

## LAB 7

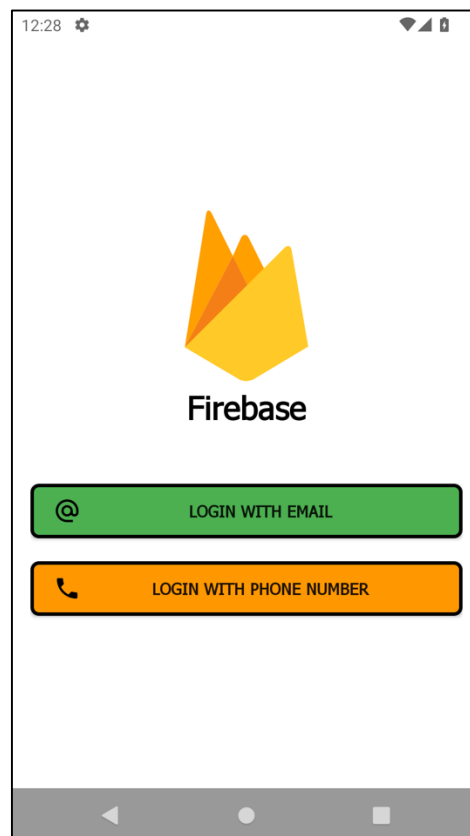
### MỤC TIÊU

Kết thúc bài thực hành sinh viên có khả năng:

- ✓ Biết cách sử dụng Firebase Authentication để đăng nhập đăng ký bằng email hay hay bằng số điện thoại

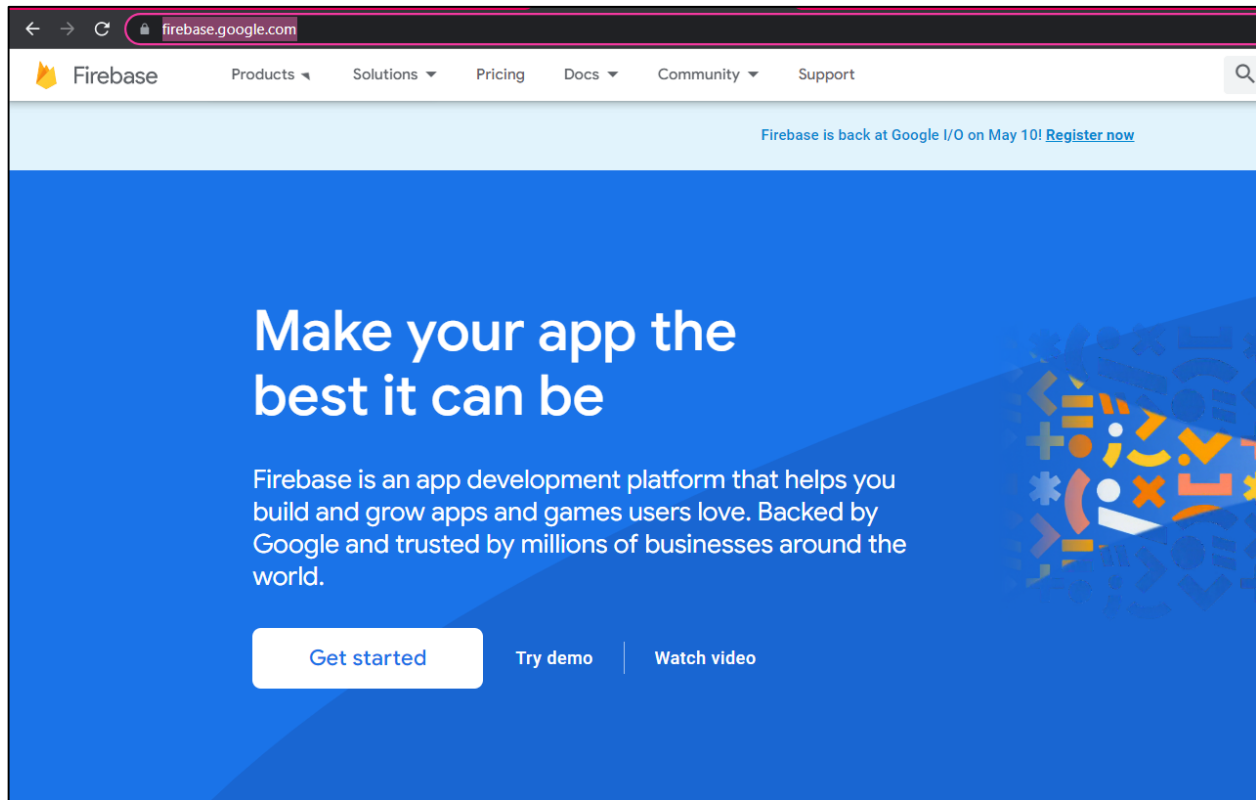
### NỘI DUNG

Thực hiện chức năng đăng nhập bằng Email, số điện thoại sử dụng Firebase Authentication

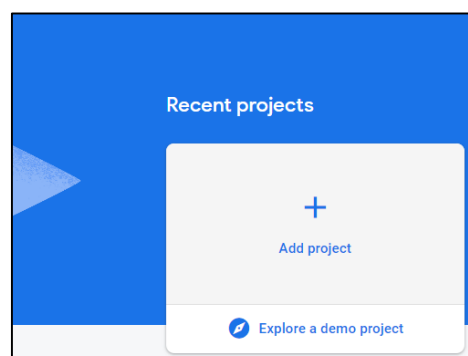


## BÀI 1: Setup Project Firebase

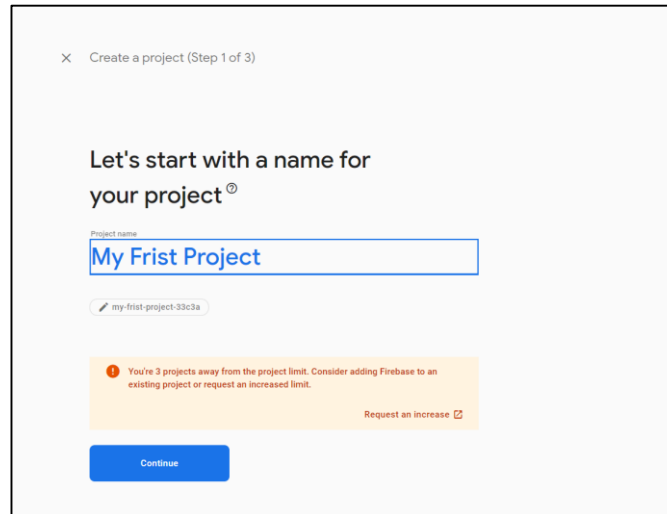
**Bước 1:** Truy cập vào <https://firebase.google.com/> đăng nhập tài khoản. Sau đó chọn **Get started**



Chọn **Add project**



## Đặt tên project và nhấn **Continue**



X Create a project (Step 1 of 3)

Let's start with a name for your project<sup>®</sup>

Project name

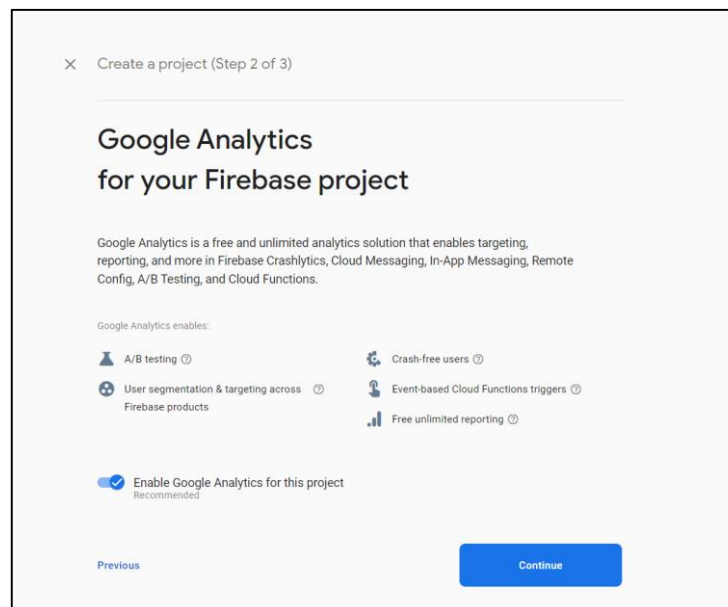
My Frist Project

my-frist-project-33c3a

**!** You're 3 projects away from the project limit. Consider adding Firebase to an existing project or request an increased limit. [Request an increase](#)

Continue

## Tiếp tục chọn **Continue**



X Create a project (Step 2 of 3)

### Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

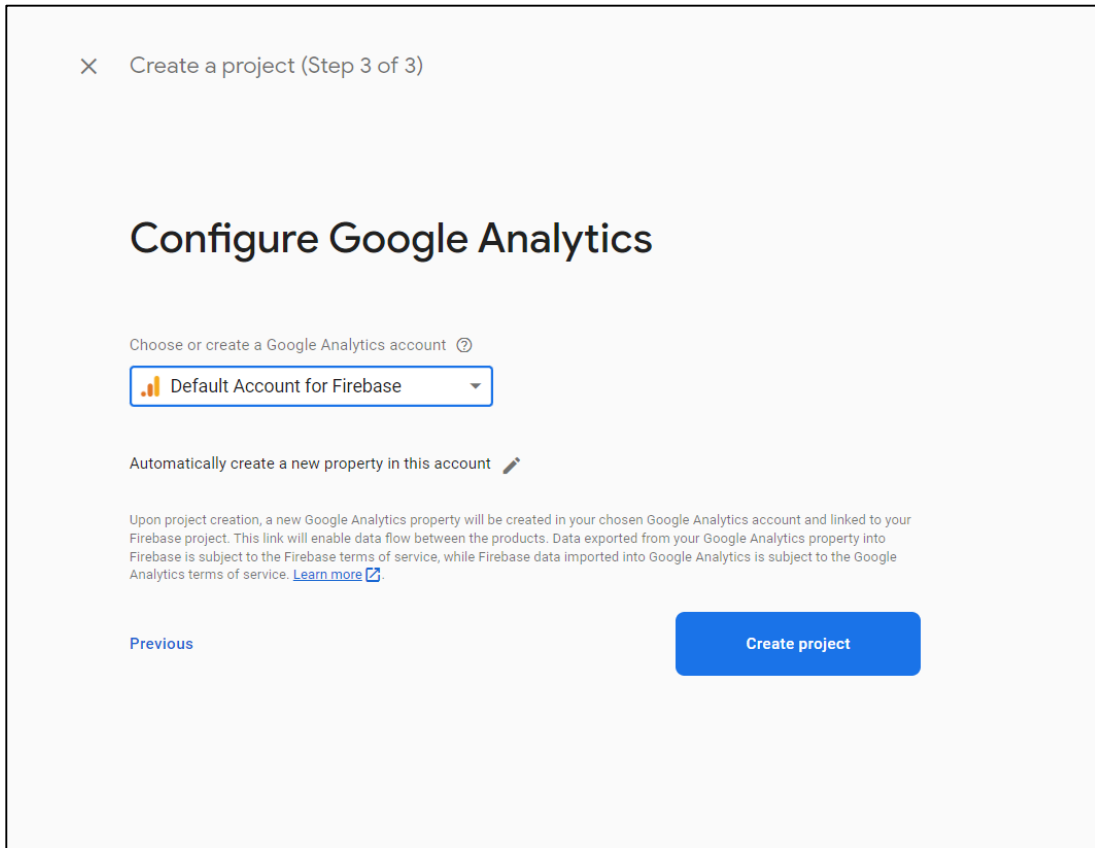
Google Analytics enables:

- A/B testing
- User segmentation & targeting across Firebase products
- Crash-free users
- Event-based Cloud Functions triggers
- Free unlimited reporting

☒ Enable Google Analytics for this project Recommended

Previous Continue


## Chọn **Default Account for Firebase** và nhấn **Create Project**




× Create a project (Step 3 of 3)

### Configure Google Analytics

Choose or create a Google Analytics account ?

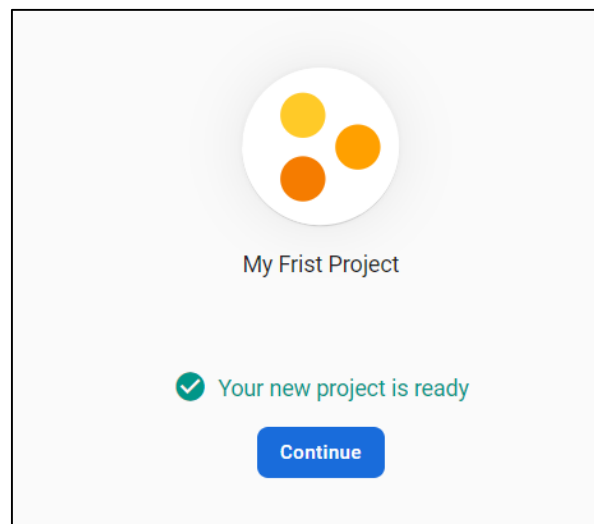
 Default Account for Firebase ▼

Automatically create a new property in this account ✎

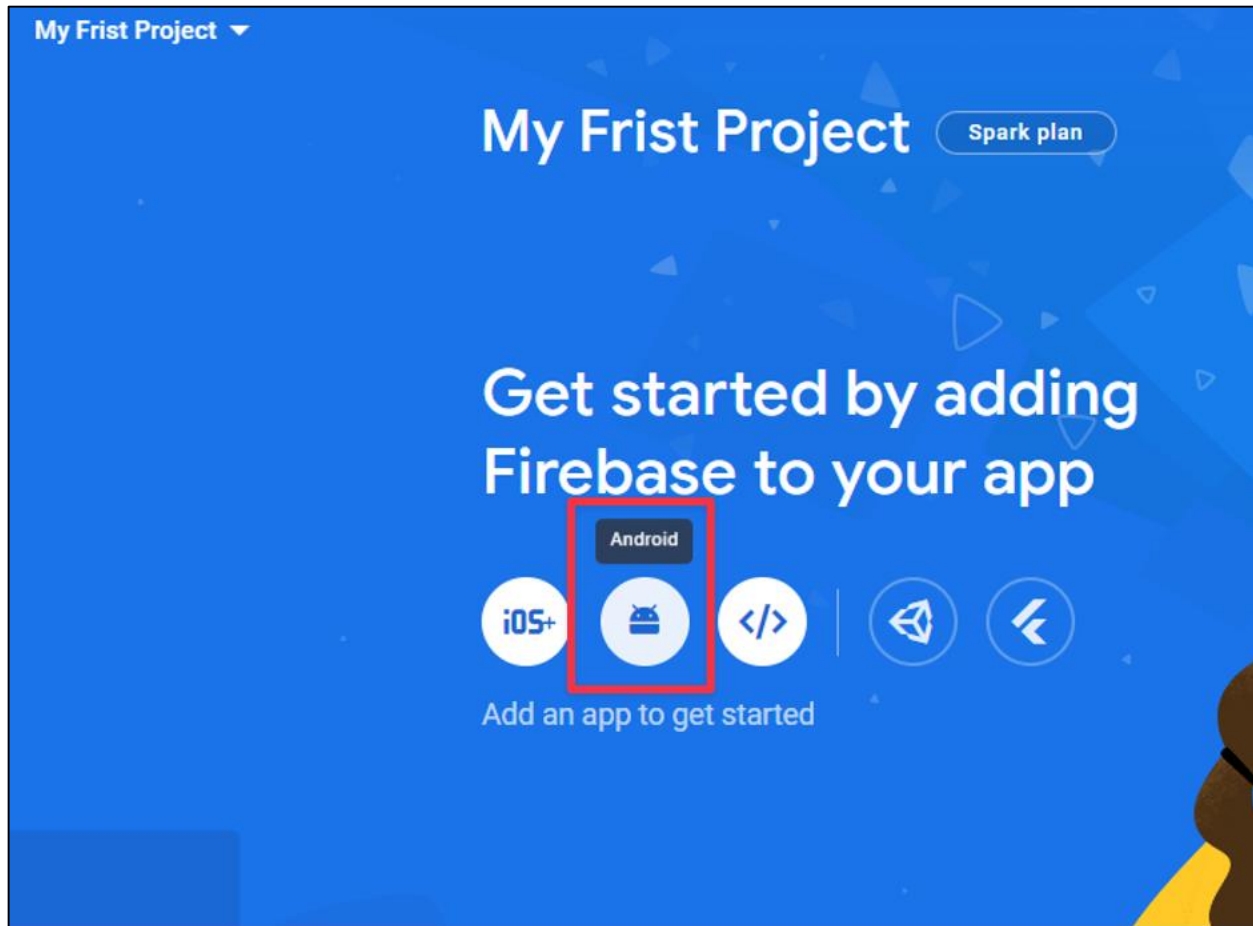
Upon project creation, a new Google Analytics property will be created in your chosen Google Analytics account and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#) .

[Previous](#) [Create project](#)

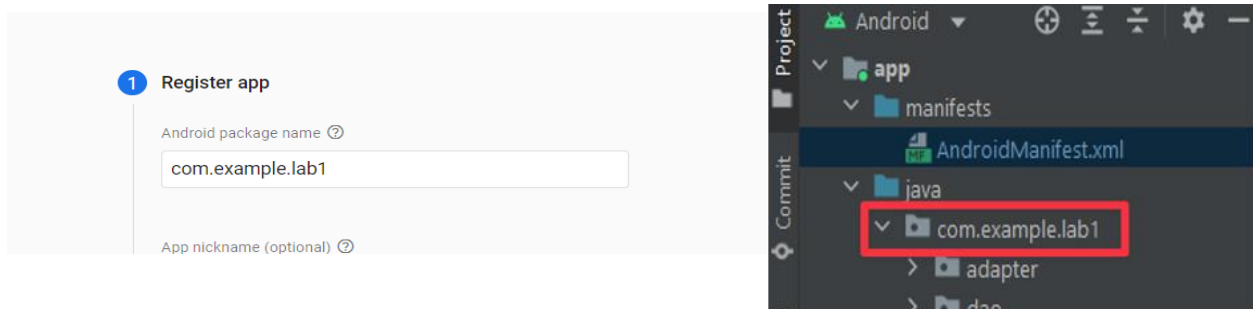
Sau khi xong khi Firebase tạo xong Project, ta chọn **Continue** để bắt đầu sử dụng



Chọn vào biểu tượng Android để bắt đầu setup



- Điền tên package của app



- App nickname đặt tên tùy ý

App nickname (optional) 

Authentication

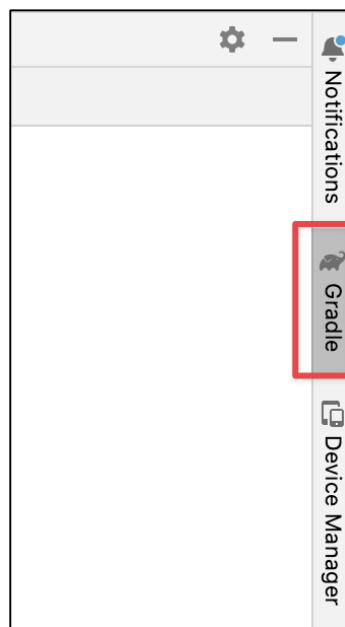
- Điền thông tin mã **SHA-1**, chúng ta sẽ lấy được mã này trong Android Studio

Debug signing certificate SHA-1 (optional) ?

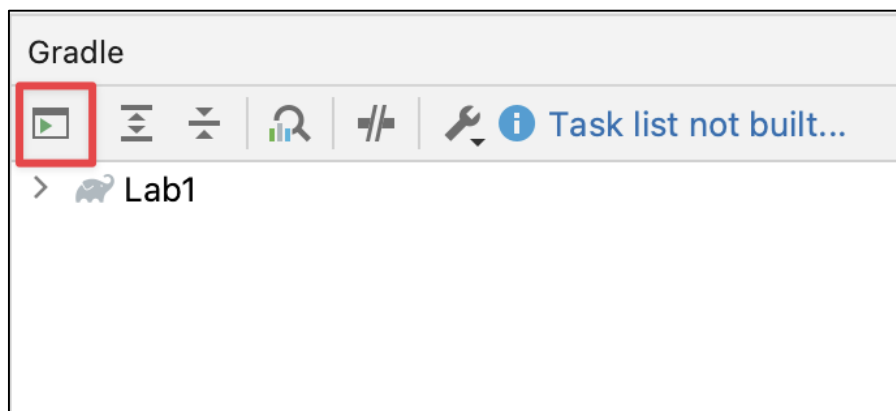
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:

i Required for Dynamic Links, and Google Sign-In or phone number support in Auth.  
Edit SHA-1s in Settings.

Trong Android Studio, ở góc phải màn hình chọn và tag **Gradle**



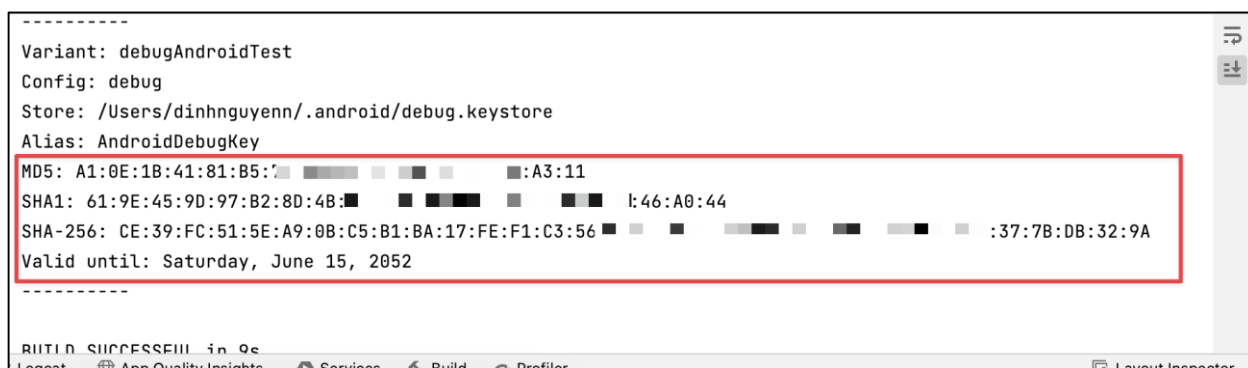
Chọn biểu tượng **Execute Gradle Task**



Gõ vào ô tìm kiếm **signReport** và nhấn enter



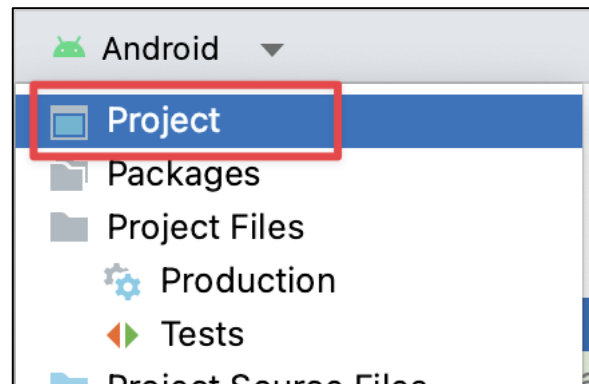
Sau đó sẽ có cửa sổ Terminal hiện lên copy SHA-1



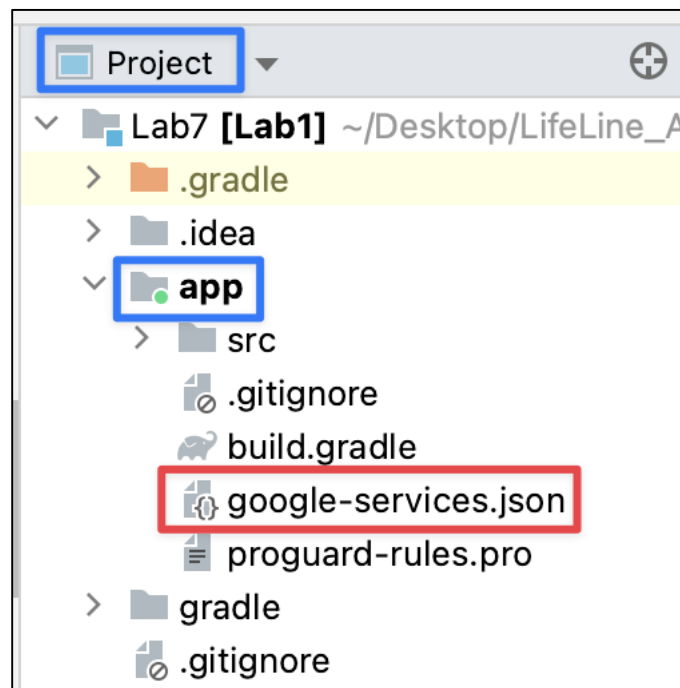




Paste file config vừa tải vào project Android bằng cách thay đổi cấu trúc view của project từ **Android** sang **Project**



Copy file config vừa tải vào thư mục **app**



### Bước 3: Thêm các thông số, thư viện cần thiết

3 Add Firebase SDK

Instructions for Gradle | [Unity C++](#)

1. To make the `google-services.json` config values accessible to Firebase SDKs, you need the Google services Gradle plugin.

Add the plugin as a buildscript dependency to your project-level `build.gradle` file:

Root-level (project-level) Gradle file (`<project>/build.gradle`):

```

buildscript {
    repositories {
        // Make sure that you have the following two repositories
        google() // Google's Maven repository
        mavenCentral() // Maven Central repository
    }
    dependencies {
        ...
        // Add the dependency for the Google services Gradle plugin
        classpath 'com.google.gms:google-services:4.3.15'
    }
}

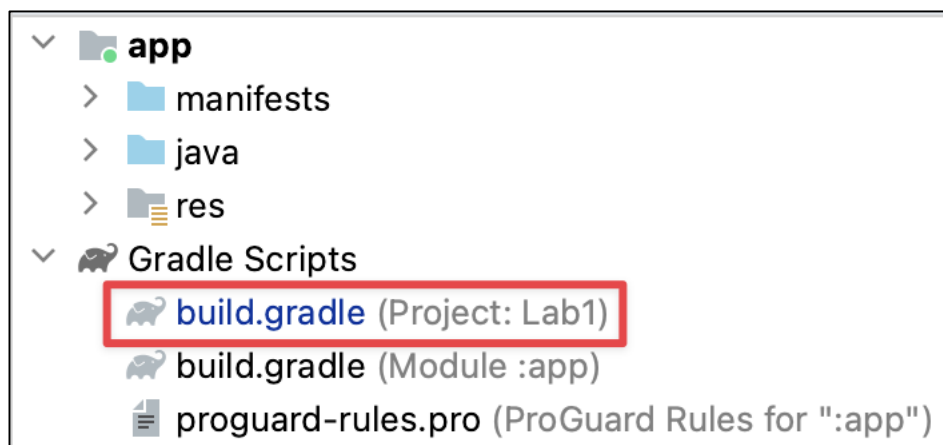
allprojects {
    ...
    repositories {
        // Make sure that you have the following two repositories
        google() // Google's Maven repository
        mavenCentral() // Maven Central repository
    }
}

```

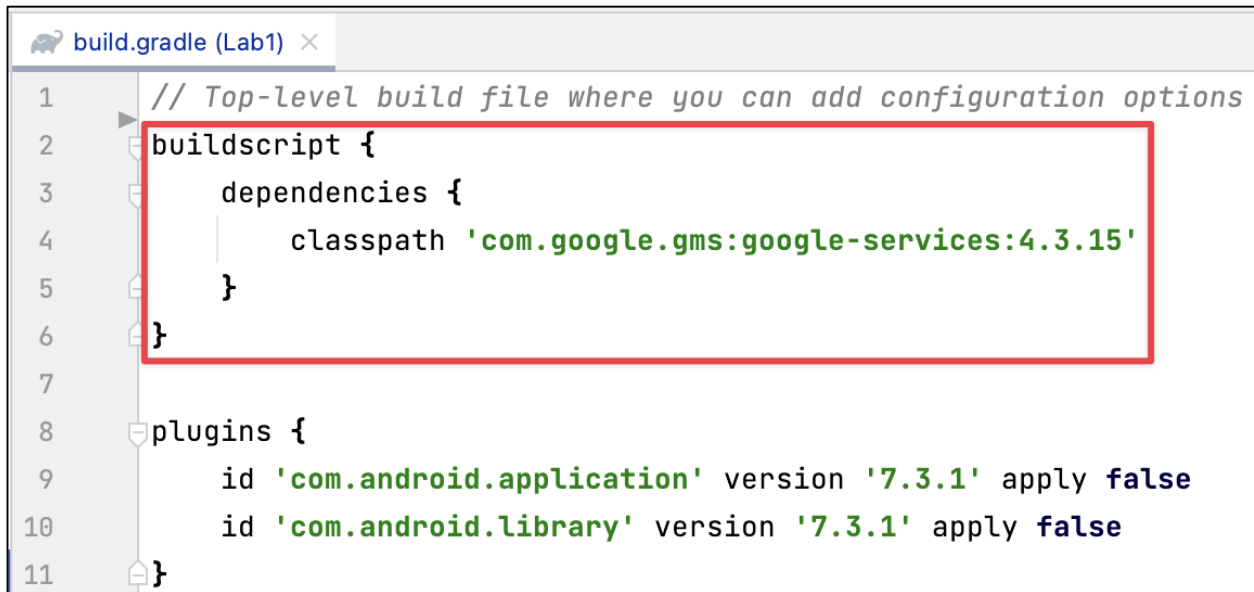
2. Then, in your module (app-level) `build.gradle` file, add both the `google-services` plugin and any Firebase SDKs that you want to use in your app:

☐ Kotlin
 ☒ Java

### Trong `build.gradle` (Module: Project)

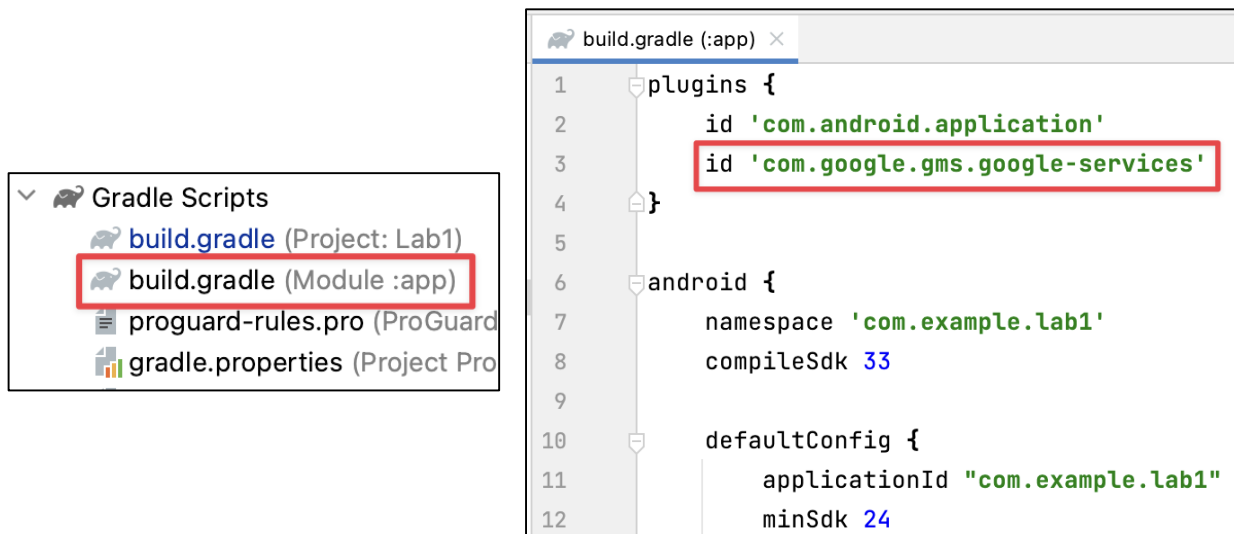


Thêm đoạn code sau (copy ở bước 3 trên Firebase)



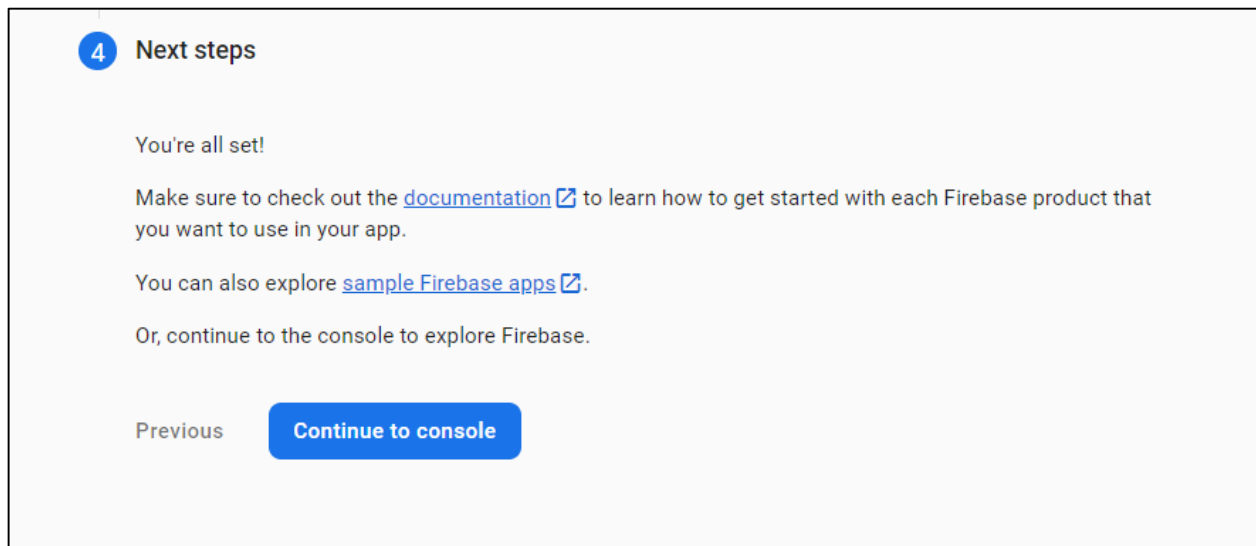
```
1 // Top-level build file where you can add configuration options
2 buildscript {
3     dependencies {
4         classpath 'com.google.gms:google-services:4.3.15'
5     }
6 }
7
8 plugins {
9     id 'com.android.application' version '7.3.1' apply false
10    id 'com.android.library' version '7.3.1' apply false
11 }
```

Tiếp theo trong file **build.gradle (Module)**, phần plugins thêm đoạn code:



```
1 plugins {
2     id 'com.android.application'
3     id 'com.google.gms.google-services'
4 }
5
6 android {
7     namespace 'com.example.lab1'
8     compileSdk 33
9
10    defaultConfig {
11        applicationId "com.example.lab1"
12        minSdk 24
13    }
14 }
```

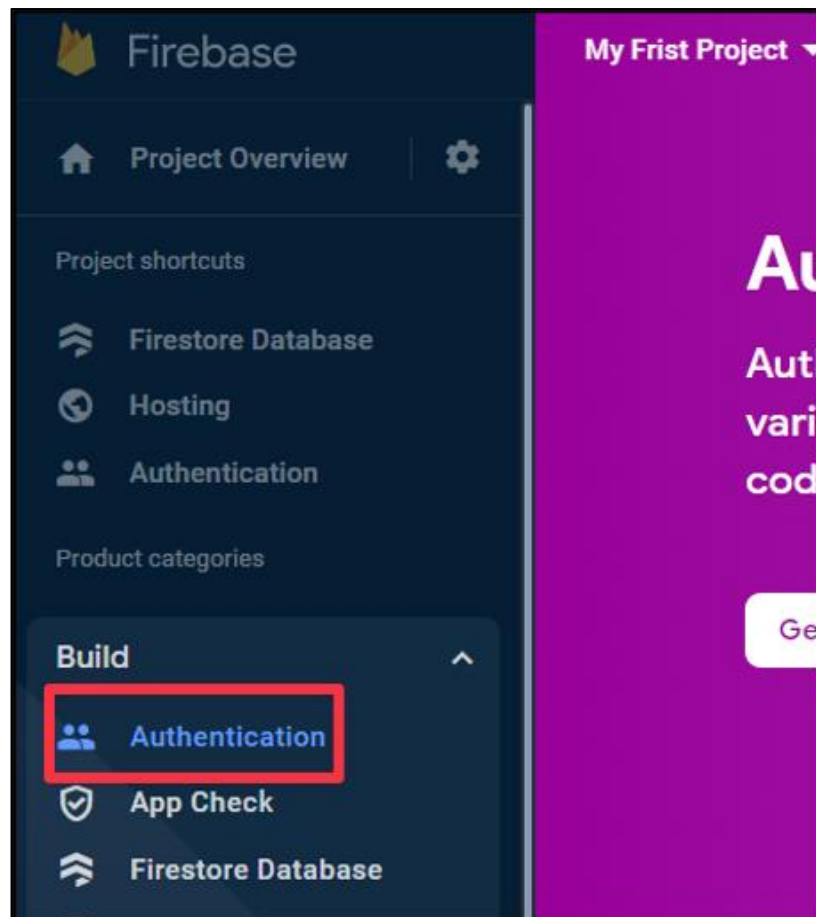
Trên Firebase nhấn **Next** và tiếp tục nhấn **Continue to console**



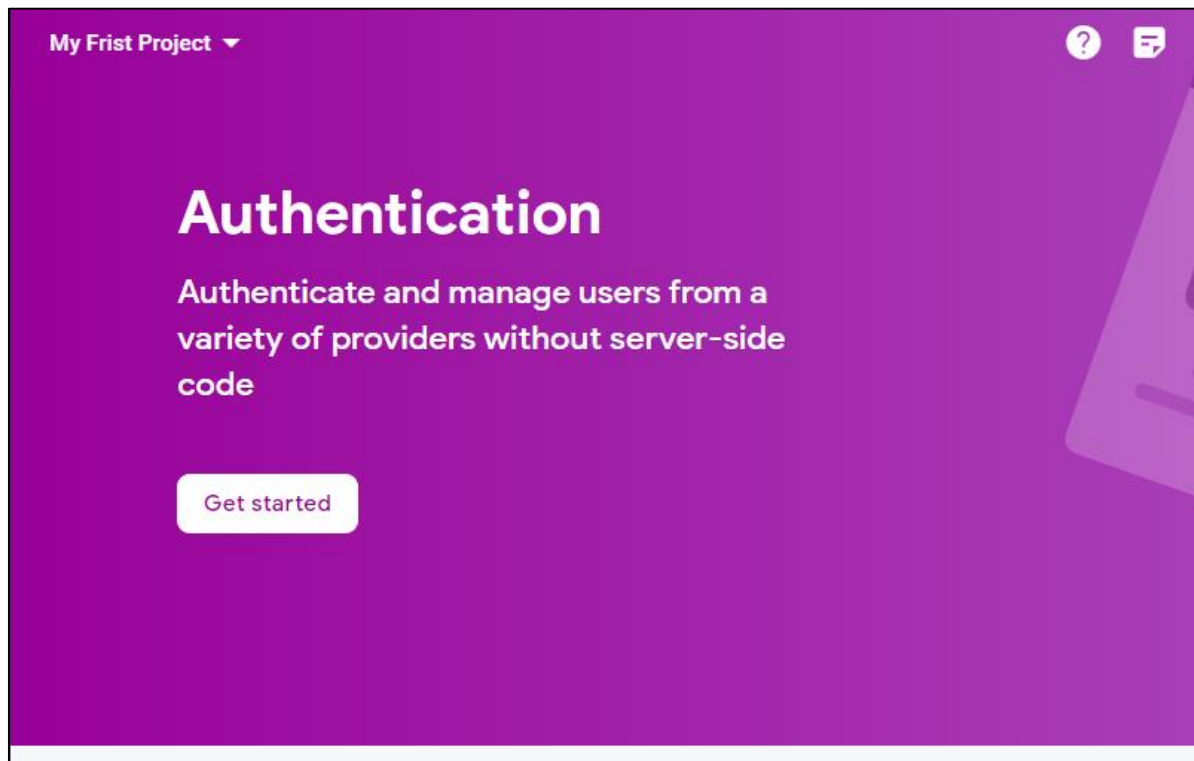
Ta đã hoàn thành xong bước tạo project trên Firebase.

## BÀI 2: Thực hiện chức năng nhập, đăng ký tài khoản bằng email sử dụng Firebase Authentication

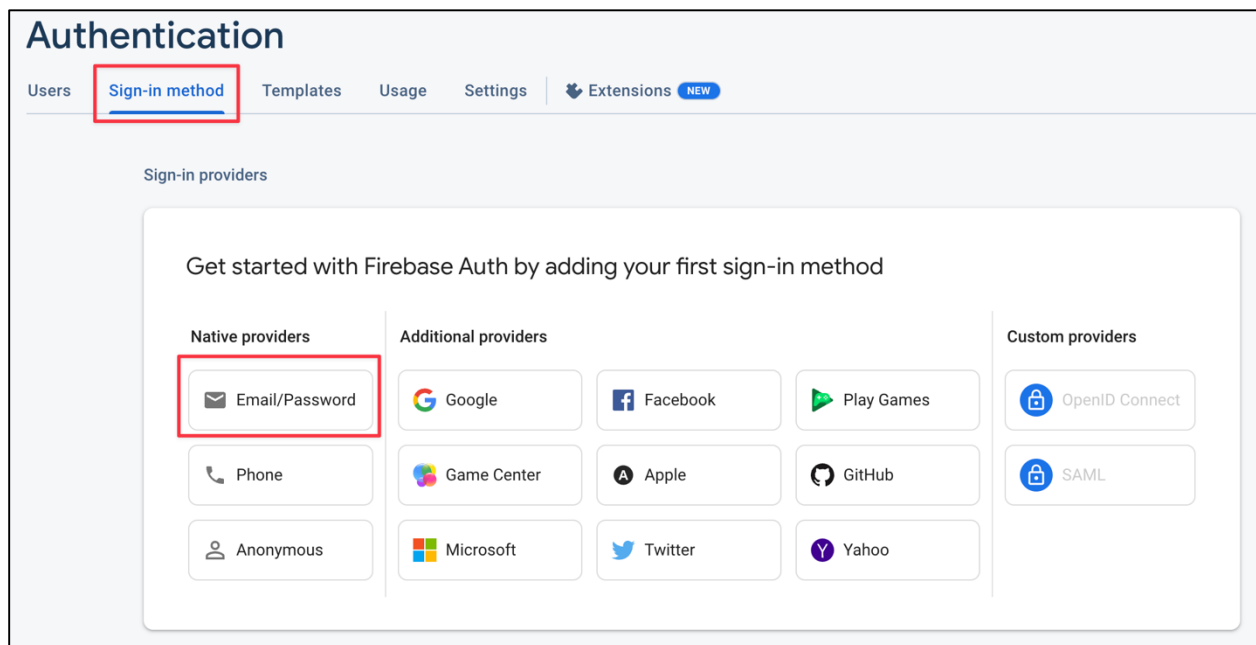
Bước 1: Tại console Firebase, ta chọn **Build** chọn **Authentication**



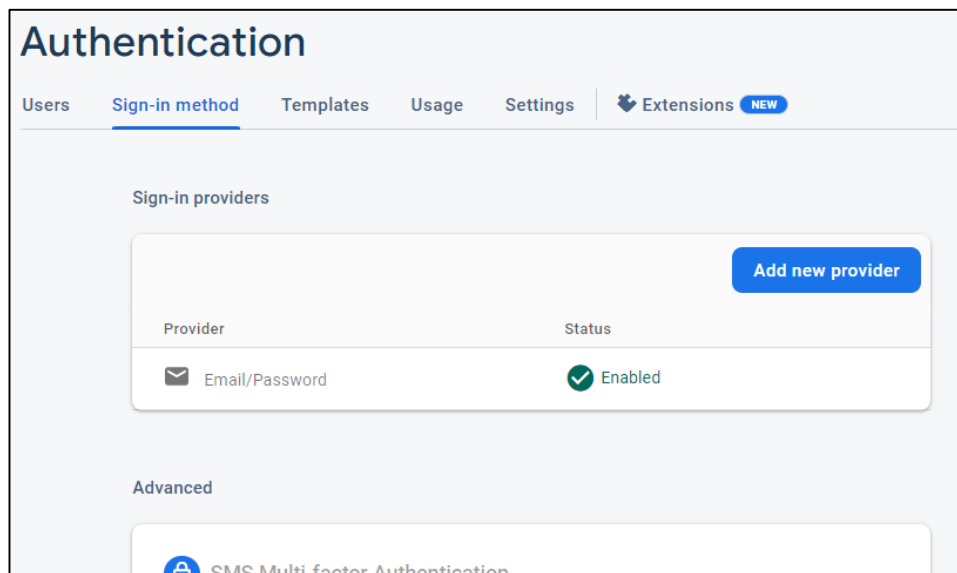
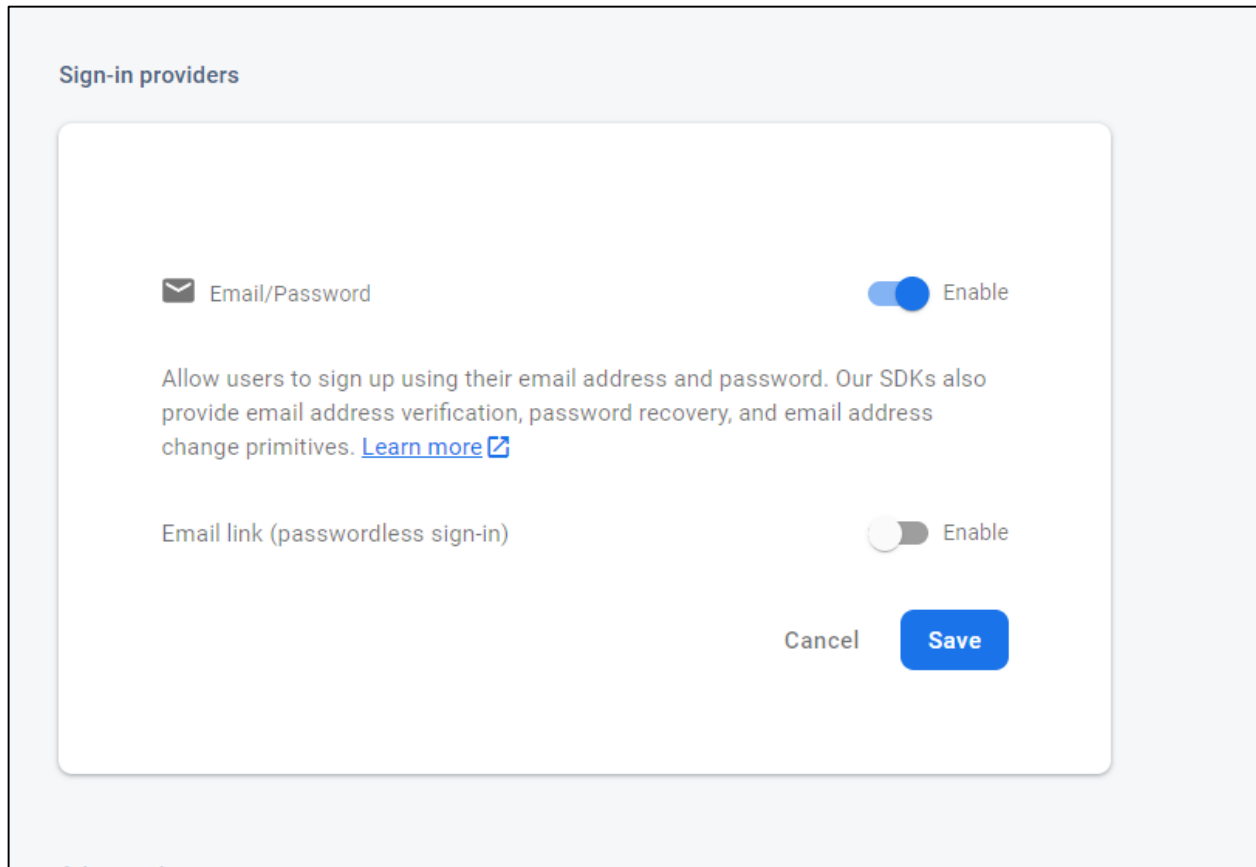
## Chọn **Get started**



Tại tab **Sign-in method** ta chọn vào **Email/Password**



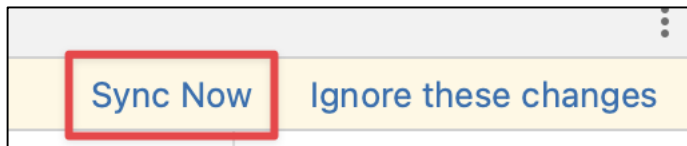
**Enable** phương thức đăng nhập bằng email/password sau đó nhấn nút **Save**



Tại **dependencies** trong **build.gradle (Moule)** thêm thư viện:

```
//Firebase
implementation platform('com.google.firebase:firebase-bom:31.2.3')
implementation 'com.google.firebase:firebase-analytics'
//Firebase Authentication
implementation 'com.google.firebase:firebase-auth-ktx'
```

Sau đó nhấn **Sync Now**



**Bước 2:** Tạo 3 giao diện đăng ký, đăng nhập và đăng xuất như hình bên dưới



**Bước 3:** Thực hiện chức năng đăng ký tài khoản



Tại activity đăng ký, khởi tạo **FirebaseAuth** và gán giá trị cho nó ở hàm **onCreate**

```
public class SignUpActivity extends AppCompatActivity {
    1 usage
    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_up);
        mAuth = FirebaseAuth.getInstance();
    }
}
```

Sau đó ta viết hàm đăng ký tài khoản

```
mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if (task.isSuccessful()) {
            //lấy thông tin tài khoản mới vừa đăng ký
            FirebaseUser user = mAuth.getCurrentUser();
            Toast.makeText( context: SignUpActivity.this, text: "Đăng ký thành công", Toast.LENGTH_SHORT).show();
            finish();
        } else {
            Log.w(TAG, msg: "createUserWithEmail:failure", task.getException());
            Toast.makeText( context: SignUpActivity.this, text: "Đăng ký thất bại", Toast.LENGTH_SHORT).show();
        }
    }
});
```

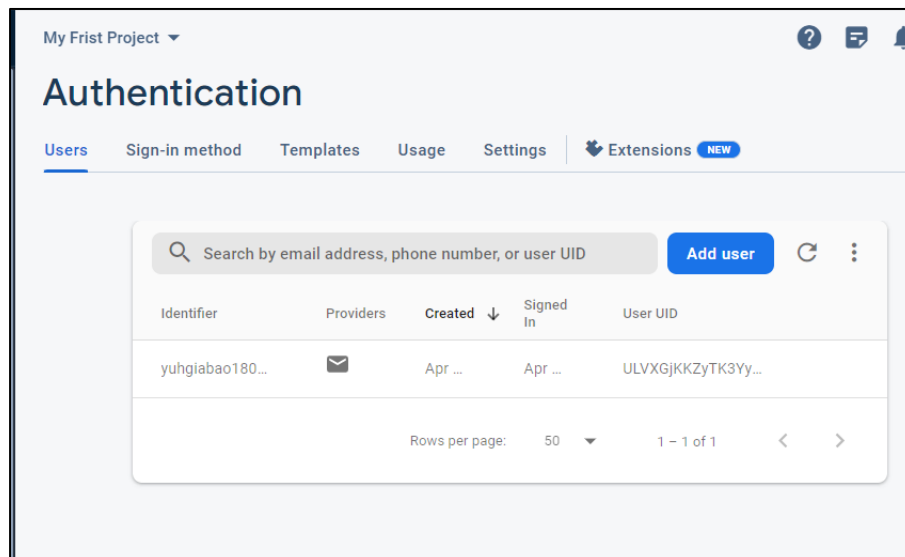
#### Bước 4: Tương tự ở bước 3, ta thực hiện chức năng đăng nhập

```

mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if (task.isSuccessful()) {
            //lấy thông tin tài khoản mới vừa đăng nhập
            FirebaseUser user = mAuth.getCurrentUser();
            Toast.makeText( context: LoginActivity.this, text: "Đăng nhập thành công", Toast.LENGTH_SHORT).show();
            startActivity(new Intent( packageContext: LoginActivity.this, LogoutActivity.class));
        } else {
            // If sign in fails, display a message to the user.
            Log.w(TAG, msg: "signInWithEmail:failure", task.getException());
            Toast.makeText( context: LoginActivity.this, text: "Đăng nhập thất bại", Toast.LENGTH_SHORT).show();
        }
    }
});

```

Ta có thể kiểm tra thông tin các tài khoản đã đăng ký ở Console Firebase



The screenshot shows the 'Authentication' section of the Firebase console for 'My First Project'. The 'Users' tab is selected, displaying a table of registered users. The table has columns for Identifier, Providers, Created, Signed In, and User UID. One user is listed with the identifier 'yuhgiabao180...', a provider of 'Email', and a creation date of 'Apr ...'. The 'Signed In' status is also 'Apr ...'. The 'User UID' is 'ULVXGjKKZyTK3Yy...'. At the bottom, there are controls for 'Rows per page' (set to 50) and pagination (1 - 1 of 1).

Identifier	Providers	Created	Signed In	User UID
yuhgiabao180...	Email	Apr ...	Apr ...	ULVXGjKKZyTK3Yy...

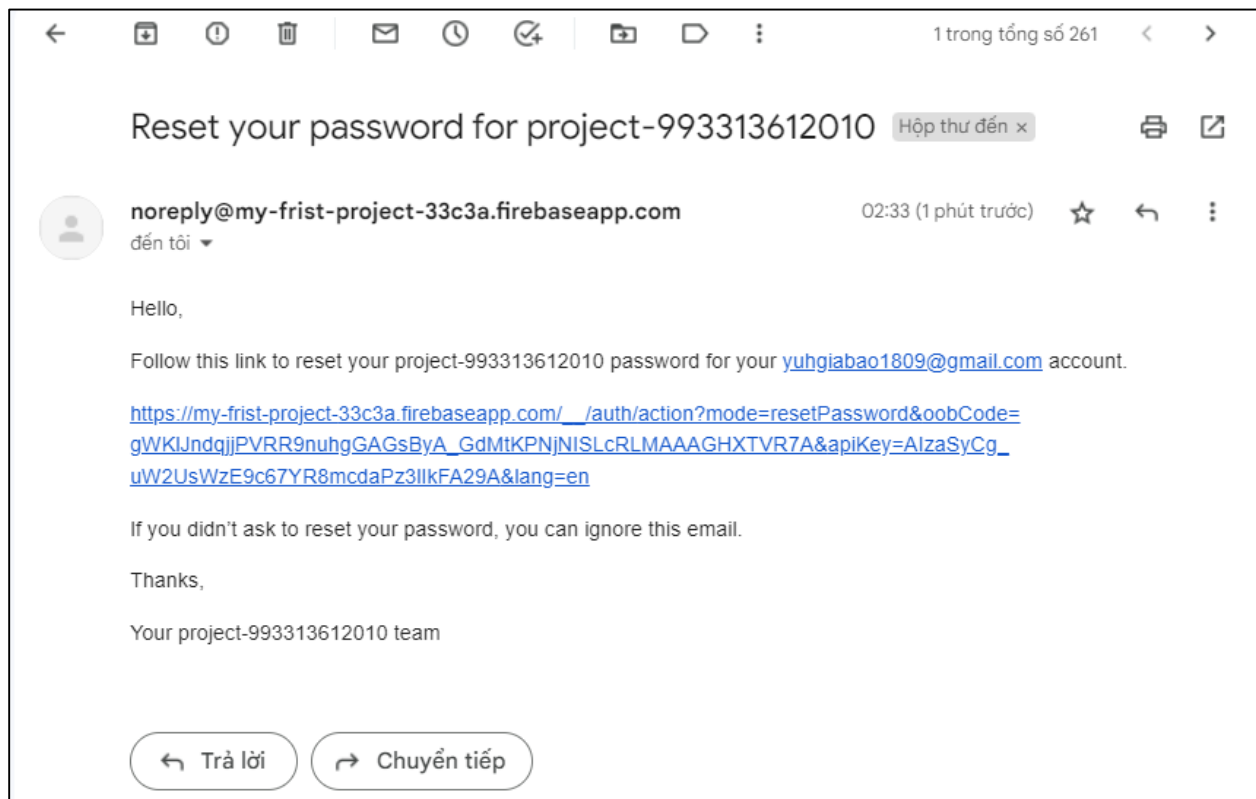
**Bước 5:** Thực hiện chức năng đăng xuất ta chỉ cần gọi phương thức signOut

```
FirebaseAuth.getInstance().signOut();
```

**Bước 6:** Ngoài ra chúng ta có thể xử lý thêm chức năng quên mật khẩu

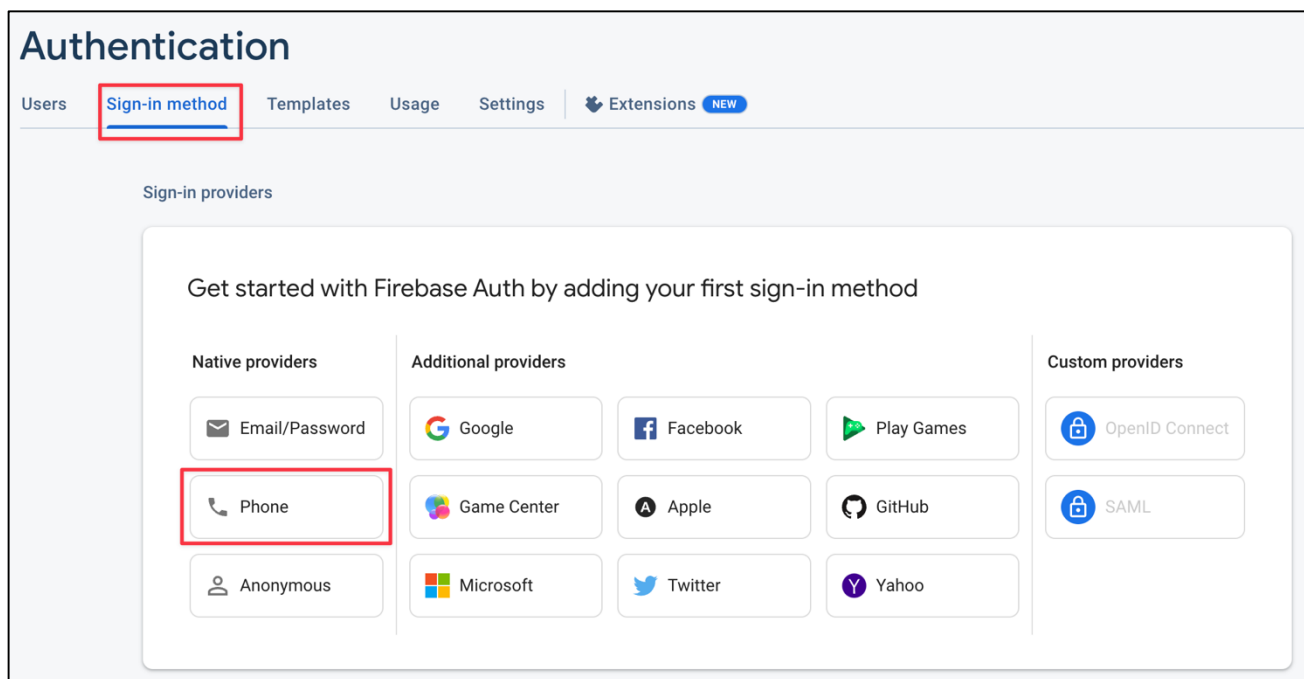
```
auth.sendPasswordResetEmail(emailAddress).addOnCompleteListener(new OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if (task.isSuccessful()) {
            Toast.makeText(context: LoginActivity.this, text: "Vui lòng kiểm tra hộp thư để cập nhật lại mật khẩu", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(context: LoginActivity.this, text: "Lỗi gửi mail", Toast.LENGTH_SHORT).show();
        }
    }
});
```

Sau khi gọi hàm quên mật khẩu thành công, ta sẽ nhận được mail để reset password

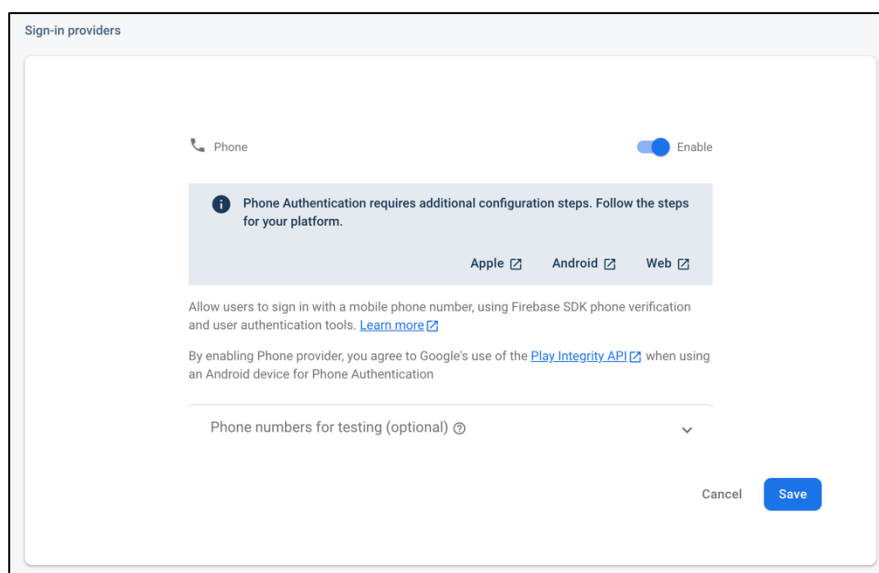


## BÀI 2: Thực hiện chức năng nhập thông qua số điện thoại và xác thực bằng OTP

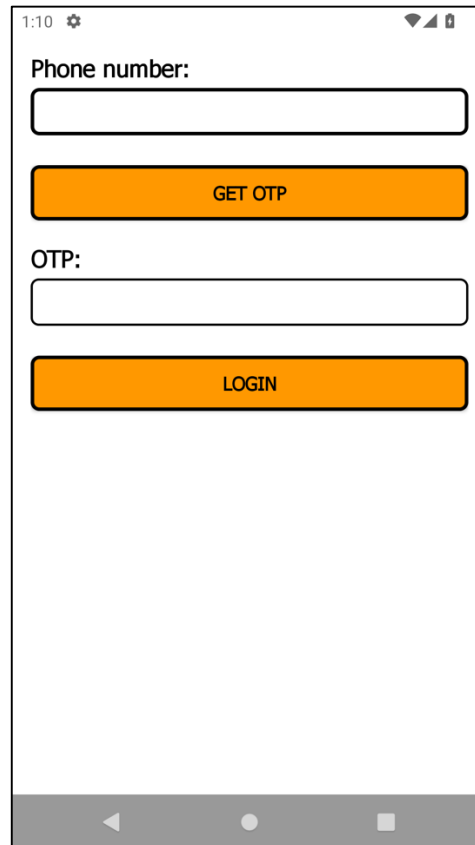
### Bước 1: Tại tab **Sign-in method** ta chọn vào **Phone**



### Enable phương thức đăng nhập bằng số điện thoại sau đó nhấn nút **Save**



### Bước 2: Tạo một giao diện cơ bản như hình



### Bước 3: Khởi tạo FirebaseAuth và PhoneAuthProvider

```
public class LoginWithPhoneActivity extends AppCompatActivity {  
  
    1 usage  
    private FirebaseAuth mAuth;  
    private PhoneAuthProvider.OnVerificationStateChangedCallbacks mCallbacks;  
    private String mVerificationId;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_login_with_phone);  
  
        mAuth = FirebaseAuth.getInstance();  
    }  
}
```

### Bước 4: Tạo hàm **getOTP** để yêu cầu Firebase gửi OTP xác thực về số điện thoại

```
private void getOTP(String phoneNumber) {
    PhoneAuthOptions options =
        PhoneAuthOptions.newBuilder(mAuth)
            .setPhoneNumber("+84"+phoneNumber) // Phone number to verify
            .setTimeout(timeout: 60L, TimeUnit.SECONDS) // Timeout and unit
            .setActivity(this) // (optional) Activity for callback binding
            // If no activity is passed, reCAPTCHA verification can not be used.
            .setCallbacks(mCallbacks) // OnVerificationStateChangedCallbacks
            .build();
    PhoneAuthProvider.verifyPhoneNumber(options);
}
```

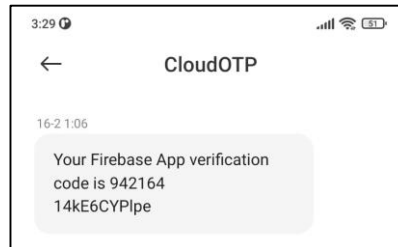
Trong **onCreate** ta khởi tạo **PhoneAuthProvider**

```
//Hàm này sẽ tự động được gọi khi chúng ta gửi yêu cầu lấy OTP
mCallbacks = new PhoneAuthProvider.OnVerificationStateChangedCallbacks() {
    @Override
    public void onVerificationCompleted(@NonNull PhoneAuthCredential credential) {
        //Khi số điện thoại đăng nhập của bạn nằm trong máy bạn thực hiện getOTP
        //Chúng có thể lập tức lấy mã otp và setText lên ô nhập otp mà không cần phải nhập
        otp.setText(credential.getSmsCode());
    }

    @Override
    public void onVerificationFailed(@NonNull FirebaseException e) {
        //Khi fail sẽ chạy vào hàm này
    }

    @Override
    public void onCodeSent(@NonNull String verificationId,
        PhoneAuthProvider.@NonNull ForceResendingToken token) {
        //Hàm này sẽ được chạy khi otp gửi thành công, ta sẽ lấy verificationId
        mVerificationId = verificationId;
    }
};
```

Ta được kết quả sau khi Firebase gửi OTP về số điện thoại:



### Bước 5: Viết hàm thực hiện xác thực OTP

```
private void verifyOTP(String code)
{
    PhoneAuthCredential credential = PhoneAuthProvider.getCredential(mVerificationId, code);
    signInWithPhoneAuthCredential(credential);
}
1 usage
private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {
    mAuth.signInWithCredential(credential)
        .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the signed-in user's information
                    Toast.makeText( context: LoginWithPhoneActivity.this, text: "Đăng nhập thành công", Toast.LENGTH_SHORT).show();
                    FirebaseUser user = task.getResult().getUser();
                    startActivity(new Intent( packageContext: LoginWithPhoneActivity.this, LogoutActivity.class));
                    // Update UI
                } else {
                    // Sign in failed, display a message and update the UI
                    Log.w(TAG, msg: "signInWithCredential:failure", task.getException());
                    if (task.getException() instanceof FirebaseAuthInvalidCredentialsException) {
                        // The verification code entered was invalid
                    }
                }
            }
        });
}
```