,
borderRadius:
BorderRadius.circular(15),
border: Border.all(color:
Colors.white24),
),
child: TextFormField(
  controller: _passwordController,
  style: const TextStyle(color:
Colors.white),
obscureText:
!_isPasswordVisible,
decoration: InputDecoration(
  labelText: 'Password',
labelStyle: const
TextStyle(color: Colors.white70),
border: OutlineInputBorder(
  borderRadius:
BorderRadius.circular(15),
borderSide: BorderSide.none,
),
prefixIcon: const
Icon(Icons.lock, color: Colors.white70),
suffixIcon: IconButton(
  icon: Icon(
    _isPasswordVisible
? Icons.visibility
: Icons.visibility_off,
color: Colors.white70,
),
onPressed: () {

```

```

        setState() {
          _isPasswordVisible =
!_isPasswordVisible;
        });
      },
    ),
    validator: (value) {
      if (value?.isEmpty ?? true) {
        return 'Please enter password';
      }
      return null;
    },
  ),
  const SizedBox(height: 10),
  Align(
    alignment:
Alignment.centerRight,
    child: TextButton(
      onPressed: () {
        // Add forgot password
functionality
      },
      child: const Text(
        'Forgot Password?',
        style: TextStyle(color:
Colors.white70),
      ),
    ),
    const SizedBox(height: 20),
    Container(
      decoration: BoxDecoration(
        gradient: const LinearGradient(
          colors: [Colors.blue,
Colors.lightBlue],
        ),
        borderRadius:
BorderRadius.circular(15),
        boxShadow: [
          BoxShadow(
            color:
Colors.blue.withOpacity(0.3),
            blurRadius: 10,
            spreadRadius: 2,
          ),
        ],
      ),
      child: ElevatedButton(
        onPressed: () {
          if

```

```

(_formKey.currentState?.validate() ?? false) {
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) =>
OTPVerificationScreen(
          email:
_emailController.text,
        ),
      );
    },
    style: ElevatedButton.styleFrom(
      backgroundColor:
Colors.transparent,
      shadowColor:
Colors.transparent,
      minimumSize: const
Size.fromHeight(55),
      shape:
RoundedRectangleBorder(
        borderRadius:
BorderRadius.circular(15),
      ),
    ),
    child: const Text(
      'Login',
      style: TextStyle(
        fontSize: 18,
        fontWeight: FontWeight.bold,
        color: Colors.white,
      ),
    ),
    const SizedBox(height: 20),
    Row(
      mainAxisAlignment:
MainAxisAlignment.center,
      children: [
        Container(
          height: 1,
          width: 50,
          color: Colors.white24,
        ),
        const Padding(
          padding:
EdgeInsets.symmetric(horizontal: 10),
          child: Text(
            'Or continue with',
            style: TextStyle(color:

```



## register\_screen.dart

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

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

  @override
  State<RegisterScreen> createState() =>
    _RegisterScreenState();
}

class _RegisterScreenState extends
  State<RegisterScreen> {
  final _formKey = GlobalKey<FormState>();
  final _nameController = TextEditingController();
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();
  bool _isPasswordVisible = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: const BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [
              Color(0xFF1A237E),
              Color(0xFF0D47A1),
              Color(0xFF01579B),
            ],
          ),
        ),
        child: SafeArea(
          child: SingleChildScrollView(
            child: Padding(
              padding: const EdgeInsets.all(20.0),
              child: Form(
                key: _formKey,
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    IconButton(
                      icon: const Icon(Icons.arrow_back_ios, color:
                        Colors.white),
                      onPressed: () => Navigator.pop(context),
                    ),
                    const SizedBox(height: 20),
                    const Text(
                      'Create Account',
                      style: TextStyle(
                        fontSize: 32,
```

```
fontWeight: FontWeight.bold,
                        color: Colors.white,
                      ),
                    ),
                    const SizedBox(height: 10),
                    Text(
                      'Join SecureShield VPN',
                      style: TextStyle(
                        color: Colors.white.withOpacity(0.7),
                        fontSize: 16,
                      ),
                    ),
                    const SizedBox(height: 40),
                    // Name Field
                    Container(
                      decoration: BoxDecoration(
                        color: Colors.white10,
                        borderRadius: BorderRadius.circular(15),
                        border: Border.all(color: Colors.white24),
                      ),
                      child: TextFormField(
                        controller: _nameController,
                        style: const TextStyle(color: Colors.white),
                        decoration: InputDecoration(
                          labelText: 'Full Name',
                          labelStyle: const TextStyle(color: Colors.white70),
                          border: OutlineInputBorder(
                            borderRadius: BorderRadius.circular(15),
                            borderSide: BorderSide.none,
                          ),
                        ),
                        prefixIcon: const Icon(Icons.person, color:
                          Colors.white70),
                      ),
                      validator: (value) {
                        if (value?.isEmpty ?? true) {
                          return 'Please enter your name';
                        }
                      },
                      return null;
                    ),
                    const SizedBox(height: 20),
                    // Email Field
                    Container(
                      decoration: BoxDecoration(
                        color: Colors.white10,
                        borderRadius: BorderRadius.circular(15),
                        border: Border.all(color: Colors.white24),
                      ),
                      child: TextFormField(
                        controller: _emailController,
                        style: const TextStyle(color: Colors.white),
                        decoration: InputDecoration(
```

```

labelText: 'Email',
labelStyle: const TextStyle(color: Colors.white70),
border: OutlineInputBorder(
borderRadius: BorderRadius.circular(15),
borderSide: BorderSide.none,
),
prefixIcon: const Icon(Icons.email, color:
Colors.white70),
),
validator: (value) {
if (value?.isEmpty ?? true) {
return 'Please enter email';
}
if (!value!.contains('@')) {
return 'Please enter valid email';
}
return null;
},
),
),
const SizedBox(height: 20),
// Password Field
Container(
decoration: BoxDecoration(
color: Colors.white10,
borderRadius: BorderRadius.circular(15),
border: Border.all(color: Colors.white24),
),
child: TextFormField(
controller: _passwordController,
style: const TextStyle(color: Colors.white),
obscureText: !_isPasswordVisible,
decoration: InputDecoration(
labelText: 'Password',
labelStyle: const TextStyle(color: Colors.white70),
border: OutlineInputBorder(
borderRadius: BorderRadius.circular(15),
borderSide: BorderSide.none,
),
prefixIcon: const Icon(Icons.lock, color:
Colors.white70),
suffixIcon: IconButton(
icon: Icon(
_isPasswordVisible
? Icons.visibility
: Icons.visibility_off,
color: Colors.white70,
),
),
onPressed: () {
setState(() {
_isPasswordVisible = !_isPasswordVisible;
});
},
),
),
validator: (value) {

```

```

if (value?.isEmpty ?? true) {
return 'Please enter password';
}
if (value!.length < 6) {
return 'Password must be at least 6 characters';
}
return null;
},
),
),
const SizedBox(height: 40),
// Register Button
Container(
width: double.infinity,
decoration: BoxDecoration(
gradient: const LinearGradient(
colors: [Colors.blue, Colors.lightBlue],
),
borderRadius: BorderRadius.circular(15),
boxShadow: [
BoxShadow(
color: Colors.blue.withOpacity(0.3),
blurRadius: 10,
spreadRadius: 2,
),
],
),
child: ElevatedButton(
onPressed: () {
if (_formKey.currentState?.validate() ?? false) {
Navigator.push(
context,
MaterialPageRoute(
builder: (context) => OTPVerificationScreen(
email: _emailController.text,
),
),
);
},
style: ElevatedButton.styleFrom(
backgroundColor: Colors.transparent,
shadowColor: Colors.transparent,
padding: const EdgeInsets.symmetric(vertical: 15),
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(15),
),
),
child: const Text(
'Create Account',
style: TextStyle(
fontSize: 18,
fontWeight: FontWeight.bold,
),
),
),
),

```



```
),
const SizedBox(height: 20),
// Login Option
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    const Text(
      'Already have an account? ',
      style: TextStyle(color: Colors.white70),
    ),
    TextButton(
      onPressed: () => Navigator.pop(context),
      child: const Text(
        'Login',
        style: TextStyle(
          color: Colors.white,
          fontWeight: FontWeight.bold,
        ),
      ),
    ),
  ],
),
],
),
],
),
),
),
),
),
),
),
);
}
```

## otp\_verification\_screen.dart

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

```
import 'dart:async';
```

```
class OTPVerificationScreen extends StatefulWidget {
```

```
  final String? email;
```

```
  final String? phoneNumber;
```

```
  const OTPVerificationScreen({
```

```
    super.key,
```

```
    this.email,
```

```
    this.phoneNumber,
```

```
  }) : assert(email != null || phoneNumber != null);
```

```
  @override
```

```
  State<OTPVerificationScreen> createState() =>
```

```
    _OTPVerificationScreenState();
```

```
}
```

```
class _OTPVerificationScreenState extends
```

```
  State<OTPVerificationScreen> {
```

```
    final List<TextEditingController> _controllers =
```

```
    List.generate(
```

```
      4,
```

```
      (index) => TextEditingController(),
```

```
    );
```

```
    final List<FocusNode> _focusNodes = List.generate(
```

```
      4,
```

```
      (index) => FocusNode(),
```

```
    );
```

```
    Timer? _timer;
```

```
    int _timeLeft = 30;
```

```
    @override
```

```
    void initState() {
```

```
      super.initState();
```

```
      startTimer();
```

```
    }
```

```
    void startTimer() {
```

```
      _timer = Timer.periodic(const Duration(seconds: 1),
```

```
      (timer) {
```

```
        if (_timeLeft > 0) {
```

```
          setState(() {
```

```
            _timeLeft--;
```

```
          });
```

```
        } else {
```

```
          _timer?.cancel();
```

```
        }
```

```
      });
```

```
    }
```

```

@override

void dispose() {
  _timer?.cancel();

  for (var controller in _controllers) {
    controller.dispose();
  }

  for (var node in _focusNodes) {
    node.dispose();
  }

  super.dispose();
}

void verifyOTP() {
  String otp = _controllers.map((e) => e.text).join();

  if (otp.length == 4) {
    // Here you would typically verify the OTP with your
    backend

    Navigator.pushReplacementNamed(context, '/main');
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(

```

```

decoration: const BoxDecoration(
  gradient: LinearGradient(
    begin: Alignment.topCenter,
    end: Alignment.bottomCenter,
    colors: [
      Color(0xFF1A237E),
      Color(0xFF0D47A1),
      Color(0xFF01579B),
    ],
  ),
),
child: SafeArea(
  child: Padding(
    padding: const EdgeInsets.all(20.0),
    child: Column(
      crossAxisAlignment:
        CrossAxisAlignment.start,
      children: [
        IconButton(
          icon: const Icon(Icons.arrow_back_ios,
            color: Colors.white),
          onPressed: () => Navigator.pop(context),
        ),
        const SizedBox(height: 30),
        const Text(
          'Verification Code',

```

```

style: TextStyle(
  fontSize: 32,
  fontWeight: FontWeight.bold,
  color: Colors.white,
),
),
const SizedBox(height: 10),
Text(
  'We have sent the verification code
to\n${ widget.email ?? widget.phoneNumber}',
  style: TextStyle(
    color: Colors.white.withOpacity(0.7),
    fontSize: 16,
  ),
),
const SizedBox(height: 50),
Row(
  mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
  children: List.generate(
    4,
    (index) => SizedBox(
      width: 60,
      child: Container(
        decoration: BoxDecoration(
          color: Colors.white.withOpacity(0.1),

```

```

borderRadius:
BorderRadius.circular(15),
border: Border.all(color:
Colors.white24),
),
child: TextField(
  controller: _controllers[index],
  focusNode: _focusNodes[index],
  textAlign: TextAlign.center,
  style: const TextStyle(
    color: Colors.white,
    fontSize: 24,
    fontWeight: FontWeight.bold,
  ),
  keyboardType: TextInputType.number,
  maxLength: 1,
  decoration: InputDecoration(
    counterText: "",
    filled: true,
    fillColor: Colors.transparent,
    border: OutlineInputBorder(
      borderRadius:
BorderRadius.circular(15),
      borderSide: BorderSide.none,
    ),
  ),
),

```

```

        onChanged: (value) {
          if (value.isNotEmpty && index < 3) {
            _focusNodes[index +
1].requestFocus();
          }
          if (value.isEmpty && index > 0) {
            _focusNodes[index -
1].requestFocus();
          }
          if (index == 3 && value.isNotEmpty)
{
          verifyOTP();
        }
      },
    ),
  ),
),
),
),
const SizedBox(height: 40),
Center(
  child: Text(
    'Time remaining: $_timeLeft seconds',
    style: const TextStyle(
      color: Colors.white70,
      fontSize: 16,
    ),
  ),
),
const SizedBox(height: 20),
Center(
  child: TextButton(
    onPressed: () {
      setState(() {
        _timeLeft = 30;
      });
      startTimer();
      // Here you would typically resend the
      OTP
    },
  ),
  child: const Text(
    'Resend Code',
    style: TextStyle(
      color: Colors.white,
      fontSize: 16,
    ),
  ),
),
const Spacer(),
Container(

```

```

width: double.infinity,
decoration: BoxDecoration(
  gradient: const LinearGradient(
    colors: [Colors.blue, Colors.lightBlue],
  ),
  borderRadius: BorderRadius.circular(15),
  boxShadow: [
    BoxShadow(
      color: Colors.blue.withOpacity(0.3),
      blurRadius: 10,
      spreadRadius: 2,
    ),
  ],
),
borderRadius: BorderRadius.circular(15),
boxShadow: [
  BoxShadow(
    color: Colors.blue.withOpacity(0.3),
    blurRadius: 10,
    spreadRadius: 2,
  ),
],
),
child: ElevatedButton(
  onPressed: verifyOTP,
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.transparent,
    shadowColor: Colors.transparent,
    padding: const
EdgeInsets.symmetric(vertical: 15),
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(15),
    ),
  ),
  child: const Text(
    'Verify',
    style: TextStyle(
      fontSize: 18,
      fontWeight: FontWeight.bold,
    ),
  ),
),
);

```

## profile\_screen.dart

```
import 'package:flutter/material.dart';
import 'dart:math';
import 'dart:async';
class ProfileScreen extends StatelessWidget {
  const ProfileScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.transparent,
      body: SingleChildScrollView(
        child: Column(
          children: [
            _buildProfileHeader(),
            _buildProfileOptions(),
          ],
        ),
      ),
    );
  }

  Widget _buildProfileHeader() {
    return Container(
      padding: const EdgeInsets.all(32),
      decoration: BoxDecoration(
        gradient: LinearGradient(
          colors: [Colors.blue.shade900,
Colors.purple.shade900],
        ),
        borderRadius: const
BorderRadius.vertical(bottom:
Radius.circular(30)),
      ),
      child: SafeArea(
        child: Column(
          children: [
            const CircleAvatar(
              radius: 50,
              backgroundColor: Colors.white,
              child: Icon(Icons.person, size: 50, color:
Colors.blue),
            ),
            const SizedBox(height: 16),
            const Text('Free User',
              style: TextStyle(
                color: Colors.white,
                fontSize: 24,
                fontWeight: FontWeight.bold,
              ),
            ),
            const SizedBox(height: 16),
            _buildPremiumButton(),
          ],
        ),
      ),
    );
  }
}
```

```
],
    ),
  );
}

Widget _buildPremiumButton() {
  return Container(
    decoration: BoxDecoration(
      gradient: LinearGradient(
        colors: [Colors.amber.shade400,
Colors.orange.shade400],
      ),
      borderRadius: BorderRadius.circular(15),
      boxShadow: [
        BoxShadow(
          color: Colors.amber.withOpacity(0.3),
          blurRadius: 10,
          spreadRadius: 2,
        ),
      ],
    ),
    child: ElevatedButton(
      onPressed: () {},
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.transparent,
        shadowColor: Colors.transparent,
        padding: const
EdgeInsets.symmetric(horizontal: 32, vertical: 12),
        shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(15)),
      ),
      child: const Row(
        mainAxisAlignment: MainAxisAlignment.min,
        children: [
          Icon(Icons.star, color: Colors.white),
          SizedBox(width: 8),
          Text('Upgrade to Premium',
            style: TextStyle(color: Colors.white,
fontWeight: FontWeight.bold),
          ),
        ],
      ),
    );
  }

  Widget _buildProfileOptions() {
    return Padding(
      padding: const EdgeInsets.all(16),
      child: Column(
        children: [
          _buildOptionTile('Connection History',
Icons.history),
          _buildOptionTile('Support',
```

```

Icons.support_agent),
    _buildOptionTile('About', Icons.info),
    _buildOptionTile('Logout', Icons.logout,
isLogout: true),
  ],
),
);
}

```

```

Widget _buildOptionTile(String title, IconData
icon, {bool isLogout = false}) {
  return Container(
    margin: const EdgeInsets.only(bottom: 12),
    decoration: BoxDecoration(
      color: Colors.white.withOpacity(0.1),
      borderRadius: BorderRadius.circular(15),
      border: Border.all(color: Colors.white24),
    ),
    child: ListTile(
      leading: Container(
        padding: const EdgeInsets.all(8),
        decoration: BoxDecoration(
          color: (isLogout ? Colors.red :
Colors.blue).withOpacity(0.2),
          borderRadius: BorderRadius.circular(10),
        ),
        child: Icon(icon,
          color: isLogout ? Colors.red : Colors.blue,
        ),
      ),
      title: Text(title,
        style: TextStyle(
          color: isLogout ? Colors.red :
Colors.white,
          fontWeight: FontWeight.bold,
        ),
      ),
      trailing: const
Icon(Icons.arrow_forward_ios,
        color: Colors.white70,
        size: 16,
      ),
    ),
  );
}
}

```

```

TextStyle(fontSize:
  18)), Column(
  children: myCrops.map((crop) => Text(crop)).toList(),
),
],
),
);
}

```

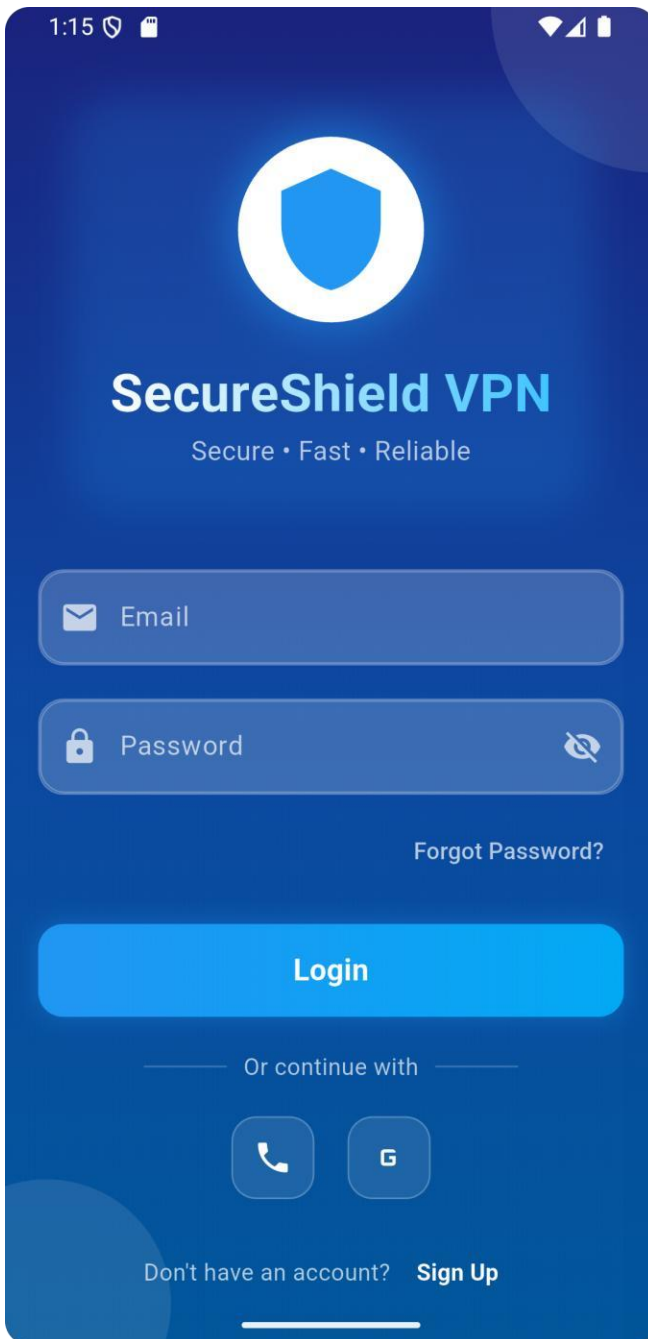
```

Widget _buildTextField({
  required String label,
  required String
  initialValue,
  required Function(String?) onSave,
}) {
  return TextFormField(
    initialValue: initialValue,
    decoration:
    InputDecoration(
      labelText: label,
      border: OutlineInputBorder(),
    ),
    onSave: onSave,
  );
}


```



**OUTPUT: -**






1:15



# SecureShield VPN

Secure • Fast • Reliable



 Email

 Password 

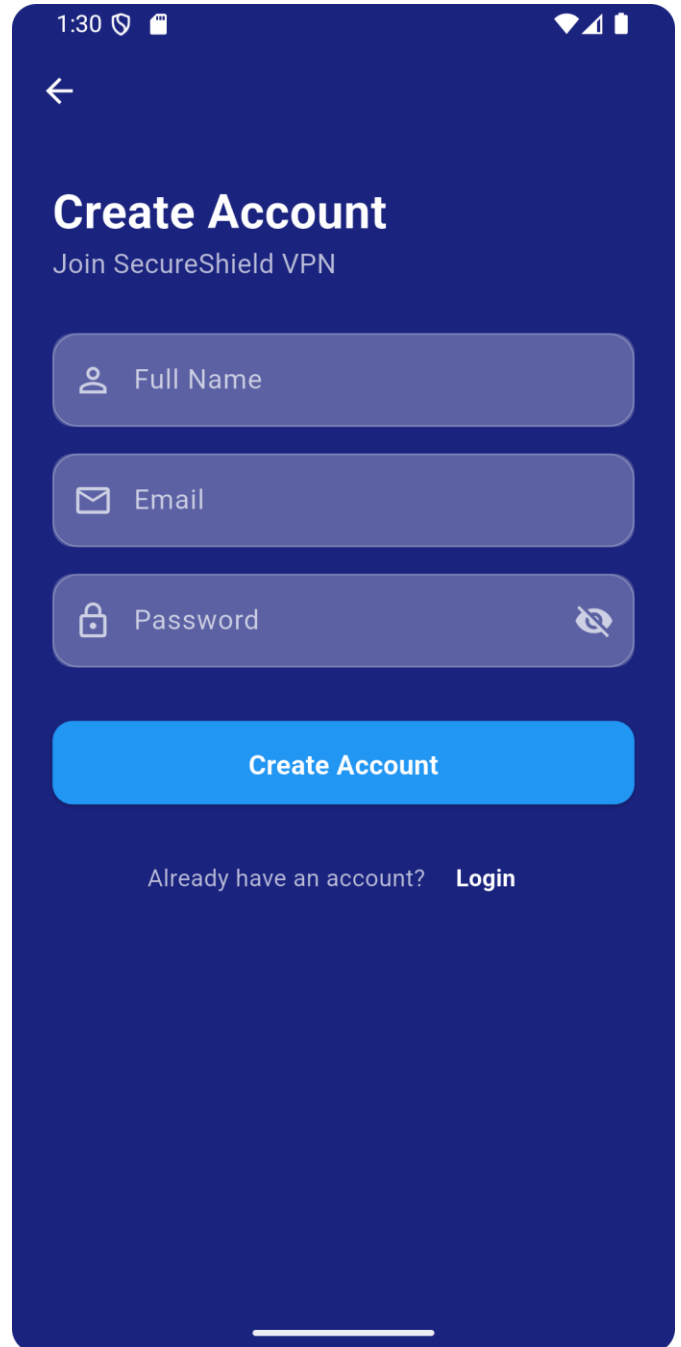
[Forgot Password?](#)

**Login**


Or continue with

Don't have an account? **Sign Up**





1:30





## Create Account

Join SecureShield VPN

 Full Name

 Email

 Password 

**Create Account**

Already have an account? **Login**

