

Experiment 04: To create an interactive Form using form widget**Theory:**

In Flutter, forms are a critical part of collecting user input. They allow validation, state tracking, and control over multiple form fields. Your SignUpPage is an excellent example of building a fully functional and interactive Form.

1. Form Widget

- Purpose: Acts as a container for grouping form fields and handling their validation.

2. TextFormField Widget

- Purpose: Used for user input like username, email, and password.
- Each field includes:
 - A controller to access input values.
 - A validator to define validation rules.

3. DropdownButtonFormField

- Used to allow the user to select a role (student or admin).

4. Form Validation & Submission

- Each input is validated based on its rules (validator) before proceeding with Firebase Auth registration.

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'homepage.dart';
import 'signin_page.dart';
import 'admin.dart';
```

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

  @override
  State<SignUpPage> createState() => _SignUpPageState();
}
```

```
class _SignUpPageState extends State<SignUpPage> {
```

```
final FirebaseAuth _auth = FirebaseAuth.instance;
final FirebaseFirestore _firestore = FirebaseFirestore.instance;

final TextEditingController _usernameController = TextEditingController();
final TextEditingController _emailController = TextEditingController();
final TextEditingController _passwordController = TextEditingController();
final _formKey = GlobalKey<FormState>();
bool _isLoading = false;

String _selectedRole = 'student'; // Default role

Future<void> _signUp() async {
  if (!_formKey.currentState!.validate()) return;

  setState(() => _isLoading = true);

  try {
    // Create user
    UserCredential userCredential =
      await _auth.createUserWithEmailAndPassword(
        email: _emailController.text.trim(),
        password: _passwordController.text.trim(),
      );

    User? user = userCredential.user;

    if (user != null) {
      // Send email verification
      await user.sendEmailVerification();

      // Store user data in Firestore
      await _firestore.collection('users').doc(user.uid).set({
        'username': _usernameController.text.trim(),
        'email': _emailController.text.trim(),
        'role': _selectedRole,
        'createdAt': Timestamp.now(),
      });

      // Show verification message
      _showEmailVerificationDialog();
    }
  }
```

```
} on FirebaseAuthException catch (e) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text("Error: ${e.message}")),
  );
}

setState(() => _isLoading = false);
}

void _showEmailVerificationDialog() {
  showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: const Text("Verify Your Email"),
      content: const Text(
        "A verification email has been sent to your email address. Please verify your email before logging in."),
      actions: [
        TextButton(
          onPressed: () {
            Navigator.pushReplacement(
              context,
              MaterialPageRoute(builder: (context) => const SignInPage()),
            );
          },
          child: const Text("OK"),
        ),
      ],
    ),
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      width: double.infinity,
      padding: const EdgeInsets.all(24.0),
      decoration: const BoxDecoration(
        gradient: LinearGradient(
          colors: [Color(0xFF4A00E0), Color(0xFF8E2DE2)],

```

```
begin: Alignment.topCenter,
end: Alignment.bottomCenter,
),
),
child: Center(
  child: SingleChildScrollView(
    child: Form(
      key: _formKey,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          // Quizzify Logo
          Image.asset("assets/images/quizzify_logo.png", height: 100),
          const SizedBox(height: 10),

          // App Name
          const Text(
            "QUIZZIFY",
            style: TextStyle(
              fontSize: 32,
              fontWeight: FontWeight.bold,
              color: Colors.white),
          ),
          const SizedBox(height: 20),

          const Text("Sign Up",
            style: TextStyle(
              fontSize: 28,
              fontWeight: FontWeight.bold,
              color: Colors.white)),
          const SizedBox(height: 8),
          const Text("Create your account",
            style: TextStyle(color: Colors.white70, fontSize: 16)),
          const SizedBox(height: 24),

          // Username Field
          TextFormField(
            controller: _usernameController,
            decoration: _buildInputDecoration("Username", Icons.person),
            style: const TextStyle(color: Colors.white, fontSize: 14),
            validator: (value) =>
```

```
        value!.isEmpty ? "Please enter a username" : null,
    ),
    const SizedBox(height: 16),

    // Email Field
    TextFormField(
      controller: _emailController,
      decoration: _buildInputDecoration("Email", Icons.email),
      keyboardType: TextInputType.emailAddress,
      style: const TextStyle(color: Colors.white, fontSize: 16),
      validator: (value) {
        if (value == null || value.isEmpty)
          return "Please enter your email";
        if (!value.contains("@") || !value.contains("."))
          return "Enter a valid email";
        return null;
      },
    ),
    const SizedBox(height: 16),

    // Password Field
    TextFormField(
      controller: _passwordController,
      decoration: _buildInputDecoration("Password", Icons.lock),
      obscureText: true,
      style: const TextStyle(color: Colors.white, fontSize: 16),
      validator: (value) {
        if (value == null || value.isEmpty)
          return "Please enter a password";
        if (value.length < 4 || value.length > 8)
          return "Password must be 4-8 characters long";
        return null;
      },
    ),
    const SizedBox(height: 24),

    // Role Selection
    DropdownButtonFormField<String>(
      value: _selectedRole,
      decoration: _buildInputDecoration(
        "Select Role", Icons.person_outline),
```

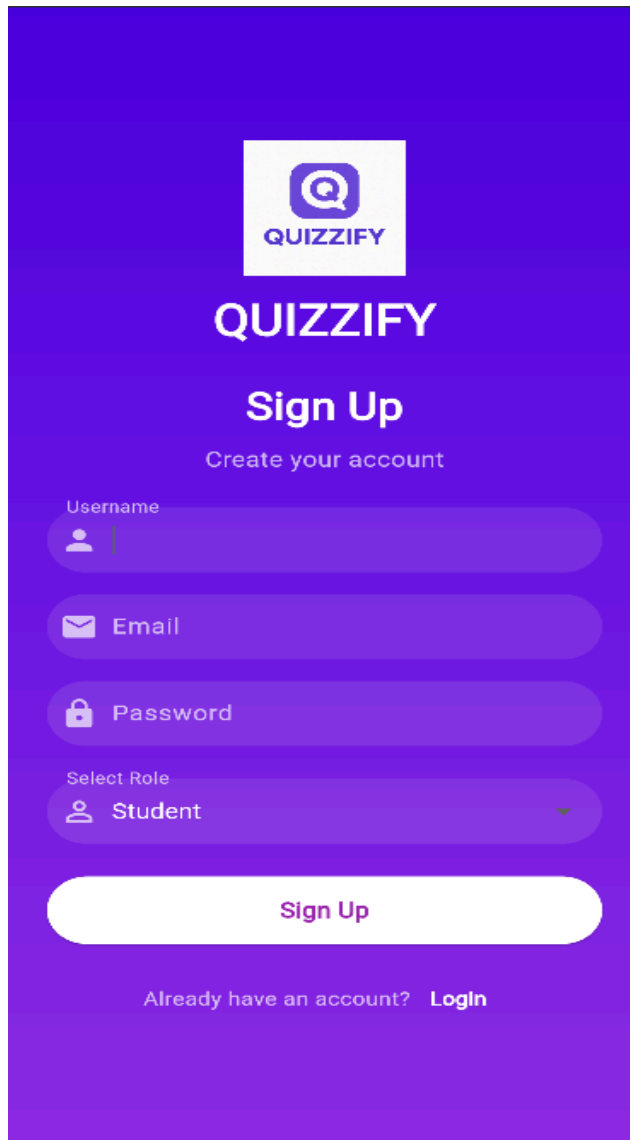
```
dropdownColor: Colors.deepPurple,
style: const TextStyle(color: Colors.white),
items: const [
  DropdownMenuItem(
    value: "student",
    child: Text("Student",
      style: TextStyle(
        color: Colors.white, fontSize: 16))),
  DropdownMenuItem(
    value: "admin",
    child: Text("Admin",
      style: TextStyle(
        color: Colors.white, fontSize: 16))),
],
onChanged: (value) {
  setState() {
    _selectedRole = value!;
  });
},
),

const SizedBox(height: 24),

// Sign Up Button
SizedBox(
  width: double.infinity,
  height: 50,
  child: ElevatedButton(
    onPressed: _isLoading ? null : _signUp,
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.white,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(30)),
    ),
    child: _isLoading
      ? const CircularProgressIndicator()
      : const Text(
          "Sign Up",
          style: TextStyle(
            color: Colors.purple,
            fontWeight: FontWeight.bold,
```

```
InputDecoration _buildInputDecoration(String label, IconData icon) {
  return InputDecoration(
    labelText: label,
    labelStyle: const TextStyle(color: Colors.white70),
    filled: true,
```

```
fillColor: Colors.white.withOpacity(0.1),  
prefixIcon: Icon(icon, color: Colors.white70),  
border: OutlineInputBorder(  
  borderRadius: BorderRadius.circular(30), borderSide: BorderSide.none),  
);  
}  
}
```

The image shows a mobile application screen for signing up. At the top, there is a white square logo with a blue 'Q' and the word 'QUIZZIFY' in blue. Below the logo, the word 'QUIZZIFY' is written in large, bold, white capital letters. Underneath, 'Sign Up' is written in bold white text, followed by 'Create your account' in a smaller white font. The form consists of four rounded rectangular input fields with a light blue background and a darker blue border. The first field is labeled 'Username' and has a person icon. The second is labeled 'Email' and has an envelope icon. The third is labeled 'Password' and has a lock icon. The fourth is labeled 'Select Role' and has a dropdown menu with 'Student' selected and a green arrow icon. Below the form is a large, rounded rectangular button with a white background and blue text that says 'Sign Up'. At the bottom, there is a link that says 'Already have an account? Login' in white text.**Conclusion:**

The SignUpPage in your Quiz App demonstrates an interactive form using Flutter's Form and TextFormField widgets. It efficiently handles user input through controllers and validations for fields like username, email, and password. Dropdowns add interactivity by allowing role selection within the form. The form ensures user input is validated before Firebase sign-up is triggered. Overall, it provides a user-friendly and functional interface for account creation.