

# Firebase Authentication

**Sinh viên:** Trần Việt Thắng

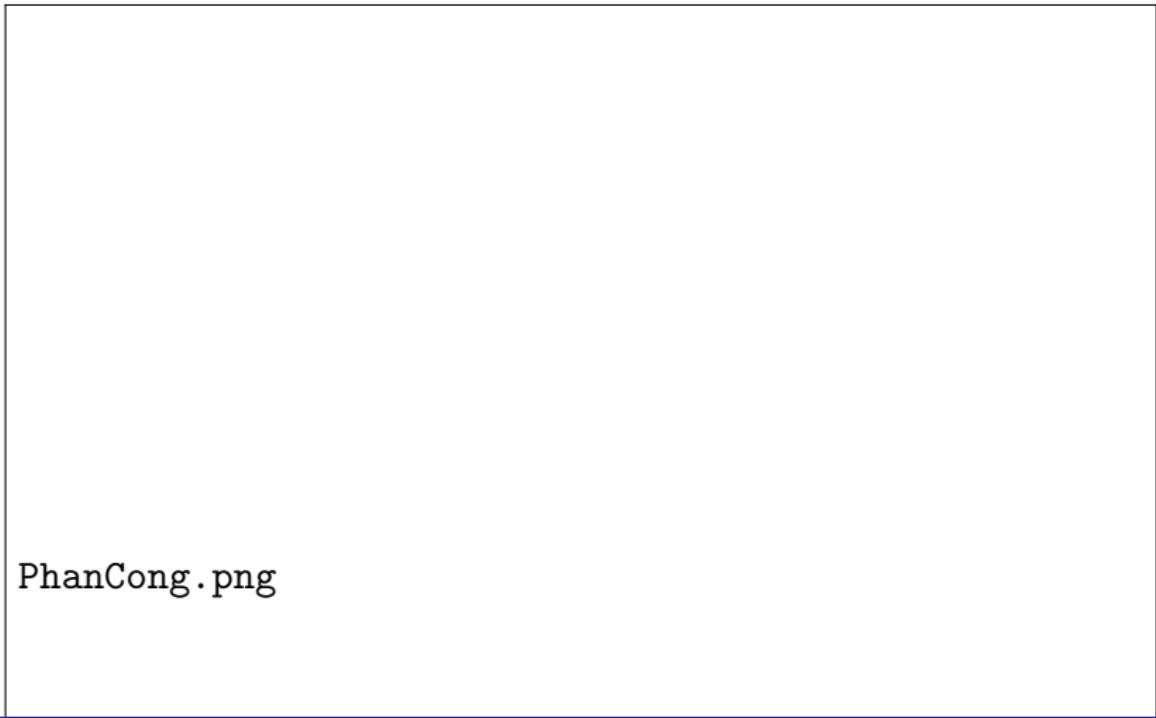
**Sinh viên:** Trần Duy Nguyên

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

Ngày 11 tháng 11 năm 2025



# Phân công công việc



PhanCong.png



# Outline

- ① Overview
- ② Email/Password login
- ③ Social Login
  - Google Login
- ④ User Management
- ⑤ Giới thiệu về Icons trong Flutter
- ⑥ Tài liệu tham khảo



## Table of Contents

- ① Overview
  - ② Email/Password login
  - ③ Social Login
    - Google Login
  - ④ User Management
  - ⑤ Giới thiệu về Icons trong Flutter
  - 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
- ④ User Management
- ⑤ Giới thiệu về Icons trong Flutter
- ⑥ Tài liệu tham khảo



# Bật quyền cho các nhà cung cấp

Sign-in providers

Add new provider

Provider	Status
Email/Password	Enabled
Google	Enabled
Facebook	Enabled

- Ta nhấn vào nút "Add new provider" ở mục signed-in provider.
- Chọn nhà cung cấp mong muốn và nhấn vào nút "Enable".
- Thành công sẽ cho ta thấy thanh "Status" như trên hình.
- 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 ký 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
```

- 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 );
```

- 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.signInWithEmailAndPassword(  
2   email: _emailController.text.trim(),  
3   password: _passwordController.text,  
4 );
```



# Đăng nhập và đăng ký bằng mail

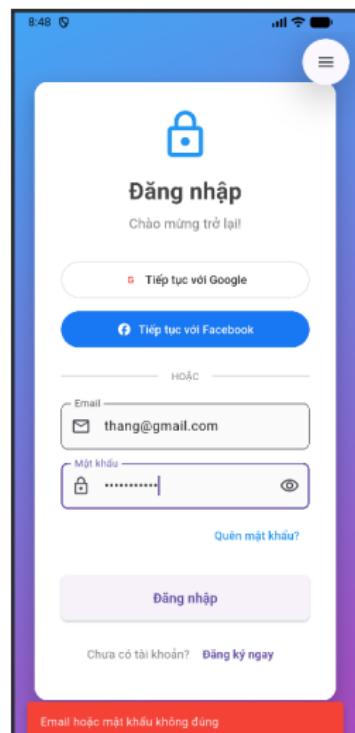
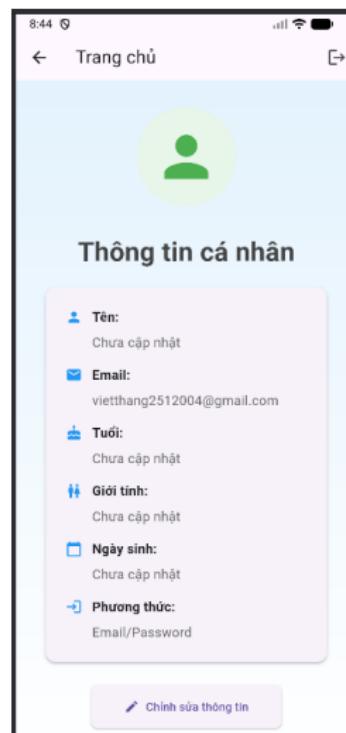
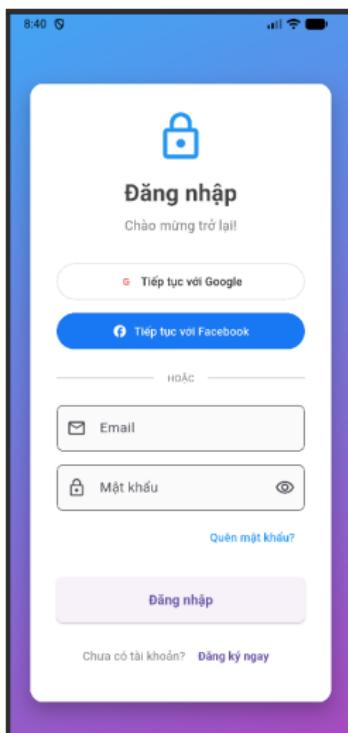
- Sử dụng try catch cho 2 hàm đăng nhập và đăng kí

```
1 try{  
2   //  
3 } on FirebaseAuthException catch (e) {  
4   String message = 'Da xay ra loi';  
5   if (e.code == 'user-not-found') message = 'Khong tim thay tai khoan';  
6   else if (e.code == 'wrong-password') message = 'sai mat khau';  
7   // ...  
8   _showError(message);  
9 }
```

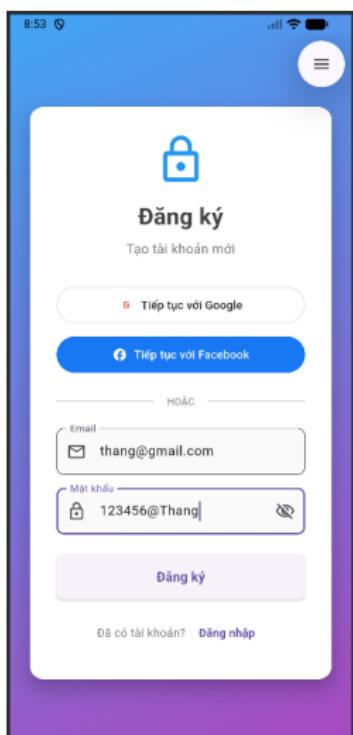
- FirebaseAuthException trả về các loại lỗi:
  - user-not-found: Email chưa được đăng ký.
  - wrong-password: Mật khẩu sai.
  - email-already-in-use: Email đã có tài khoản.
  - weak-password: Mật khẩu quá yếu.
  - invalid-email: Email sai định dạng.
  - invalid-credential: Thông tin không hợp lệ.



# Demo



# Demo



Identifier	Providers	Created ↓	Signed In	User UID
thang@gmail.com	✉	Nov 11, 2025	Nov 11, 2025	3AcNhuM27zTzqqglfOrZfxbH...
vietthang2512004@gm...	✉	Nov 10, 2025	Nov 11, 2025	3G31o1gR4dPlyud3OxiNqPDF...
duynguyen25092004@...	G	Nov 10, 2025	Nov 10, 2025	d2lFYjsveNbkh6bJ34rzTEcmE...
hellosigma@gmail.com	✉	Nov 10, 2025	Nov 10, 2025	mvFzrLedq20eoBZaAkJ2BfGe...
trannguyendt@gmail.c...	✉	Nov 10, 2025	Nov 10, 2025	tTsGNtoJvbfC8Rii1kle6SHlyr1
mhang0181@gmail.com	✉	Nov 10, 2025	Nov 10, 2025	GvJZR6g21InVpUyAUCYfiDoxY...
minhangdo2004@gm...	✉	Nov 10, 2025	Nov 10, 2025	fcZUPeYIE1d6sLgf6Y3G9KPT...
thangtran2512004@gm...	G	Nov 9, 2025	Nov 10, 2025	k8sapje7e2YB7xXHOcoC0j8m...



# Table of Contents

- ① Overview
- ② Email/Password login
- ③ Social Login
  - Google Login
- ④ User Management
- ⑤ Giới thiệu về Icons trong Flutter
- ⑥ Tài liệu tham khảo

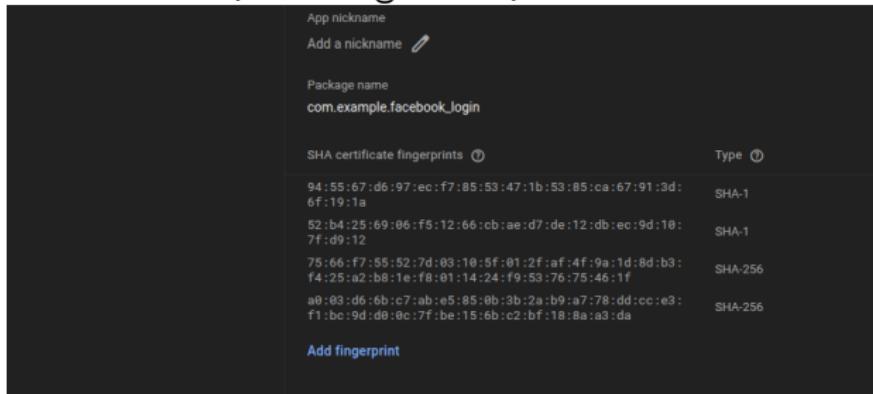


# Cấu hình trên Firebase

- 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
```

- Thêm vào mục setting của dự án Firebase:



# Table of Contents

- ① Overview
- ② Email/Password login
- ③ Social Login
  - Google Login
- ④ User Management
- ⑤ Giới thiệu về Icons trong Flutter
- 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 **Firebase Authentication**, nhà phát triển không cần tự xây dựng hệ thống xác thực và phân quyền từ đầu.
- Firebase 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

users	3G31o1gR4dPluyd30xiNqPDFsWh1
+ Add document	+ Start collection
3G31o1gR4dPluyd30xiNqPDFsWh1	+ Add field
GvJZR6g21nVpUyAUCYF1doxYWe82	age: null
PtUV07PotZ02bASNzjqxBsrIRNs2	birthDate: null
d2fIFYjSveNbKx6bJ34rzTEcmEDD3	createdAt: November 10, 2025 at 10:07:27 PM UTC+7
fcZUPeY1E1d6sLgfY3G9KPToXm1	displayName: "
k8sapje7e2YB7xKH0coC0j8e2yE2	email: "vietthang2512004@gmail.com"
tTsGntoJjVbFC8Rii1kie6SHiyr1	gender: null
	isActive: true
	isAdmin: false
	photoURL: "
	updatedAt: November 10, 2025 at 10:07:27 PM UTC+7

- 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
```

- 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



# Error handling

- **errorBuilder:** Hàm được xây dựng trong flutter, sẽ được gọi trong quá trình tải ảnh nếu có lỗi xảy ra.

```
1     ImageErrorWidgetBuilder = Widget Function(  
2       BuildContext context,  
3       Object error,  
4       StackTrace? stackTrace)
```

- Đối số error chứa các thông tin về lỗi
- Đối số stackTrace giúp truyền đạt cho người dùng về các lệnh đã tạo ra ngoại lệ.



# Demo Error handling

- Source code error handling (thay bằng URL không có thật):

```
1 void main() {  
2   runApp(  
3     MaterialApp(  
4       home: Scaffold(  
5         body: Center(  
6           child: CachedNetworkImage(  
7             imageUrl: 'https://picsum.photos/1500/1000' ,  
8             errorWidget: (context, url, error) => Center(  
9               child: Icon(Icons.error, color: Colors.red,  
10                  size: 100,)  
11                ),  
12              ),  
13            ),  
14          )  
15      );  
16 }
```



## Demo Error handling (Con't)

demoErrorHandler.png



# Table of Contents

- 1 Overview
- 2 Email/Password login
- 3 Social Login
  - Google Login
- 4 User Management
- 5 Giới thiệu về Icons trong Flutter
- 6 Tài liệu tham khảo



# Icon Widget

- Để có thể hiện thị biểu tượng trong Flutter, ta sử dụng Icon Widget. Có thể tìm thấy các Icons có sẵn của Flutter ở trang web : [Google Fonts](#).
- Flutter cung cấp 2 thư viện để truy cập Icon:
  - Material Icons
  - Cupertino Icons
- Để có thể sử dụng biểu tượng Icon của thư viện Cupertino, ta cần phải thêm gói *cupertino\_icons* vào tệp *pubspec.yaml*.

```
1 dependencies:  
2   cupertino_icons: ^1.0.8
```

- Đối với thư viện Material Icons, ta phải đảm bảo gói *uses-material-design* được chỉ định trong tệp *pubspec.yaml*.

```
1 flutter:  
2   uses-material-design: true
```

# Icon Widget

- Sau khi đã chỉ định các gói của material và cupertino trong tệp *pubspec.yaml*
- ta cần "import" 2 gói này vào mã nguồn để sử dụng

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter/cupertino.dart';
```

# Demo code sử dụng Material Icons

```
1 void main() {  
2     runApp(  
3         MaterialApp(  
4             home: Scaffold(  
5                 body: Center(  
6                     child: Row(  
7                         mainAxisAlignment: MainAxisAlignment.center,  
8                         children: [  
9                             Icon(  
10                                Icons.mail, color: Colors.blue, size: 50.0,),  
11                             Icon(  
12                                Icons.mail, color: Colors.red, size: 30.0,),  
13                             ],  
14                         ),  
15                         ),  
16                         ),  
17                         ),  
18                     ),  
19             );  
}
```



# Demo code sử dụng Material Icons (Con't)

 images/MaterialIcon\_Demo.png



# Demo code sử dụng Cupertino Icons

```
1 void main() {  
2     runApp( MaterialApp(  
3         home: Scaffold(  
4             body: Center(  
5                 child: Row(  
6                     mainAxisAlignment: MainAxisAlignment.center,  
7                     children: [  
8                         Icon(  
9                             CupertinoIcons.mail, color: Colors.blue, size:  
10                                : 50.0,),  
11                         Icon(  
12                             CupertinoIcons.mail, color: Colors.red, size:  
13                                : 30.0,),  
14                         ],  
15                         ),  
16                         ),  
17                         ),  
18                     );  
}
```

## Demo code sử dụng Cupertino Icons (Con't)

 images/cupertinoIcons\_demo.png

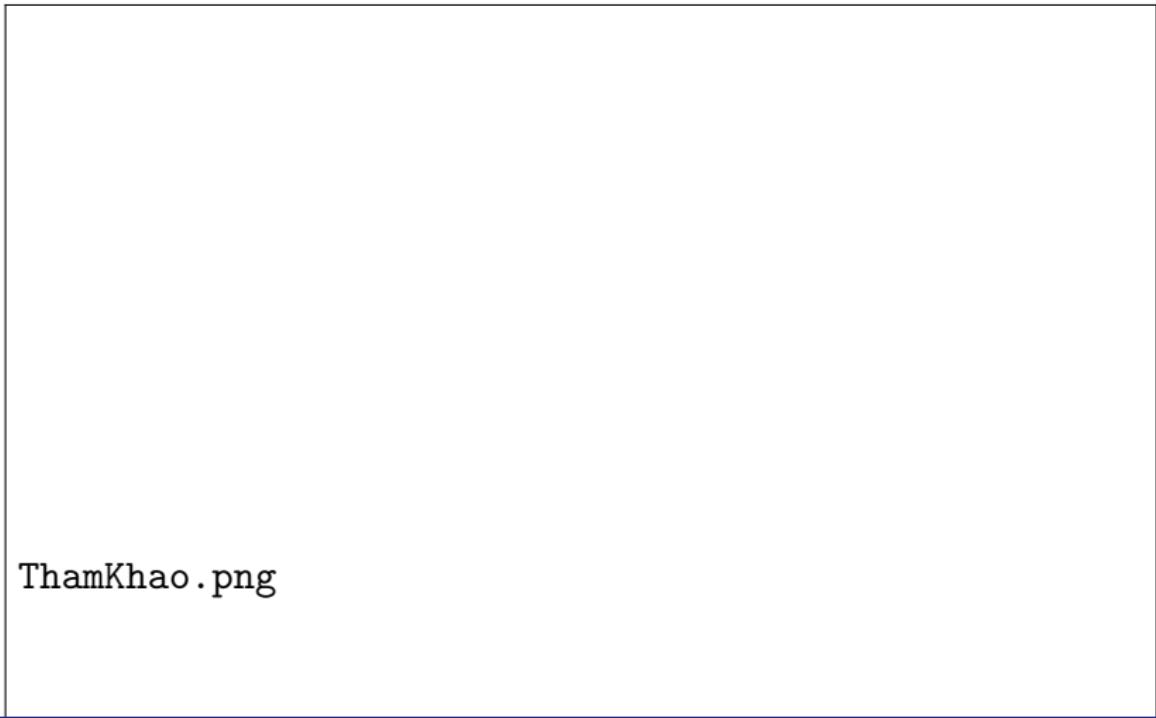


# Table of Contents

- ① Overview
- ② Email/Password login
- ③ Social Login
  - Google Login
- ④ User Management
- ⑤ Giới thiệu về Icons trong Flutter
- ⑥ Tài liệu tham khảo



# Tài liệu tham khảo



ThamKhao.png



# THANK YOU FOR LISTENING

