Google-services.json

```
{
 "project_info": {
   "project_number": "663509823079",
   "project_id": "comaplaint1",
   "storage_bucket": "comaplaint1.firebasestorage.app"
 },
 "client": [
   {
     "client_info": {
       "mobilesdk_app_id": "1:663509823079:android:f919160ed188bc2ee42ec2",
       "android_client_info": {
         "package_name": "com.example.new_complaint"
       }
     },
     "oauth_client": [],
     "api_key": [
       {
         "current_key": "AIzaSyALbN6aQftIodwXpZmdIRzO0JDSUhD4ojk"
       }
     ],
     "services": {
       "appinvite_service": {
         "other_platform_oauth_client": []
       }
     }
   }
 ],
 "configuration_version": "1"
}
```

Login Page

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'home_page.dart';
import 'signup_page.dart';

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

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

class _LoginPageState extends State<LoginPage> {
```

```
final FirebaseAuth _auth = FirebaseAuth.instance;
final TextEditingController emailController = TextEditingController();
final TextEditingController passwordController = TextEditingController();
bool isLoading = false;

Future<void> _login() async {
  final email = emailController.text.trim();
  final password = passwordController.text.trim();

  if (email.isEmpty || password.isEmpty) {
    _showSnackbar('Please enter both email and password');
    return;
  }

  setState(() => isLoading = true);

  try {
    await _auth.signInWithEmailAndPassword(email: email, password: password);
    _showSnackbar('Login Successful');

    // Navigate to HomePage after successful login
    Navigator.pushReplacement(
      context,
      MaterialPageRoute(builder: (context) => const HomePage()),
    );
  } on FirebaseAuthException catch (e) {
    _handleFirebaseError(e);
  } finally {
    setState(() => isLoading = false);
  }
}

void _handleFirebaseError(FirebaseAuthException e) {
  String message = 'Login failed';
  if (e.code == 'user-not-found') {
    message = 'No account found for this email';
  } else if (e.code == 'wrong-password') {
    message = 'Incorrect password';
  } else if (e.code == 'invalid-email') {
    message = 'Invalid email format';
  }
  _showSnackbar(message);
}

void _showSnackbar(String message) {
  ScaffoldMessenger.of(context).showSnackBar(SnackBar(content:
Text(message)));
}
```

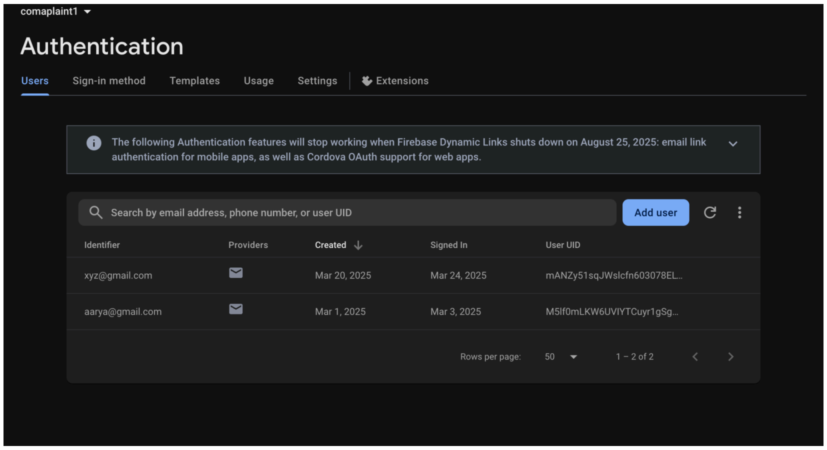
```
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SingleChildScrollView(
      child: Container(
        padding: const EdgeInsets.all(16.0),
        height: MediaQuery.of(context).size.height,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.blue.shade200, Colors.purple.shade200],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            // App Logo or Icon
            const Icon(
              Icons.lock_person,
              size: 100,
              color: Colors.white,
            ),
            const SizedBox(height: 20),
            const Text(
              'Welcome Back!',
              style: TextStyle(
                fontSize: 28,
                fontWeight: FontWeight.bold,
                color: Colors.white,
              ),
            ),
            const SizedBox(height: 10),
            const Text(
              'Login to continue',
              style: TextStyle(
                fontSize: 16,
                color: Colors.white70,
              ),
            ),
            const SizedBox(height: 40),
            // Email Field
            TextField(
              controller: emailController,
              decoration: InputDecoration(
                labelText: 'Email',
                labelStyle: const TextStyle(color: Colors.white),
                filled: true,
                fillColor: Colors.white.withOpacity(0.1),
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(10),
                  borderSide: BorderSide.none,
                ),
                prefixIcon: const Icon(Icons.email, color: Colors.white),
              ),
              style: const TextStyle(color: Colors.white),
              keyboardType: TextInputType.emailAddress,
            ),
            const SizedBox(height: 20),
            // Password Field
            TextField(
              controller: passwordController,
              obscureText: true,
              decoration: InputDecoration(
                labelText: 'Password',
                labelStyle: const TextStyle(color: Colors.white),
                filled: true,
                fillColor: Colors.white.withOpacity(0.1),
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(10),
                  borderSide: BorderSide.none,
                ),
                prefixIcon: const Icon(Icons.lock, color: Colors.white),
              ),
              style: const TextStyle(color: Colors.white),
            ),
            const SizedBox(height: 30),
            // Login Button
            isLoading
                ? const CircularProgressIndicator(color: Colors.white)
                : ElevatedButton(
              onPressed: _login,
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.white,
                padding: const EdgeInsets.symmetric(
                    horizontal: 50, vertical: 15),
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(10),
                ),
              ),
              child: const Text(
                'Login',
                style: TextStyle(
                  fontSize: 18,
                  color: Colors.blue,
                  fontWeight: FontWeight.bold,
                ),
              ),
             ),
             const SizedBox(height: 20),
             // Signup Link
             TextButton(
               onPressed: () {
                 Navigator.push(
                   context,
                   MaterialPageRoute(builder: (context) => const SignupPage()),
                 );
               },
               child: const Text(
                 'Don\'t have an account? Sign up',
                 style: TextStyle(
                   color: Colors.white,
                   fontSize: 16,
                 ),
               ),
             ),
           ],
         ),
       ),
     ),
   );
 }
}
```

Signup Page:

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'login_page.dart';

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

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

class _SignupPageState extends State<SignupPage> {
 final FirebaseAuth _auth = FirebaseAuth.instance;
 final TextEditingController emailController = TextEditingController();
 final TextEditingController passwordController = TextEditingController();
 final TextEditingController confirmPasswordController =
TextEditingController();
 bool isLoading = false;

 Future<void> _signup() async {
   final email = emailController.text.trim();
   final password = passwordController.text.trim();
   final confirmPassword = confirmPasswordController.text.trim();

   if (email.isEmpty || password.isEmpty || confirmPassword.isEmpty) {
     _showSnackbar('Please fill all fields');
     return;
   }

   if (password != confirmPassword) {
     _showSnackbar('Passwords do not match');
     return;
   }

   setState(() => isLoading = true);

   try {
     await _auth.createUserWithEmailAndPassword(email: email, password:
password);
     _showSnackbar('Signup Successful');

     // Navigate to LoginPage after successful signup
     Navigator.pushReplacement(
       context,
       MaterialPageRoute(builder: (context) => const LoginPage()),
     );
   } on FirebaseAuthException catch (e) {
     _handleFirebaseError(e);
   } finally {
     setState(() => isLoading = false);
   }
 }

 void _handleFirebaseError(FirebaseAuthException e) {
   String message = 'Signup failed';
   if (e.code == 'email-already-in-use') {
     message = 'Email is already in use';
   } else if (e.code == 'weak-password') {
     message = 'Password is too weak';
   } else if (e.code == 'invalid-email') {
     message = 'Invalid email format';
   }
   _showSnackbar(message);
 }

 void _showSnackbar(String message) {
   ScaffoldMessenger.of(context).showSnackBar(SnackBar(content:
Text(message)));
 }

 @override
```

```
Widget build(BuildContext context) {
  return Scaffold(
    body: SingleChildScrollView(
      child: Container(
        padding: const EdgeInsets.all(16.0),
        height: MediaQuery.of(context).size.height,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.blue.shade200, Colors.purple.shade200],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            // App Logo or Icon
            const Icon(
              Icons.person_add,
              size: 100,
              color: Colors.white,
            ),
            const SizedBox(height: 20),
            const Text(
              'Create an Account',
              style: TextStyle(
                fontSize: 28,
                fontWeight: FontWeight.bold,
                color: Colors.white,
              ),
            ),
            const SizedBox(height: 10),
            const Text(
              'Sign up to get started',
              style: TextStyle(
                fontSize: 16,
                color: Colors.white70,
              ),
            ),
            const SizedBox(height: 40),
            // Email Field
            TextField(
              controller: emailController,
              decoration: InputDecoration(
                labelText: 'Email',
                labelStyle: const TextStyle(color: Colors.white),
                filled: true,
                fillColor: Colors.white.withOpacity(0.1),
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(10),
                  borderSide: BorderSide.none,
                ),
                prefixIcon: const Icon(Icons.email, color: Colors.white),
              ),
              style: const TextStyle(color: Colors.white),
              keyboardType: TextInputType.emailAddress,
            ),
            const SizedBox(height: 20),
            // Password Field
            TextField(
              controller: passwordController,
              obscureText: true,
              decoration: InputDecoration(
                labelText: 'Password',
                labelStyle: const TextStyle(color: Colors.white),
                filled: true,
                fillColor: Colors.white.withOpacity(0.1),
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(10),
                  borderSide: BorderSide.none,
                ),
                prefixIcon: const Icon(Icons.lock, color: Colors.white),
              ),
              style: const TextStyle(color: Colors.white),
            ),
            const SizedBox(height: 20),
            // Confirm Password Field
            TextField(
              controller: confirmPasswordController,
              obscureText: true,
              decoration: InputDecoration(
                labelText: 'Confirm Password',
                labelStyle: const TextStyle(color: Colors.white),
                filled: true,
                fillColor: Colors.white.withOpacity(0.1),
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(10),
                  borderSide: BorderSide.none,
                ),
                prefixIcon: const Icon(Icons.lock, color: Colors.white),
              ),
              style: const TextStyle(color: Colors.white),
            ),
            const SizedBox(height: 30),
            // Signup Button
            isLoading
                ? const CircularProgressIndicator(color: Colors.white)
                : ElevatedButton(
                           onPressed: _signup,
                           style: ElevatedButton.styleFrom(
                             backgroundColor: Colors.white,
                             padding: const EdgeInsets.symmetric(
                                 horizontal: 50, vertical: 15),
                             shape: RoundedRectangleBorder(
                               borderRadius: BorderRadius.circular(10),
                             ),
                           ),
                           child: const Text(
                             'Sign Up',
                             style: TextStyle(
                               fontSize: 18,
                               color: Colors.blue,
                               fontWeight: FontWeight.bold,
                             ),
                           ),
                         ),
                         const SizedBox(height: 20),
                         // Login Link
                         TextButton(
                           onPressed: () {
                             Navigator.pushReplacement(
                               context,
                               MaterialPageRoute(builder: (context) => const LoginPage()),
                             );
                           },
                           child: const Text(
                             'Already have an account? Login',
                             style: TextStyle(
                               color: Colors.white,
                               fontSize: 16,
                             ),
                           ),
                         ),
                       ],
                     ),
                   ),
                 ),
               );
             }
            }
```

Firebase Screenshots:

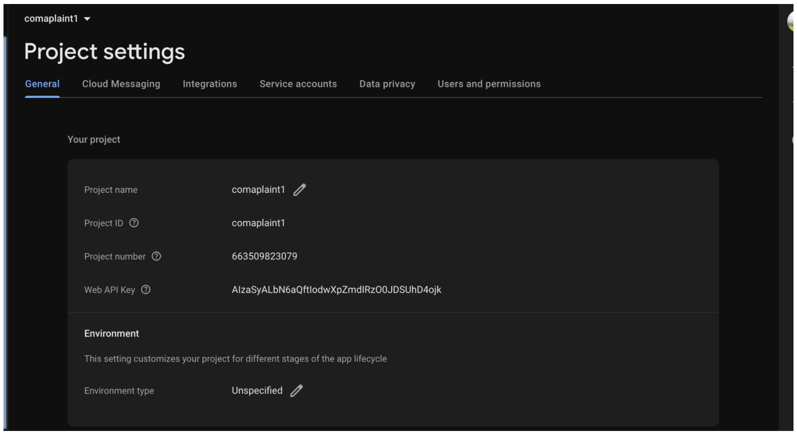


Authentication

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.

Identifier	Providers	Created ↓	Signed In	User UID
xyz@gmail.com	✉	Mar 20, 2025	Mar 24, 2025	mANZy51sqJWslcfn603078EL...
aarya@gmail.com	✉	Mar 1, 2025	Mar 3, 2025	M5lf0mLKW6UVIYTCuyr1gSg...

Rows per page: 50 ▾ 1 – 2 of 2



Project settings

Your project

Project name comaplaint1

Project ID comaplaint1

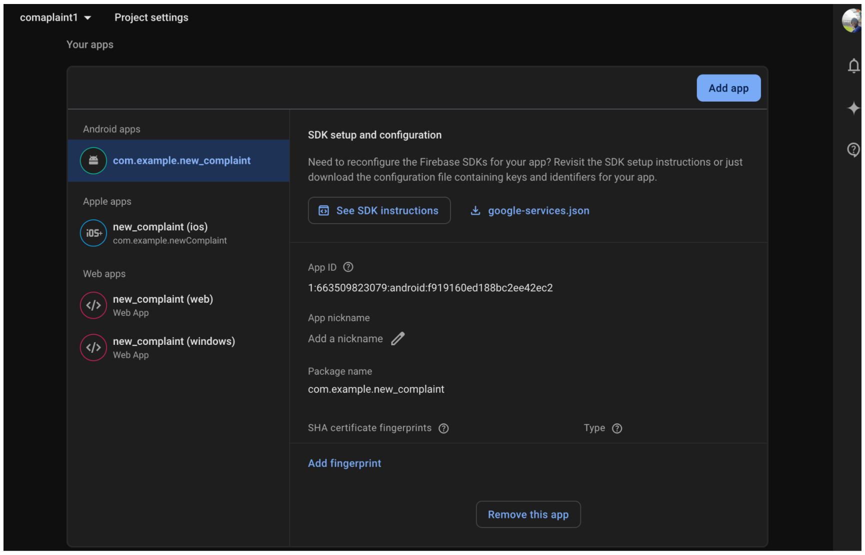
Project number 663509823079

Web API Key AIzaSyALbN6aQftIodwXpZmdIRzO0JDSUhD4ojk

Environment

This setting customizes your project for different stages of the app lifecycle

Environment type Unspecified



Your apps

Android apps

com.example.new_complaint

Apple apps

new_complaint (ios)
com.example.newComplaint

Web apps

new_complaint (web)
Web App

new_complaint (windows)
Web App

SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

App ID
1:663509823079:android:f919160ed188bc2ee42ec2

App nickname
Add a nickname

Package name
com.example.new_complaint

SHA certificate fingerprints Type