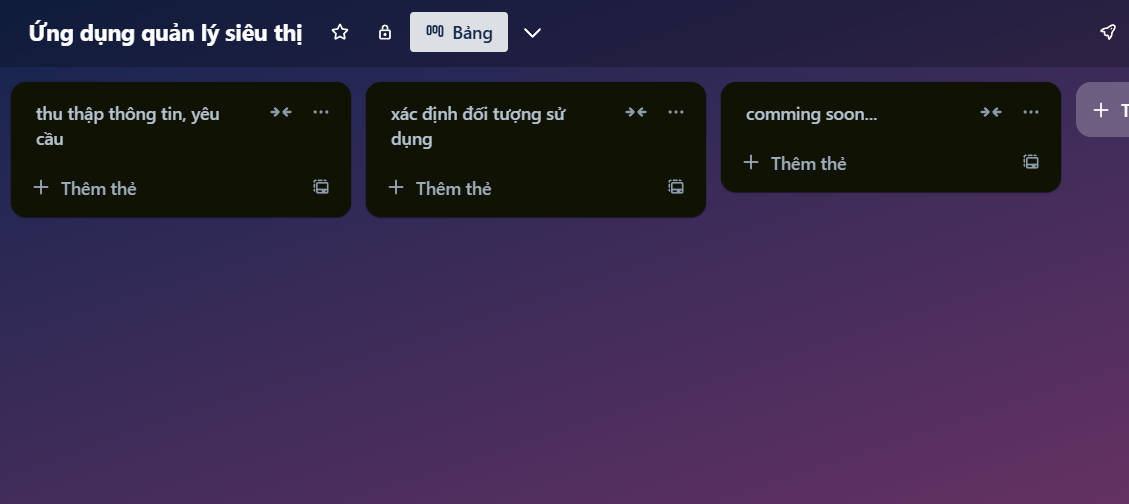
Nguyễn Thế Yên

20211066

Đề 15: PHÂN TÍCH, ĐẶC TẢ, THIẾT KẾ VÀ PHÁT TRIỂN ỨNG DỤNG QUẢN LÝ SIÊU THỊ

Các tools dùng trong 3 pha:

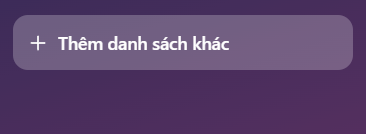
**Pha lập kế hoạch**  
- Trello : Công cụ quản lý dự án đơn giản sử dụng hệ thống bảng Kanban, dùng để lên kế hoạch và hướng giải quyết lâu dài



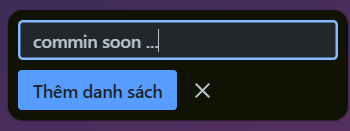
Pic1.1: giao diện Trello

-cách dùng:

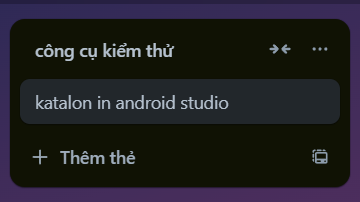
Chọn thêm danh sách và điền thông tin để đính như ảnh trên:



Khi nhập thông tin xong thì bấm thêm danh sách :



Có thể thêm thẻ bên dưới danh sách công việc cần làm :



* Ngoài ra còn dung word, file txt để làm lưu trữ tạm thời, note thêm nháp…
* Dùng thêm các công cụ thiết kế mô hình như : Draw.io

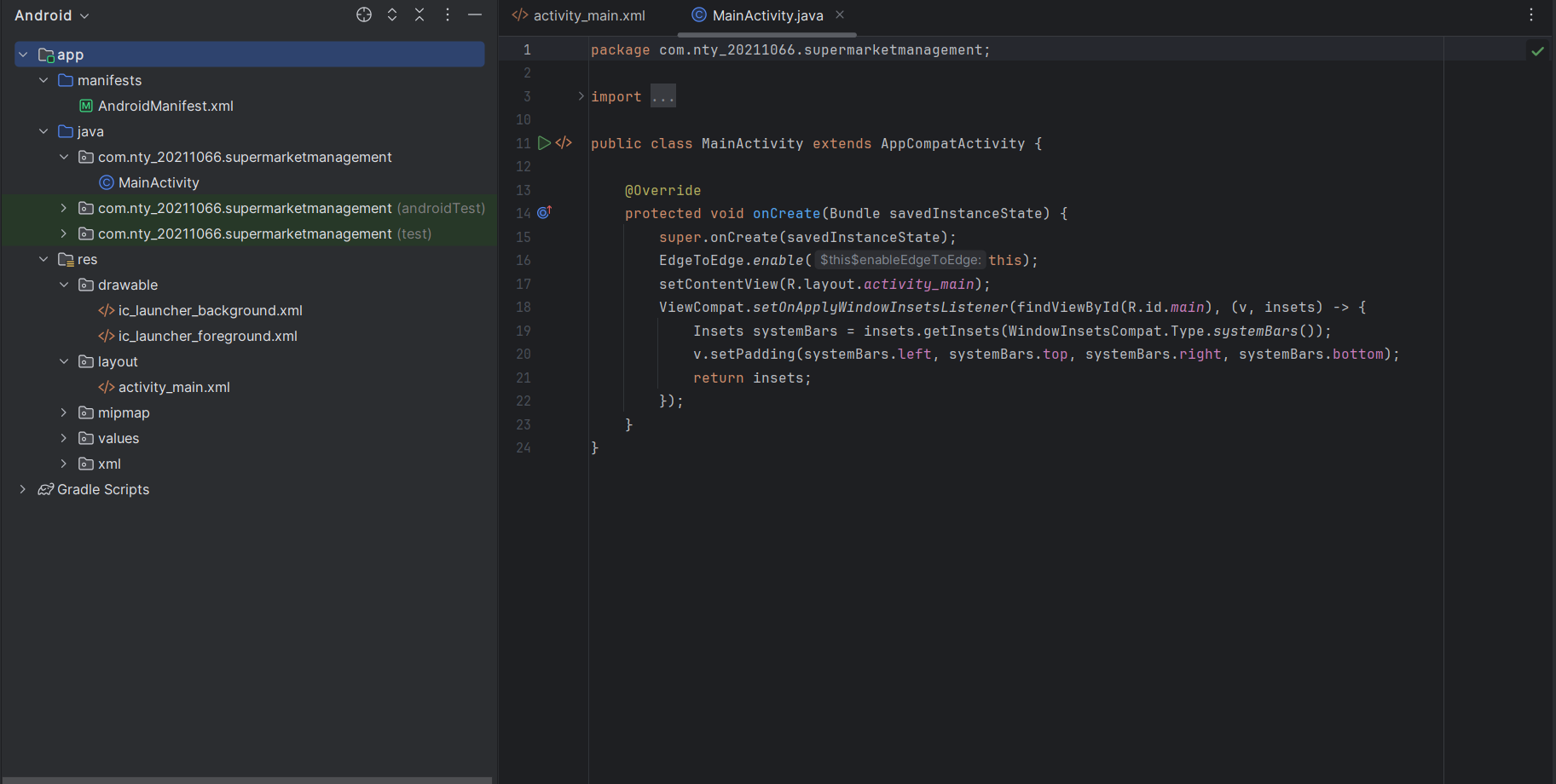
# **Pha phát triển:**

Em đang thử giữa android studio , visual studio, và bên pycham.

## *Android studio :*

* Mục đích chính: Android Studio là môi trường phát triển tích hợp (IDE) chính thức của Google dành cho việc phát triển ứng dụng Android.
* Các tính năng nổi bật:
* Hỗ trợ đầy đủ công cụ để phát triển, thử nghiệm, và gỡ lỗi ứng dụng Android.
* Trình mô phỏng (emulator) để chạy thử ứng dụng trên nhiều thiết bị.
* Công cụ thiết kế giao diện kéo-thả (drag-and-drop).
* Tích hợp sẵn Gradle để quản lý dự án và các dependency.
* Ngôn ngữ hỗ trợ: Java, Kotlin là hai ngôn ngữ chính, nhưng cũng có thể sử dụng C++ với Android NDK.

Dự án dự kiến :

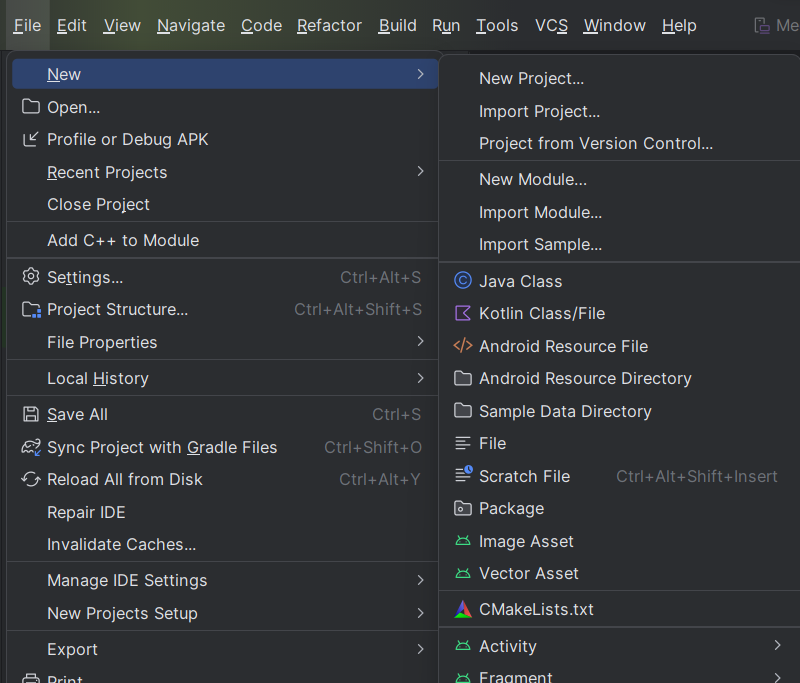


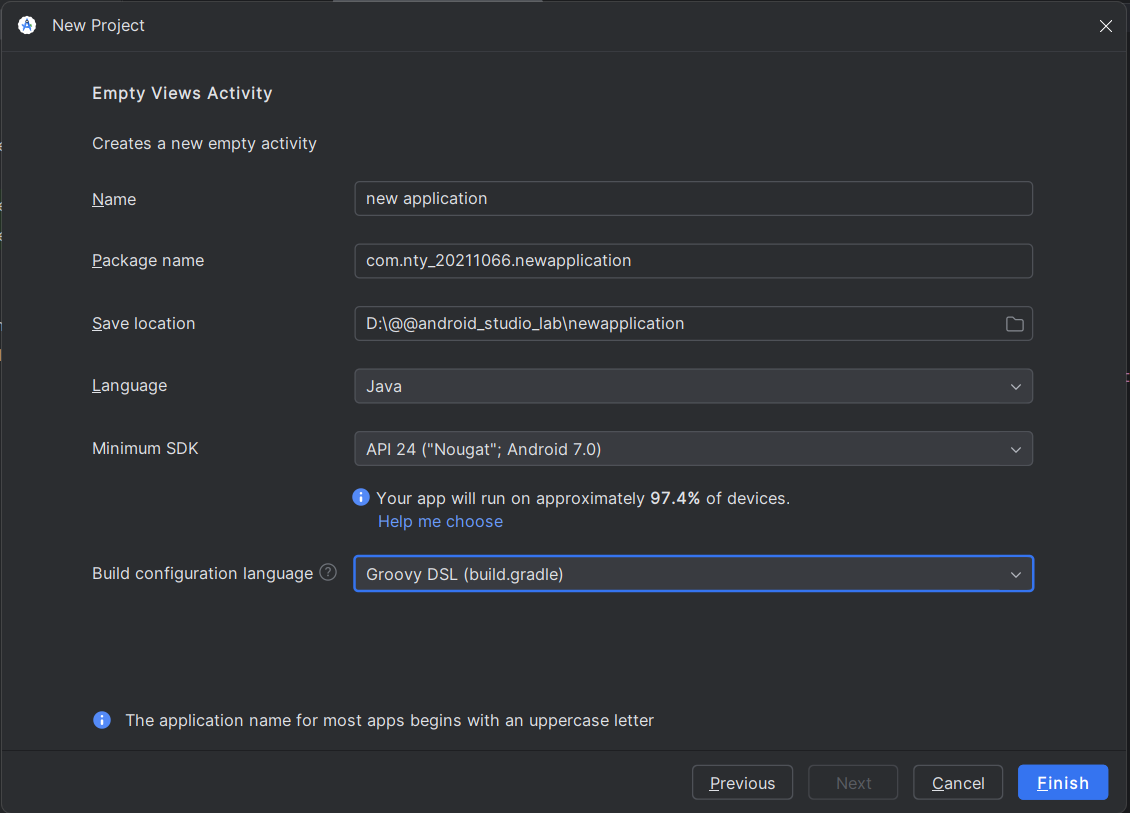
* Cách dùng Android Studio:

### 1. Tạo Project Mới

#### Bước 1: Tạo Project

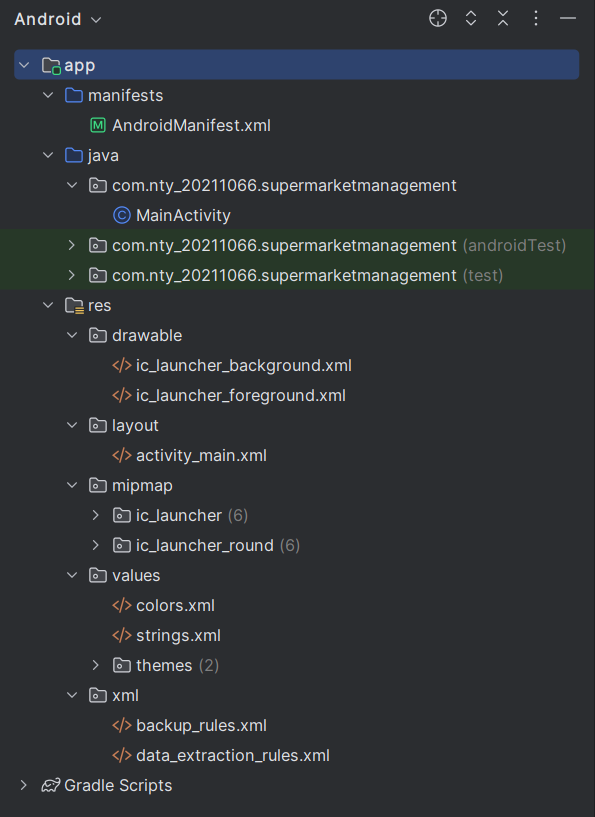
Mở Android Studio và chọn New Project. Chọn loại giao diện mặc định, ví dụ: Empty Activity. Nhập thông tin Project: Tên ứng dụng, tên package, vị trí lưu trữ, ngôn ngữ lập trình (Java hoặc Kotlin), và phiên bản SDK.





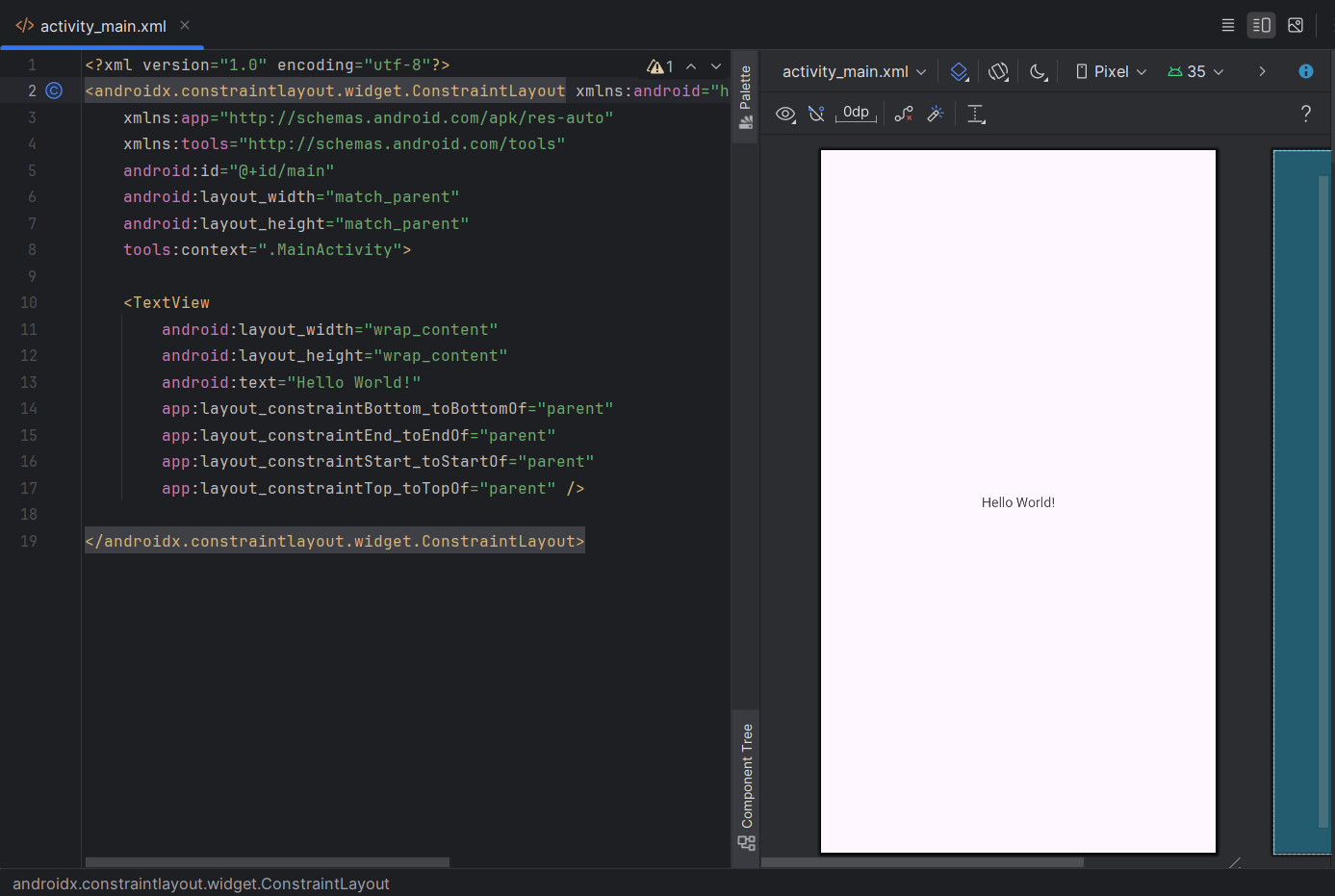
#### Bước 2: Cấu trúc của Project

app: Chứa mã nguồn ứng dụng Android, bao gồm các file Java/Kotlin, XML.  
res: Thư mục chứa tài nguyên như hình ảnh, layout, chuỗi văn bản, etc.  
AndroidManifest.xml: File cấu hình ứng dụng, định nghĩa các Activity, quyền truy cập, etc.  
Gradle Scripts: Quản lý thư viện và các thiết lập biên dịch.



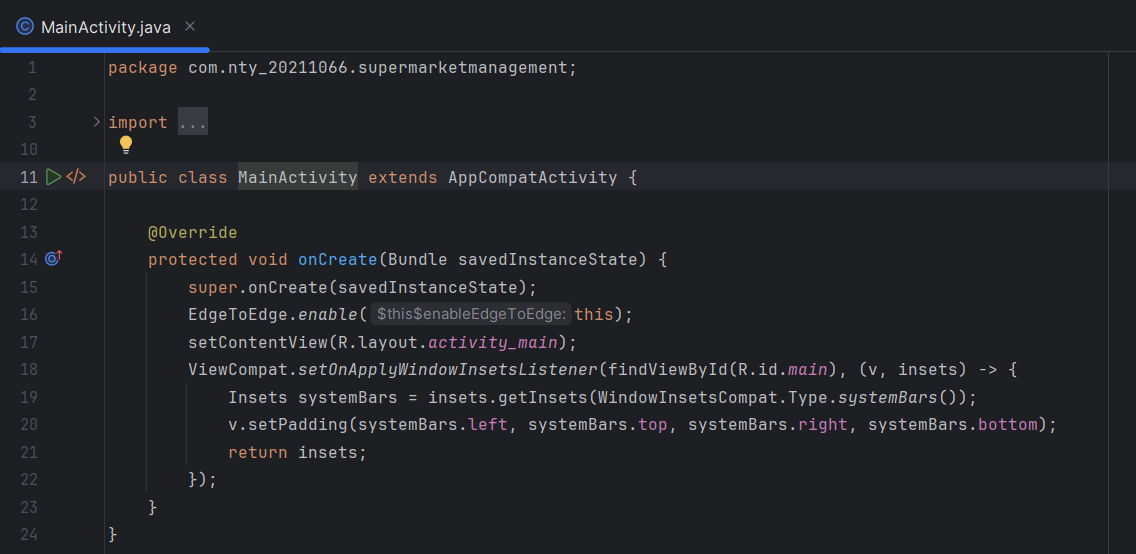
### 2. Tạo Giao Diện Người Dùng (UI)

#### Bước 1: Sử dụng XML để tạo Layout

Mở file activity\_main.xml trong thư mục res/layout. Dùng các thành phần như TextView, Button, ImageView để xây dựng giao diện. Bạn có thể kéo thả các thành phần UI vào giao diện hoặc chỉnh sửa trực tiếp file XML

#### Bước 2: Kết nối UI với Code

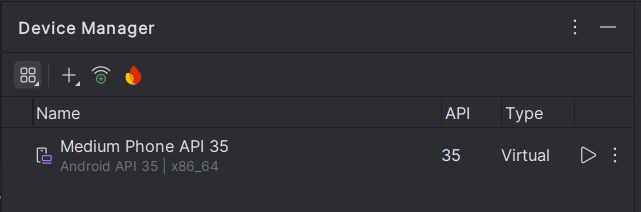
Mở file MainActivity.java và kết nối các thành phần UI với mã Java thông qua R.layout.



### 3. Chạy Ứng Dụng Trên Thiết Bị Thật Hoặc Trình Giả Lập

#### Bước 1: Thiết lập Emulator (Trình Giả Lập)

Trong Android Studio, chọn Tools > Device Manager để tạo và quản lý các thiết bị giả lập. Chọn cấu hình giả lập với phiên bản Android mong muốn.

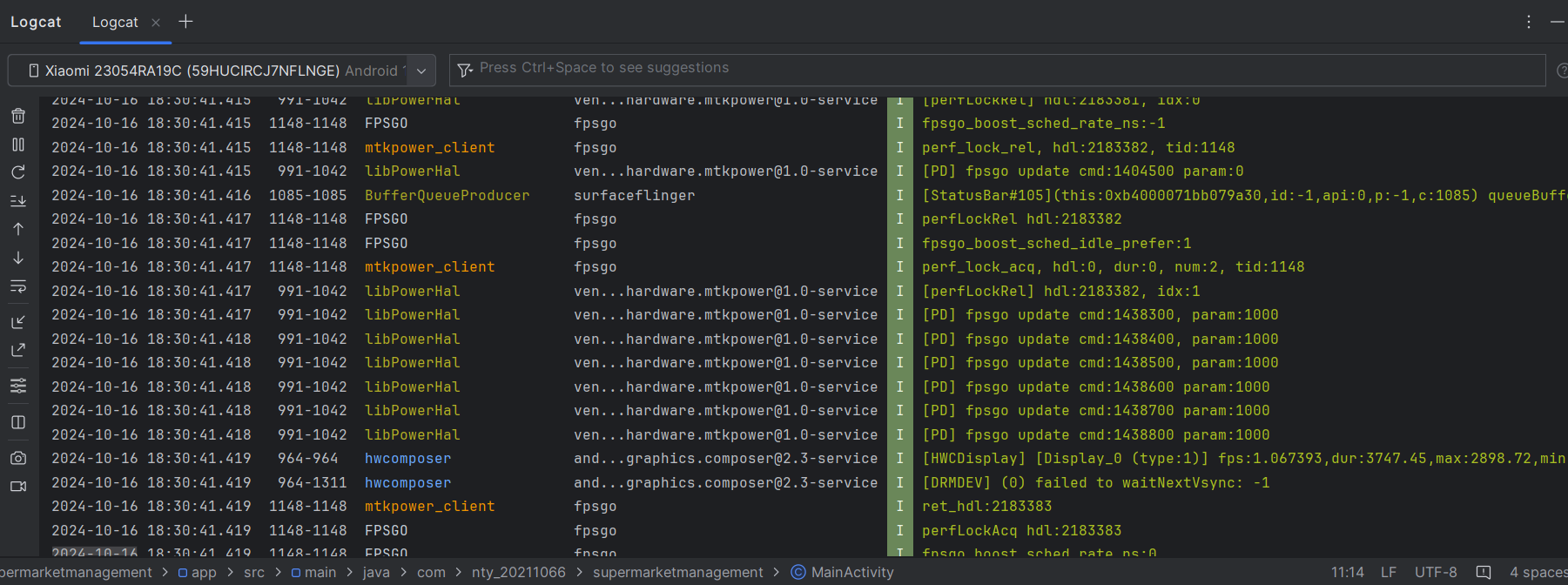


#### Bước 2: Chạy ứng dụng

Kết nối thiết bị Android với máy tính (nếu bạn sử dụng thiết bị thật, hãy bật tùy chọn Developer Options và USB Debugging). Nhấn nút Run (hình tam giác màu xanh) trên thanh công cụ để biên dịch và chạy ứng dụng.

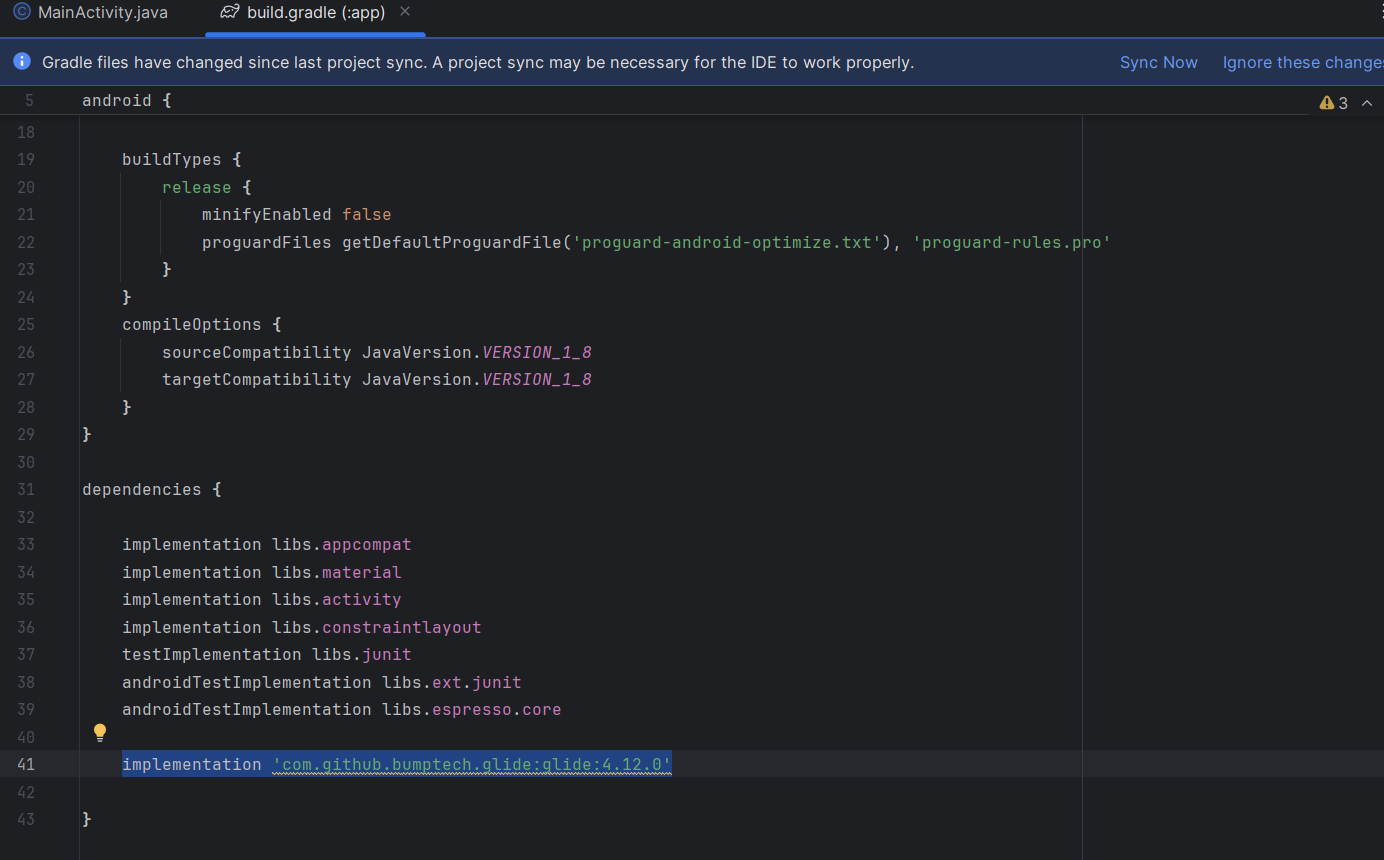
### 4. Debug và Logcat

Logcat: Công cụ hiển thị các bản ghi (log) từ thiết bị. Sử dụng Log.d() trong mã nguồn để ghi lại thông tin trong quá trình chạy ứng dụng.  
  
Log.d('MainActivity', 'Ứng dụng đã chạy thành công');  
  
Debug: Sử dụng công cụ Debug của Android Studio để đặt các điểm dừng (breakpoint) và kiểm tra các giá trị biến trong quá trình chạy ứng dụng.



### 5. Quản lý Thư Viện với Gradle

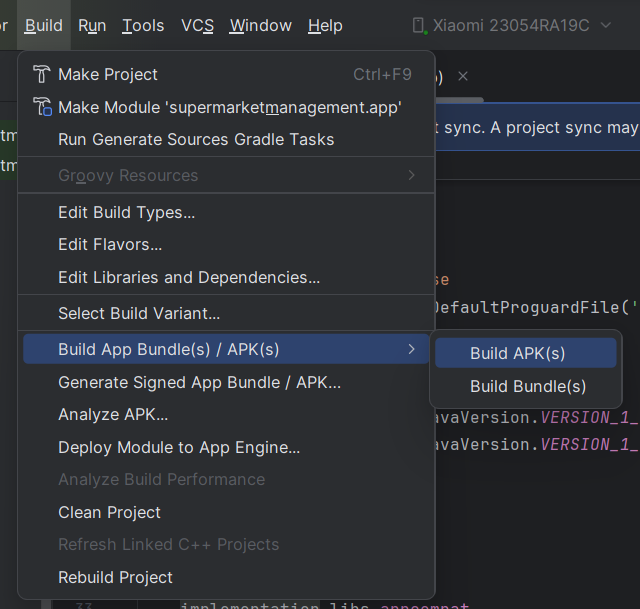
build.gradle: Tập tin này quản lý các thư viện và các thông số cấu hình của project. Để thêm một thư viện vào project, bạn có thể thêm dòng mã trong file build.gradle ở module app.  
  
Ví dụ: Thêm thư viện Glide để tải ảnh từ Internet.  
  
implementation 'com.github.bumptech.glide:glide:4.12.0'



### 6. Xuất Bản Ứng Dụng

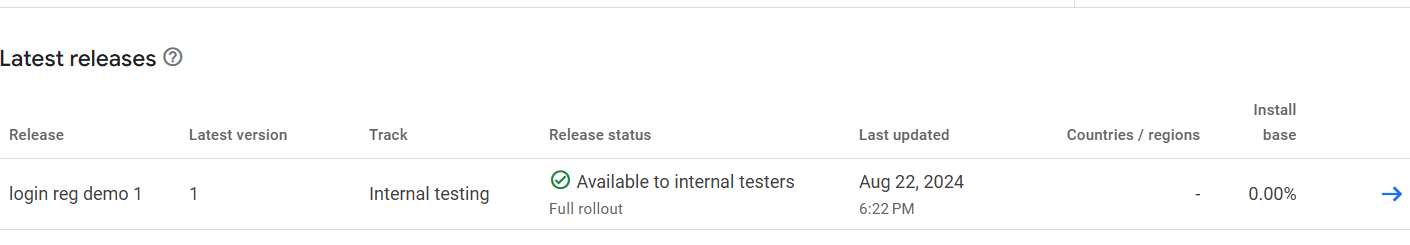
#### Bước 1: Tạo APK hoặc AAB

Chọn Build > Build Bundle(s) / APK(s) > Build APK(s). Android Studio sẽ biên dịch và tạo file APK trong thư mục app/build/outputs/apk.



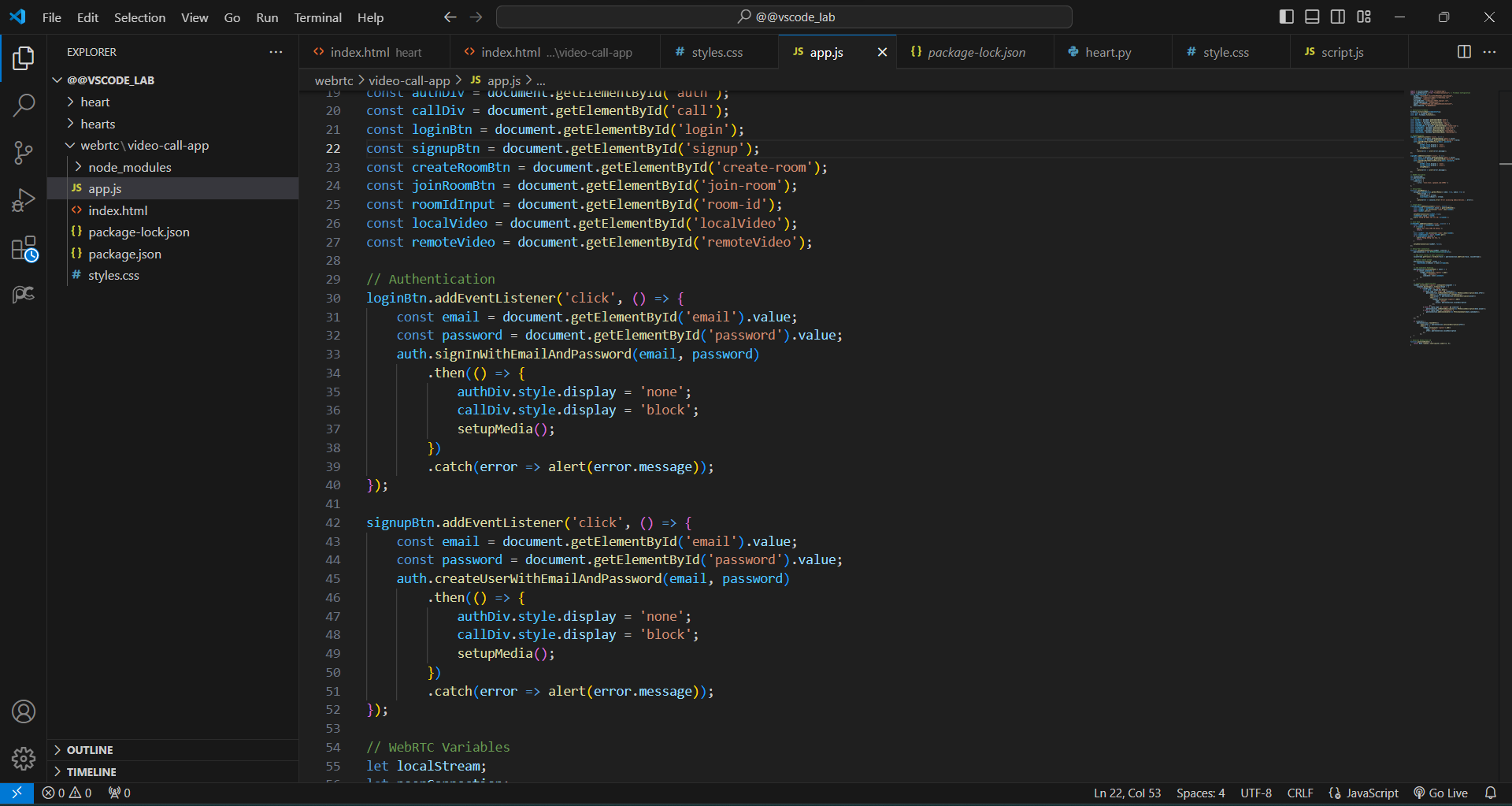
#### Bước 2: Đăng tải lên Google Play

Tạo tài khoản Google Play Console. Chuẩn bị các thông tin cần thiết (biểu tượng ứng dụng, ảnh chụp màn hình, mô tả). Tải lên file APK hoặc AAB và cấu hình phiên bản.



## *Visual studio code (VsCode):*

* Mục đích chính:Visual Studio Code là một trình biên tập mã nguồn nhẹ, hỗ trợ nhiều ngôn ngữ lập trình khác nhau.
* Các tính năng nổi bật:
* Giao diện đơn giản, nhẹ, và dễ tùy chỉnh.
* Có kho tiện ích mở rộng (extensions) rất phong phú, giúp tích hợp với nhiều công nghệ và ngôn ngữ lập trình khác nhau (JavaScript, Python, Java, C#, HTML, CSS, v.v.).
* Tính năng IntelliSense (gợi ý mã thông minh) hỗ trợ tự động hoàn thành mã và phát hiện lỗi nhanh chóng.
* Tích hợp Git và GitHub, giúp quản lý phiên bản dự án dễ dàng.
* Ngôn ngữ hỗ trợ:Rất đa dạng, từ JavaScript, Python, C++, Java, Go, đến các ngôn ngữ web như HTML, CSS.
* Giao diện mô phỏng:

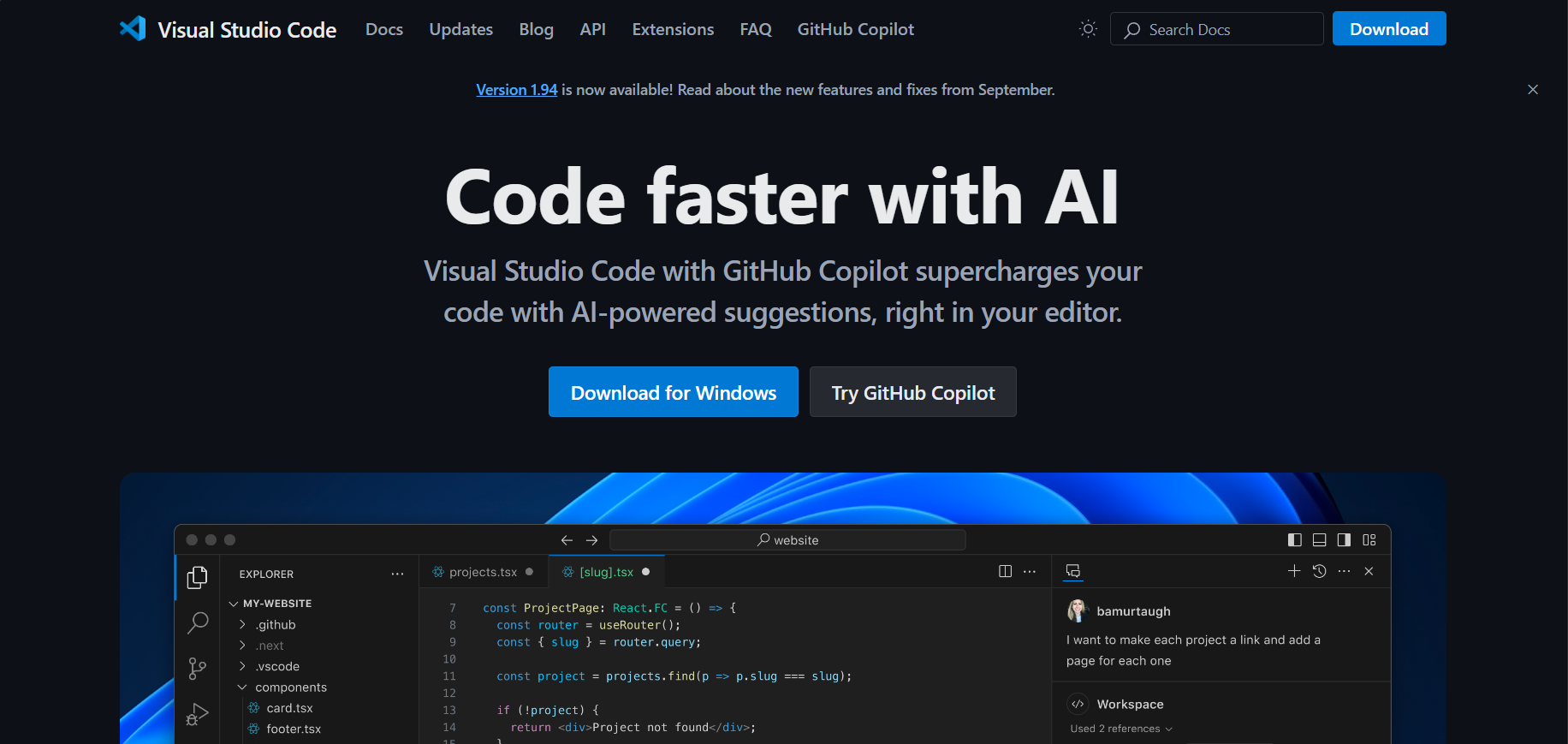


* Cách dùng visual studio code:

### 1. Cài đặt VSCode

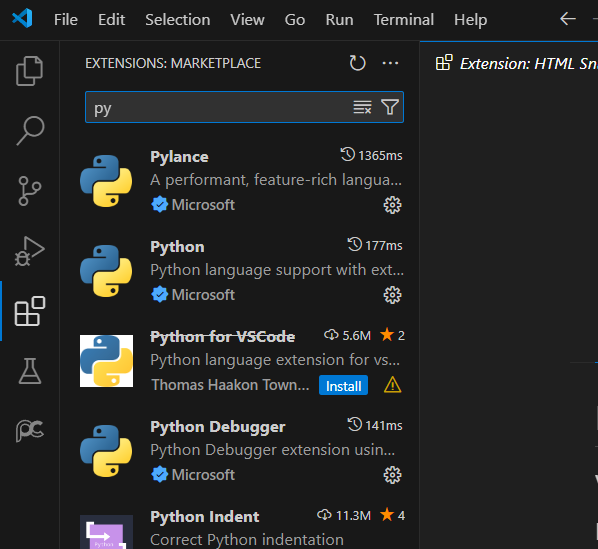
#### Bước 1: Tải và cài đặt VSCode

Truy cập trang chủ của Visual Studio Code: https://code.visualstudio.com/. Tải phiên bản phù hợp với hệ điều hành của bạn (Windows, macOS, Linux). Sau khi tải xuống, làm theo hướng dẫn để cài đặt VSCode.



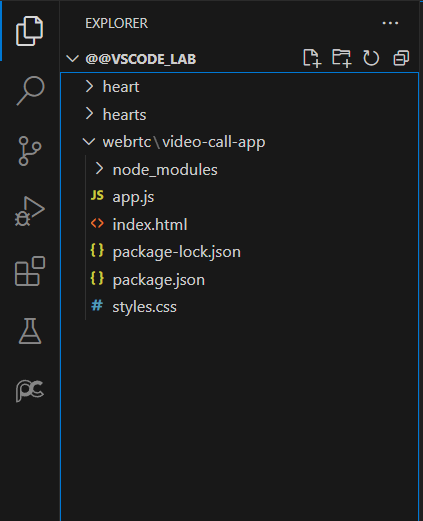
#### Bước 2: Cài đặt các tiện ích mở rộng (Extensions)

Mở VSCode và nhấn vào biểu tượng Extensions (hình cái hộp ở thanh công cụ bên trái). Tìm kiếm các tiện ích mở rộng phù hợp với ngôn ngữ hoặc công cụ bạn đang sử dụng, ví dụ như Python, JavaScript, Prettier, v.v.

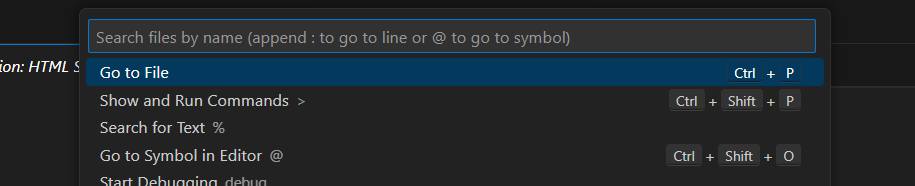


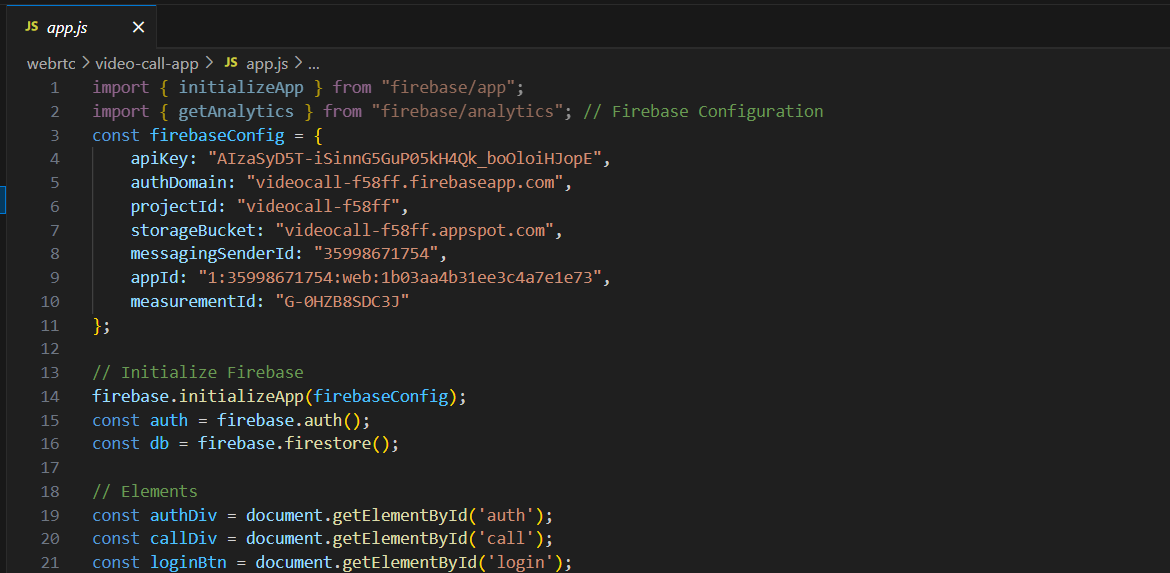
### 2. Giao diện của VSCode

Explorer: Quản lý các tệp và thư mục của dự án.

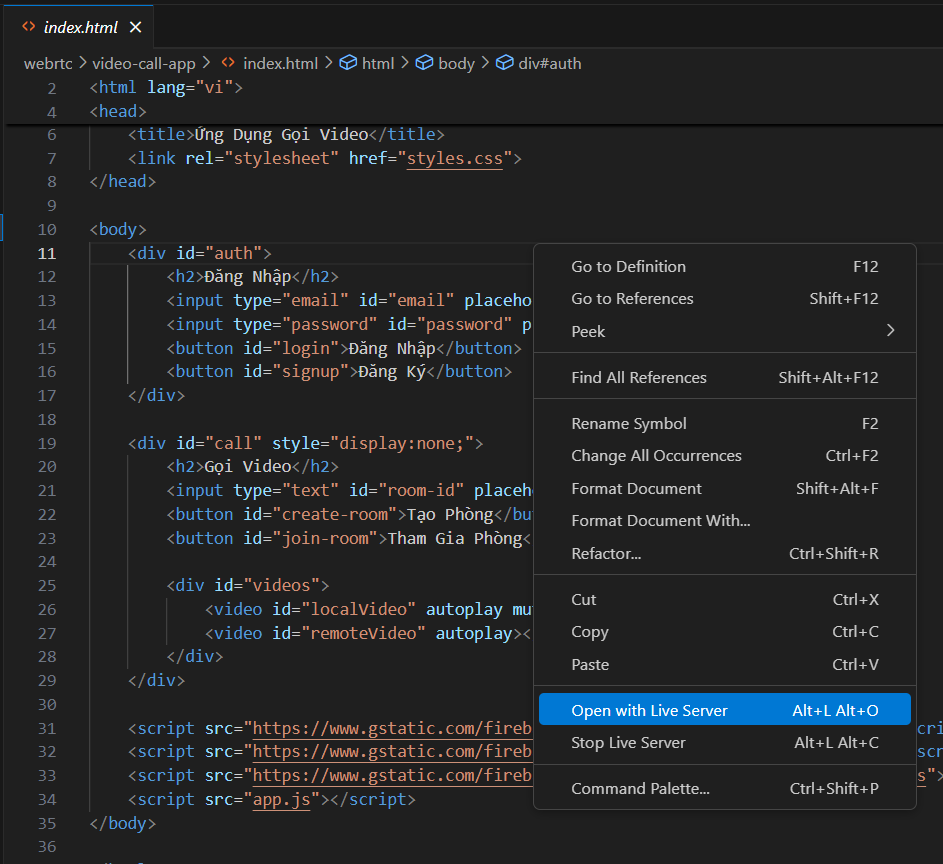


Search: Tìm kiếm và thay thế nội dung trong toàn bộ dự án.

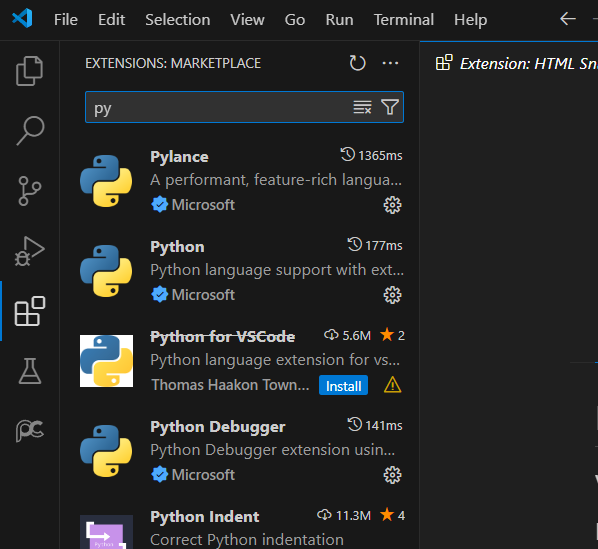
  
Source Control: Tích hợp với Git và các hệ thống quản lý mã nguồn.

  
Run and Debug: Chạy và debug ứng dụng.

Vd chạy file index.html bằng cách nhấn chuột phải chố trống và chọn open with live server



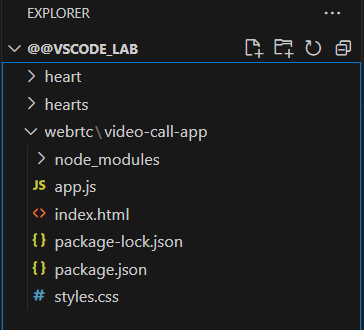
Extensions: Cài đặt và quản lý các tiện ích mở rộng.



### 3. Mở và quản lý dự án

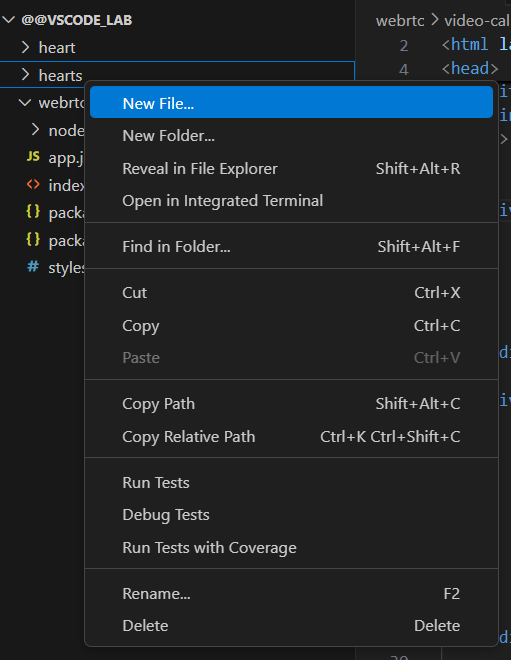
#### Bước 1: Mở thư mục dự án

Trong menu File, chọn Open Folder để mở thư mục chứa mã nguồn của bạn. Toàn bộ tệp và thư mục trong dự án sẽ được hiển thị ở thanh Explorer bên trái.



#### Bước 2: Quản lý tệp và thư mục

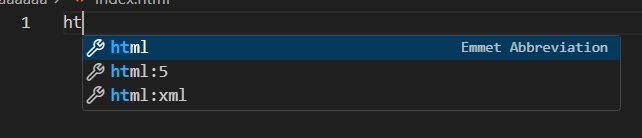
Bạn có thể nhấn chuột phải vào thanh Explorer để tạo tệp mới, thư mục mới, hoặc xóa tệp.



### 4. Viết và chỉnh sửa mã nguồn

#### Bước 1: Viết mã nguồn

Mở tệp mã nguồn bạn muốn chỉnh sửa bằng cách nhấp đúp vào tệp trong Explorer. VSCode hỗ trợ các tính năng như tự động hoàn thành mã, gợi ý đoạn mã, và highlight cú pháp.

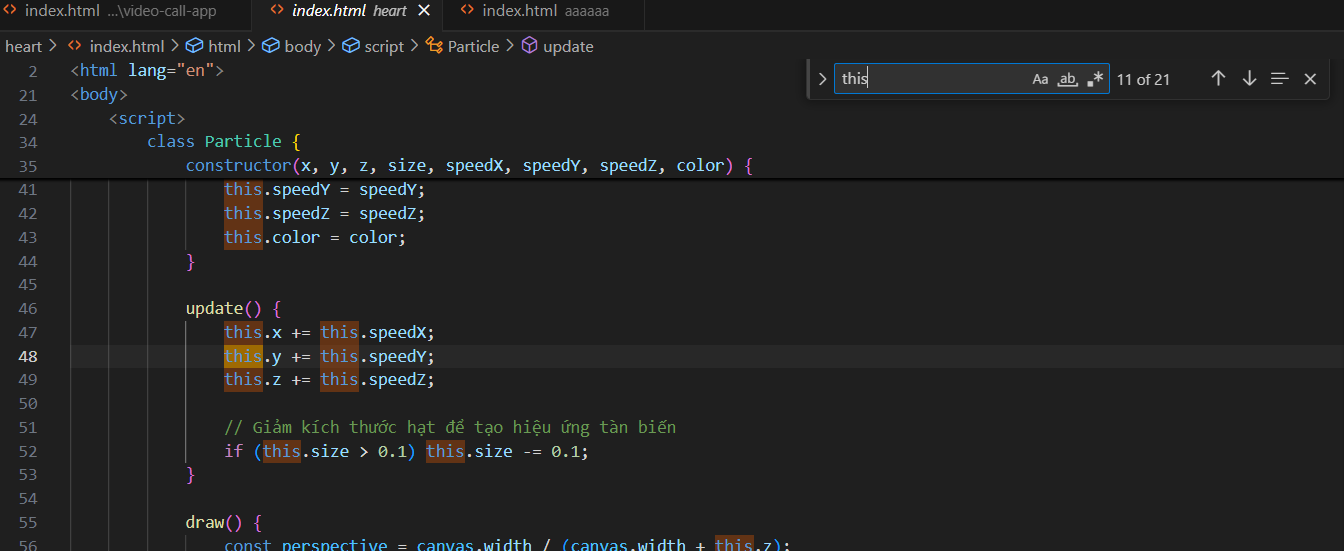


#### Bước 2: Định dạng mã

Sử dụng các tiện ích như Prettier để định dạng mã theo các quy tắc chuẩn. Nhấn tổ hợp phím Shift + Alt + F để định dạng tự động tệp hiện tại.

#### Bước 3: Tìm kiếm và thay thế

Nhấn tổ hợp phím Ctrl + F để mở thanh tìm kiếm, nhập từ khóa bạn muốn tìm. Nếu bạn muốn tìm kiếm và thay thế toàn bộ, nhấn Ctrl + Shift + H.



### 5. Debug và Chạy chương trình

#### Bước 1: Cấu hình Run/Debug

Nhấn vào biểu tượng Run and Debug. Tạo file cấu hình launch.json để thiết lập môi trường chạy và debug. VSCode sẽ tự động gợi ý các cấu hình phù hợp cho các ngôn ngữ khác nhau.

#### Bước 2: Thêm Breakpoints

Bạn có thể nhấp vào bên cạnh số dòng mã để thêm Breakpoint (điểm dừng) trong khi debug.

#### Bước 3: Chạy chương trình

Sau khi cấu hình xong, nhấn F5 hoặc chọn Run để chạy chương trình và bắt đầu quá trình debug.

### 6. Tích hợp Git và quản lý phiên bản

#### Bước 1: Cài đặt Git

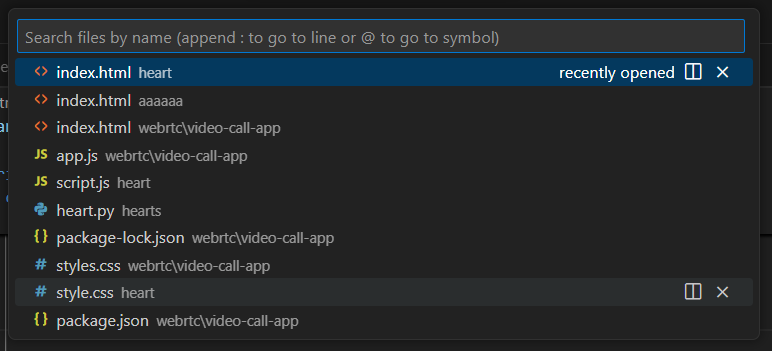
Nếu chưa cài đặt Git, hãy tải về và cài đặt từ https://git-scm.com/. Sau khi cài đặt Git, bạn có thể quản lý mã nguồn trực tiếp trong VSCode.

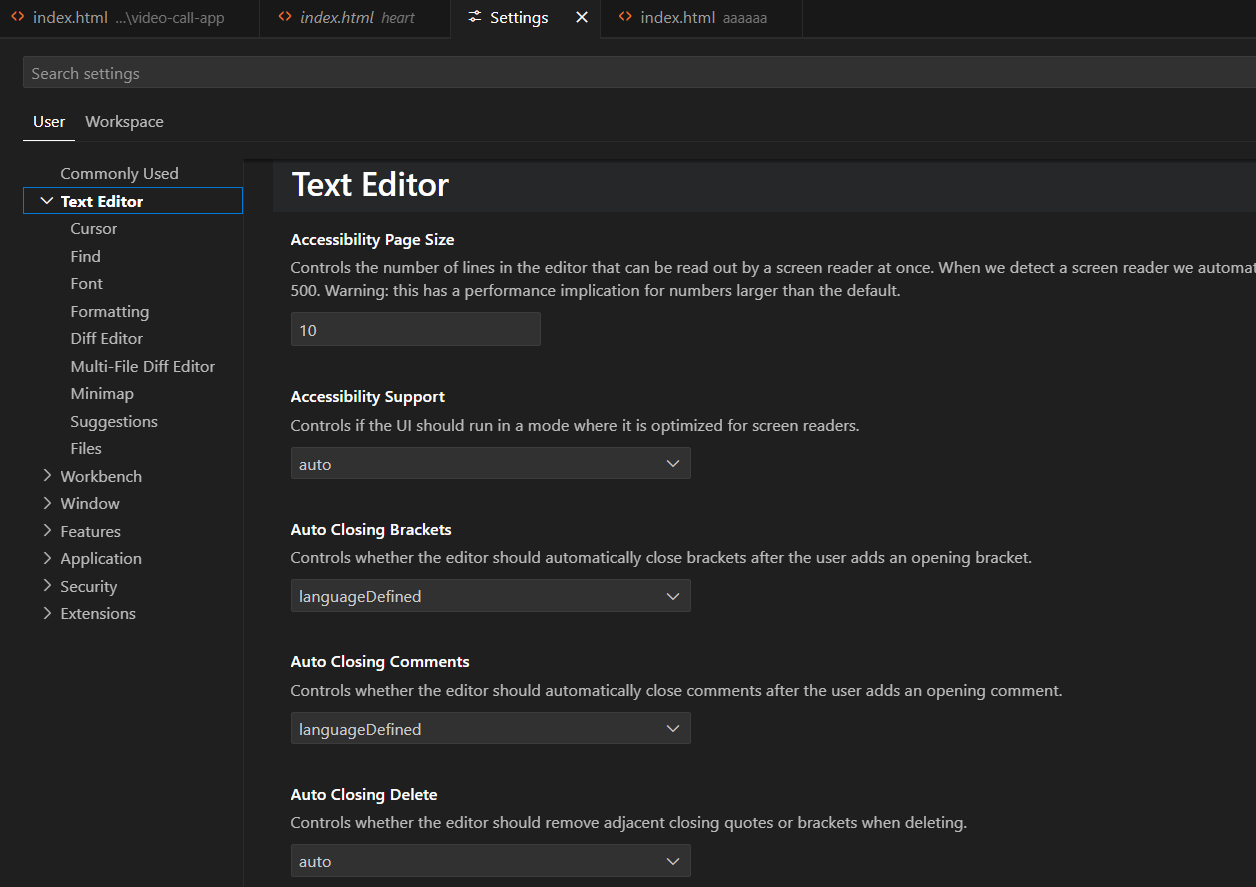
#### Bước 2: Sử dụng Git trong VSCode

Chọn tab Source Control (biểu tượng hình nhánh cây) để quản lý các thay đổi mã nguồn. Bạn có thể commit, push, pull, và xem lịch sử commit trực tiếp từ đây.

### 7. Cài đặt phím tắt (Keybindings)

VSCode hỗ trợ rất nhiều phím tắt để giúp lập trình viên thao tác nhanh hơn.  
  
Một số phím tắt mặc định phổ biến:  
Ctrl + P: Tìm kiếm tệp nhanh.

  
Ctrl + B: Ẩn/hiện thanh Explorer.  
Ctrl + `: Mở terminal tích hợp.



### 8. Cài đặt Terminal tích hợp

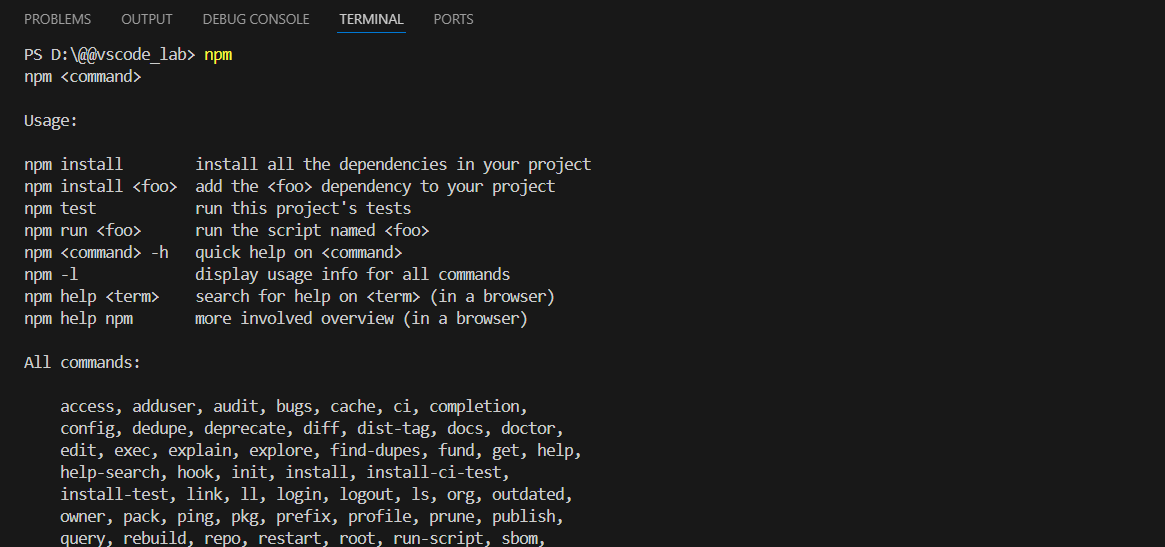
#### Bước 1: Mở Terminal

Trong VSCode, bạn có thể mở terminal tích hợp bằng cách nhấn Ctrl + `.



#### Bước 2: Sử dụng Terminal

Terminal tích hợp trong VSCode hỗ trợ các lệnh shell như git, npm, python trực tiếp.



### 9. Cài đặt Linter và Formatter

Linter giúp kiểm tra và phát hiện lỗi cú pháp, còn Formatter giúp định dạng mã nguồn.  
  
Cài đặt các tiện ích Linter như ESLint, Pylint từ tab Extensions. Sử dụng Prettier để định dạng mã JavaScript, HTML, CSS.

## *Firebase FireStore:*

* **Firebase Firestore** (Cloud Firestore) là một cơ sở dữ liệu NoSQL thuộc hệ sinh thái của Google Firebase, được sử dụng rộng rãi cho các ứng dụng web và mobile để lưu trữ và đồng bộ hóa dữ liệu theo thời gian thực.

### Kiến trúc NoSQL:

* Firestore lưu trữ dữ liệu dưới dạng các tài liệu (document) trong các bộ sưu tập (collection).
* Tài liệu là một đối tượng chứa các cặp key-value, có thể bao gồm dữ liệu dạng số, chuỗi, mảng, đối tượng, và thậm chí các tài liệu con (sub-collections).
* Không giống như các cơ sở dữ liệu quan hệ (RDBMS), Firestore không sử dụng bảng và hàng, mà sử dụng các tài liệu được tổ chức trong các bộ sưu tập.

### Tính năng đồng bộ thời gian thực:

* Firestore cung cấp tính năng đồng bộ hóa dữ liệu theo thời gian thực, giúp các ứng dụng phản hồi tức thì khi dữ liệu thay đổi mà không cần tải lại trang hay ứng dụng.
* Điều này rất hữu ích cho các ứng dụng yêu cầu dữ liệu cập nhật liên tục, như chat, các ứng dụng hợp tác, hay các trò chơi online.

### Tích hợp mạnh mẽ với Firebase

* Firestore tích hợp trực tiếp với các dịch vụ khác của Firebase như Authentication, Cloud Functions, và Firebase Hosting, giúp dễ dàng quản lý người dùng, thực thi logic backend, và triển khai ứng dụng nhanh chóng.
* Ngoài ra, Firestore hỗ trợ lưu trữ và truy vấn dữ liệu ở quy mô lớn với hiệu suất cao, nhờ vào khả năng mở rộng tự động.

### Khả năng truy vấn mạnh mẽ

* Firestore hỗ trợ truy vấn linh hoạt, bao gồm các phép lọc theo điều kiện, sắp xếp, giới hạn, và các truy vấn phức tạp như kết hợp nhiều điều kiện.
* Dữ liệu trong Firestore có thể được truy vấn dựa trên chỉ mục, giúp tăng hiệu suất.

### Bảo mật và kiểm soát quyền truy cập

* Sử dụng Firestore Security Rules, người phát triển có thể kiểm soát chi tiết quyền truy cập của người dùng đến dữ liệu. Các quy tắc này dựa trên các điều kiện như quyền hạn của người dùng và cấu trúc dữ liệu.
* Firestore cũng hỗ trợ Firebase Authentication để xác thực người dùng và bảo mật dữ liệu.

### Chế độ làm việc ngoại tuyến

* Firestore hỗ trợ chế độ ngoại tuyến. Khi không có kết nối mạng, Firestore sẽ lưu trữ tạm thời các thay đổi trên thiết bị và đồng bộ lại khi kết nối được phục hồi, giúp ứng dụng có thể tiếp tục hoạt động mượt mà.

### Quy mô và mở rộng

* Firestore được thiết kế để mở rộng linh hoạt, từ các dự án nhỏ đến các ứng dụng có hàng triệu người dùng mà không gặp phải các vấn đề về hiệu suất.

### Ưu điểm của Firestore

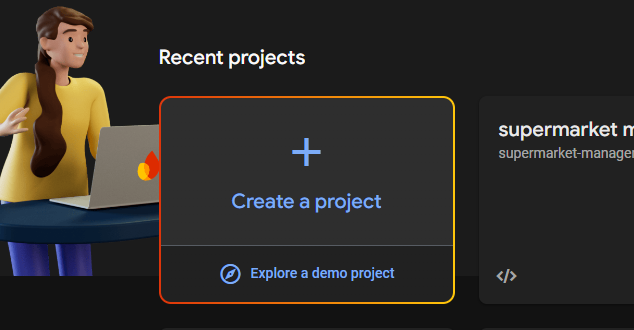
* Dễ sử dụng: Firestore có cú pháp đơn giản, tích hợp dễ dàng với các nền tảng web và mobile.
* Tính mở rộng: Khả năng xử lý lượng dữ liệu lớn và mở rộng theo nhu cầu.
* Đồng bộ hóa thời gian thực: Ứng dụng có thể cập nhật dữ liệu liên tục mà không cần tải lại.

### Cách dùng Firebase FireStore

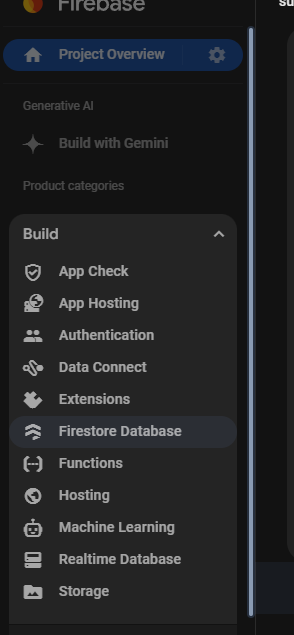
#### Cài đặt Firebase Firestore

##### Bước 1: Tạo dự án Firebase

1. Truy cập Firebase Console: https://console.firebase.google.com/ và tạo dự án.

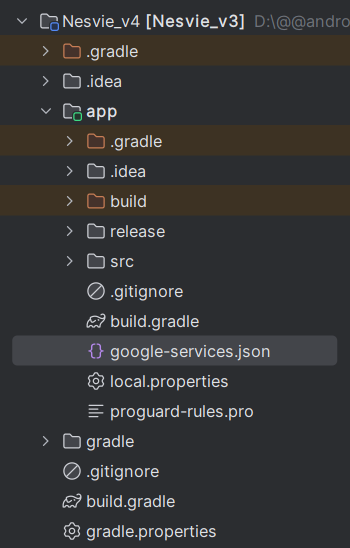


1. Kích hoạt Firestore trong phần Firestore Database.



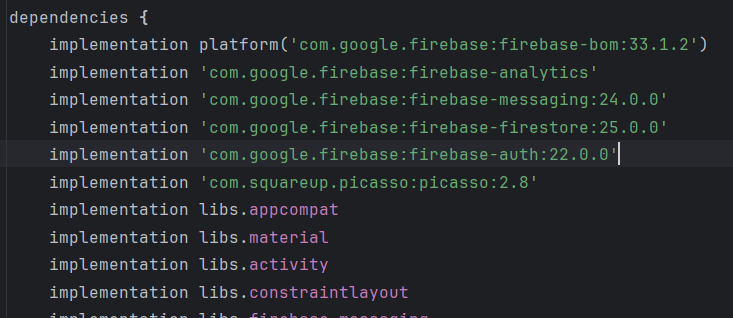
##### Bước 2: Thêm Firebase SDK vào Android

* + Thêm file google-services.json vào thư mục app/.



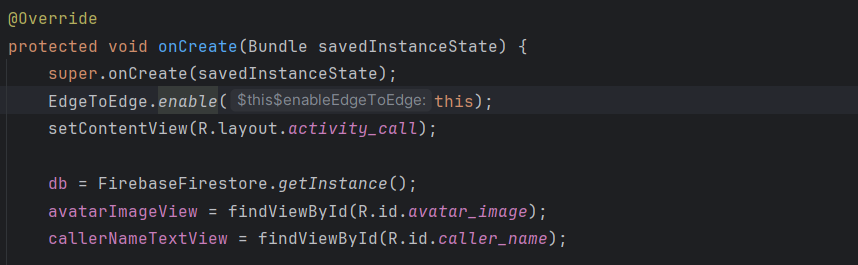
##### Bước 3: Cấu hình Gradle:

dependencies {  
 implementation platform('com.google.firebase:firebase-bom:32.0.0')  
 implementation 'com.google.firebase:firebase-firestore'  
}



##### Bước 4: Khởi tạo Firestore

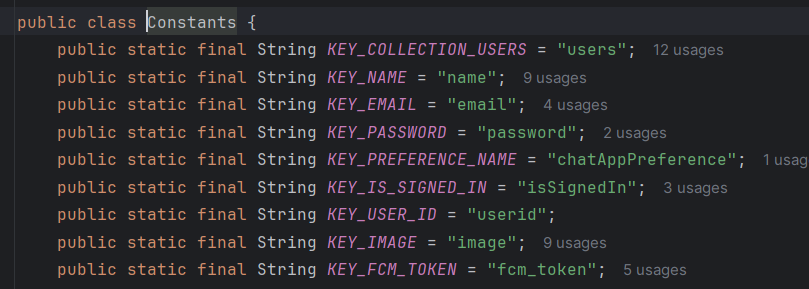
Để sử dụng Firestore, khởi tạo đối tượng Firestore:  
FirebaseFirestore db = FirebaseFirestore.getInstance();



#### Các thao tác cơ bản với Firestore

Thêm dữ liệu vào Firestore:

1. Tạo một HashMap để lưu dữ liệu:

2. Thêm dữ liệu vào Firestore:

db.collection("users").document("userID")  
.set(user)  
.addOnSuccessListener(aVoid -> Log.d("TAG", "DocumentSnapshot successfully written!"))  
.addOnFailureListener(e -> Log.w("TAG", "Error writing document", e));

Đọc dữ liệu từ Firestore:

db.collection("users").document("userID")  
.get()  
.addOnCompleteListener(task -> {  
 if (task.isSuccessful()) {  
 DocumentSnapshot document = task.getResult();  
 if (document.exists()) {  
 Log.d("TAG", "Document data: " + document.getData());  
 } else {  
 Log.d("TAG", "No such document");  
 }  
 } else {  
 Log.d("TAG", "get failed with ", task.getException());  
 }  
});

Cập nhật dữ liệu trong Firestore:

db.collection("users").document("userID")  
.update("born", 1820)  
.addOnSuccessListener(aVoid -> Log.d("TAG", "DocumentSnapshot successfully updated!"))  
.addOnFailureListener(e -> Log.w("TAG", "Error updating document", e));

Xóa dữ liệu trong Firestore:

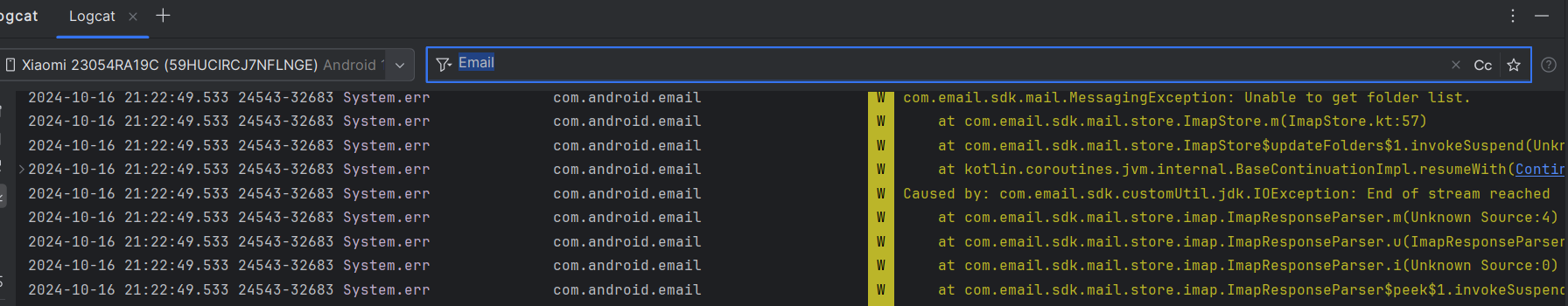
db.collection("users").document("userID")  
.delete()  
.addOnSuccessListener(aVoid -> Log.d("TAG", "DocumentSnapshot successfully deleted!"))  
.addOnFailureListener(e -> Log.w("TAG", "Error deleting document", e));

#### Lưu ý về bảo mật

Cấu hình quy tắc bảo mật trong Firestore bằng cách vào Firestore Database > Rules trong Firebase Console để điều chỉnh quyền truy cập dữ liệu.

#### Debugging và quản lý

Firebase Console cung cấp giao diện trực quan để kiểm tra dữ liệu đã lưu và kiểm tra log để phát hiện lỗi khi thao tác với Firestore.



# **Pha kiểm thử:**

* + Pha kiểm thử em vẫn đang tìm hiểu thêm về mấy tool nên chưa chọn được tool chính