

## LẬP TRÌNH ĐA NỀN TẢNG 2

BÀI 7: GIỚI THIỆU VỀ FIREBASE, ĐĂNG  
NHẬP EMAIL/PASSWOR VÀ GOOGLE

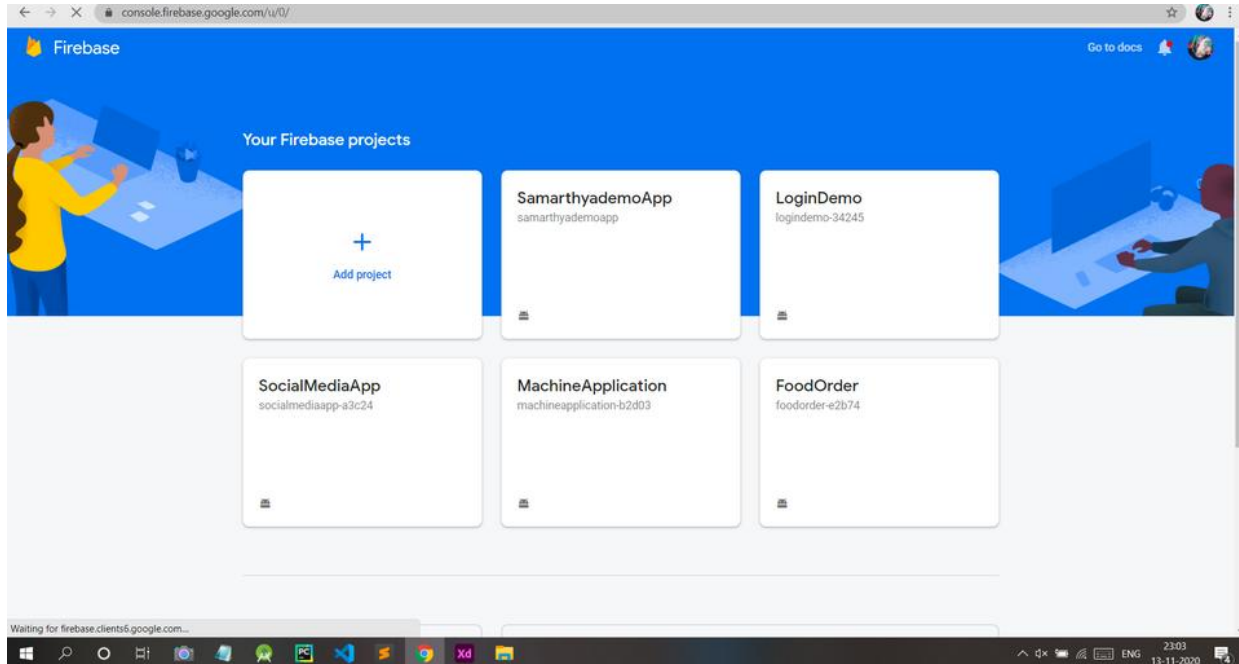
PHẦN 1: GIỚI THIỆU VỀ FIREBASE

- ☐ Giới thiệu về firebase
- ☐ Setup Firebase console

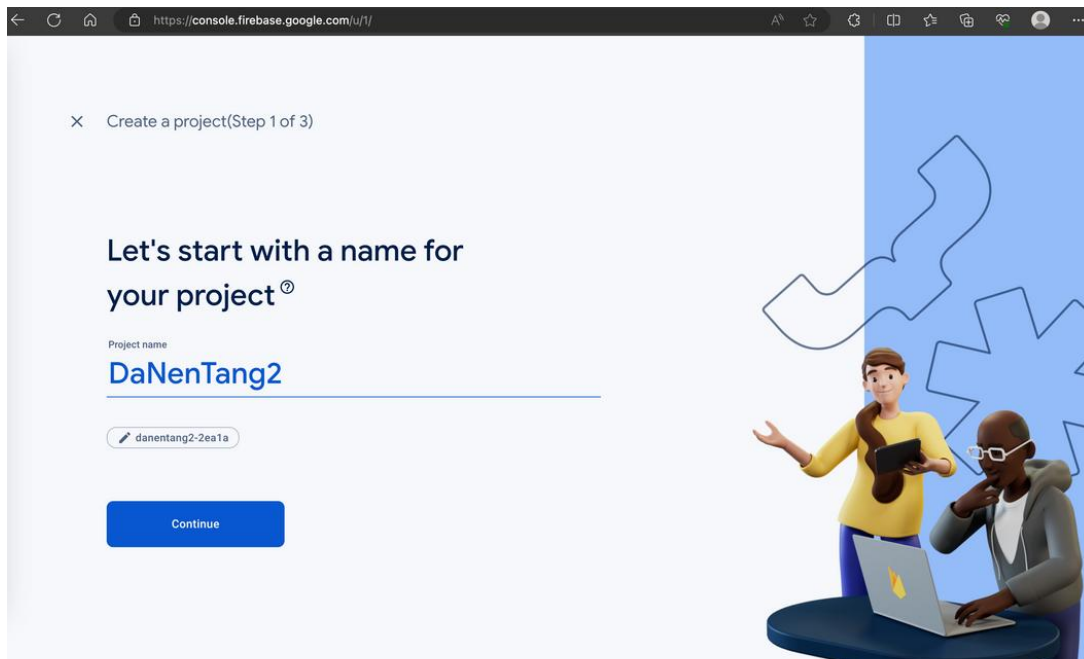
- ☐ Firebase là một nền tảng dịch vụ đám mây của Google cung cấp các tính năng cho việc phát triển ứng dụng di động và web.  
Firebase cung cấp cho các nhà phát triển các công cụ để phát triển và triển khai ứng dụng di động và web, cũng như quản lý người dùng, dữ liệu và thông tin đăng nhập.
  
- ☐ Các tính năng của Firebase bao gồm:
  - ❖ **Authentication:** Firebase cung cấp tính năng xác thực người dùng, giúp cho việc đăng ký và đăng nhập trở nên dễ dàng hơn. Điều này giúp cho các lập trình viên có thể tập trung vào việc phát triển tính năng thay vì việc quản lý người dùng.

- ❖ **Realtime Database:** Firebase cung cấp cơ sở dữ liệu thời gian thực, cho phép các ứng dụng có thể truy cập và cập nhật dữ liệu một cách nhanh chóng và liên tục. Điều này giúp cho các ứng dụng có thể đồng bộ hóa dữ liệu trên nhiều thiết bị cùng một lúc.
- ❖ **Hosting: Firebase:** cung cấp tính năng hosting cho các ứng dụng web, cho phép các lập trình viên có thể đăng ký và triển khai các ứng dụng web của mình trên Firebase một cách dễ dàng.
- ❖ **Storage:** Firebase cung cấp dịch vụ lưu trữ đám mây, cho phép lưu trữ các tệp tin như hình ảnh, video và âm thanh.

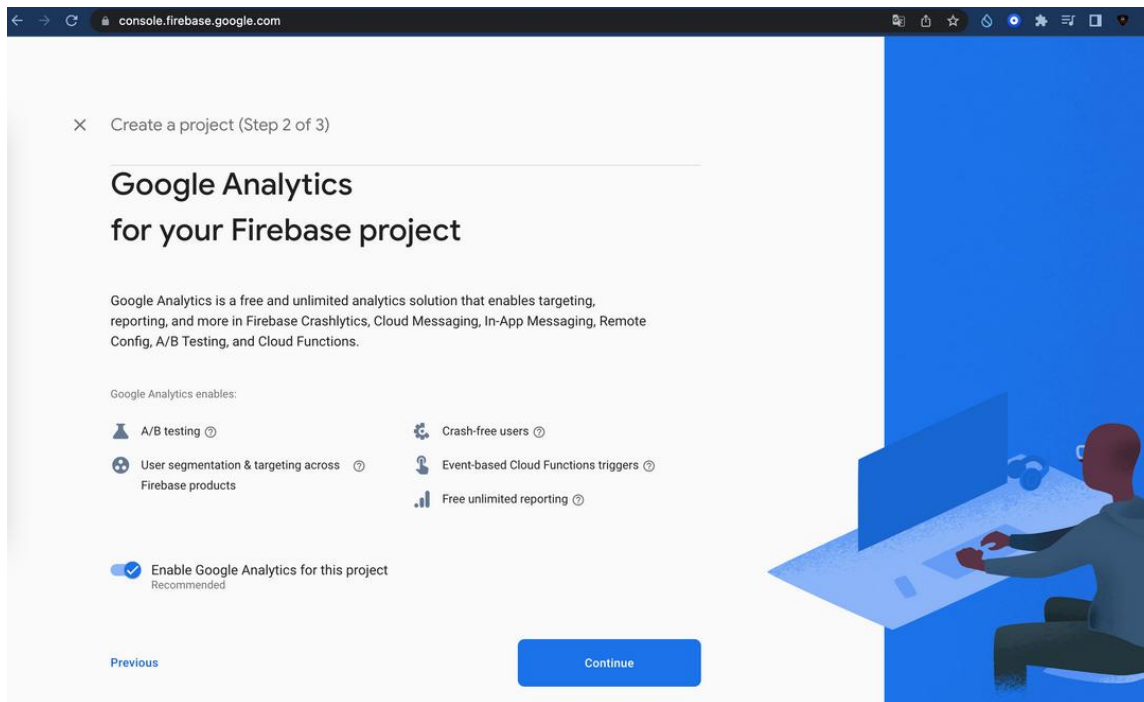
❖ **Cloud Functions:** Firebase cung cấp tính năng Cloud Functions, cho phép các lập trình viên viết các chức năng backend của ứng dụng của mình bằng các ngôn ngữ như JavaScript hoặc TypeScript.



## ❖ Bước 2: Đặt tên cho dự án của bạn và nhấn nút **Continue**

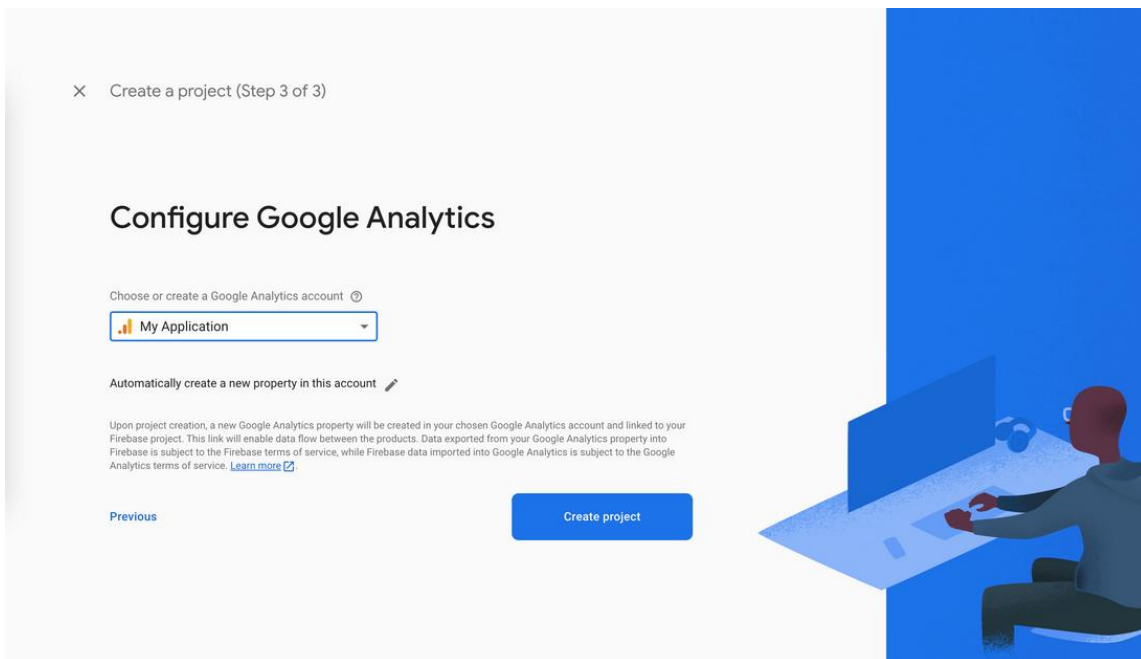


## ❖ Bước 3: Nhấn nút Continue

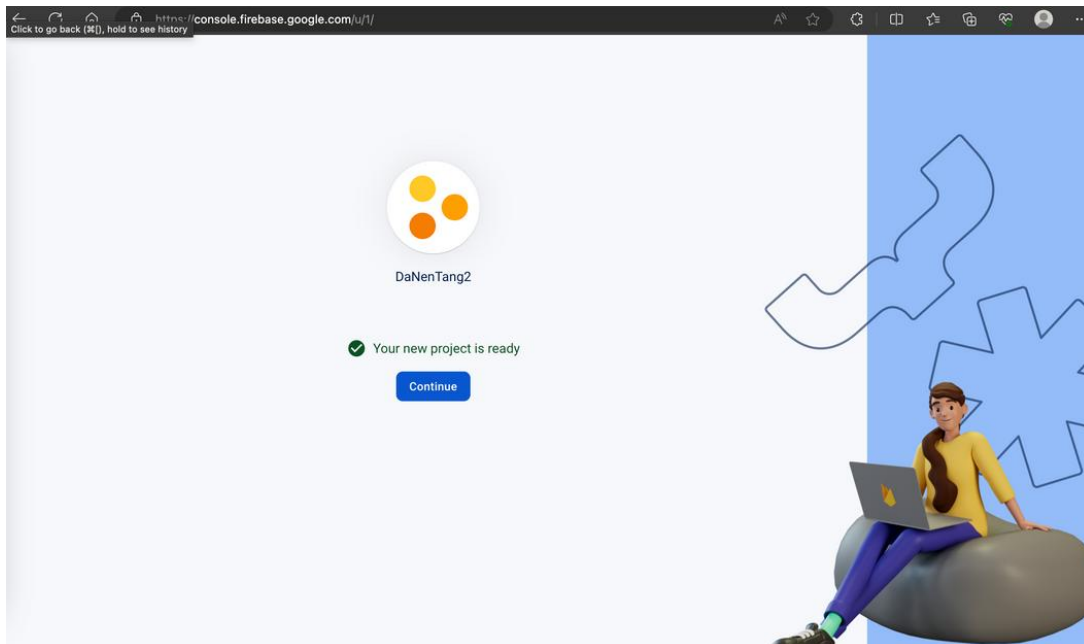




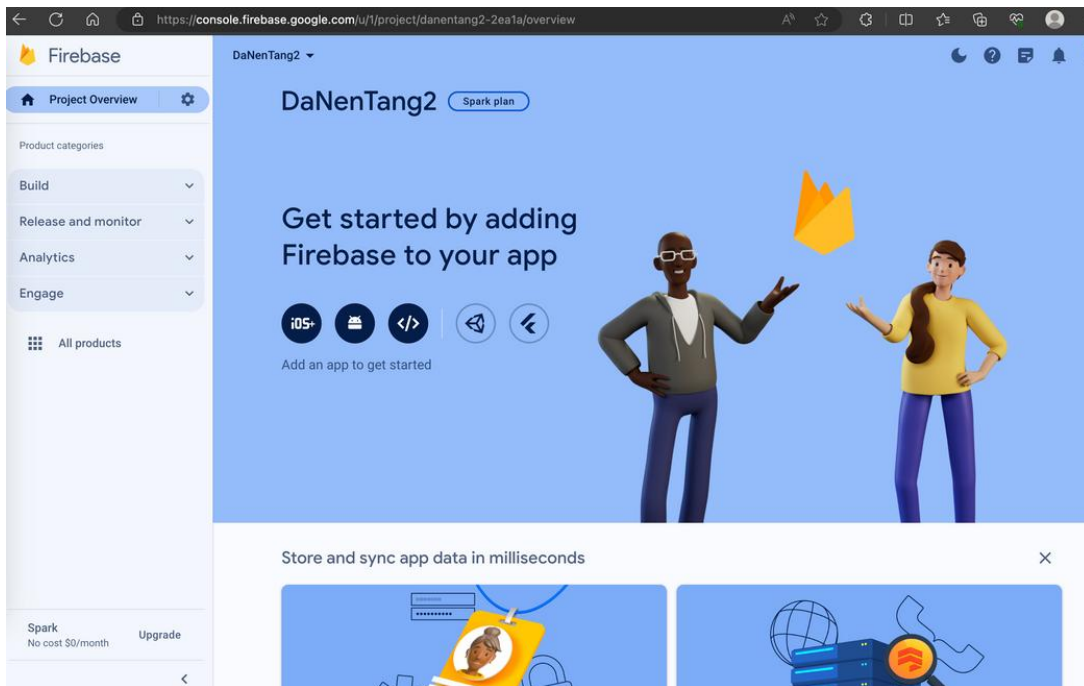
## ❖ Bước 4: Chọn bất kỳ tài khoản nào bạn muốn và nhấn nút Create project



❖ **Bước 5:** Bây giờ ứng dụng của bạn đã được tạo và tiếp tục làm thêm vài bước nữa nhé.



## ❖ Bước 6: Nhấn icon thứ 2, nút có biểu tượng Android



❖ **Bước 7:** Điền package name của project của bạn, ví dụ com.package\_name. Để lấy được certificate SHA-1 chạy lệnh **cd android && ./gradlew signinReport**

11

## ❖ Bước 8: Tải file **google-services.json** và lưu nó trong thư mục **android/app**.

× Add Firebase to your Android app

✓ Register app  
Android package name: com.danentang2

2 Download and then add config file Instructions for Android Studio below | [Unity](#) [C++](#)

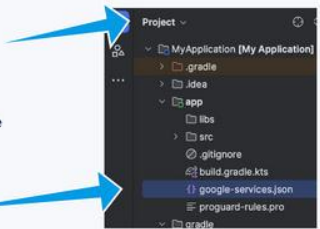
[Download google-services.json](#)

Switch to the **Project** view in Android Studio to see your project root directory.

Move your downloaded `google-services.json` file into your module (app-level) root directory.

`google-services.json`

[Next](#)



### ❖ **Bước 9:** Thêm firebase SDK .

Để làm cho **SDK Firebase** có thể truy cập các giá trị config **google-services.json**.

Thêm trình **google-services.json** vào tệp **build.gradle**  
(android/app)

- ☐ Để sử dụng **Firestore** trong React Native, bạn phải cài đặt package của firebase

```
npm install --save @react-native-firebase/app
```

Package **@react-native-firebase/app** phải được cài đặt trước khi sử dụng bất kỳ dịch vụ **Firestore** nào khác.

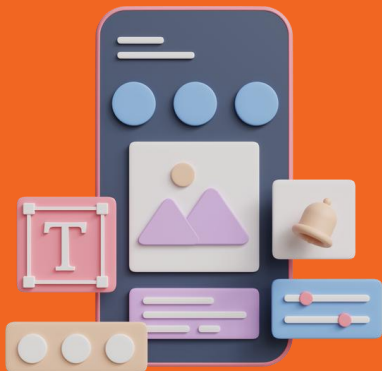
- ❖ Để thực thi plugin thêm dòng sau  
(**/android/app/build.gradle**):

```
apply plugin: 'com.android.application'  
apply plugin: 'com.google.gms.google-services' // <- Add this line
```

- ❖ Đầu tiên, thêm plugin google-services làm phụ thuộc, ở bên trong tệp **/android/build.gradle** của bạn:

```
buildscript {  
    dependencies {  
        // ... other dependencies  
        classpath 'com.google.gms:google-services:4.4.1'  
        // Add me --- ^\n    }  
}
```





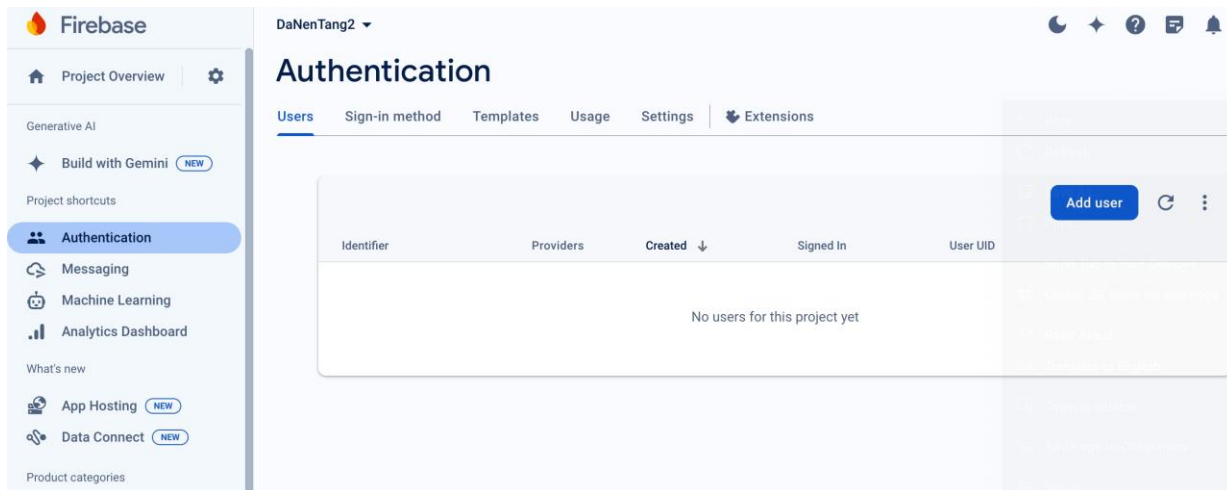
## LẬP TRÌNH ĐA NỀN TẢNG 2

BÀI 7: GIỚI THIỆU VỀ FIREBASE, ĐĂNG  
NHẬP EMAIL/PASSWOR VÀ GOOGLE

PHẦN 2: ĐĂNG NHẬP EMAIL/PASSWOR VÀ  
GOOGLE

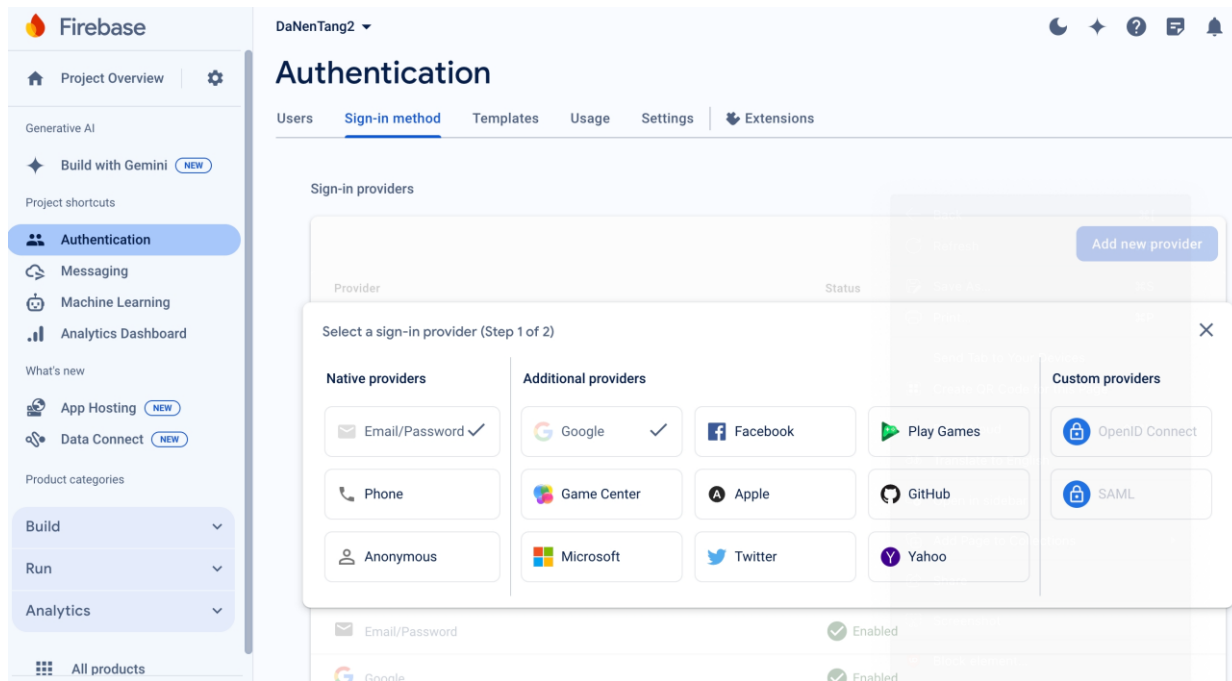
- ☐ Đăng nhập **Google** với firebase
- ☐ Đăng nhập, đăng ký với tài khoản Email/Password.

## ❖ Bước 1: Vào menu Authentication



The screenshot shows the Firebase console interface for the project 'DaNenTang2'. The left sidebar contains the 'Authentication' menu item, which is highlighted. The main content area displays the 'Authentication' page with the 'Users' tab selected. A table with columns 'Identifier', 'Providers', 'Created', 'Signed In', and 'User UID' is shown, but it contains no data, with the message 'No users for this project yet' displayed below the table. A 'Add user' button is visible in the top right corner of the table area.

## ❖ Bước 2: Tại phần Sign-in-method chọn Google



The screenshot shows the Firebase Authentication console for the project 'DaNenTang2'. The 'Sign-in method' tab is active, displaying a list of configured providers. A modal window titled 'Select a sign-in provider (Step 1 of 2)' is open, showing three categories of providers:

- Native providers:** Email/Password (checked), Phone, Anonymous.
- Additional providers:** Google (checked), Game Center, Microsoft, Facebook, Apple, Twitter, Play Games, GitHub, Yahoo.
- Custom providers:** OpenID Connect, SAML.

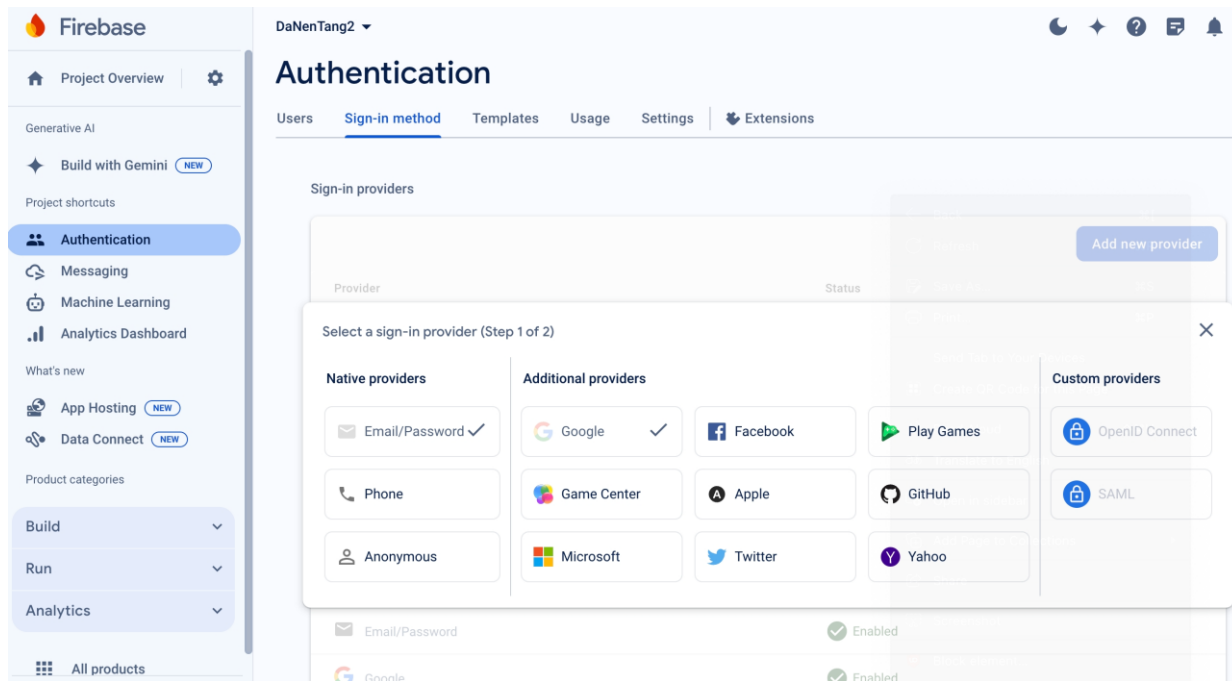
Below the modal, the main configuration table shows the status of the providers:

Provider	Status
Email/Password	Enabled
Google	Enabled

❖ **Bước 3:** Nhấn **Enable**, sau đó chọn mail **Project support email**, đây là email dùng để hỗ trợ người dùng, cuối cùng nhấn **Save**.



## ❖ Bước 4: Chọn Email/Password



The screenshot shows the Firebase Authentication console for the project 'DaNenTang2'. The 'Sign-in method' tab is active, displaying a list of providers. The 'Email/Password' provider is highlighted and marked as 'Enabled'.



**Sign-in providers**

Provider	Status
Email/Password	Enabled
Google	Enabled

**Select a sign-in provider (Step 1 of 2)**

Native providers	Additional providers	Custom providers
<ul style="list-style-type: none"> <li>Email/Password ✓</li> <li>Phone</li> <li>Anonymous</li> </ul>	<ul style="list-style-type: none"> <li>Google ✓</li> <li>Game Center</li> <li>Microsoft</li> </ul>	<ul style="list-style-type: none"> <li>Facebook</li> <li>Play Games</li> <li>GitHub</li> <li>Yahoo</li> <li>OpenID Connect</li> <li>SAML</li> </ul>

## ❖ Bước 5: Enable Email/Password sau đó nhấn nút Save

Provider	Status
<div>  <b>Email/Password</b> </div> <p>Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. <a href="#">Learn more</a></p> <div>            Delete provider         </div>	<div> <input checked="" type="checkbox"/> Enable         </div> <div> <input type="checkbox"/> Enable         </div>





Cancel
Save

❖ Hoàn tất, giao diện trên Firebase console như sau:

## Authentication

Users Sign-in method Templates Usage Settings Extensions

Sign-in providers

Provider		Status		<div>Add new provider</div>
 Email/Password		 Enabled		
 Google		 Enabled		



- ☐ Đăng nhập với Email/Password và đăng nhập bằng **google**
- ☐ Để có thể đăng nhập bằng tài khoản google, cần tải thêm thư viện của firebase **@react-native-firebase/app**

```
npm i @react-native-firebase/auth
```

## ☐ Giới thiệu về **Authentication**

Xác thực **Firestore** cung cấp các dịch vụ **backend** và SDK để sử dụng để xác thực người dùng với ứng dụng của bạn. Nó hỗ trợ xác thực bằng mật khẩu, số điện thoại, các nhà cung cấp danh tính liên kết phổ biến như **Google**, **Facebook** và **Twitter**, v.v.

Xác thực **Firestore** tích hợp chặt chẽ với các dịch vụ **Firestore** khác và nó tận dụng các tiêu chuẩn ngành như **OAuth 2.0** và **OpenID Connect**, vì vậy nó có thể dễ dàng tích hợp với phụ trợ tùy chỉnh của bạn.

## ☐ Lắng nghe trạng thái xác thực

Mô-đun cung cấp một phương thức được gọi là **onAuthStateChanged** cho phép bạn đăng ký để lắng nghe trạng thái xác thực hiện tại của người dùng và nhận một sự kiện bất cứ khi nào trạng thái đó thay đổi.

Điều quan trọng cần nhớ là trình nghe **onAuthStateChanged** không đồng bộ và sẽ kích hoạt trạng thái ban đầu sau khi kết nối với Firebase đã được thiết lập. Do đó, điều quan trọng là phải thiết lập trạng thái 'initializing'.

thái 'initializing' chặn kết xuất ứng dụng chính của chúng tôi trong khi kết nối được thiết lập:

```
const [initializing, setInitializing] = useState(true);  
const [user, setUser] = useState();
```

Tạo 2 state, state **initializing** được dùng để biết trạng thái đang kiểm tra trạng thái đăng nhập

Tiếp theo import **auth** từ **@react-native-firebase/auth**. Sử dụng **onAuthStateChanged** để xác định trạng thái đăng nhập

```
function onAuthStateChanged(user) {  
  setUser(user);  
  if (initializing) {  
    setInitializing(false);  
  }  
}  
  
useEffect(() => {  
  const subscriber = auth().onAuthStateChanged(onAuthStateChanged);  
  return subscriber; // unsubscribe on unmount  
}, []);
```

Dựa vào state chúng ta render ra giao diện phù hợp với trạng thái đăng nhập cho người dùng

```
function onAuthStateChanged(user) {  
  setUser(user);  
  if (initializing) {  
    setInitializing(false);  
  }  
}  
  
useEffect(() => {  
  const subscriber = auth().onAuthStateChanged(onAuthStateChanged);  
  return subscriber; // unsubscribe on unmount  
}, []);
```

## Đăng nhập bằng Email/Mật khẩu

---

- ☐ Đăng nhập email/mật khẩu là một phương pháp phổ biến để người dùng đăng nhập trên các ứng dụng. Điều này yêu cầu người dùng cung cấp địa chỉ email và mật khẩu an toàn. Người dùng có thể đăng ký và đăng nhập bằng phương thức có tên **createUserWithEmailAndPassword** hoặc đăng nhập vào tài khoản hiện có bằng **signInWithEmailAndPassword**.
- ☐ Đảm bảo bạn đã bật đăng nhập "Email/Password" được bật trên Bảng điều khiển Firebase.

## Đăng nhập bằng Email/Mật khẩu

---

- ☐ Đăng nhập email/mật khẩu là một phương pháp phổ biến để người dùng đăng nhập trên các ứng dụng. Điều này yêu cầu người dùng cung cấp địa chỉ email và mật khẩu an toàn. Người dùng có thể đăng ký và đăng nhập bằng phương thức có tên **createUserWithEmailAndPassword** hoặc đăng nhập vào tài khoản hiện có bằng **signInWithEmailAndPassword**.
- ☐ Đảm bảo bạn đã bật đăng nhập "Email/Password" được bật trên Bảng điều khiển Firebase.



- Tạo hàm tạo tài khoản, bạn gọi **createUserWithEmailAndPassword** của Firebase để tạo tài khoản. Truyền vào 2 prop, prop đầu tiên là email bạn muốn tạo, prop thứ 2 là mật khẩu

```
const onSignUpWithPassword = () => {  
  auth().createUserWithEmailAndPassword(  
    'jane.doe@example.com',  
    'SuperSecretPassword!',  
  );  
};
```

- ☐ Để bắt được thái tạo tài khoản thành công hay chưa, sử dụng **then** và **catch** ngay sau **createUserWithEmailAndPassword**

```
.then(() => {  
  console.log('Tài khoản đã được tạo và đăng nhập');  
}).catch(error => {  
  if (error.code === 'auth/email-already-in-use') {  
    console.log('Email đã tồn tại');  
  }  
  if (error.code === 'auth/invalid-email') {  
    console.log('Email của bạn không hợp lệ!');  
  }  
  console.error(error);  
});
```

- Gọi hàm **signInWithEmailAndPassword** để đăng nhập tài khoản với **email** và **password**

```
const onSignInWithPassword = () => {  
  auth()  
    .signInWithEmailAndPassword(  
      'jane.doe@example.com',  
      'SuperSecretPassword!')  
    .then(() => {  
      console.log('Tài khoản đã đăng nhập');  
    })  
    .catch(error => { console.error(error) });  
};
```

- ☐ Thư viện đăng nhập **google** cung cấp một trình bọc xung quanh thư viện đăng nhập chính thức của Google, cho phép bạn tạo thông tin đăng nhập và đăng nhập vào Firebase.
- ☐ Cài đặt thư viện

```
npm i @react-native-google-signin/google-signin
```

- ☐ Trước khi kích hoạt yêu cầu đăng nhập, bạn phải khởi tạo **Google SDK** bằng cách sử dụng bất kỳ required scopes và **webClientId** có thể tìm thấy trong tệp **android/app/google-services.json** dưới dạng thuộc tính **client/oauth\_client/client\_id** (id kết thúc bằng **.apps.googleusercontent.com**). Đảm bảo chọn **client\_id** có **client\_type: 3**

- Gọi hàm **GoogleSignin** từ package và config **webClientId** từ file **google-service.json** của bạn

```
import { GoogleSignin } from '@react-native-google-signin/google-signin';
GoogleSignin.configure({
  webClientId:
    '182511633460-
    p38tuh1aeakf26gu9g1lnukt0ep5hvua.apps.googleusercontent.com',
});
```

- Tiếp theo gọi hàm **hasPlayServices** để kiểm tra thiết bị có hỗ trợ Google Play không, sau đó gọi hàm **signIn** để lấy **idToken**. Cuối cùng gọi hàm **signInWithCredential** để đăng nhập lên Firebase

```

async function onGoogleButtonPress() {
  await GoogleSignin.hasPlayServices({ showPlayServicesUpdateDialog:
true});
  const {idToken} = await GoogleSignin.signIn().catch(error =>
    console.log('ERROR ', error),
  );
  const googleCredential = auth.GoogleAuthProvider.credential(idToken);

  return auth().signInWithCredential(googleCredential);
}

```

- ☐ Để đăng xuất khỏi tài khoản google, gọi **GoogleSignin.signOut()**

- ☐ Giới thiệu về firebase
- ☐ Setup Firebase console
- ☐ Đăng nhập **Google** với firebase
- ☐ Đăng nhập, đăng ký với tài khoản Email/Password.



# Kết thúc