

NAME - KARAN MISHRA

DIV - D15B

ROLL NO - 33

MPL EXP 5

auth_screen.dart

```
import 'dart:math';
import 'package:flutter/material.dart';
import '../services/auth_service.dart';
import '../screens/menu_screen.dart';

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

  @override
  _AuthScreenState createState() => _AuthScreenState();
}

class _AuthScreenState extends State<AuthScreen> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final TextEditingController _captchaController = TextEditingController();
  final AuthService _authService = AuthService();

  final List<String> captchaWords = ["apple", "secure", "verify", "random", "trust"];
  late String captchaWord;

  @override
  void initState() {
    super.initState();
    _generateNewCaptcha();
  }

  void _generateNewCaptcha() {
    setState(() {
      captchaWord = captchaWords[Random().nextInt(captchaWords.length)];
    });
  }

  void _authenticate(bool isLogin) async {
    if (_captchaController.text.trim().toLowerCase() != captchaWord.toLowerCase()) {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text("Incorrect CAPTCHA. Try again!")),
      );
    }
  }
}
```

```

    _generateNewCaptcha();
    return;
}

String? userId = isLogin
    ? await _authService.login(_emailController.text.trim(), _passwordController.text.trim())
    : await _authService.register(_emailController.text.trim(), _passwordController.text.trim());

if (userId != null) {
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text(isLogin ? "Login Successful!" : "Registration Successful!")),
    );

    Future.delayed(const Duration(milliseconds: 500), () {
        Navigator.pushReplacement(
            context,
            MaterialPageRoute(
                builder: (context) => MenuScreen(userEmail: _emailController.text.trim()),
            ),
        );
    });
} else {
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text("Authentication Failed!")),
    );
}
}

```

```

@override
Widget build(BuildContext context) {
    return Scaffold(
        body: Stack(
            fit: StackFit.expand,
            children: [
                Image.asset("assets/images/background.png", fit: BoxFit.cover),
                Center(
                    child: Padding(
                        padding: const EdgeInsets.all(24.0),
                        child: Container(
                            padding: const EdgeInsets.all(20),
                            decoration: BoxDecoration(
                                color: Colors.white.withOpacity(0.9),
                                borderRadius: BorderRadius.circular(15),
                            ),
                        ),
                    ),
                ),
            ],
        ),
    );
}

```

```

    child: Column(
      mainAxisAlignment: MainAxisAlignment.min,
      children: [
        const Text("Welcome!", style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
        const SizedBox(height: 10),
        _buildTextField(_emailController, "Email"),
        _buildTextField(_passwordController, "Password", isPassword: true),
        const SizedBox(height: 10),
        Text("Enter the word: "$captchaWord",
          style: const TextStyle(fontSize: 16, fontWeight: FontWeight.bold)),
        _buildTextField(_captchaController, "Enter CAPTCHA"),
        const SizedBox(height: 10),
        _buildButton("Login", () => _authenticate(true), Colors.blue),
        _buildButton("Register", () => _authenticate(false), Colors.green),
      ],
    ),
  ),
),
],
),
);
}

```

```

Widget _buildTextField(TextEditingController controller, String hint, {bool isPassword = false}) {
  return Padding(
    padding: const EdgeInsets.symmetric(vertical: 8),
    child: TextField(
      controller: controller,
      obscureText: isPassword,
      decoration: InputDecoration(
        labelText: hint,
        border: OutlineInputBorder(borderRadius: BorderRadius.circular(12)),
        fillColor: Colors.white,
        filled: true,
      ),
    ),
  );
}

```

```

Widget _buildButton(String text, VoidCallback onPressed, Color color) {
  return Padding(
    padding: const EdgeInsets.symmetric(vertical: 8),
    child: ElevatedButton(

```

```

        style: ElevatedButton.styleFrom(backgroundColor: color, shape:
RoundedRectangleBorder(borderRadius: BorderRadius.circular(10))),
        onPressed: onPressed,
        child: Text(text, style: const TextStyle(fontSize: 18, color: Colors.white)),
      ),
    );
  }
}

```

auth.service.dart

```

import 'package:firebase_auth/firebase_auth.dart';

class AuthService {
  final FirebaseAuth _auth = FirebaseAuth.instance;

  Future<String?> login(String email, String password) async {
    try {
      UserCredential userCredential = await _auth.signInWithEmailAndPassword(
        email: email,
        password: password,
      );
      return userCredential.user?.uid;
    } catch (e) {
      return null;
    }
  }

  Future<String?> register(String email, String password) async {
    try {
      UserCredential userCredential = await _auth.createUserWithEmailAndPassword(
        email: email,
        password: password,
      );
      return userCredential.user?.uid;
    } catch (e) {
      return null;
    }
  }

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

```

```
User? getCurrentUser() {  
  return _auth.currentUser;  
}  
}
```

Output:-

Before registering:-

Authentication

[Users](#) | [Sign-in method](#) | [Templates](#) | [Usage](#) | [Settings](#) | [Extensions](#)

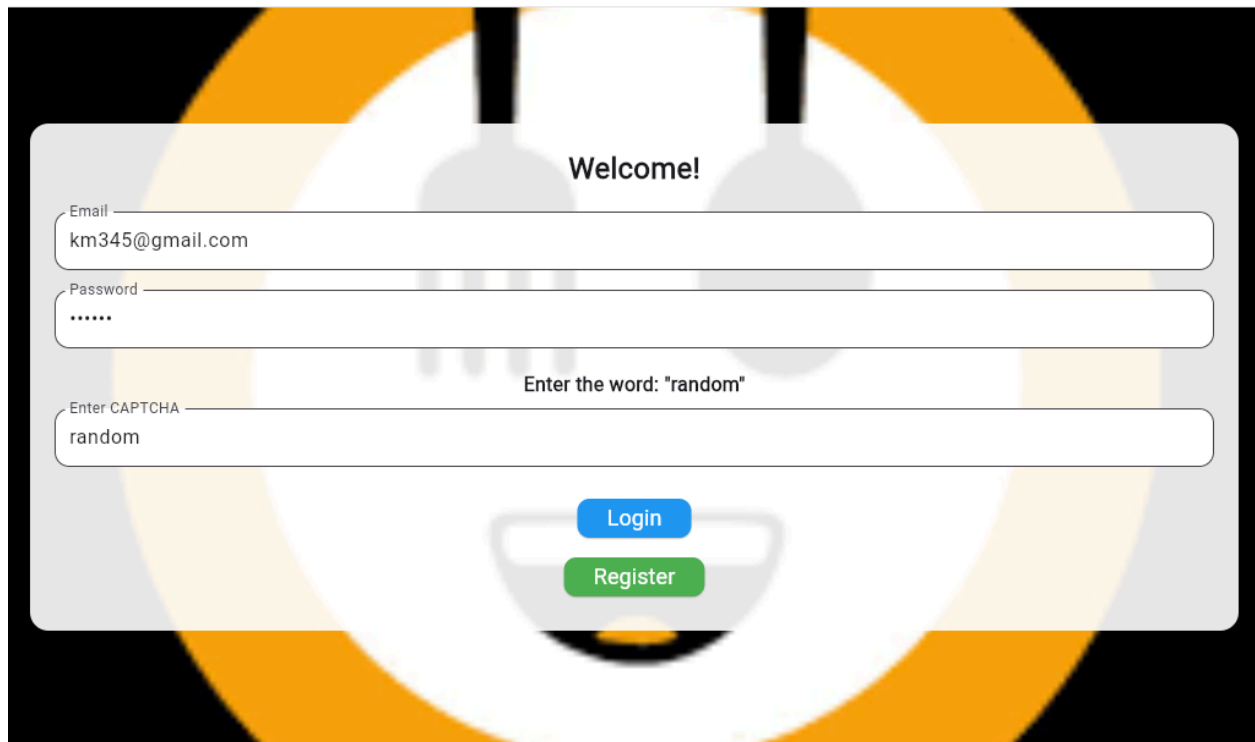
i The following Authentication features will stop working when Firebase Dynamic Links shuts down on August 25, 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps. [v](#)

[Add user](#) [Refresh](#) [More](#)

| Identifier | Providers | Created v | Signed In | User UID |
|-----------------|-----------|---------------------------|--------------|------------------------------|
| km234@gmail.com | | Mar 4, 2025 | Mar 17, 2025 | I6FHpczNVhTz8qSapUTXFzifL... |
| km123@gmail.com | | Mar 4, 2025 | Mar 18, 2025 | vnhZiqv4uAga3JXkLZ0FTuQO... |

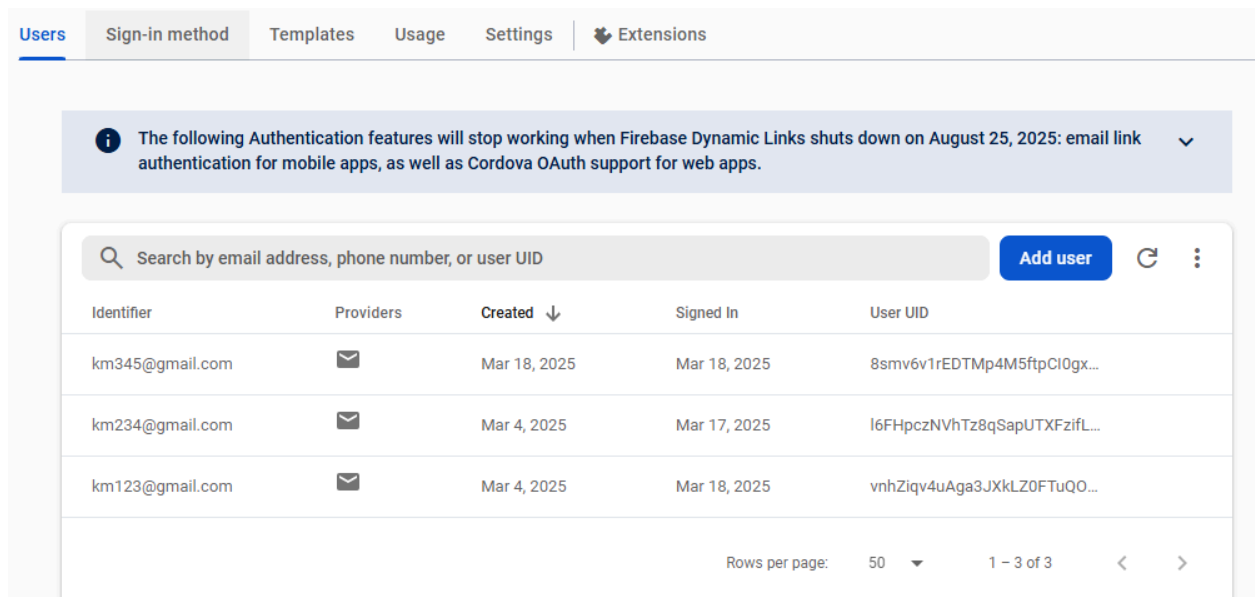
Rows per page: 50 [v](#) 1 – 2 of 2 [v](#)

Register and Login page






A login and registration form titled "Welcome!". It features three input fields: "Email" with the value "km345@gmail.com", "Password" with masked characters "*****", and "Enter CAPTCHA" with the value "random". The instruction "Enter the word: 'random'" is displayed above the CAPTCHA field. Below the inputs are two buttons: a blue "Login" button and a green "Register" button. The background shows a stylized yellow smiley face with a black outline.

After Registration:-



The screenshot shows the "Users" page in the Firebase Authentication console. It includes a navigation bar with links for "Users", "Sign-in method", "Templates", "Usage", "Settings", and "Extensions". A warning banner at the top states: "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." Below the banner is a search bar with the placeholder "Search by email address, phone number, or user UID" and an "Add user" button. The main content is a table of users.

| Identifier | Providers | Created ↓ | Signed In | User UID |
|-----------------|---|--------------|--------------|------------------------------|
| km345@gmail.com |  | Mar 18, 2025 | Mar 18, 2025 | 8smv6v1rEDTMp4M5ftpCI0gx... |
| km234@gmail.com |  | Mar 4, 2025 | Mar 17, 2025 | l6FHpczNVhTz8qSapUTXFzifL... |
| km123@gmail.com |  | Mar 4, 2025 | Mar 18, 2025 | vnhZlqv4uAga3JXkLZ0FTuQO... |

At the bottom right, there is a pagination control showing "Rows per page: 50" and "1 - 3 of 3" with navigation arrows.