MPL Experiment 4

Name: Prathamesh Palve **Class:** D15A **Roll no:**31

Aim: To create an interactive Form using form widget

Theory:

Creating an interactive form in Flutter requires using form-related widgets to collect and validate user input efficiently. The `Form` widget, combined with `TextFormField`, provides a structured way to manage input fields. Various input widgets like `TextField`, `DropdownButton`, `Checkbox`, `Radio`, and `Switch` allow users to enter data in different formats.

Validation and state management can be handled using the `GlobalKey<FormState>` to validate inputs before submission. Wrapping the form in a `SingleChildScrollView` ensures smooth scrolling when multiple fields are present.

A `RaisedButton` (deprecated) or `ElevatedButton` can trigger validation and submission logic. To enhance usability and create a responsive experience, proper padding, spacing, and `InputDecoration` should be applied.

Steps:

Step 1: Create a new Flutter project or open an existing one.

Step 2: Define a `Form` widget inside a StatefulWidget to manage user input.

Step 3: Use `TextFormField` for text input fields with validation logic.

Step 4: Include other input widgets such as `DropdownButton`, `Checkbox`, `Radio`, and `Switch` for additional user selections.

Step 5: Wrap the form inside a `SingleChildScrollView` to ensure smooth scrolling.

Step 6: Implement an `ElevatedButton` to trigger form validation and submission.

Step 7: Use `GlobalKey<FormState>` to manage form validation.

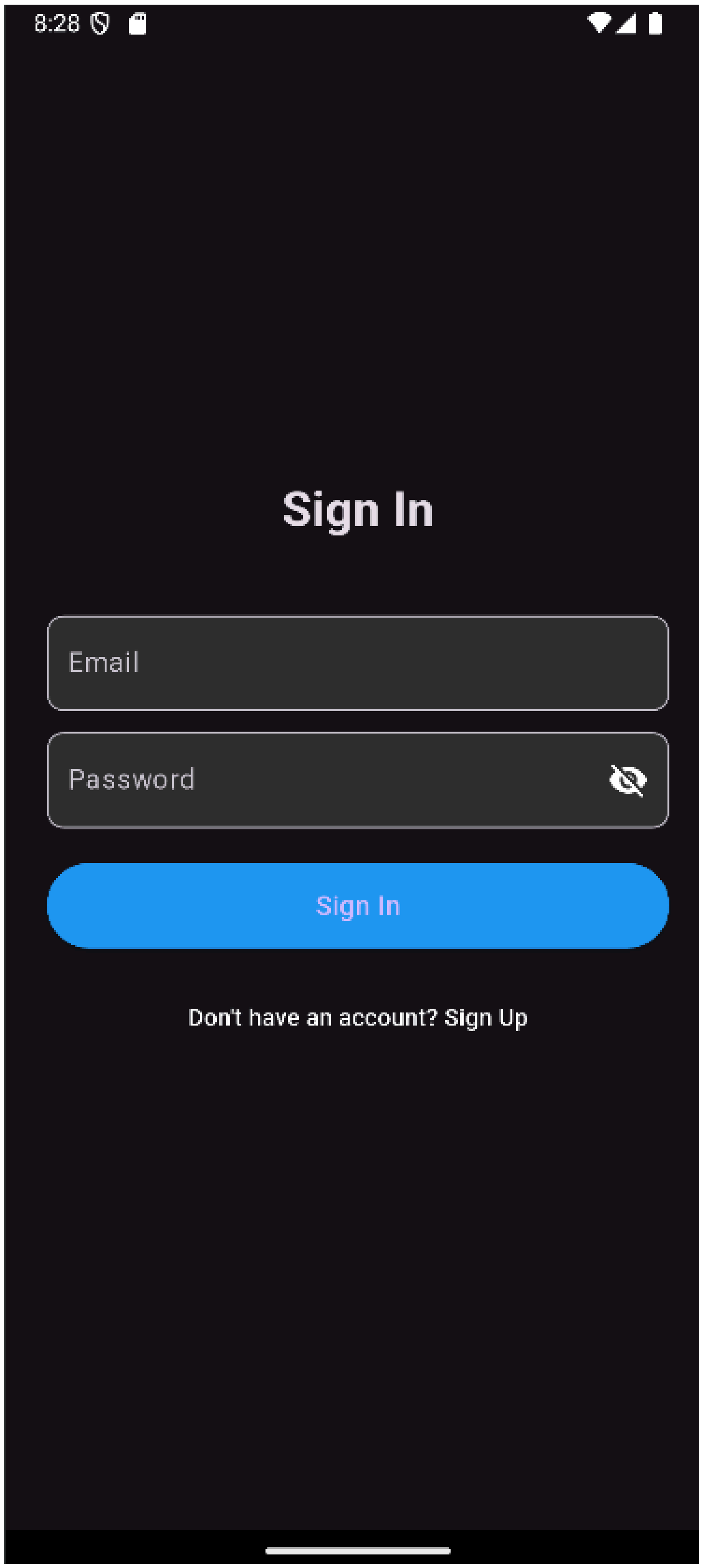
Code:

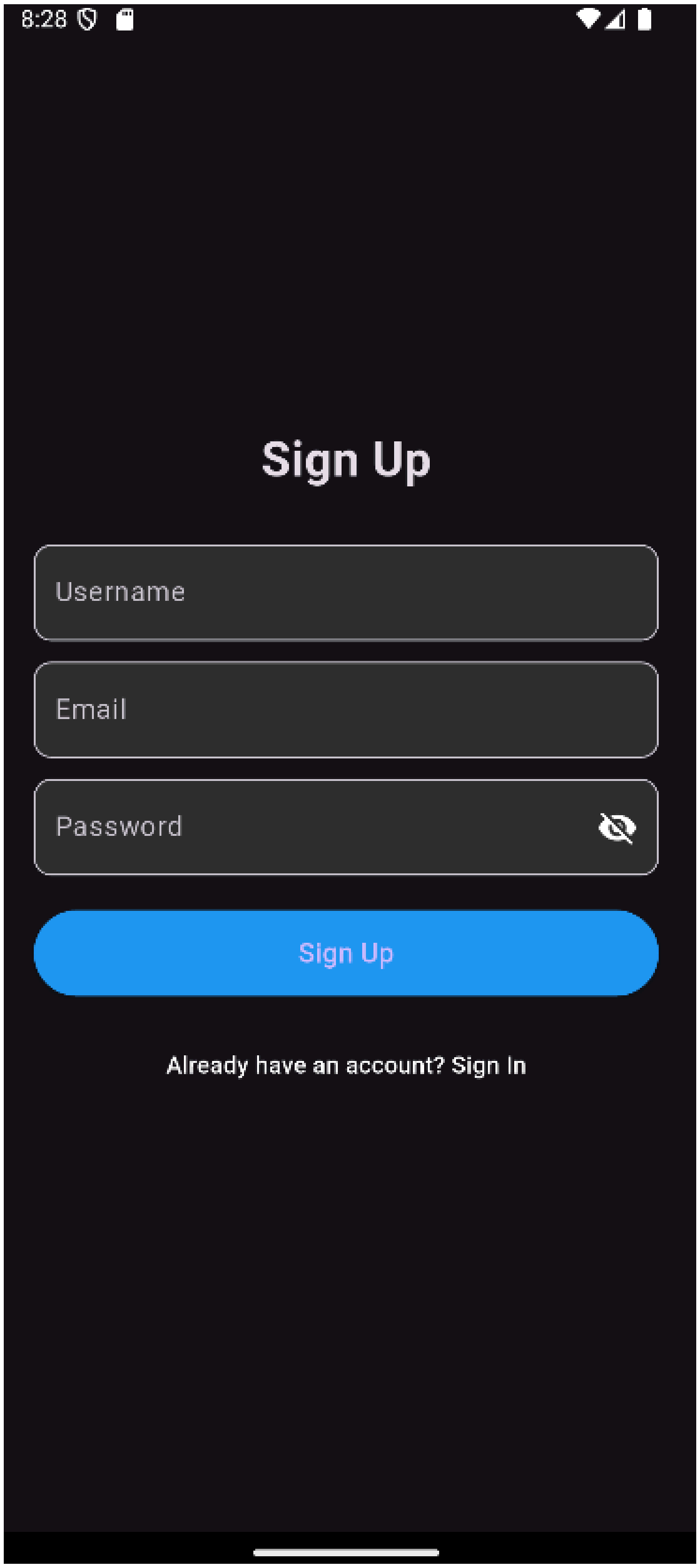
```dart
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'home.dart'; // Import HomeScreen

class AuthScreen extends StatefulWidget {
 @override
 _AuthScreenState createState() => _AuthScreenState();
```

```
}

class _AuthScreenState extends State<AuthScreen> {
 bool isSignUp = false;
 bool isPasswordVisible = false;
 bool isLoading = false;

 final FirebaseAuth _auth = FirebaseAuth.instance;
 final TextEditingController _emailController = TextEditingController();
 final TextEditingController _passwordController = TextEditingController();
 final TextEditingController _usernameController = TextEditingController();

 @override
 void initState() {
   super.initState();
   _auth.setPersistence(Persistence.SESSION); // Ensures user logs in again if app restarts
 }

 Future<void> _authenticate() async {
   setState(() => isLoading = true);

   String email = _emailController.text.trim();
   String password = _passwordController.text.trim();
   String username = _usernameController.text.trim();

   if (email.isEmpty || password.isEmpty || (isSignUp && username.isEmpty)) {
     ScaffoldMessenger.of(context).showSnackBar(
       SnackBar(content: Text("All fields are required!")),
     );
     setState(() => isLoading = false);
     return;
   }

   try {
     UserCredential userCredential;
     if (isSignUp) {
       // Sign Up
       userCredential = await _auth.createUserWithEmailAndPassword(
         email: email,
         password: password,
       );
       await userCredential.user?.updateDisplayName(username);
       ScaffoldMessenger.of(context).showSnackBar(
         SnackBar(content: Text("Account Created. Please Login!")),
       );
```

```
        setState(() => isSignUp = false); // Switch to sign-in mode
      } else {
        // Sign In
        userCredential = await _auth.signInWithEmailAndPassword(
          email: email,
          password: password,
        );
        String chatId = userCredential.user?.uid ?? "defaultChatId";

        if (mounted) {
          Navigator.pushReplacement(
            context,
            MaterialPageRoute(builder: (context) => HomeScreen(chatId: chatId)),
// Pass chatId
          );
        }
      }
    } on FirebaseAuthException catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text(e.message ?? "Authentication error")),
      );
    } finally {
      if (mounted) setState(() => isLoading = false);
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Padding(
        padding: EdgeInsets.symmetric(horizontal: 24),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            Text(
              isSignUp ? "Sign Up" : "Sign In",
              style: TextStyle(fontSize: 28, fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 30),
            if (isSignUp)
              TextField(
                controller: _usernameController,
                decoration: InputDecoration(
                  hintText: "Username",
                  filled: true,
                fillColor: Colors.grey[850],
                border: OutlineInputBorder(borderRadius:
BorderRadius.circular(10)),
              ),
            ),
            SizedBox(height: 12),
            TextField(
              controller: _emailController,
              decoration: InputDecoration(
                hintText: "Email",
                filled: true,
                fillColor: Colors.grey[850],
                border: OutlineInputBorder(borderRadius:
BorderRadius.circular(10)),
              ),
            ),
            SizedBox(height: 12),
            TextField(
              controller: _passwordController,
              obscureText: !isPasswordVisible,
              decoration: InputDecoration(
                hintText: "Password",
                filled: true,
                fillColor: Colors.grey[850],
                border: OutlineInputBorder(borderRadius:
BorderRadius.circular(10)),
                suffixIcon: IconButton(
                  icon: Icon(
                    isPasswordVisible ? Icons.visibility : Icons.visibility_off,
                    color: Colors.white,
                  ),
                  onPressed: () {
                    setState(() {
                      isPasswordVisible = !isPasswordVisible;
                    });
                  },
                ),
              ),
            ),
            SizedBox(height: 20),
            isLoading
                ? CircularProgressIndicator()
                : ElevatedButton(
              onPressed: _authenticate,
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.blue,
                minimumSize: Size(double.infinity, 50),
              ),
              child: Text(isSignUp ? "Sign Up" : "Sign In", style:
TextStyle(fontSize: 16)),
            ),
            SizedBox(height: 16),
            TextButton(
              onPressed: () => setState(() => isSignUp = !isSignUp),
              child: Text(
                isSignUp ? "Already have an account? Sign In" : "Don't have an
account? Sign Up",
                style: TextStyle(color: Colors.white),
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

Output:
Login
Page




8:28
Sign Up
Username
Email
Password
Sign Up
Already have an account? Sign In