

**TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN TỬ - VIỄN THÔNG**

Báo cáo Tuần 15
Môn Lập Trình Đa Nền Tảng

Firebase Cloud Firestore

SVTH: 1. Lê Văn Minh Trí
2. Nguyễn Phan Hiếu Minh

Đà Nẵng, 2025

Nội Dung

Mô tả: Sử dụng Firestore cho NoSQL database với Flutter

1. Setup Firestore và authentication
2. Complex queries với where, orderBy, limit
3. Real-time listeners và offline support
4. Pagination và data modeling

Phân công công việc

TT	Họ và Tên	Nội dung	Tỷ lệ đóng góp
1	Lê Văn Minh Trí	Làm Slide, tạo code Real-time listeners và offline support; Pagination và data modeling	50%
2	Nguyễn Phan Hiếu Minh	Làm Slide, tạo code Setup Firestore và authentication; Complex queries với where, orderBy, limit	50%

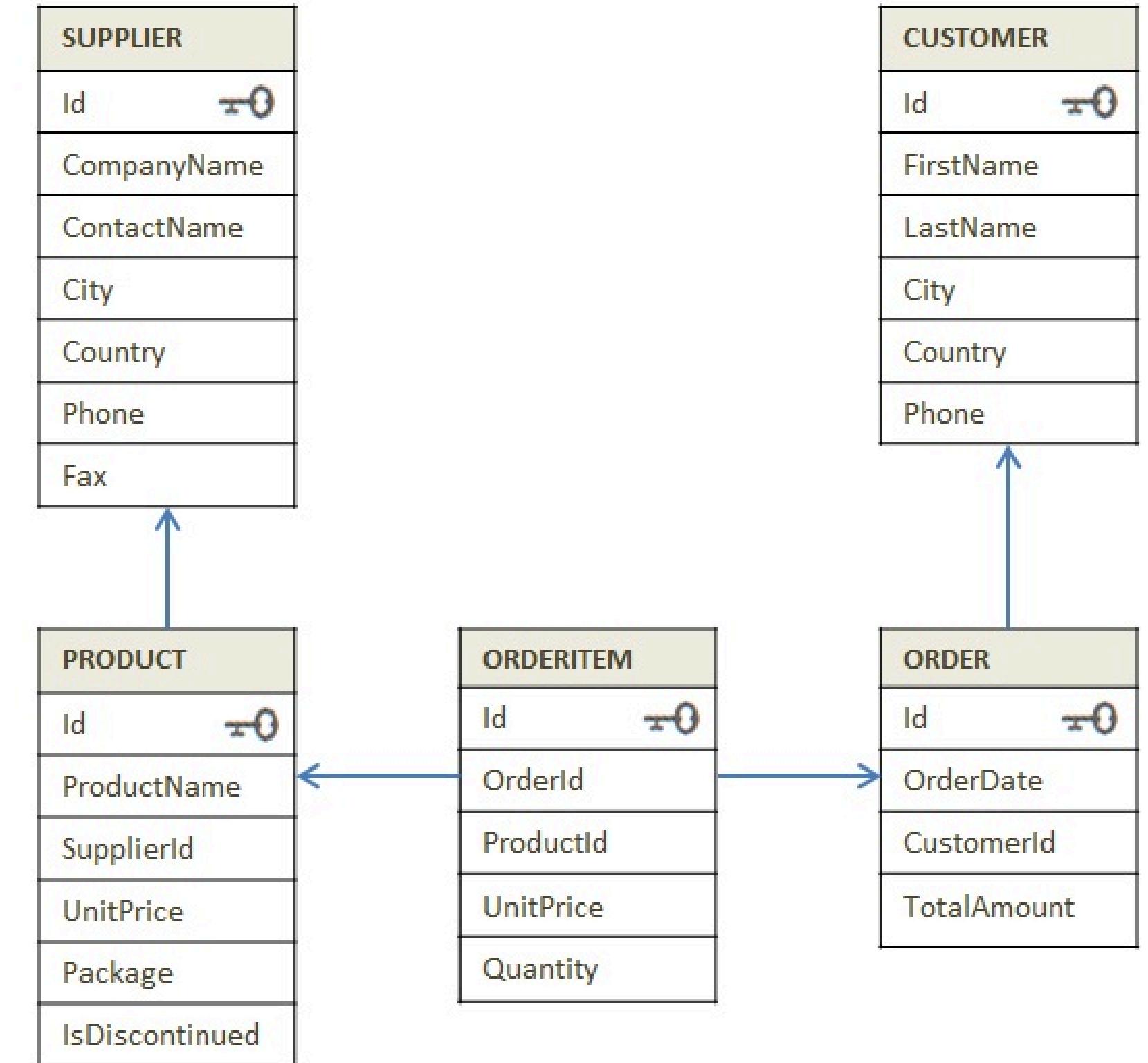
1. Setup Firestore và authentication

SQL Database là gì ?

-SQL Database hay còn gọi là cơ sở dữ liệu quan hệ, là hệ thống lưu trữ dữ liệu có cấu trúc dưới dạng các bảng, mỗi bảng bao gồm các hàng (bản ghi) và cột (thuộc tính)

-Đây là dạng cơ sở dữ liệu truyền thống với các trường là dữ liệu độc nhất, không có sự trùng lặp với nhau

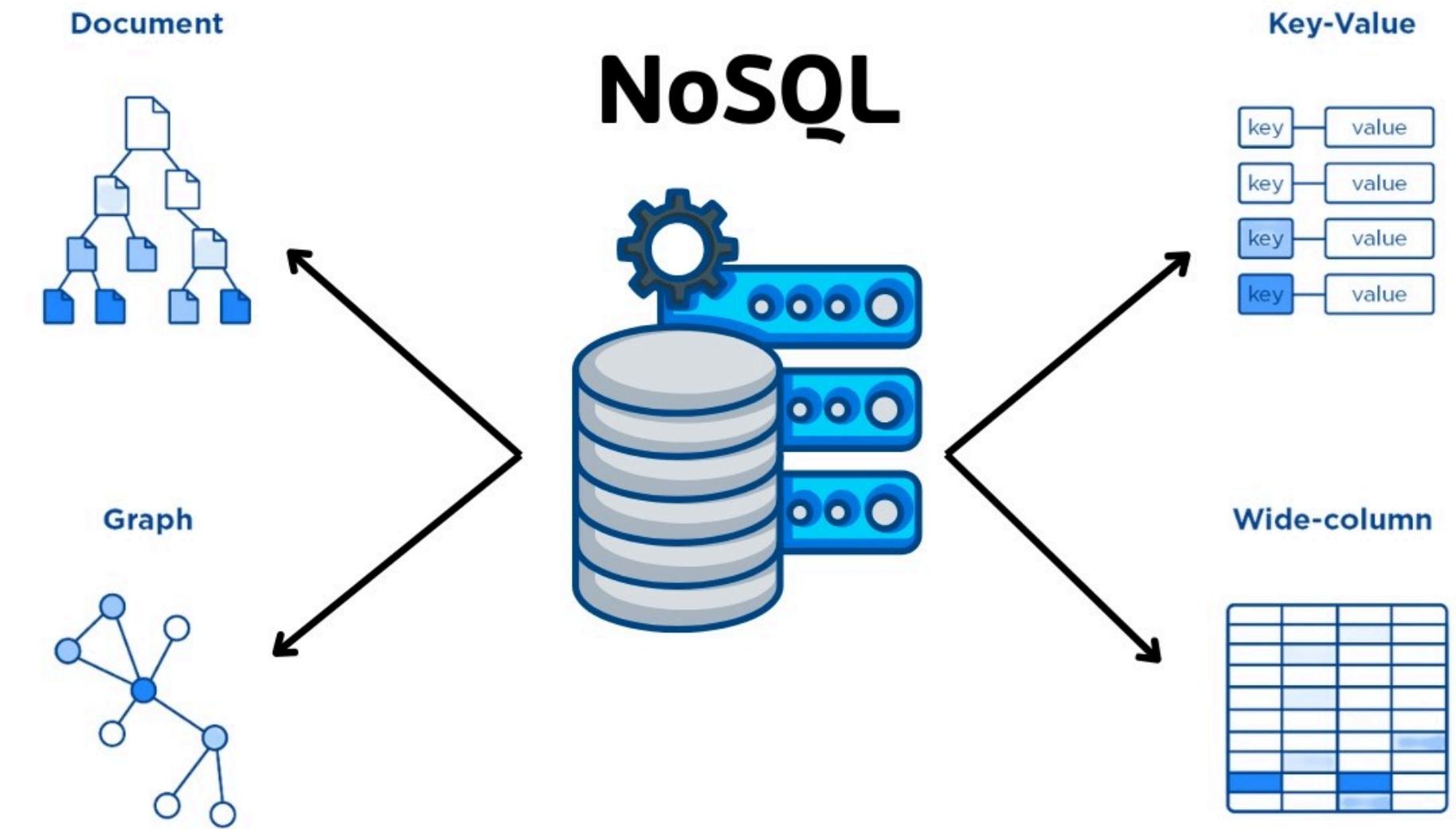
→ Điều này dẫn đến việc khó khăn khi đọc dữ liệu từ nhiều bảng khác nhau



1. Setup Firestore và authentication

NoSQL Database là gì ?

- NoSQL Database là loại cơ sở dữ liệu phi quan hệ, không yêu cầu lược đồ cố định, được thiết kế để quản lý và lưu trữ lượng lớn dữ liệu phi cấu trúc hoặc bán cấu trúc, với khả năng mở rộng và hiệu suất cao
- Đây là dạng cơ sở dữ liệu mới hỗ trợ nhiều mô hình dữ liệu linh hoạt như khóa-giá trị, tài liệu, cột, và đồ thị.
- Điều này dẫn đến việc khó khăn khi chỉnh sửa dữ liệu trong database



1. Setup Firestore và authentication

SQL hay NoSQL ?

-SQL hay NoSQL đều có những ưu điểm riêng tuy nhiên đa số ứng dụng trên thị trường hiện nay đều có tỉ lệ đọc nhiều hơn viết rất nhiều.

-Vì vậy từ những lợi ích đó thì NoSQL database đang phát triển cực kì mạnh mẽ



1. Setup Firestore và authentication

Firebase cloud Firestore



- Một trong những cloud NoSQL database phổ biến nhất hiện nay là Firebase cloud Firestore
- Được sử dụng rộng rãi với Webapp hay android/ios

1. Setup Firestore và authentication

Firebase cloud Firestore

- Firestore lưu trữ dữ liệu dưới dạng các Collections và Documents
- Các documents này bao gồm các cặp khóa-giá trị và có thể chứa bất kỳ loại dữ liệu nào, từ chuỗi đến dữ liệu nhị phân đến các đối tượng giống với cây JSON
- Các documents, lần lượt, được nhóm thành các collections

Collection

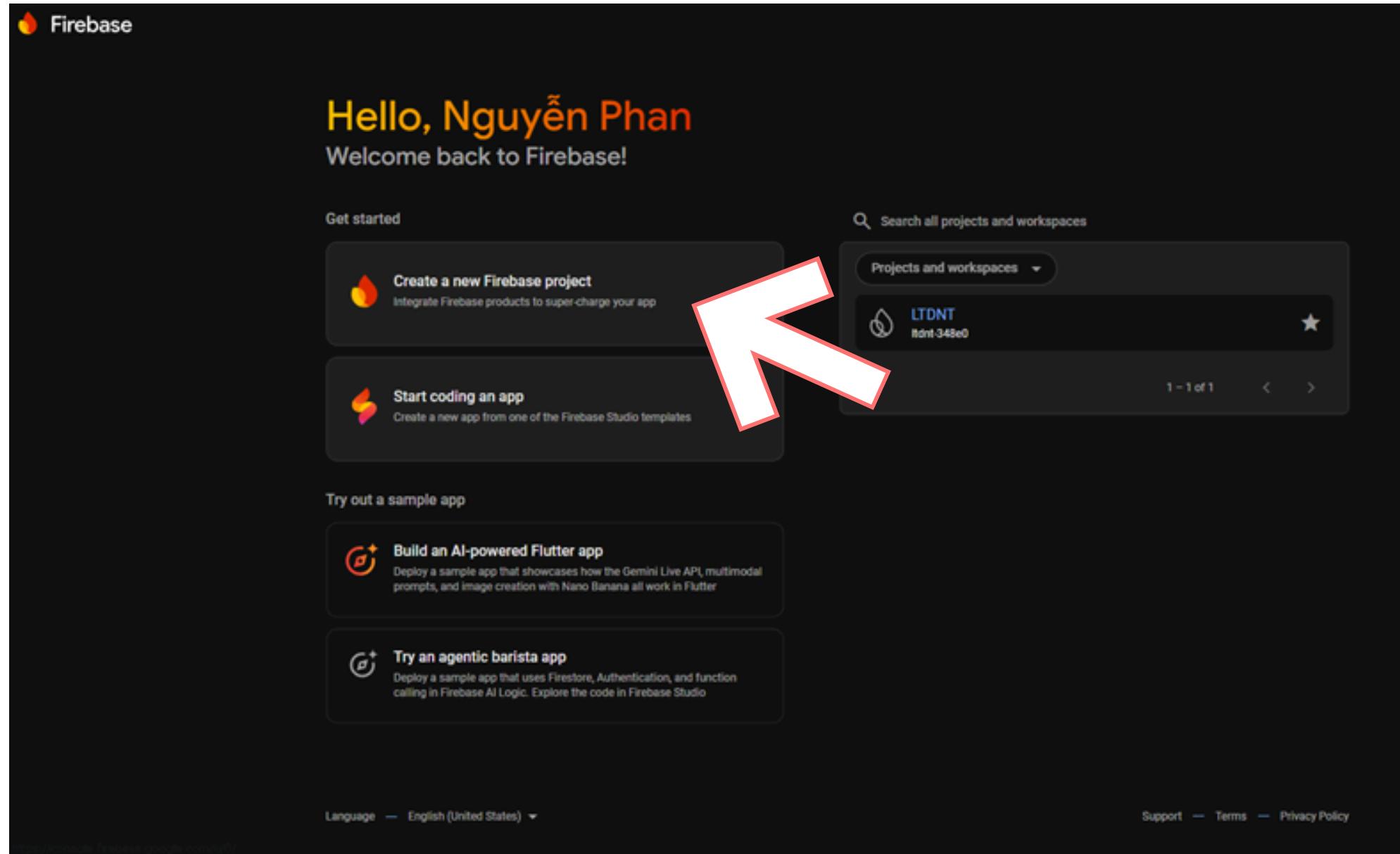
Document

Document

Document

1. Setup Firestore và authentication

Các bước Setup

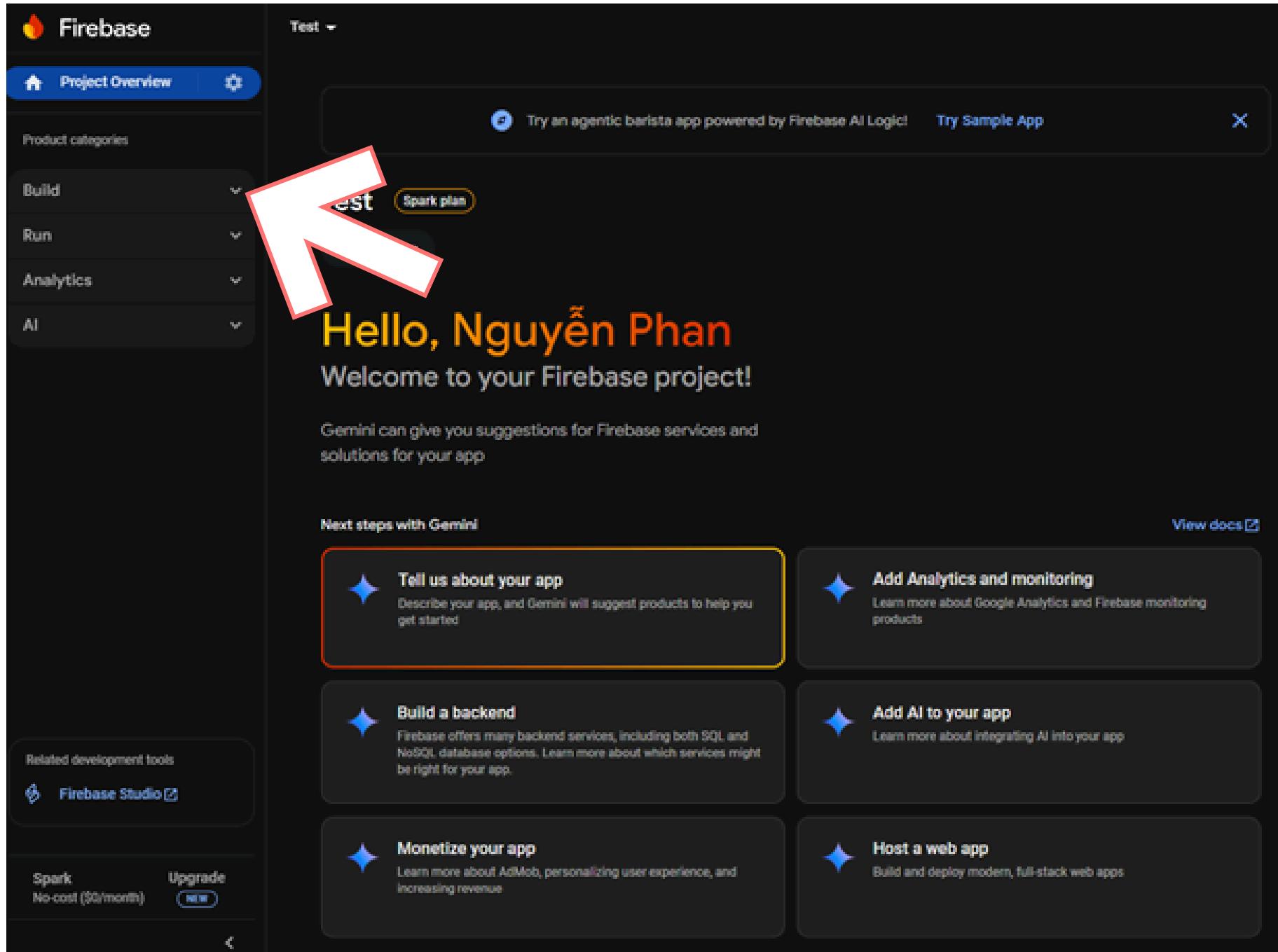


1/Đăng nhập vào Firebase

2/Tạo Project mới

1. Setup Firestore và authentication

Các bước Setup



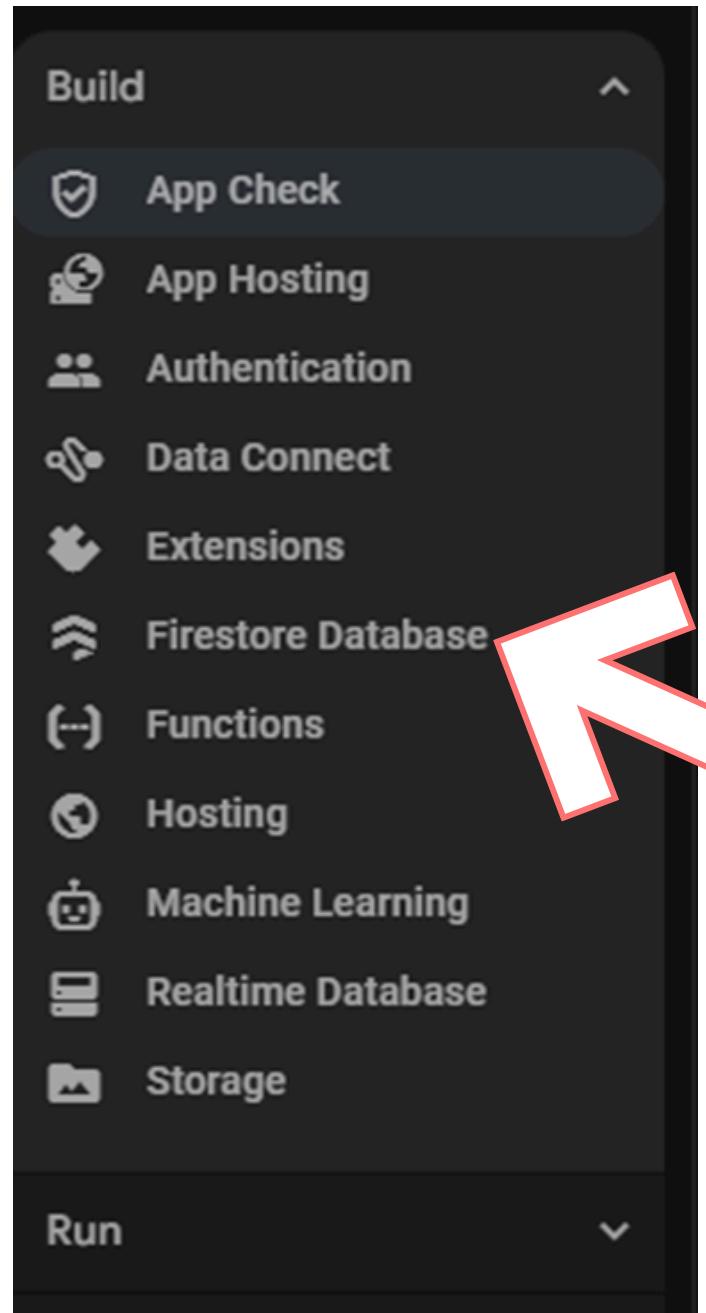
1/Đăng nhập vào Firebase

2/Tạo Project mới

3/Trong mục build chọn
Firestore Database

1. Setup Firestore và authentication

Các bước Setup



Firebase

Project Overview

Build

App Check

App Hosting

Authentication

Data Connect

Extensions

Firestore Database

Functions

Hosting

Machine Learning

Realtime Database

Storage

Run

Cloud Firestore

Realtime updates, powerful queries, automatic scaling, and MongoDB compatibility

Create database Ask Gemini

Learn more

How do I get started? View the docs

How much will Cloud Firestore cost? View pricing

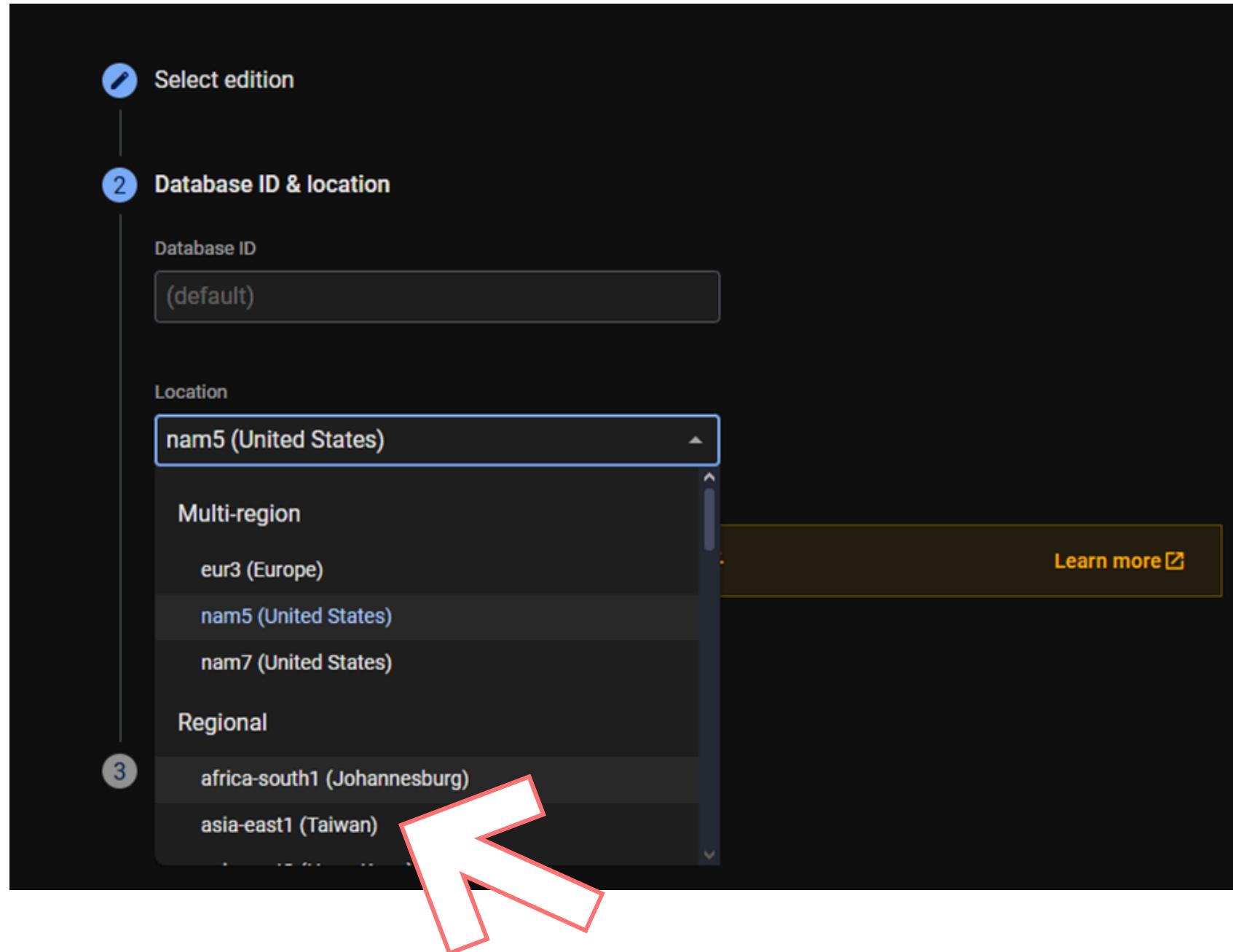
What can Cloud Firestore do for me? Learn more

Cloud Firestore

- 1/ Đăng nhập vào Firebase
- 2/ Tạo Project mới
- 3/ Trong mục build chọn Firestore Database
- 4/ Create Database và chọn vị trí của Database vật lý

1. Setup Firestore và authentication

Các bước Setup



1/Đăng nhập vào Firebase

2/Tạo Project mới

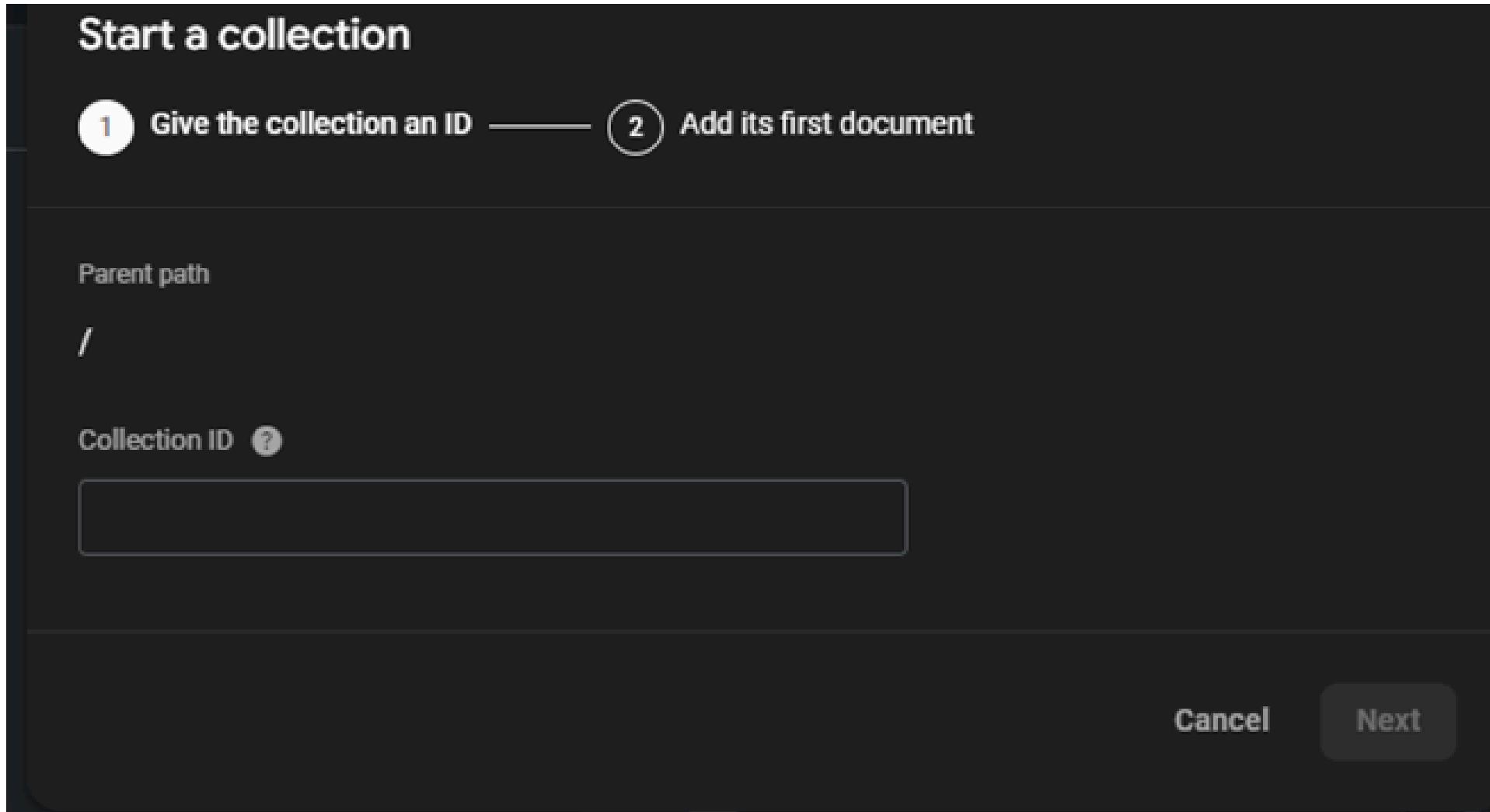
3/Trong mục build chọn
Firestore Database

4/Create Database và chọn
vị trí của Database vật lý

Lưu ý: Đây là vị trí vật lý của database nên chúng ta nên chọn địa điểm tối ưu, vì chúng ta ở Việt Nam nên chúng ta nên chọn vị trí ở Taiwan để có đường truyền tốt nhất

1. Setup Firestore và authentication

Các bước Setup



1/Đăng nhập vào Firebase

2/Tạo Project mới

3/Trong mục build chọn
Firestore Database

4/Create Database và chọn
vị trí của Database vật lý

5/Thêm dữ liệu vào
database thủ công

1. Setup Firestore và authentication

Các bước Setup

The screenshot shows the Firebase Firestore interface. On the left, there's a sidebar with a 'Start collection' button and a 'SinhVien' collection listed. The main area shows a list of document IDs. One document, 'NS1Ri9xU5HrOHLN1Bx7', is selected and expanded. It contains nested fields: 'address', 'email', 'name', and 'school'. The 'school' field is an array containing a single element.

Field	Type	Value
address	Object	city: "Hue" house number: "55" street: "Nguyen Hue" ward: "Phu Nhuan"
email	String	"ThiB@gmail.com"
name	String	"Tran Thi B"
school	Array	[{"address": "Hue", "name": "Hue University", "yearIn": 2021, "yearOut": 2026}]

1/Đăng nhập vào Firebase

2/Tạo Project mới

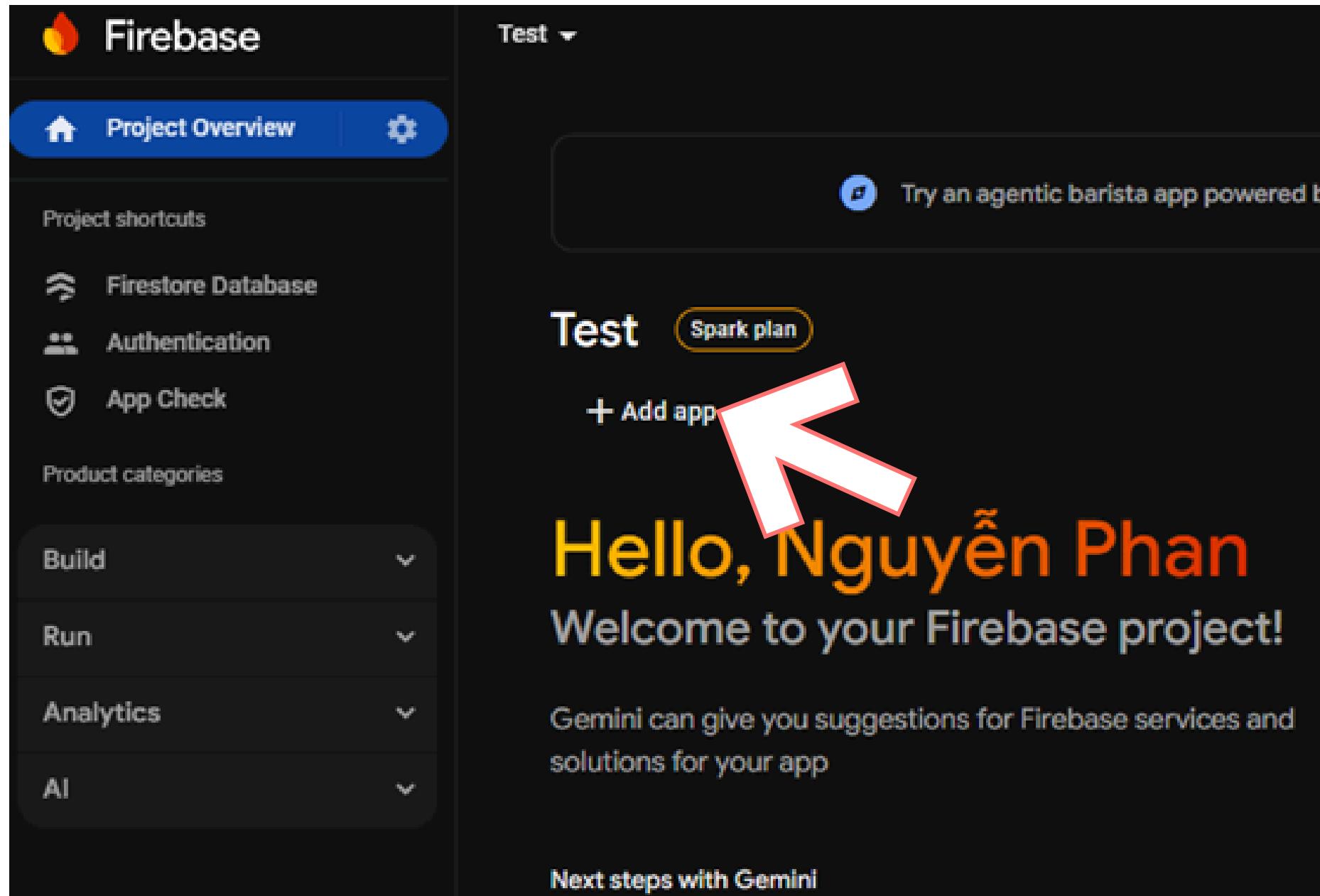
3/Trong mục build chọn
Firestore Database

4/Create Database và chọn
vị trí của Database vật lý

5/Thêm dữ liệu vào
database thủ công

1. Setup Firestore và authentication

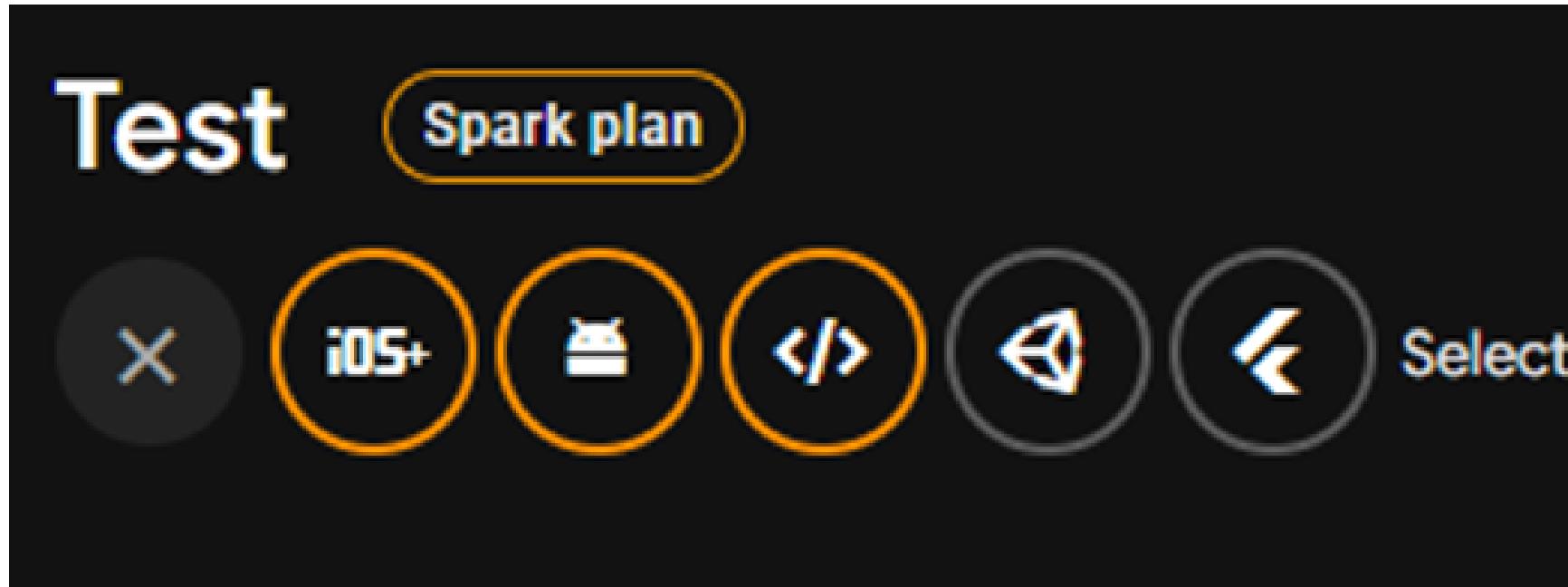
Các bước Setup



6/Kết nối app vào Firebase

1. Setup Firestore và authentication

Các bước Setup



6/Kết nối app vào Firebase
Package name là applicationId của file dự án ở
trong thư mục \android\app\build.gradle.kts

```
defaultConfig {  
    applicationId = "com.example.flutter_application_2_firebase"
```

× Add Firebase to your Android app

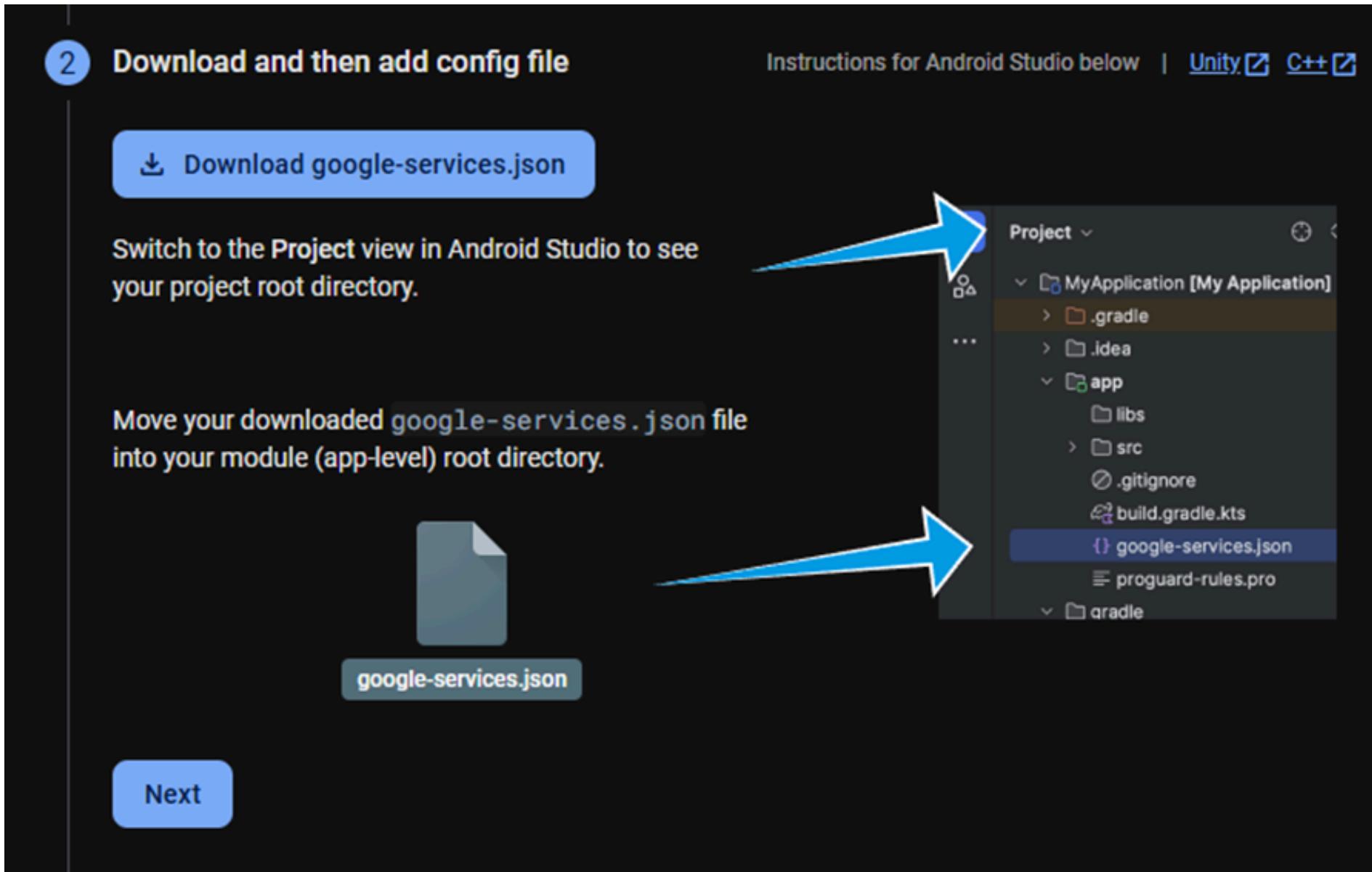
1 Register app

Android package name ?

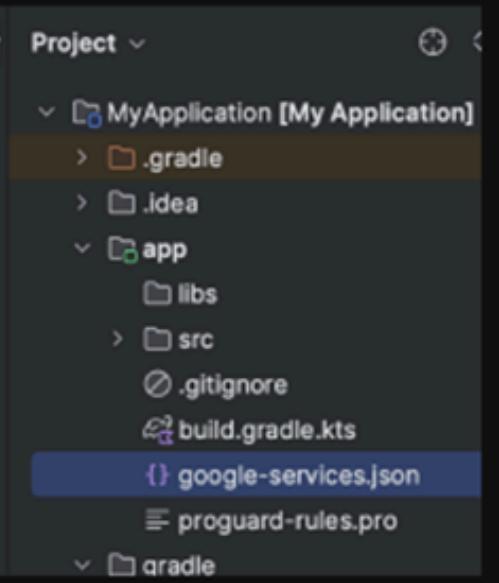
App nickname (optional) ?

1. Setup Firestore và authentication

Các bước Setup



7/Thêm file config vào dự án



1. Setup Firestore và authentication

Các bước Setup

The screenshot shows a step-by-step guide for adding the Firebase SDK. Step 3, titled "Add Firebase SDK", provides instructions for Gradle, Unity, and C++. It includes a note about using buildscript syntax for managing plugins and links to learn how to add Firebase plugins using that syntax. Below this, it explains that to make the google-services.json config values accessible to Firebase SDKs, you need the Google services Gradle plugin. Two options are shown: "Kotlin DSL (build.gradle.kts)" (selected) and "Groovy (build.gradle)". It then instructs to add the plugin as a dependency to the project-level build.gradle.kts file. A code snippet for the Root-level (project-level) Gradle file (<project>/build.gradle.kts) is provided:

```
plugins {
    // ...

    // Add the dependency for the Google services Gradle plugin
    id("com.google.gms.google-services") version "4.4.4" apply false
}
```

7/Thêm file config vào dự án
8/Thêm Firebase SDK

1. Setup Firestore và authentication

Các bước Setup

2. Then, in your **module (app-level)** `build.gradle.kts` file, add both the `google-services` plugin and any Firebase SDKs that you want to use in your app:

Module (app-level) Gradle file (`<project>/<app-module>/build.gradle.kts`):

```
plugins {
    id("com.android.application")
    // Add the Google services Gradle plugin
    id("com.google.gms.google-services")
    ...
}

dependencies {
    // Import the Firebase BoM
    implementation(platform("com.google.firebase:firebase-bom:34.5.0"))
    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation("com.google.firebase:firebase-analytics")

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

3. After adding the plugin and the desired SDKs, sync your Android project with Gradle files.

7/Thêm file config vào dự án

8/Thêm Firebase SDK và Plugins

dependencies:

flutter:

sdk: flutter

firebase_core: ^4.2.1

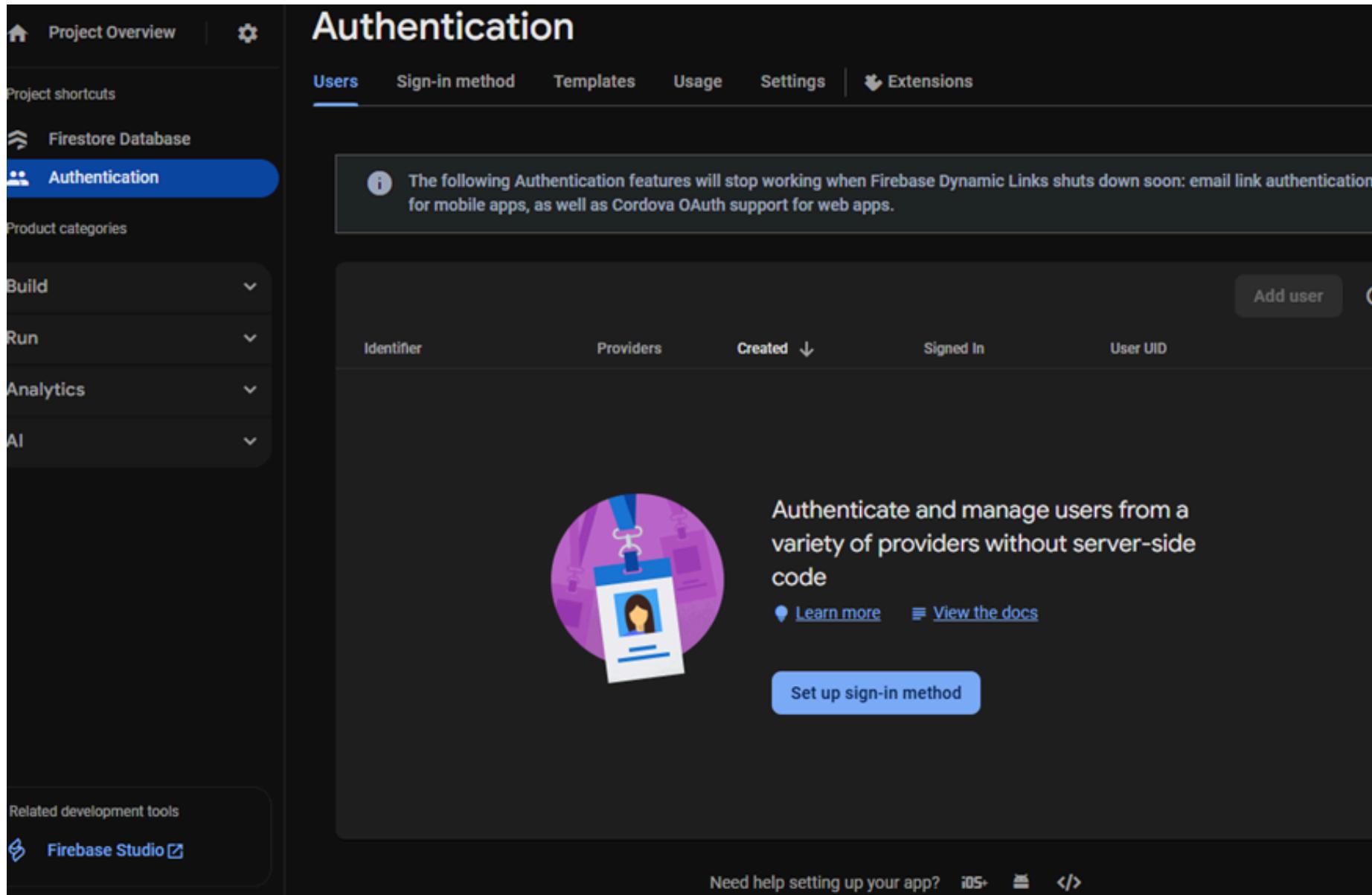
firebase_auth: ^6.1.2

cloud_firestore: ^6.1.0

9/Thêm các Package

1. Setup Firestore và authentication

Authentication

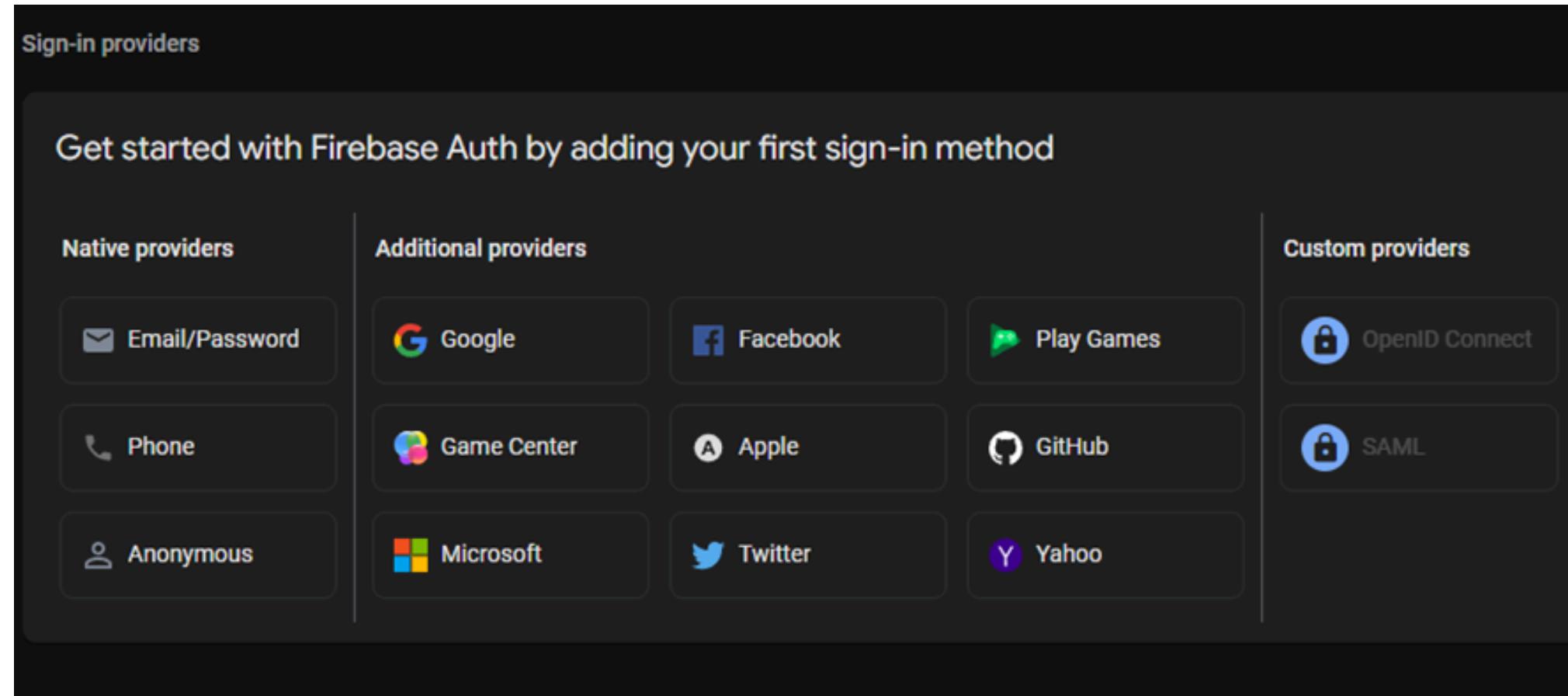


The screenshot shows the Firebase Authentication console. On the left, there's a sidebar with project settings like Project Overview, Firestore Database, Authentication (which is selected and highlighted in blue), Product categories, Build, Run, Analytics, and AI. The main area is titled 'Authentication' and contains a sub-section for 'Users'. It shows a table with columns: Identifier, Providers, Created (sorted by descending date), Signed In, and User UID. There's a button to 'Add user' and a 'C' icon for bulk actions. Below the table, there's a section titled 'Authenticate and manage users from a variety of providers without server-side code' with links to 'Learn more' and 'View the docs'. At the bottom, there's a button to 'Set up sign-in method'. The footer of the page includes links for 'Related development tools' (Firebase Studio) and help resources for iOS, Android, and web development.

Firebase Authentication là một dịch vụ cung cấp giải pháp xác thực người dùng an toàn và dễ sử dụng cho ứng dụng di động và web. Dịch vụ này hỗ trợ nhiều phương thức đăng nhập như email/mật khẩu, số điện thoại, và các nhà cung cấp bên thứ ba phổ biến như Google, Facebook, Twitter, và GitHub.

1. Setup Firestore và authentication

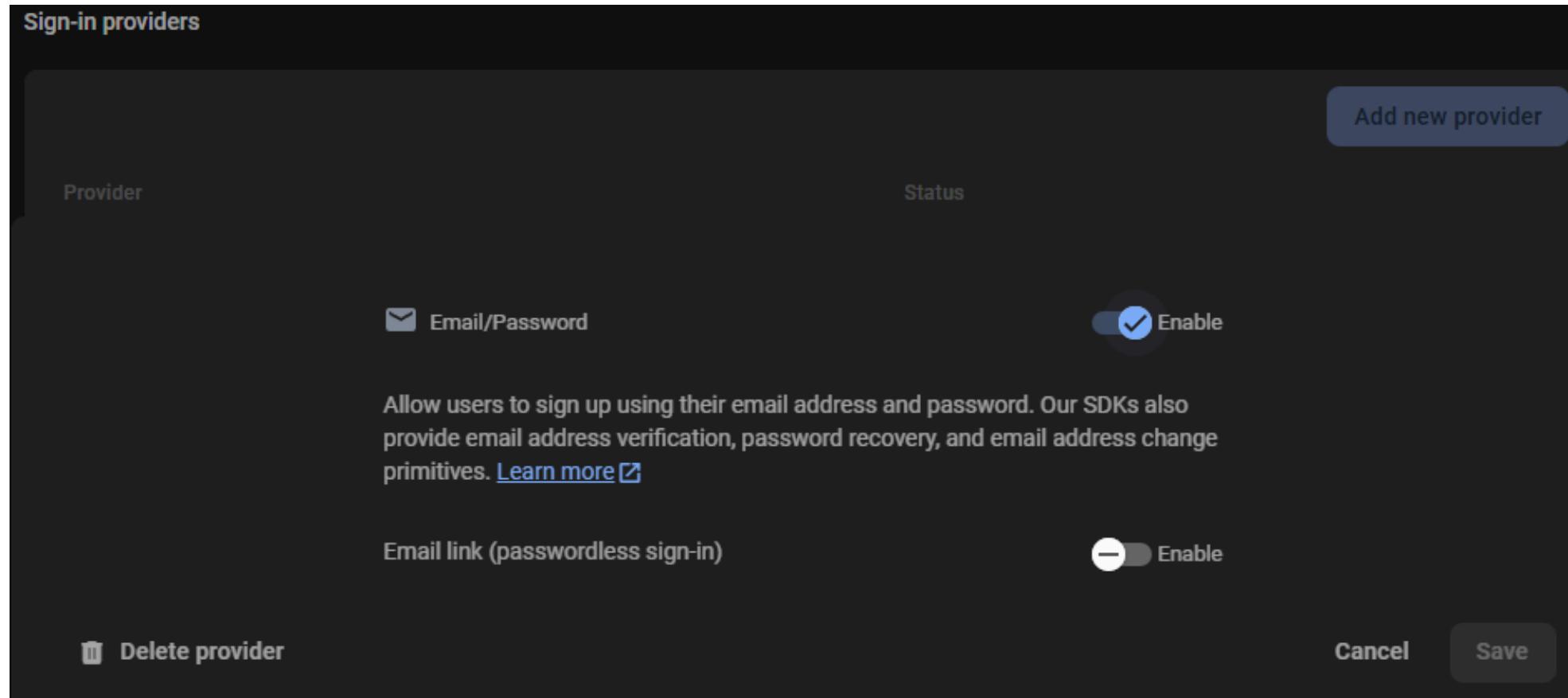
Authentication



Firebase Authentication là một dịch vụ cung cấp giải pháp xác thực người dùng an toàn và dễ sử dụng cho ứng dụng di động và web. Dịch vụ này hỗ trợ nhiều phương thức đăng nhập như email/mật khẩu, số điện thoại, và các nhà cung cấp bên thứ ba phổ biến như Google, Facebook, Twitter, và GitHub.

1. Setup Firestore và authentication

Authentication



Firebase Authentication là một dịch vụ cung cấp giải pháp xác thực người dùng an toàn và dễ sử dụng cho ứng dụng di động và web. Dịch vụ này hỗ trợ nhiều phương thức đăng nhập như email/mật khẩu, số điện thoại, và các nhà cung cấp bên thứ ba phổ biến như Google, Facebook, Twitter, và GitHub.

1. Setup Firestore và authentication

Authentication

The screenshot shows the Firebase Authentication 'Users' screen. At the top, there are tabs for 'Users', 'Sign-in method', 'Templates', 'Usage', 'Settings', and 'Extensions'. A warning message in a box states: 'The following Authentication features will stop working when Firebase Dynamic Links shuts down soon: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.' Below the tabs, there are columns for 'Identifier', 'Providers', 'Created', 'Signed In', and 'User UID'. A modal window is open in the center, titled 'Add an Email/Password user'. It has two input fields: 'Email' containing 'np.hieuminhqh@gmail.com' and 'Password' containing '123456'. At the bottom of the modal are 'Cancel' and 'Add user' buttons. Below the modal, the main screen displays the message 'No users for this project yet'.

-Bằng cách Add user, chúng ta có thể thêm người đăng nhập mới thủ công hoặc tạo một trang đăng ký trên app để thêm người truy cập mới

1. Setup Firestore và authentication

Authentication

Đăng nhập

-Lúc này, người dùng có thể đăng nhập vào app sử dụng tài khoản Email và mật khẩu

Email

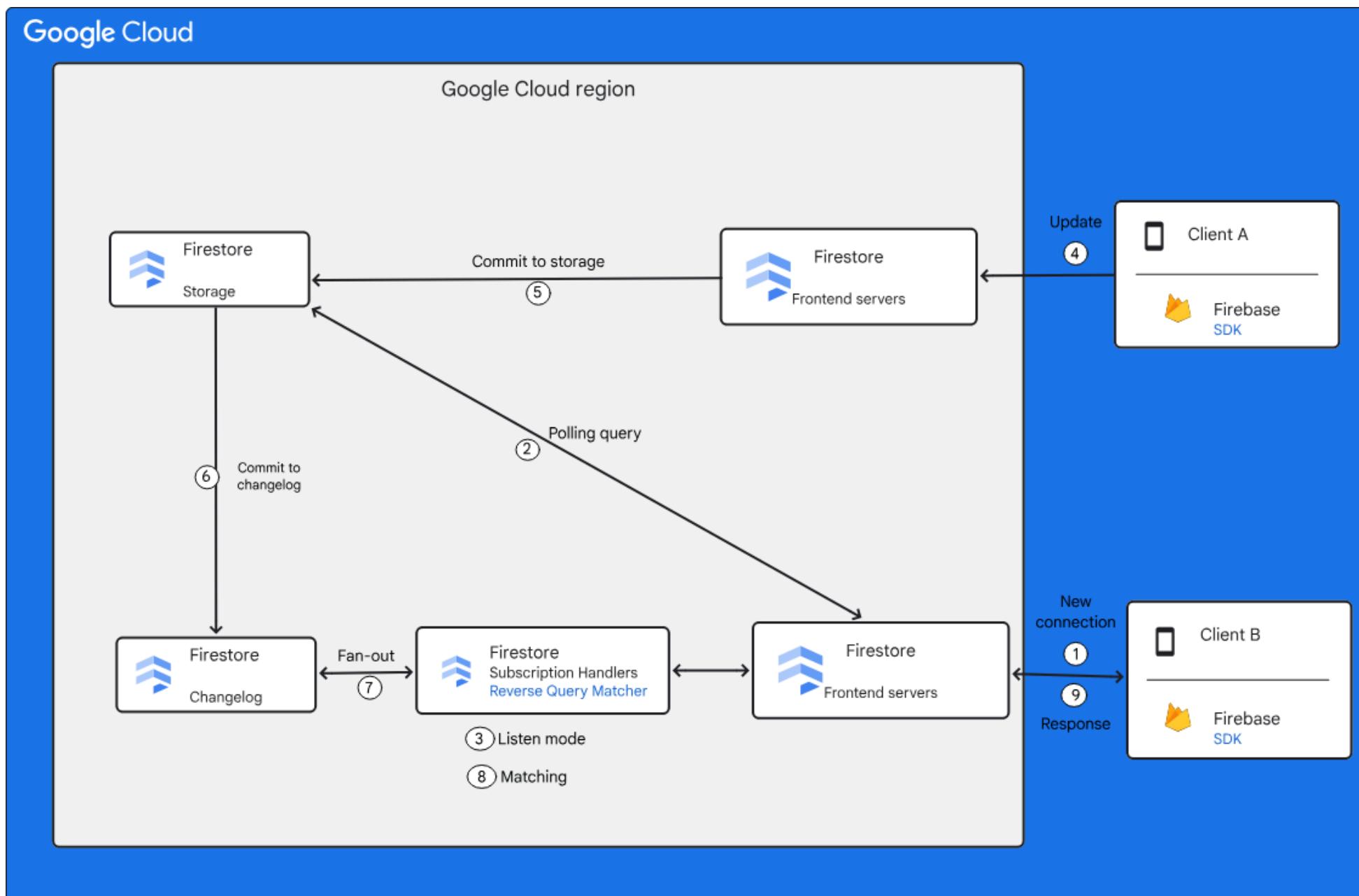
Mật khẩu

Đăng nhập

Đăng ký

2.Complex queries với where, orderBy, limit

Query



Query hay còn gọi là truy vấn, là các phương thức được sử dụng để lọc và truy xuất dữ liệu cụ thể từ cơ sở dữ liệu Firebase, thay vì tải toàn bộ dữ liệu. Các truy vấn này cho phép bạn kết hợp các điều kiện khác nhau để tìm kiếm dữ liệu một cách hiệu quả, giúp giảm lượng dữ liệu được tải và tăng tốc độ ứng dụng.

2.Complex queries với where, orderBy, limit

Cú pháp Query

```
final QuerySnapshot snapshot = await FirebaseFirestore.instance  
    .collection('SinhVien')  
    .where("city", isEqualTo: "Da Nang")  
    .orderBy("name", descending: true)  
    .limit(3)  
    .get();
```

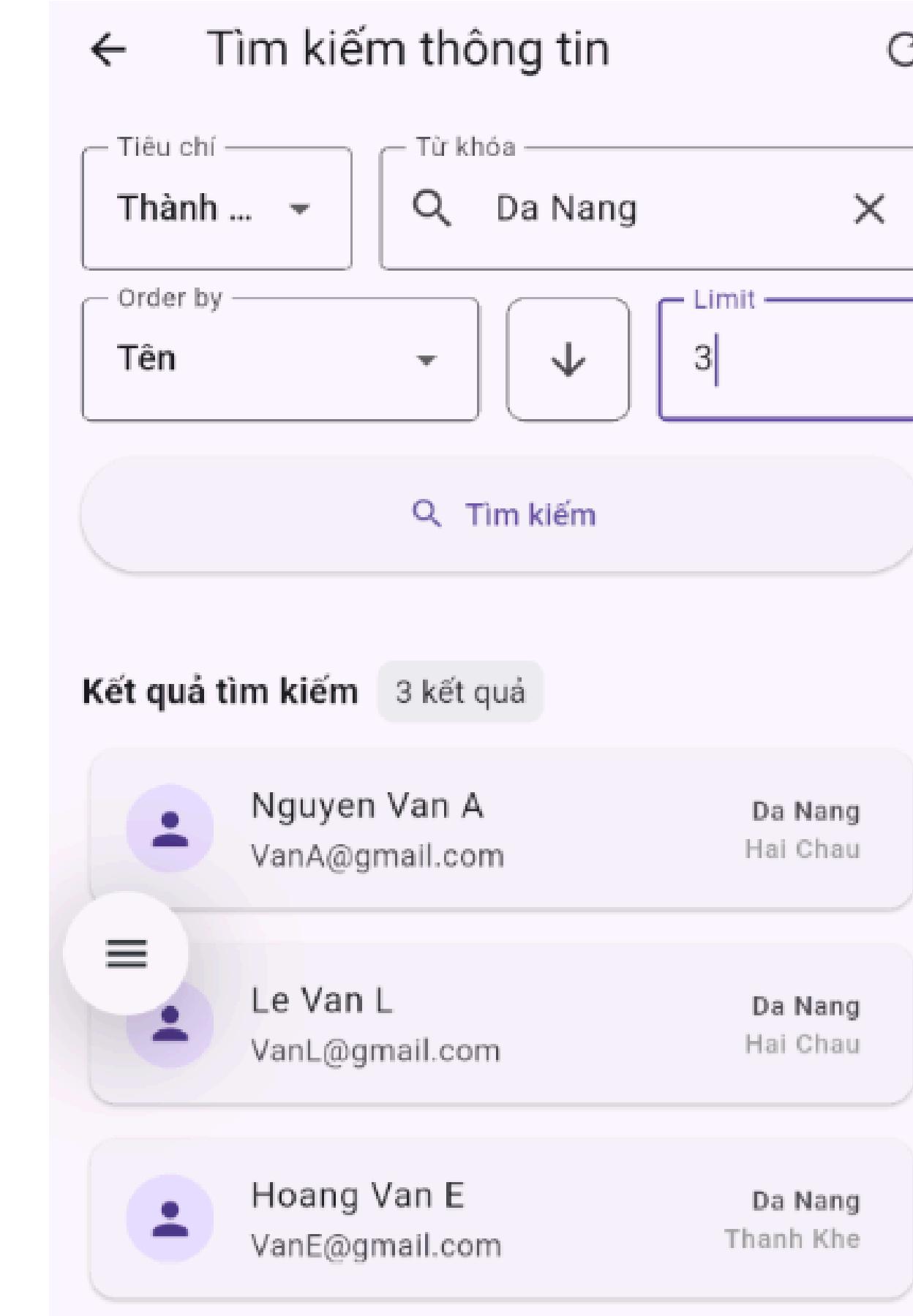
```
Query<Map<String, dynamic>> where(  
    Object field, {  
        Object? isEqualTo,  
        Object? isNotEqualTo,  
        Object? isLessThan,  
        Object? isLessThanOrEqualTo,  
        Object? isGreaterThanOrEqualTo,  
        Object? arrayContains,  
        Iterable<Object?>? arrayContainsAny,  
        Iterable<Object?>? whereIn,  
        Iterable<Object?>? whereNotIn,  
        bool? isNull,
```

- Collection: chỉ tên của collection mà chúng ta muốn lấy document
- OrderBy: sắp xếp những document lấy về theo tiêu chí nào đó theo chiều tăng hoặc giảm
- Limit: giới hạn số lượng document lấy về
- Where: chỉ lấy những document mà có trường có giá trị nào đó

2.Complex queries với where, orderBy, limit

Ví dụ với app tìm kiếm thông tin

- Các document tìm kiếm theo tiêu chí thành phố và giới hạn là 3
- Đối với một số queries phức tạp hơn chúng ta có thể lồng thêm nhiều hàm where() với các giá trị: “greater than”, “contains”,... để thực hiện tìm kiếm sâu hơn



3. Real-time listeners và offline support

Realtime Listeners:

- Mục đích: Đảm bảo dữ liệu hiển thị trên ứng dụng khách (client) luôn được đồng bộ với dữ liệu trên máy chủ (Server) mà không cần làm mới thủ công (polling).
- Triển khai trong Flutter: dùng phương thức `.snapshot()` trên 1 Query hoặc DocumentReference. Kết quả trả về là 1 Stream (luồng dữ liệu).

```
final Stream<QuerySnapshot> studentsStream =  
  FirebaseFirestore.instance.collection('SinhVien').snapshots();
```

3. Real-time listeners và offline support

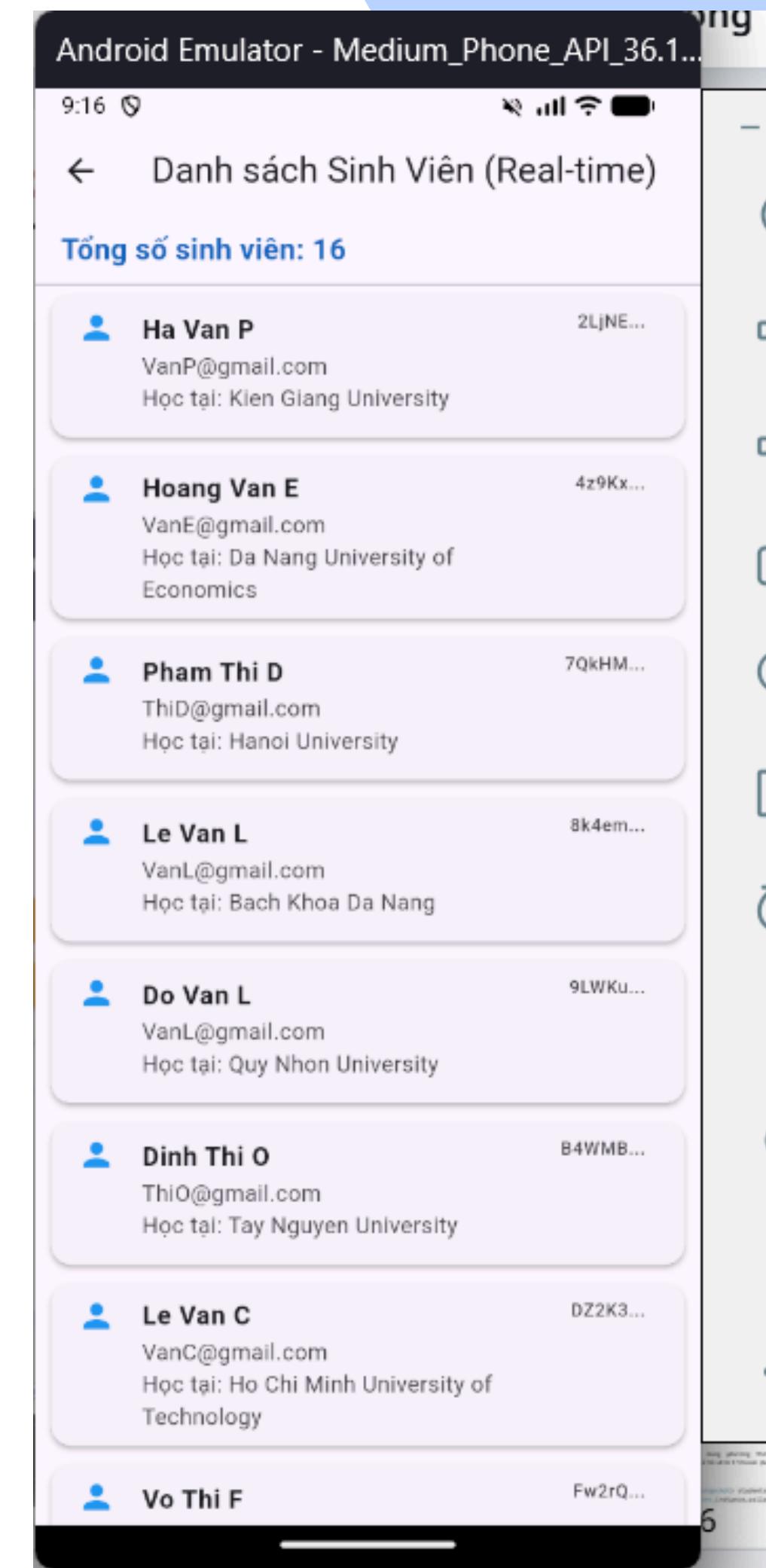
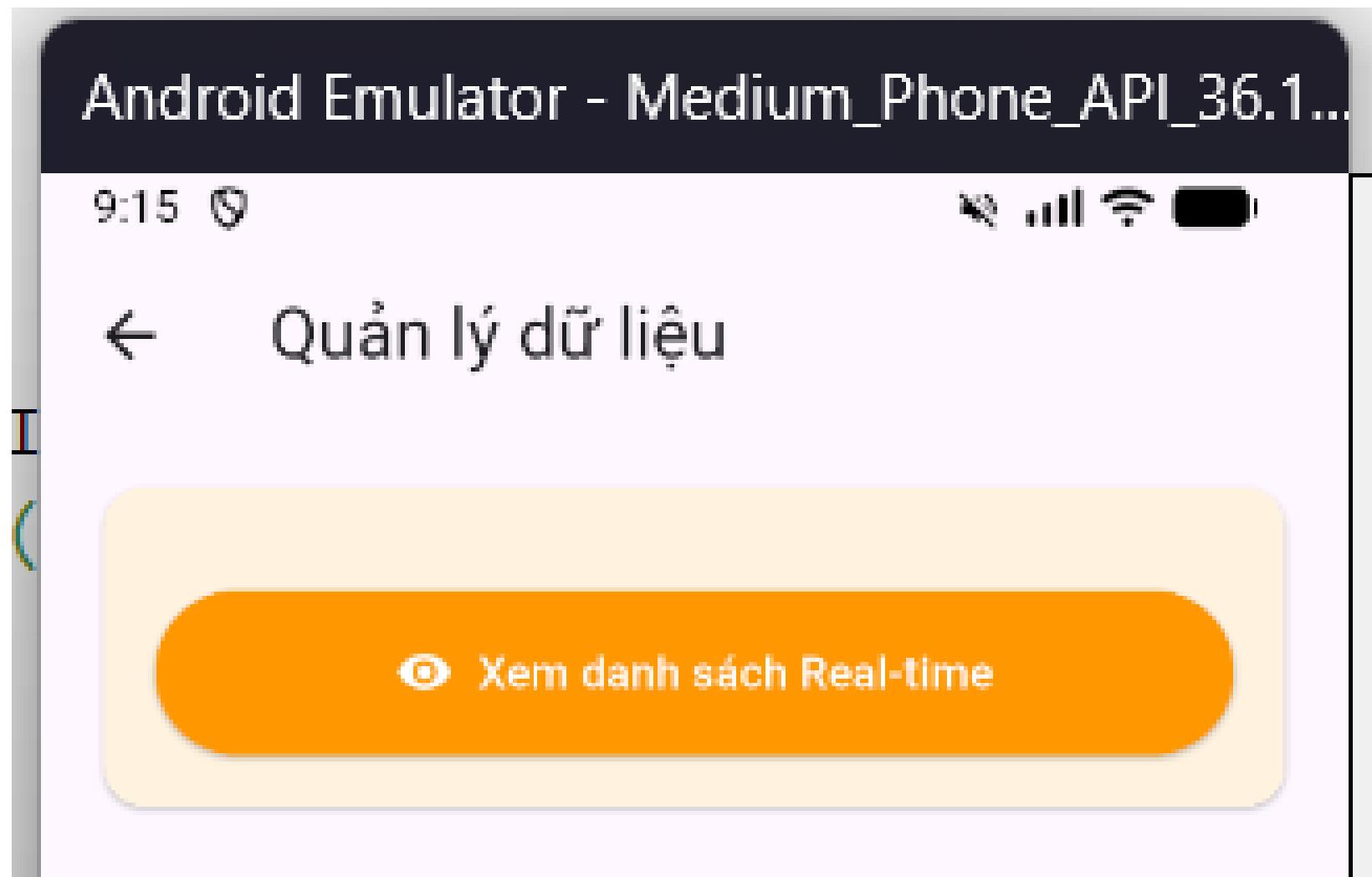
Widget chính: dùng StreamBuilder để lắng nghe Stream này. Khi có bất kỳ thay đổi nào (thêm, sửa, xóa), StreamBuilder sẽ nhận được một snapshot mới và tự động xây dựng lại widget (ví dụ: ListView).

```
StreamBuilder<QuerySnapshot>(  
    stream: studentsStream,  
    builder: (context, snapshot) {  
        final int currentTotal = snapshot.hasData ? snapshot.data!.docs.length : 0;
```

StreamBuilder	Widget được thiết kế để lắng nghe các sự kiện từ 1 Stream
Stream: studentsStream	Liên kết StreamBuilder với luồng dữ liệu realtime đã tạo
builder:(context,snapshot):	Mỗi khi Firestore gửi dữ liệu mới (do có thay đổi), hàm builder này được gọi lại với snapshot mới, và widget sẽ tự động được vẽ lại để hiển thị số lượng mới nhất (currentTotal)

3. Real-time listeners và offline support

App Demo



3. Real-time listeners và offline support

Offline Support:

- Là tính năng tích hợp sẵn của Firebase Firestore, cho phép ứng dụng hoạt động trơn tru ngay cả khi không có kết nối Internet.
- Bộ nhớ đệm (Caching): Firestore SDK tự động lưu trữ cục bộ (caches) dữ liệu mà ứng dụng đã truy vấn gần đây. Các thao tác ghi dữ liệu (.add(), .set(), .delete()) được xếp vào hàng đợi cục bộ (local queue).

```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(); // Khởi tạo Firebase
  runApp(const MyApp());
}
```

4. Pagination và data modeling

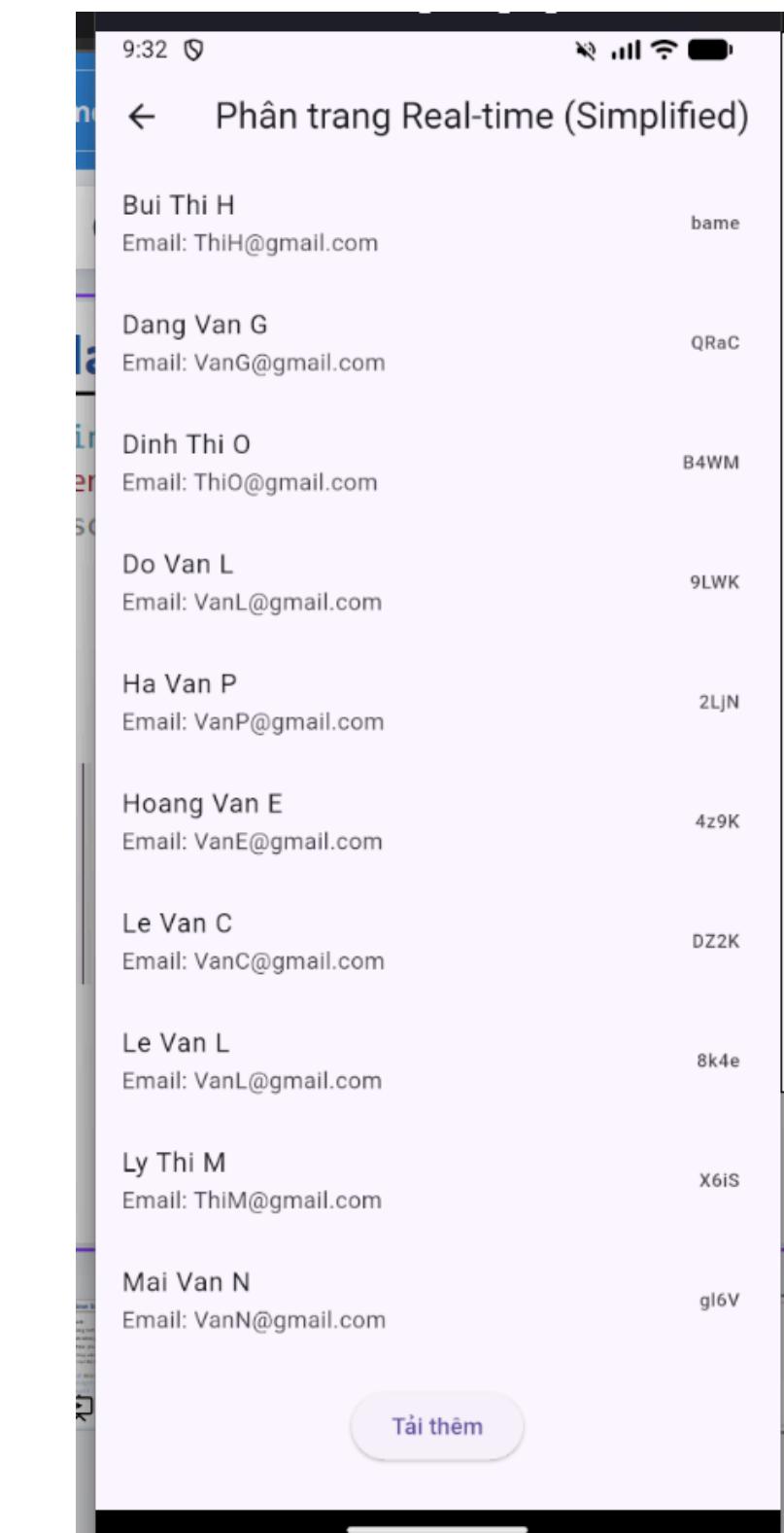
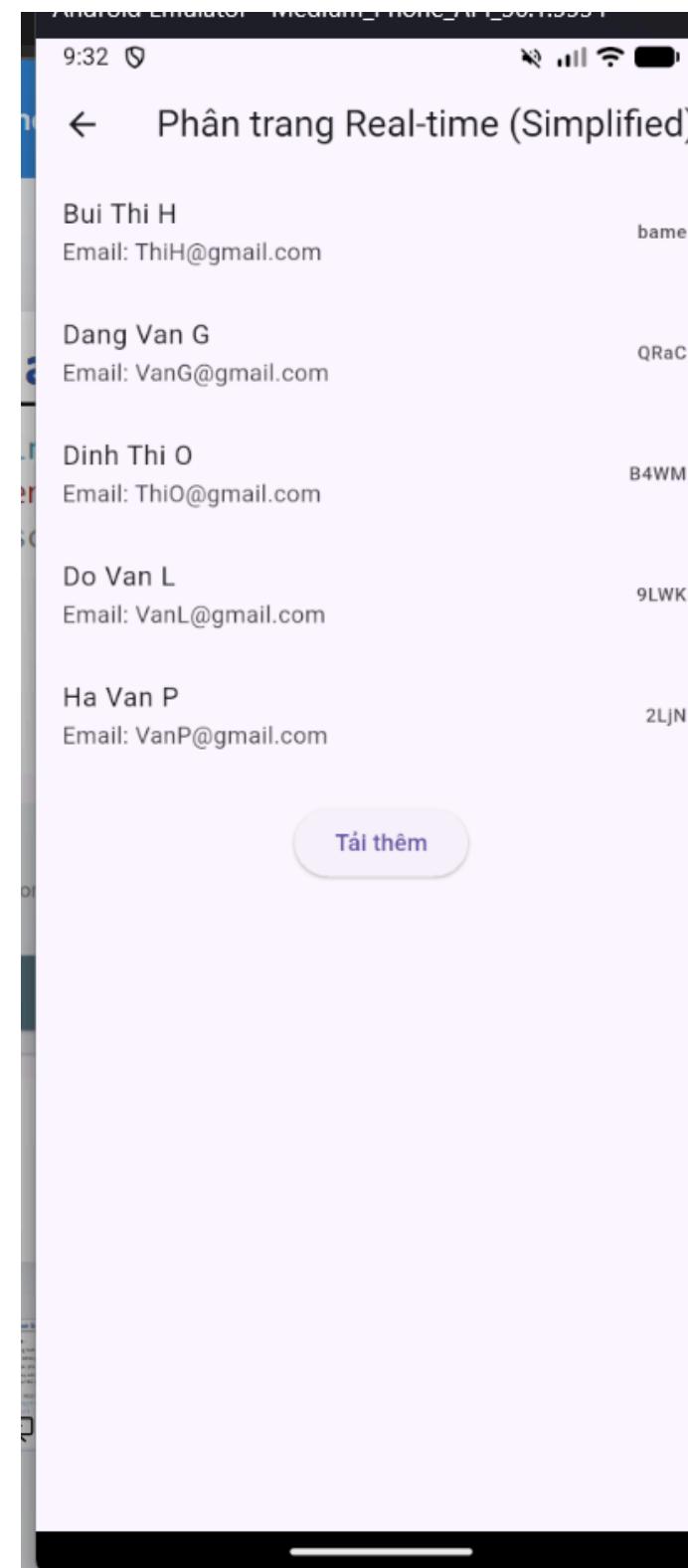
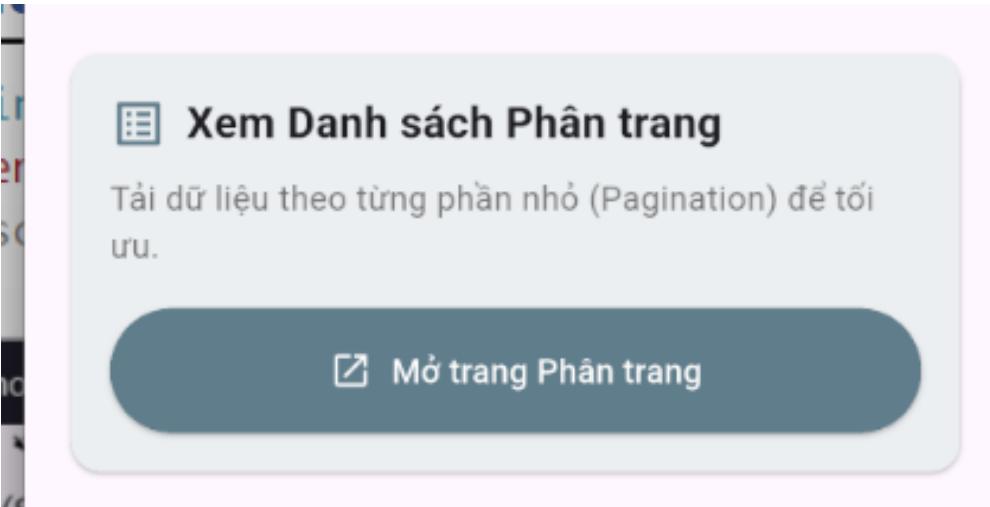
Pagination: Phân Trang

- Là kỹ thuật chia tập dữ liệu lớn thành các phần nhỏ hơn để tải và hiển thị từng phần, VD: 5 mục 1 lần nhằm tăng tốc độ tải, nâng cao trải nghiệm người dùng.

Phương thức	Chức năng	Ví dụ
limit(N)	Giới hạn số lượng tài liệu trả về	Lấy 5 mục 1 lần
orderBy('field')	Bắt buộc. Phân trang luôn cần sắp xếp để xác định điểm dừng	Sắp xếp theo tên
startAfter(doc Snapshot)	Lấy các mục sau tài liệu cuối cùng của trang trước	Đọc trang 2 bắt đầu theo mục 5

4. Pagination và data modeling

```
Query query = FirebaseFirestore.instance  
.collection('SinhVien')  
.orderBy('name', descending: false) // BẤT BUỘC phải có orderBy  
.limit(_pageSize);
```



4. Pagination và data modeling

Data Modeling:

- Trong Firestore, ưu tiên truy vấn hơn cấu trúc (Query-Driven Modeling), thiết kế theo cách ứng dụng đọc dữ liệu, tổ chức sao cho lấy được dữ liệu với số lần đọc ít nhất.
- Các relations:

Mối quan hệ	Kỹ thuật triển khai chính	Mục đích
1:1 / 1:Ít	Nhúng (Embedding) (Map/Array)	Dữ liệu nhỏ, luôn cần được tải cùng tài liệu cha.
1:N (Nhiều)	Bộ sưu tập con (Subcollection)	Dữ liệu lớn, cần truy vấn cục bộ, nhưng vẫn thuộc về tài liệu cha. (VD: Lịch sử giao dịch).
1:N (Toàn cầu)	Tham chiếu (Reference) (Lưu ID)	Cần truy vấn toàn bộ các mục con độc lập trên hệ thống. (VD: Tất cả bài viết).
N:N	Collection Liên kết (Junction)	Cần tạo mối liên kết phức tạp giữa hai Collection. (VD: Users & Groups).
Tối ưu chung	Denormalization (Lặp dữ liệu)	Bắt buộc để giảm số lần đọc. Lặp lại dữ liệu thường xuyên cần thiết.

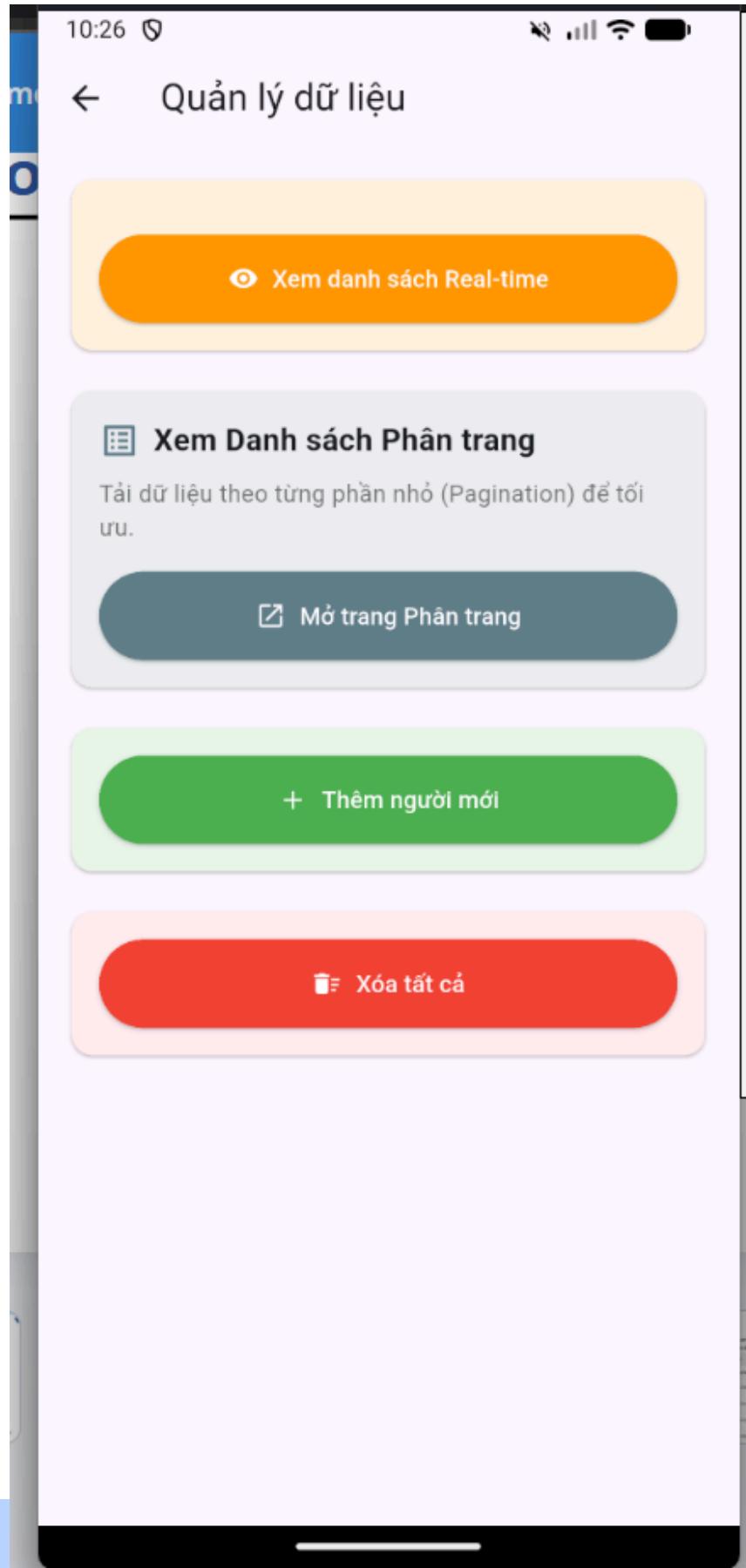
4. Pagination và data modeling

Ứng dụng Data Modeling:

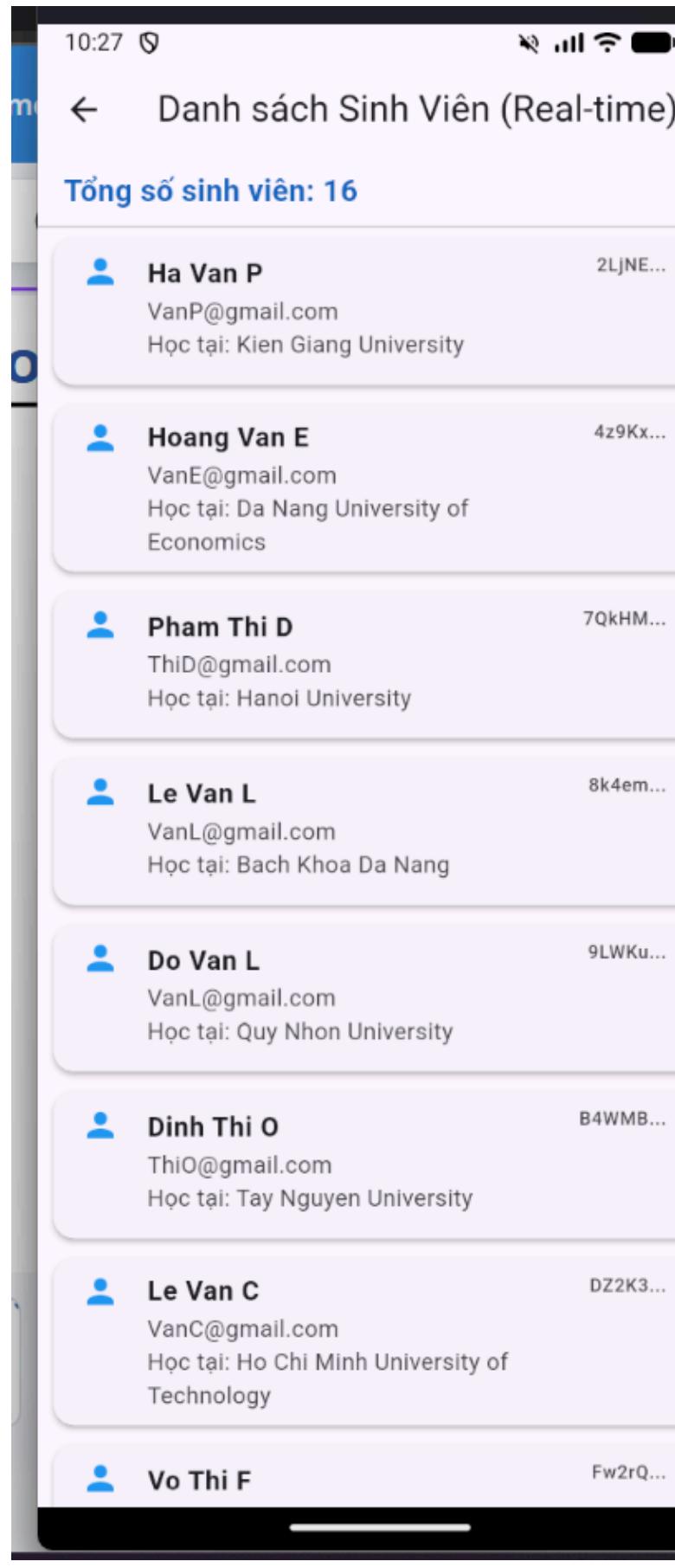
- Nhúng tài liệu (Embedded Document) cho địa chỉ Address. Địa chỉ lưu dưới dạng Map.

```
try {
    // Prepare data
    final data = {
        'name': _nameController.text.trim(),
        'email': _emailController.text.trim(),
        'address': {
            'city': _cityController.text.trim(),
            'ward': _wardController.text.trim(),
            'street': _streetController.text.trim(),
            'houseNumber': _houseNumberController.text.trim(),
        },
        'school': _schools,
    };
}
```

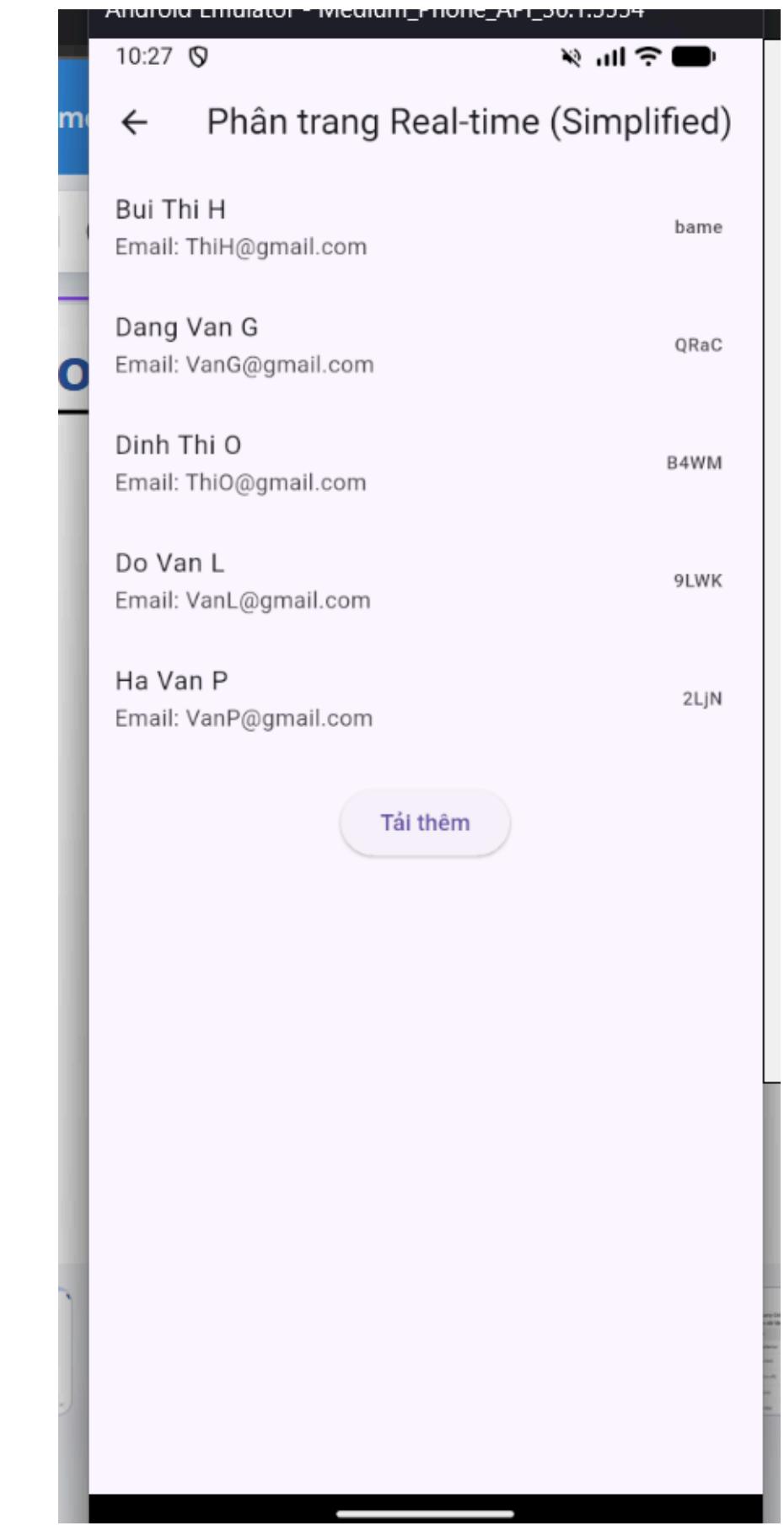
Tổng quan



Real-time



Pagination



Tài liệu tham khảo

- 1/<https://firebase.google.com/docs/firestore/rtdb-vs-firebase>
- 2/<https://www.youtube.com/playlist?list=PLIK7zZEYLuG5MCVEzXAQ7ACZBCuZgZe> (kênh của Firebase)
- 3/https://pub.dev/packages/firebase_core/example
- 4/https://pub.dev/packages/firebase_auth/example
- 5/https://pub.dev/packages/cloud_firestore/example

THE END



Thank You