

## EXPERIMENT 6

**AIM:**To Connect Flutter UI with fireBase database

**Theory:**

- **Flutter Forms Overview:** Flutter provides `Form` and `TextFormField` widgets to handle user input efficiently in structured formats like login, signup, or feedback forms.
- **Creating a Form:** Use the `Form` widget with a `GlobalKey<FormState>` to group and manage multiple form fields. This allows you to control validation and form state.
- **Input Fields (TextFormField):** `TextFormField` is commonly used for capturing text input. It can be paired with a `TextEditingController` to manage text and a `validator` for validation rules.
- **Validation Mechanism:** Use the `validator` property in `TextFormField` to define logic such as checking for empty inputs, valid email formats, or password strength.
- **Form Submission:** When the user presses a submit button (e.g., `ElevatedButton`), use `_formKey.currentState!.validate()` to check validation, and `_formKey.currentState!.save()` to save the data.
- **State Management:** Use controllers and stateful widgets to dynamically manage input values and control form UI behavior.

**CODE:**

```
import 'package:firebase_auth/firebase_auth.dart';

class AuthService {
  final FirebaseAuth _auth = FirebaseAuth.instance;

  // Sign Up Method
  Future<User?> signUpWithEmail(String email, String password) async {
    try {
      UserCredential userCredential = await _auth.createUserWithEmailAndPassword(
        email: email,
        password: password,
      );
      return userCredential.user;
    } catch (e) {
      // Handle errors
    }
  }
}
```

```
    } catch (e) {  
        print("Error: $e");  
        return null;  
    }  
}
```

// Sign In Method

```
Future<User?> signInWithEmail(String email, String password) async {  
    try {  
        UserCredential userCredential = await _auth.signInWithEmailAndPassword(  
            email: email,  
            password: password,  
        );  
        return userCredential.user;  
    } catch (e) {  
        print("Error: $e");  
        return null;  
    }  
}
```

// Sign Out Method

```
Future<void> signOut() async {  
    await _auth.signOut();  
}  
}
```

## OUTPUT :

Login

Welcome Back Donor!

Email

Password

Login

[Forgot Password?](#)

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

← Sign Up

Create an Account

Full Name

Mobile Number

Email

Password

Sign Up

[Already have an account? Login](#)

Donation ▾

Authentication

Users Sign-in method Templates Usage Settings Extensions

The following Authentication features will stop working when Firebase Dynamic Links shuts down on August 25, 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.

▾

🔍 Search by email address, phone number, or user UID

Add user ↻ ⋮

Identifier	Providers	Created ▾	Signed In	User UID
2022.sandeshkumar.ya...	✉	Mar 7, 2025	Mar 7, 2025	uPJrIxtNurEYvoyfIMojFduthZZf2
2022.atharva.shinde@v...	✉	Mar 7, 2025	Apr 11, 2025	kTHMd0AV7sRhrMbiVKHyDFc...

Rows per page: 50 ▾ 1 - 2 of 2 < >