

```

class RegisterForm extends StatefulWidget {
  @override
  RegisterFormState createState() => _RegisterFormState();
}

class _RegisterFormState extends State<RegisterForm> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _passwordController = TextEditingController();
  bool _obscurePassword = true;
  bool _obscureConfirmPassword = true;

  String? _validateFullName(String? value) {
    if (value == null || value.isEmpty) return 'Vui lòng nhập họ tên';
    if (!RegExp(r'^[a-zA-Z\sÀ-ÿ]+$').hasMatch(value)) {
      return 'Họ tên không được chứa số hoặc ký tự đặc biệt';
    }
    return null;
  }

  String? _validatePhoneNumber(String? value) {
    if (value == null || value.isEmpty) return 'Vui lòng nhập số điện thoai';
    if (!RegExp(r'^(0[3|5|7|8|9])[0-9]{8}$').hasMatch(value)) {
      return 'Số điện thoai không hợp lệ';
    }
    return null;
  }

  String? _validateEmail(String? value) {
    if (value == null || value.isEmpty) return 'Vui lòng nhập email';
    if (!EmailValidator.validate(value)) {
      return 'Email không hợp lệ';
    }
    return null;
  }
}

```

```
String? _validatePassword(String? value) {  
    if (value == null || value.isEmpty) return 'Vui lòng nhập mật khẩu';  
    if (value.length < 8 || value.length > 20) {  
        return 'Mật khẩu phải từ 8 đến 20 ký tự';  
    }  
    if (!RegExp(r'(?=.*[a-z])').hasMatch(value)) {  
        return 'Mật khẩu phải chứa ít nhất 1 chữ thường';  
    }  
    if (!RegExp(r'(?=.*[A-Z])').hasMatch(value)) {  
        return 'Mật khẩu phải chứa ít nhất 1 chữ hoa';  
    }  
    if (!RegExp(r'(?=.*\d)').hasMatch(value)) {  
        return 'Mật khẩu phải chứa ít nhất 1 chữ số';  
    }  
    if (!RegExp(r'(?=.*[!@#\$%^&*(),.?":{}|<>])').hasMatch(value)) {  
        return 'Mật khẩu phải chứa ít nhất 1 ký tự đặc biệt';  
    }  
    return null;  
}
```

```
String? _validateConfirmPassword(String? value) {  
    if (value == null || value.isEmpty) return 'Vui lòng nhập lại mật khẩu';  
    if (value != _passwordController.text) {  
        return 'Mật khẩu không khớp';  
    }  
    return null;  
}
```

```

child: Padding(
  padding: const EdgeInsets.all(16.0),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Image.asset('assets/logo.png', height: 100),
      SizedBox(height: 20),
      Form(
        key: _formKey,
        child: Column(
          children: [
            _buildTextField('Full name', _validateFullName),
            _buildTextField('Phone number', _validatePhoneNumber, keyboardType: TextInputType.phone),
            _buildTextField('Email', _validateEmail, keyboardType: TextInputType.emailAddress),
            _buildTextField('Password', _validatePassword, controller: _passwordController, obscureText: _obscurePassword, toggleObscure: () {
              setState(() {
                _obscurePassword = !_obscurePassword;
              });
            }),
            _buildTextField('Confirm password', _validateConfirmPassword, obscureText: _obscureConfirmPassword, toggleObscure: () {
              setState(() {
                _obscureConfirmPassword = !_obscureConfirmPassword;
              });
            }),
            SizedBox(height: 20),
            ElevatedButton(
              style: ElevatedButton.styleFrom(
                backgroundColor: Color(0xFFC04030),
                shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(8)),
                minimumSize: Size(double.infinity, 50),
              ),
              onPressed: () {
                if (_formKey.currentState!.validate()) {
                  ScaffoldMessenger.of(context).showSnackBar(
                    SnackBar(content: Text('Đăng ký thành công')),
                  );
                }
              },
              child: Text('SIGN UP', style: TextStyle(color: Colors.white, fontWeight: FontWeight.bold)),
            ), // ElevatedButton
            SizedBox(height: 10),
            TextButton(
              onPressed: () {},
              child: Text('Already have account? Log in', style: TextStyle(color: Color(0xFFC04030))),
            ),
          ],
        ),
      ),
    ],
  ),
),

```

```
Widget _buildTextField(String label, String? Function(String?) validator, {TextInputType? keyboardType, bool obscureText = false, TextEditingController? controller, VoidCallback? onPressed}) {
  return Padding(
    padding: const EdgeInsets.symmetric(vertical: 8.0),
    child: TextFormField(
      controller: controller,
      decoration: InputDecoration(
        labelText: label,
        filled: true,
        fillColor: Colors.white,
        border: OutlineInputBorder(borderRadius: BorderRadius.circular(8)),
        suffixIcon: toggleObscure != null
          ? IconButton(
              icon: Icon(obscureText ? Icons.visibility_off : Icons.visibility),
              onPressed: toggleObscure,
            ) // IconButton
          : null,
      ), // InputDecoration
      keyboardType: keyboardType,
      obscureText: obscureText,
      validator: validator,
    ), // TextFormField
  ); // Padding
}
```




SIGN UP

Already have account? [Log in](#)