

Firebase Authentication

Sinh viên: Trần Viết Thắng

Sinh viên: Trần Duy Nguyên

Khoa điện tử viễn thông
Đại học bách khoa
Đại học Đà Nẵng

Ngày 16 tháng 11 năm 2025



Phân công công việc

Họ và tên	Nhiệm vụ	Phần trăm
Trần Duy Nguyên	<ul style="list-style-type: none">- Quản lí người dùng và các phương thức bảo mật.- Xây dựng giao diện app.	50%
Trần Viết Thắng	<ul style="list-style-type: none">- Các cách thức đăng nhập lên Firebase bằng Email, Google và Facebook.- Xây dựng giao diện ứng dụng.	50%



Outline

- ① Overview
- ② Email/Password login
- ③ Social Login
 - Google Login
 - Facebook Login
- ④ User Management
- ⑤ Security
- ⑥ Tài liệu tham khảo



Table of Contents

- 1 Overview
- 2 Email/Password login
- 3 Social Login
 - Google Login
 - Facebook Login
- 4 User Management
- 5 Security
- 6 Tài liệu tham khảo



Sơ lược về firebase

- Khái niệm:
 - Backend-as-a-Service (BaaS) - phát triển bởi google.
 - Cho phép người dùng xây dựng ứng dụng mà không cần quản lý server.
 - Cung cấp nhiều dịch vụ Authentication, Database, Storage, Hosting.
- Lợi ích:
 - Set up nhanh, không cần quan tâm đến backend API.
 - Dịch vụ miễn phí, phù hợp cho các dự án nhỏ.
 - Cho phép đăng tải dữ liệu thời gian thực.
 - Tài liệu rõ ràng, dễ dùng.



Firebase Authentication



- 1 Thông tin xác thực (gmail, password hoặc mã xác thực từ các nhà cung cấp).
- 2 Các thông tin xác thực gửi đến firebase SDK.
- 3 Xác thực thông tin và phản hồi lại.
- 4 Thành công : trả về thông tin cơ bản của người dùng và token xác thực.

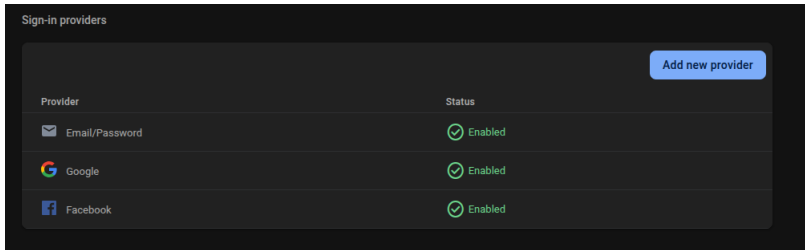


Table of Contents

- 1 Overview
- 2 Email/Password login
- 3 Social Login
 - Google Login
 - Facebook Login
- 4 User Management
- 5 Security
- 6 Tài liệu tham khảo



Bật quyền cho các nhà cung cấp



- 1 Ta nhấn vào nút "Add new provider" ở mục signed-in provider.
- 2 Chọn nhà cung cấp mong muốn và nhấn vào nút "Enable".
- 3 Thành công sẽ cho ta thấy thanh "Status" như trên hình.
- 4 Ứng với mỗi nhà cung cấp, ta cần làm theo các bước trong Firebase để có thể kết nối được với Firebase.



Đăng nhập và đăng kí bằng mail

- Thêm các gói cần thiết:
 - ① Khai báo trong pubspec.yaml :

```
1  firebase_core: ^3.15.2  
2  firebase_auth: ^5.7.0  
3
```

- Thực hiện các bước kết nối với Firebase.
- **Chú ý:** Các bước này cần thực hiện theo hướng dẫn để có kết nối với dịch vụ của Firebase.



Đăng nhập và đăng kí bằng email

- Xử lí đăng nhập:

```
1 final userCredential = await FirebaseAuth.instance.signInWithEmailAndPassword(  
2     email: _emailController.text.trim(),  
3     password: _passwordController.text,  
4 );  
5
```

- Sử dụng `FirebaseAuth.instance.signInWithEmailAndPassword`.
 - `_emailController.text.trim()` : Lấy nội dung người dùng nhập vào ô email, rồi loại bỏ khoảng trắng ở đầu/cuối.
- Firebase trả về `UserCredential` → Giúp cho công việc xác thực email người dùng.
- Xử lí đăng kí:

```
1 final userCredential = await FirebaseAuth.instance.  
2     createUserWithEmailAndPassword(  
3         email: _emailController.text.trim(),  
4         password: _passwordController.text,  
5     );
```



Đăng nhập và đăng kí bằng mail

- Sử dụng try catch cho 2 hàm đăng nhập và đăng kí


```
1 try{  
2 //  
3 } on FirebaseAuthException catch (e) {  
4 String message = 'Đã xảy ra lỗi';  
5 if (e.code == 'user-not-found') message = 'Không tìm thấy tài khoản';  
6 else if (e.code == 'wrong-password') message = 'sai mật khẩu';  
7 // ...  
8 _showError(message);}  
9
```

- FirebaseAuthException trả về các loại lỗi:
 - user-not-found: Email chưa được đăng ký.
 - wrong-password: Mật khẩu sai.
 - email-already-in-use: Email đã có tài khoản.
 - weak-password: Mật khẩu quá yếu.
 - invalid-email: Email sai định dạng.
 - invalid-credential: Thông tin không hợp lệ.




Demo đăng nhập


8:40




Đăng nhập



Chào mừng trở lại

 Tiếp tục với Google

 Tiếp tục với Facebook

HOẶC

 Email

 Mật khẩu 


[Quên mật khẩu?](#)

Đăng nhập


Chưa có tài khoản? [Đăng ký ngay](#)


8:44


← Trang chủ →





Thông tin cá nhân


 **Tên:**
Chưa cập nhật


 **Email:**
vietthang2512004@gmail.com

 **Tuổi:**
Chưa cập nhật


 **Giới tính:**
Chưa cập nhật

 **Ngày sinh:**
Chưa cập nhật

 **Phương thức:**
Email/Password


 [Chỉnh sửa thông tin](#)


8:48




Đăng nhập



Chào mừng trở lại

 Tiếp tục với Google

 Tiếp tục với Facebook

HOẶC

Email  **thang@gmail.com**

Mật khẩu  **.....** 

[Quên mật khẩu?](#)

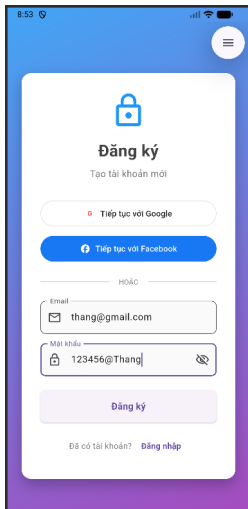
Đăng nhập

Chưa có tài khoản? [Đăng ký ngay](#)

[Email hoặc mật khẩu không đúng](#)



Demo đăng kí



8:53

Đăng ký

Tạo tài khoản mới

▶ Tiếp tục với Google

▶ Tiếp tục với Facebook

HOẶC

Email

thang@gmail.com

Mật khẩu

123456@Thang

Đăng ký

Đã có tài khoản? Đăng nhập

Identifier	Providers	Created ↓	Signed In	User UID
thang@gmail.com	✉	Nov 11, 2025	Nov 11, 2025	3AcNhuM27zTzqgIfOrZfbxH...
vietthang2512004@gm...	✉	Nov 10, 2025	Nov 11, 2025	3G31o1gR4dPlyud30xiNgPDF...
duynguyen25092004@...	Google	Nov 10, 2025	Nov 10, 2025	d2lFYjSveNbkx6bJ34rzTEcmE...
hellosigma@gmail.com	✉	Nov 10, 2025	Nov 10, 2025	mvFzrLedq20eoBZaAkJ2BfGe...
trannguyendut@gmail.c...	✉	Nov 10, 2025	Nov 10, 2025	tTsGNtoJJVbFC8Rii1kie6SHiy1
mhang0181@gmail.com	✉	Nov 10, 2025	Nov 10, 2025	GvJZR6g21nVpUyAUCYfiDoxY...
minhhangdo2004@gm...	✉	Nov 10, 2025	Nov 10, 2025	fcZUPeYIE1d6sLgf6Y3G9KPT...
thangtran2512004@gm...	Google	Nov 9, 2025	Nov 10, 2025	k8sapje7e2YB7xXH0coCOj8m...



Table of Contents

- 1 Overview
- 2 Email/Password login
- 3 Social Login**
 - Google Login
 - Facebook Login
- 4 User Management
- 5 Security
- 6 Tài liệu tham khảo

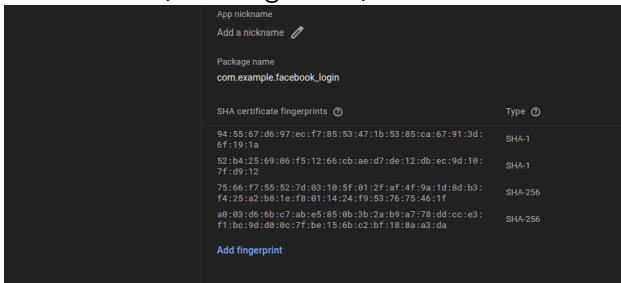


Cấu hình trên Firebase

- Để có thể kết nối với phương thức này ta cần 2 mã SHA1 và SHA256 của máy đang chạy ứng dụng.
- Sử dụng các câu lệnh như :

```
1 keytool -list -v \-alias androiddebugkey -keystore ~/.android/debug.keystore  
2
```

- Thêm vào mục setting của dự án Firebase:



Google login

```
1 Future<void> _signInWithGoogle() async {
2     //... UI handling ... //
3     try {
4         final GoogleSignIn googleSignIn = GoogleSignIn();
5         final GoogleSignInAccount? googleUser = await googleSignIn.signIn();
6         if (googleUser == null) {
7             setState(() => _isLoading = false);
8             return;
9         }
10        final GoogleSignInAuthentication googleAuth = await googleUser.authentication;
11        final credential = GoogleAuthProvider.credential(
12            accessToken: googleAuth.accessToken,
13            idToken: googleAuth.idToken,
14        );
15        await FirebaseAuth.instance.signInWithCredential(credential);
16        } on FirebaseAuthException catch (e) {
17            if (e.code == 'account-exists-with-different-credential') {
18                _pendingCredential = e.credential;
19                await _handleAccountConflict(e, _pendingCredential);
20            } else {
21                _showError('Đăng nhập google thất bại: ${e.toString()}');
22            }
23        } catch (e) {
24            _showError('Đăng nhập google thất bại: ${e.toString()}');
25        } finally { // UI handling //}
26    }
```



Google login

- Đăng nhập vào Google:

```
1 final GoogleSignIn googleSignIn = GoogleSignIn();  
2 final GoogleSignInAccount? googleUser = await googleSignIn.signIn();
```

- GoogleSignIn : Tạo một instance của class GoogleSignIn từ package.
- googleSignIn.signIn(): trả về một đối tượng của lớp GoogleSignInAccount. Nếu người dùng hủy đăng nhập thì trả về null.



Google login

- Lấy token xác thực và tạo Firebase Credential.

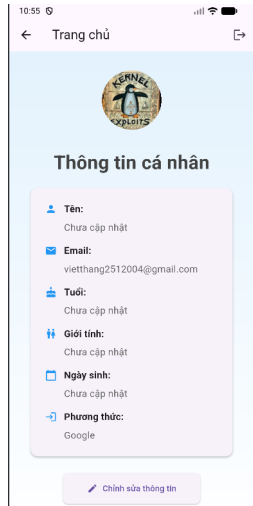
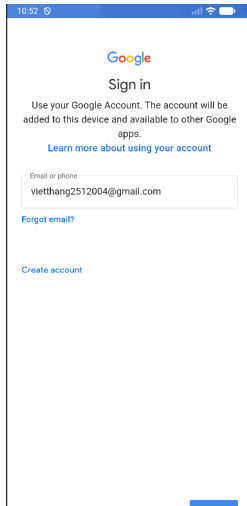
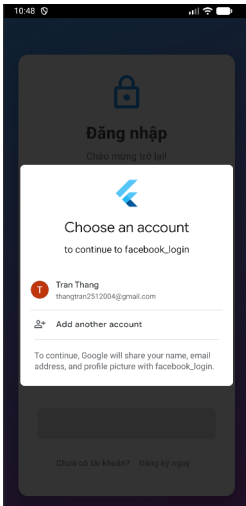
```
1 final GoogleSignInAuthentication googleAuth = await googleUser.authentication;  
2 final credential = GoogleAuthProvider.credential(  
3     accessToken: googleAuth.accessToken,  
4     idToken: googleAuth.idToken,  
5 );  
6 await FirebaseAuth.instance.signInWithCredential(credential);
```

- Đăng nhập thành công, Google sẽ cấp 2 loại token:
 - Access Token : truy cập vào Google API.
 - Id Token : Xác thực người dùng.
- Phương thức .credential() chuyển đổi token thành Firebase credential.
- Phương thức .signInWithCredential() với credential vừa tạo để truy cập vào Firebase.

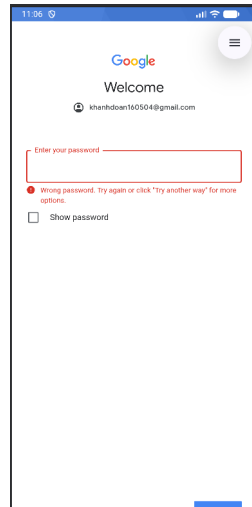
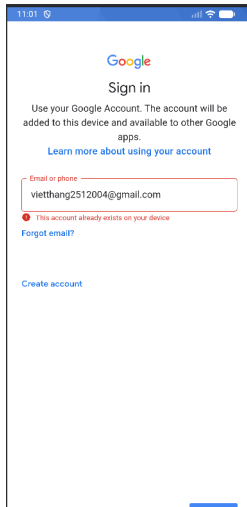
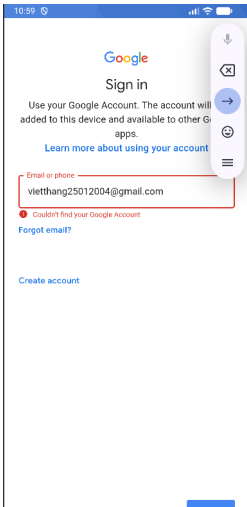


Google Login

Demo đăng nhập bằng google



Demo một số lỗi xảy ra



Facebook Login

```
1 Future<void> _signInWithFacebook() async {
2     // UI Handling ....
3     try {
4         final LoginResult result = await FacebookAuth.instance.login();
5         if (result.status == LoginStatus.success) {
6             final OAuthCredential credential = FacebookAuthProvider.credential(result.
              accessToken!.token);
7             await FirebaseAuth.instance.signInWithCredential(credential);
8         } else if (result.status == LoginStatus.cancelled) {
9             _showError('Đang nhập Facebook bị huy');
10            } else {
11                _showError('Đang nhập Facebook thất bại');
12            }
13    } on FirebaseAuthException catch (e) {
14        if (e.code == 'account-exists-with-different-credential') {
15            _pendingCredential = e.credential;
16            await _handleAccountConflict(e, _pendingCredential);
17        } else {
18            _showError('Lỗi Facebook: ${e.toString()}');
19        }
20    } catch (e) {
21        _showError('Lỗi Facebook: ${e.toString()}');
22    } finally { // UI Handling ...
23    }
24 }
```



Facebook login

- Đăng nhập vào Facebook:

```
1 final LoginResult result = await FacebookAuth.instance.login();
```

- Thực hiện đăng nhập trả về đối tượng kiểu - LoginResult.
 - status : success, failed, cancelled, operationInProgress.
 - accessToken : token trả về nếu khởi tạo thành công.
 - message : lỗi.
- Ta dựa vào 3 thuộc tính trên để xây dựng:

```
1 if (result.status == LoginStatus.success) {  
2   // .. process for success .  
3 }  
4 else {  
5   // .. process for failure .  
6 }
```



Facebook login

- Xử lí khi đăng nhập thành công:

```
1  if (result.status == LoginStatus.success) {  
2      final OAuthCredential credential = FacebookAuthProvider.credential(result.  
        accessToken!.token);  
3      await FirebaseAuth.instance.signInWithCredential(credential);  
4  }
```

- FacebookAuthProvider.credential() :
 - Chuyển đổi thành OAuthCredential cho Firebase.
- FirebaseAuth.instance.signInWithCredential():
 - Gửi credential lên Firebase.
 - Firebase verify với Facebook server.
 - Tạo/cập nhật user trong Firebase Auth.



Demo đăng nhập bằng Facebook

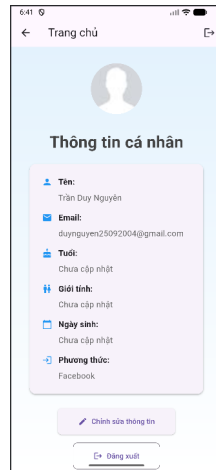


Table of Contents

- 1 Overview
- 2 Email/Password login
- 3 Social Login
 - Google Login
 - Facebook Login
- 4 User Management**
- 5 Security
- 6 Tài liệu tham khảo



Tại sao lại cần quản lý người dùng ?

- Hầu hết các ứng dụng cần biết danh tính của người dùng để:
 - Lưu dữ liệu người dùng an toàn trên đám mây.
 - Cung cấp trải nghiệm cá nhân hoá nhất quán trên mọi thiết bị.
- Việc quản lý người dùng là phần không thể thiếu khi xây dựng ứng dụng, nhưng thường phức tạp và tiềm ẩn rủi ro bảo mật.
- Với **Firestore Authentication**, nhà phát triển không cần tự xây dựng hệ thống xác thực và phân quyền từ đầu.
- Firestore cung cấp các thư viện và dịch vụ xác thực an toàn, giúp quản lý người dùng dễ dàng và giảm thiểu lỗi bảo mật.



Các phương thức quản lý người dùng

Quản lý từ phía Người dùng (User)

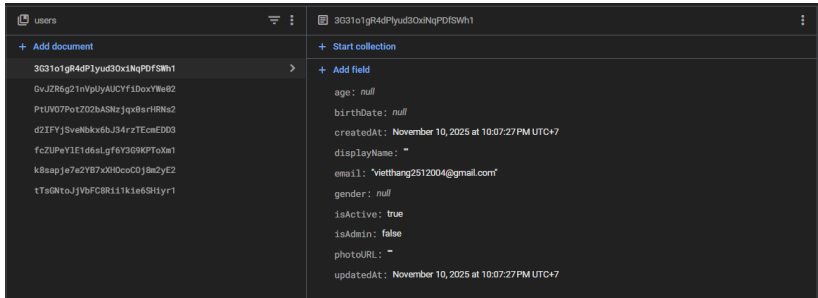
- Đăng ký tài khoản (Email/Password hoặc Google, Facebook).
- Hiển thị/cập nhật thông tin cá nhân.
- Gửi yêu cầu xác thực qua Email (xác minh tài khoản, đặt lại mật khẩu).

Quản lý từ phía Quản trị viên (Admin)

- Quản lý người dùng: xem danh sách, tìm kiếm, kích hoạt/vô hiệu hóa tài khoản.
- Xem chi tiết người dùng trong Firestore (email, tuổi, giới tính, trạng thái, ngày tạo,...).
- Xóa tài khoản người dùng.



Cấu trúc cơ sở dữ liệu trong Firebase



- Trong Firestore Database, mỗi user (tương ứng với một tài khoản Email) sẽ được phân biệt bằng UID (User ID) duy nhất.
- Mỗi UID chứa dữ liệu bao gồm các **trường (fields)** và giá trị tương ứng của chúng, được tổ chức dưới dạng key-value.



Hiển thị thông tin từ Firestore Database

- Thêm các gói cần thiết trong `pubspec.yaml`:

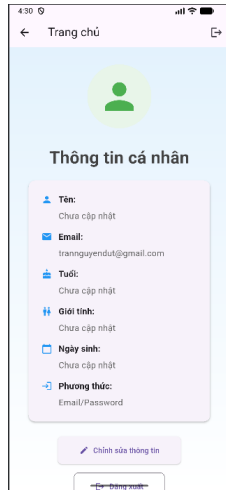
```
1 dependencies:  
2   flutter:  
3     sdk: flutter  
4   firebase_auth: ^5.7.0  
5   cloud_firestore: ^5.6.0  
6
```

- Các cách lấy thông tin người dùng hiện tại:
 - Qua các luồng (streams): `authStateChanges()`, `idTokenChanges()`, `userChanges()`
 - Qua đối tượng trả về từ phương thức đăng nhập: `UserCredential.user`
 - Qua thuộc tính hiện tại của `FirebaseAuth`:
`FirebaseAuth.instance.currentUser`
 - Thông qua Firestore Database:
`_firestore.collection('users').doc(uid).get()`,
`_firestore.collection('users').doc(uid).snapshots()`



Hiển thị thông tin từ Firestore Database (Con't)

```
1 Future<void> _initializeUserData() async {  
2   setState(() => _isLoading = true);  
3   final user = FirebaseAuth.instance.currentUser;  
4   if (user != null) {  
5     try {  
6       final docRef = _firestore.collection('users').doc(  
7         user.uid);  
8       final doc = await docRef.get();  
9       if (!doc.exists) { //...  
10      } else {  
11        print('User document exists');  
12        final data = doc.data();  
13        setState(() {  
14          _userData = data;  
15          _isAdmin = data?['isAdmin'] == true;  
16          _isLoading = false;  
17        });  
18      } catch (e) { //...  
19      }  
20    } else {  
21      setState(() => _isLoading = false);  
22    }  
23  }
```



Cập nhật thông tin lên Firestore Database

```
1 Future<void> _saveChanges() async {  
2   setState(() => _isLoading = true);  
3   try {  
4     final user = FirebaseAuth.instance.currentUser;  
5     final updateData = {  
6       'displayName': _nameController.text.trim(),  
7       'age': _ageController.text.trim().isNotEmpty  
8         ? int.tryParse(_ageController.text.trim())  
9         : null,  
10      'gender': _selectedGender,  
11      'birthDate': _birthDateController.text.trim().  
12        isEmpty  
13        ? _birthDateController.text.trim()  
14        : null,  
15      'updatedAt': FieldValue.serverTimestamp(),  
16    };  
17    await _firestore.collection('users').doc(user.uid).  
18      update(updateData);  
19  } catch (e) { //...  
20  }
```

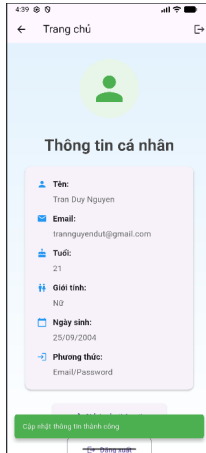
The screenshot shows a mobile app interface titled "Chỉnh sửa thông tin" (Edit Information). It features a light blue background with a white card containing the following elements:

- Thông tin cá nhân** (Personal Information): A section header.
- Họ và tên** (Full Name): A text input field with a person icon.
- Tuổi** (Age): A text input field with a calendar icon.
- Giới tính** (Gender): A dropdown menu with a person icon and a downward arrow.
- Ngày sinh** (Date of Birth): A dropdown menu with a calendar icon and a downward arrow.
- Đổi mật khẩu** (Change Password): A toggle switch.
- Lưu thay đổi** (Save Changes): A button.
- Hủy** (Cancel): A button.

The status bar at the top shows the time 4:36, signal strength, and battery level.



Cập nhật thông tin lên Firestore Database (Con't)



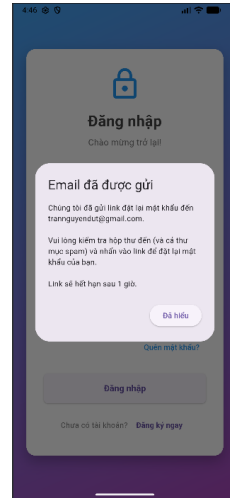
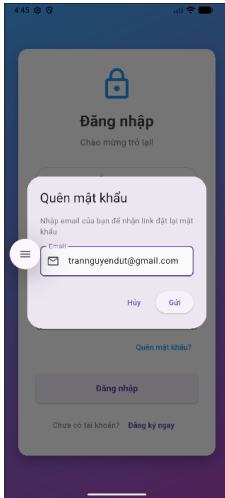
Quên mật khẩu

```
1 Future<void> _showForgotPasswordDialog()  
2   async {  
3     final emailController =  
4       TextEditingController();  
5     final formKey = GlobalKey<FormState>()  
6     ;  
7     final result = await showDialog<bool>(  
8       context: context,  
9       builder: (context) => AlertDialog(  
10        title: Text('Quên mật khẩu'),  
11      ),  
12      //...  
13    );  
14  
15    if (result == true) {  
16      await _sendPasswordResetEmail(  
17        emailController.text.trim());  
18    }  
19    // ...  
20  }
```

```
1 Future<void> _sendPasswordResetEmail(  
2   String email) async {  
3   try {  
4     await FirebaseAuth.instance.  
5       sendPasswordResetEmail(email:  
6         email);  
7     _showDialog(  
8       title: 'Email đã được gửi',  
9       // ...  
10    );  
11    } on FirebaseAuthException catch (e) {  
12      //...  
13    } catch (e) { //...  
14    }  
15  }
```



Quên mật khẩu (Con't)



Quên mật khẩu (Con't)

Reset your password for project-474081122253 Spam x



noreply@fir-auth-2626d.firebaseio.com

to me ▾

4:46 PM (1 minute ago)



Why is this message in spam? This message was marked as spam because previous messages from firebaseapp.com were marked as spam.

Report not spam



Hello,

Follow this link to reset your project-474081122253 password for your trannguyendut@gmail.com account.

https://fir-auth-2626d.firebaseio.com/_/auth/v1/action?mode=reset&passwordResetCode=xjFIRwbQbYCEPwYn75Ame_45qFlu158HG1MPC7QbHsaoAAAGakHb9kaQKeyzAlzaSyC425d07C0ZpoIc0y25UTXxy56Q8LMiangm0

If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your project-474081122253 team

↩ Reply

➦ Forward



Reset your password

for trannguyendut@gmail.com

New password



SAVE



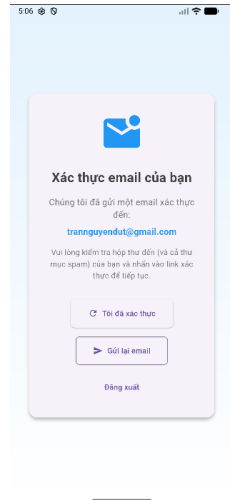
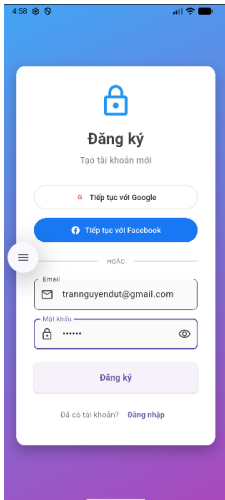
Xác minh Email

```
1 final userCredential = await
  FirebaseAuth.instance.
    createUserWithEmailAndPassword(
2     email: _emailController.text.trim(),
3     password: _passwordController.text,
4   );
5
6 if (userCredential.user != null) {
7   await userCredential.user!.
    sendEmailVerification();
8   setState(() {
9     _isWaitingForVerification = true;
10    _isLoading = false;
11  });
12 }
13
```

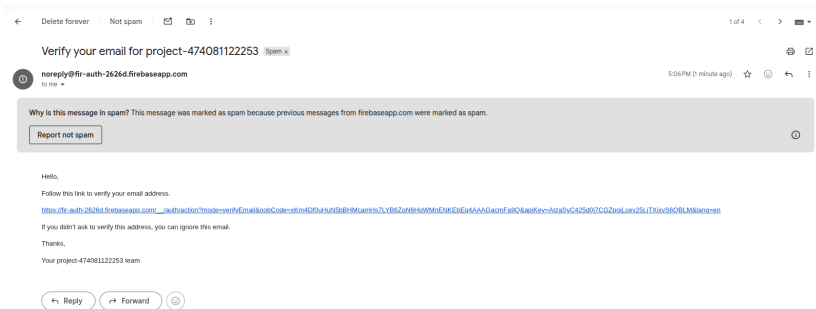
```
1 ElevatedButton.icon(
2   onPressed: () async {
3     await user.reload();
4     final currentUser = FirebaseAuth.
      instance.currentUser;
5     if (currentUser != null &&
      currentUser.emailVerified) {
6       ScaffoldMessenger.of(context).
        showSnackBar(
7         SnackBar(
8           content: Text('Email đã được
          xác thực!'),
9           backgroundColor: Colors.green,
10          ),
11        );
12        setState(() {});
13      } else { //.. send error
14      }
15    },
16    icon: Icon(Icons.refresh),
17    label: Text('Tôi đã xác thực'),
18    // ...
19  ),
20
```



Xác minh Email (Con't)



Xác minh Email (Con't)



Your email has been verified

You can now sign in with your new account

Table of Contents

- 1 Overview
- 2 Email/Password login
- 3 Social Login
 - Google Login
 - Facebook Login
- 4 User Management
- 5 Security
- 6 Tài liệu tham khảo



Security

- **Firebase Authentication** chỉ hỗ trợ việc đăng nhập và xác thực người dùng.
- Để bảo vệ dữ liệu an toàn, cần kết hợp với **Firestore Database** và thiết lập **Security Rules** nhằm:
 - Phân quyền giữa người dùng thường và quản trị viên
 - Đảm bảo xác thực an toàn qua email hoặc các phương thức khác
 - Quản lý trạng thái, cấp quyền hoặc xóa người dùng khi cần thiết



Phân quyền người dùng trong Firestore

- **Mục tiêu:** Chỉ cho phép người có quyền truy cập hoặc chỉnh sửa dữ liệu phù hợp.
- **Cách cấu hình:**
Vào *Firebase Console* → *Firestore Database* → *Rules* để thiết lập quy tắc cho cơ sở dữ liệu.

```
1 rules_version = '2';
2 service cloud.firestore {
3   match /databases/{database}/documents {
4     function isAdmin() {
5       return request.auth != null &&
6         get(/databases/{database}/documents/
7           users/{request.auth.uid})
8           .data.isAdmin == true;
9     }
10    function isSignedIn() {
11      return request.auth != null;
12    }
13    function isOwner(userId) {
14      return request.auth != null && request.auth.
15        uid == userId;
16    }
17    match /users/{userId} {
18      allow read: if isOwner(userId) || isAdmin();
19      allow create: if isSignedIn() && isOwner(
20        userId);
21      allow update: if isOwner(userId) || isAdmin();
22      allow delete: if isAdmin();
23    }
24  }
25 }
```



Phân quyền người dùng trong Firestore (Con't)

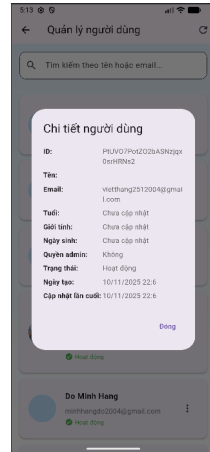
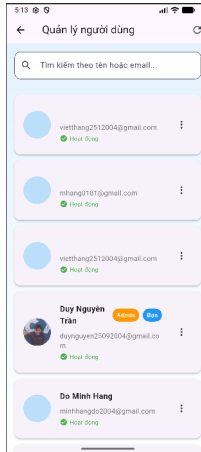
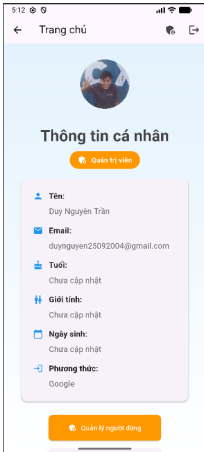
- Sau khi thiết lập các Rules trong database xong, ta sang mục Data để thay đổi quyền của người dùng.

The screenshot shows the Google Cloud Firestore console. The top navigation bar includes tabs for Data, Rules, Indexes, Disaster Recovery, Usage, and Extensions. A security warning is visible: "Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing". The left sidebar shows the "users" collection. The main area displays a list of documents, with one document selected, showing its fields: age, birthDate, createdAt, displayName, email, gender, isActive, isAdmin, photoURL, and updatedAt.

Collection	Document ID	Fields
users	3G31o1gR4dP1yud30x1NqP0fSMh1	
	0vJZR6g21nVpUyAUCYf1DoxYWe02	
	PTUV07PotZ02bASnzJqx0srRHRNz2	
	d2IFYjSveNbKx6bJ34rzTEcmEDD3	age: null birthDate: null createdAt: November 10, 2025 at 9:48:57 PM UTC+7 displayName: "Duy Nguyễn Trần" email: "duynguyen25092004@gmail.com" gender: null isActive: true isAdmin: true photoURL: "https://lh3.googleusercontent.com/a/ACg8ocLZ5QP8_Wd8JXEozGK8ZnUSDMMWMyH1t9jlyhtc" updatedAt: November 10, 2025 at 9:48:57 PM UTC+7
	fcZUPeV1E1d6sLgF6Y3G9KPT0xM1	



Demo phân quyền người dùng



Vô hiệu hóa và xóa tài khoản

- Để xóa 1 dữ liệu trên Firebase, ta sử dụng phương thức:
`_firestore.collection('users').doc(userId).delete()`.

```
1 Future<void> _deleteUser(String userId,  
2 String email) async {  
3   final confirm = await showDialog<  
4     bool>(  
5     context: context,  
6     builder: (context) => AlertDialog(  
7       title: Text('Xác nhận xóa'),  
8       content: Text(  
9         'Bạn có chắc chưa?.',  
10      ),  
11      actions: [  
12        TextButton(  
13          onPressed: () => Navigator.  
pop(context, false),  
        child: Text('Hủy'),  
      ),  
    ],  
  ),  
  );  
}
```

```
1 ElevatedButton(  
2   onPressed: () => Navigator.  
3   pop(context, true),  
4   style: ElevatedButton.  
5   styleFrom(background-color: Colors.  
6   red),  
7   child: Text('Xóa'),  
8   ),  
9 );  
10 if (confirm != true) return;  
11 try {  
12   await _firestore.collection('users  
13   ').doc(userId).delete();  
14   _showSuccess('Đã xóa');  
15 } catch (e) {  
16   _showError('Lỗi: ${e.toString()}')  
  }  
}
```



Security Best Practices

- **Xác thực người dùng:** Kiểm tra xác thực bằng email cho các hoạt động nhạy cảm (ví dụ: khôi phục mật khẩu) để tránh tạo tài khoản giả hoặc truy cập trái phép.
- **Phân quyền truy cập:** Thiết lập các *security rules* trong Firestore Database để giới hạn quyền truy cập và chỉnh sửa dữ liệu phù hợp với từng loại người dùng.
- **Quản lý tài khoản:** Quyền *xóa tài khoản* chỉ nên được cấp cho **admin** nhằm đảm bảo tính toàn vẹn và an toàn của dữ liệu.



Table of Contents

- 1 Overview
- 2 Email/Password login
- 3 Social Login
 - Google Login
 - Facebook Login
- 4 User Management
- 5 Security
- 6 Tài liệu tham khảo



- [1] Nguyễn Duy Nhật Viễn, *Slide môn học lập trình đa nền tảng*.
- [2] Google, "Firebase Authentication," *Firebase Documentation*, 2025.
- [3] P. X. Nam, "Tìm hiểu sơ lược về Firebase," *Viblo*, 2021.
- [4] GeeksforGeeks, "How to Add Firebase to Flutter App?" *GeeksforGeeks*, 2025.
- [5] Google, "Xác thực bằng Firebase bằng Tài khoản dựa trên mật khẩu trên Flutter," *Firebase Documentation*, 2025.
- [6] Google, "Firebase Login and Registration Authentication," *Viblo*, 2017.



THANK YOU FOR LISTENING

