**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**





**BÁO CÁO ĐỒ ÁN**

**MÔN SEMINAR VÀ CÁC VẤN ĐỀ HIỆN ĐẠI**

**Đề tài: Xây dựng ứng dụng quản lý công việc tích hợp**

**mô hình Serverless Service**

GVHD: Đinh Nguyễn Anh Dũng

Nhóm sinh viên thực hiện:

1. Trần Thanh Hiền MSSV: 21520230

2. Nguyễn Phúc Bình MSSV: 21520638

🙡🙢 Tp. Hồ Chí Minh, 11/2024 🙠🙣

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 20…*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

**BẢNG PHÂN CÔNG, ĐÁNH GIÁ THÀNH VIÊN:**

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên | MSSV | Phân công | Đánh giá |
| Trần Thanh Hiền | 21520230 |  |  |
| Nguyễn Phúc Bình | 21520638 |  |  |

**LỜI MỞ ĐẦU**

**MỤC LỤC**

[Chương 1: Tổng quan 7](#_Toc182970943)

[1.1 Giới thiệu 7](#_Toc182970944)

[1.2 Mục tiêu nghiên cứu 7](#_Toc182970945)

[1.3 Serverless là gì? 7](#_Toc182970946)

[1.4 Một số dịch vụ Serverless phổ biến hiện tại. 9](#_Toc182970947)

[1.5 Lý do chọn Firebase trong việc xây dựng ứng dụng 9](#_Toc182970948)

[1.6 Tìm hiểu về các Service của Firebase 10](#_Toc182970949)

[1.7 Chính sách chi phí sử dụng của Firebase 12](#_Toc182970950)

[1.8 Chức năng ứng dụng demo 15](#_Toc182970951)

[**Chương 2: Xây dựng ứng dụng demo 16**](#_Toc182970952)

[2.1 Thiết kế cơ sở dữ liệu 16](#_Toc182970953)

[2.2 Cấu hình ứng dụng Firebase 19](#_Toc182970954)

[2.3 Kết nối và khởi tạo các Firebase dưới dạng local 20](#_Toc182970955)

[2.4 Tạo tương tác giữa Client và Firebase 23](#_Toc182970956)

[2.5 Giao diện Người dùng 25](#_Toc182970957)

[2.6 Đánh giá việc triển khai ứng dụng web trên việc sử dụng dịch vụ Firebase Service 30](#_Toc182970959)

[**Chương 3: Kết luận 30**](#_Toc182970960)

[3.1 Kết quả đạt được 30](#_Toc182970962)

[3.2 Bài học kinh nghiệm 30](#_Toc182970964)

[3.3 Hướng phát triển 31](#_Toc182970966)

# Tổng quan

## Giới thiệu

Hiện nay, Serverless là một xu hướng phát triển quan trọng trong công nghệ thông tin. Nó loại bỏ yêu cầu quản lý cơ sở hạ tầng và cho phép tự động mở rộng, giúp lập trình viên có thể tập trung vào việc phát triển tính năng ứng dụng.

Việc nghiên cứu và ứng dụng Serverless, đặc biệt là về Firebase Service, áp dụng chúng vào các dự án thực tế sẽ giúp em có thể hiểu sâu hơn về cách phát triển sản phẩm theo phương pháp mới thông qua serverless bằng Firebase. Giúp nhóm tụi em sẽ có thêm kinh nghiệm và kiến thức về cách sử dụng các công nghệ mới để phát triển ứng dụng hiệu quả hơn.

Đề tài mà chúng em thực hiện sẽ nghiên cứu về mô hình Serverless và ứng dụng của chúng thông qua việc xay dựng ứng dụng thực tiễn, qua đó biết rõ được về lợi ích, thách thức và cơ hội của nó trong việc phát triển phần mềm. Ứng dụng của nhóm em được xây dựng trên nền tảng serverless với Firebase là dịch vụ serverless nhóm sử dụng và các service của Firebase trong việc hỗ trợ xây dụng các ứng dụng web về việc quản lý công việc nhóm

## Mục tiêu nghiên cứu

* Phân tích bằng cách nghiên cứu các đặc điểm, lợi ích và hạn chế của mô hình serverless trong phát triển ứng dụng.
* Tìm hiểu cách ứng dụng dịch vụ Firebase trong việc xây dựng ứng dụng serverless cho ứng dụng web.
* Xây dựng một ứng dụng Demo hoàn thiện bằng cách sử dụng các dịch vụ của Firebase
* Phân tích hiệu suất của Firebase so với các mô hình phát triển truyền thống, nhằm đưa ra những khuyến nghị cho việc áp dụng công nghệ serverless trong các dự án phát triển phần mềm.

## Serverless là gì?

Serverless về cơ bản là một mô hình kiến trúc điện toán đám mấy (Cloud computing) với mục đích có thể phát triển ứng dụng ở mô hình vi dịch vụ với các dịch vụ khác nhau ở những nền tảng khác nhau thay vì trên một server truyền thống nhờ vậy sẽ chia nhỏ các dịch vụ ở những nền tảng khác nhau nhằm tối ưu hóa được hiệu suất với ứng dụng những điểm mạnh phù hợp cho từng loại service khác nhau.

A cloud computing diagram with text

Description automatically generated

Serverless không phải là một khái niệm tồn tại độc lập mà là một phần trong hệ sinh thái Cloud computing gồm:

* FaaS (Function as a Service):
  + FaaS là mô hình Cloud Computing mà ta chỉ cần xử lí và triển khai cách function để thực hiện các chức năng cụ thể khi được gọi đến. Đối với FaaS thì mục đích chỉ xử lí và sử dụng khi cần thiết và thường sẽ chỉ chạy một chức năng nào đó trong thời gian ngắn hoạt động.
  + Ứng dụng của việc áp dụng FaaS: FaaS thường sẽ là các hàm xử lí nhanh chóng cần thực thi các việc cụ thể như có thể call function để upload ảnh hoặc xử lí ảnh từ phía người dùng khi gửi request lên.
  + Với việc sử dụng FaaS giúp ta đỡ phải quan tâm đến việc mở rộng tài nguyên xử lí các công việc phù hợp và có thể sử dụng các function đó một cách dễ dàng và trả chi phí phù hợp với lưu lượng sử dụng.
  + Các dịch vụ serverless ứng với FaaS thường được sử dụng rộng rãi như: Firebase function, AWS Lambda, Azure Function.
* BasS (Backend as a Service):
  + BaaS là kiểu mô hình Cloud Computing mà ở đó sẽ cung cấp một nền tảng dịch vụ backend hoàn chỉnh phù hợp với việc xây dựng một ứng dụng nào đó mà người dùng không cần phải quan tâm đến việc quản lý và mở rộng tài nguyên cơ sở hạ tầng của hệ thống. Nhờ vậy giúp ta tập trung hơn ở phía giao diện và front-end nhiều hơn so với việc quản lý backend của ứng dụng.
  + BaaS sẽ cung cấp các dịch vụ về cơ sở dữ liệu, xác thực người dùng, lưu trữ và các RESTFul API mà dịch vụ BaaS sẽ tự động ở rộng cơ sở hạ tầng một cách phù hợp. Nhờ vậy mà việc xây dựng một ứng dụng hoàn chỉnh sẽ nhanh hơn so với việc xây dựng Backend ở một server truyền thống.
  + Một số các dịch vụ BaaS phổ biển như: Firebase, AWS Amplìfy, Backendless.

Nhờ một số tính chất và ứng dụng trên thì nhóm cảm thấy việc ứng dụng serverless vô việc xây dựng ứng dụng thực tiễn trở nên khả thi và phù hợp.

## Một số dịch vụ Serverless phổ biến hiện tại.

Serverless đã trở thành một trong những xu hướng công nghệ quan trọng trong ngành công nghiệp điện toán đám mây, được các công ty hàng đầu thế giới như Amazon, Microsoft, và Google đẩy mạnh phát triển. Các dịch vụ serverless hiện đại được cung cấp bởi những "ông lớn" này không chỉ hỗ trợ việc triển khai ứng dụng nhanh chóng mà còn giúp giảm đáng kể chi phí và công sức quản lý hạ tầng. Một số dịch vụ serverless phổ biến hiện nay:

* **AWS (Amazon Web Services):** Đây là dịch vụ điện toán đám mây cũng có một hệ sinh thái Serverless lớn phù hợp cho người dùng sử dụng gồm các dịch vụ như:
  + **AWS Lambda:** Dịch vụ FaaS của Amazoncung cấp việc xây dựng và chạy function trực tiếp mà không cần quan tâm đến cơ sở hạ tầng hệ thống.
  + **Amazon API Gateway:** Giúp xây dựng và triển khai quản lý các dịch vụ RESTFul API và giao tiếp giữa các api đó.
* **Microsoft Azure:** Đây là dịch vụ điện toán đám mây của microsoft gồm các nhóm dịch vụ như:
  + **Azure Function:** Dịch vụ FaaS của Microsoft, cho phép chạy các function khi được kích hoạt bởi các sự kiện. Với dễ dàng tích hợp với hệ sinh thái của Azure như Azure Blob Storage, Azure Event Grid.
  + **Azure Logic Apps:** Cung cấp khả năng xây dựng quy trình tự động hóa (workflow automation) mà không cần viết mã phức tạp.
* **Google Cloud Platform (GCP):** Đây là một dịch vụ điện toán đám mây của google với các dịch vụ serverless nổi bật:
  + **Google Cloud Functions**: Dịch vụ FaaS của Google, chạy mã dựa trên sự kiện từ các dịch vụ khác như Firebase, Pub/Sub, và HTTP.
  + **Firebase Functions**: Một phần của Google Cloud, đặc biệt được tối ưu hóa cho ứng dụng di động và web. Dễ dàng thực thi các chức năng dựa trên các sự kiện trong Firebase, như thay đổi cơ sở dữ liệu hoặc xác thực người dùng.
  + **Firebase**: Là dịch vụ BaaS của Google nhằm cung cấp nền tảng xây dựng backend cho ứng dụng một cách nhanh chóng và dễ dàng tiếp cận.

## Lý do chọn Firebase trong việc xây dựng ứng dụng

Firebase là một phần của GCP và là dịch vụ serverless dưới dạng BaaS và cung cấp đầy đủ các dịch vụ hoàn chỉnh để có thể xây dựng ứng dụng một cách dễ dàng.

Từ việc đưa ra đưa ra một số service cũng như ứng dụng của các service đó ở mục trên thì nhóm dự định chọn Firebase trong việc triển khai ứng dụng cũng như nghiên cứu ở đề tài này vì các lý do sau:

* Khi so sánh giữa firebase và các ứng dụng serverless khác của các nền tảng khác như AWS hay Azure thì Firebase sẽ dễ dàng hơn trong việc xây dựng, cấu hình dự án cho người mới bắt đầu và dễ dàng triển khai đồng thời Firebase cũng cung cấp các thư viện và tài liệu đầy đủ cho người mới dễ dàng tiếp cận hơn.
* Firebase cung cấp việc tiếp cận với dạng free plan và nhờ đó giúp cho việc tiếp cận và sử dụng nó dễ dàng hơn.
* Firebase cung cấp một hệ sinh thái đầy đủ trong việc xây dựng ứng dụng như cơ sở dữ liệu (Cloud Firestore, Realtime Database), lưu trữ tệp (Cloud Storage), và thông báo đẩy (FCM). Tất cả service này đều được tích hợp vào trong Firebase nên việc triển khai và sử dụng nó khá dễ dàng.
* Tích hợp dễ dàng khi kết hợp với Frontend khi nó đã cung cấp hầu hết các thư viện và service phù hợp cho việc xây dựng ứng dụng giúp tiết kiệm thời gian cấu hình và xây dựng ứng dụng.

## Tìm hiểu về các Service của Firebase

Với việc là hệ sinh thái cũng như là cung cấp dịch vụ **BaaS (Backend as a Service)** thì Firebase có những service đầy đủ trong việc xây dựng cơ bản một ứng dụng.

1. **Firebase Authentication**

* Là dịch vụ xác thực người dùng toàn diện của Firebase. Nó cung cấp các giải pháp xác thực người dùng dễ dàng và an toàn, hỗ trợ nhiều phương thức đăng nhập như email/mật khẩu, Google, Facebook, Twitter,...
* Dịch vụ này cung cấp các API dễ dàng tích hợp vào ứng dụng, giúp quản lý người dùng, vai trò và phân quyền truy cập một cách hiệu quả. Firebase Authentication đã cung cấp một cách dễ dàng và có thể nhanh chóng xây dựng việc bảo mật đăng nhập cho ứng dụng thông qua việc đăng nhập, Quản lý phiên đăng nhập và tích hợp liền mạch với các dịch vụ khác trong Firebase.
* Dễ dàng tích hợp ở phía front-end.
* Ngoài ra có thể kết hợp với Firebase Function để tạo ra các luồng xử lí đăng nhập một cách phù hợp cho người dùng.

1. **Cloud Firestore**

* Cloud Firestore là cơ sở dữ liệu NoSQL toàn diện của nền tảng Firebase. Nó có khả năng mở rộng cao và đồng bộ hóa dữ liệu theo thời gian thực, giúp đáp ứng nhu cầu lưu trữ và quản lý dữ liệu. Firestore lưu trữ dữ liệu dưới dạng Collection và Document.
* Nhờ đó, chúng ta có thể truy vấn và lọc dữ liệu một cách hiệu quả, đồng thời đồng bộ hóa dữ liệu giữa các thiết bị nhanh chóng.
* Đồng thời với Firestore còn hỗ trợ chức năng realtime khi giúp nhận sự kiện khi Collection có sự thay đổi nào đó bên trong firestore giúp phát triển hệ thống realtime một cách dễ dàng hơn.

1. **Firebase Realtime Database**

* Firebase Realtime Database là cơ sở dữ liệu NoSQL của Firebase. Nó cung cấp việc lưu trữ và khả năng mở rộng cao. Đồng thời khác với Cloud Firestore được nêu ở trên thì Firebase Realtime được lưu trữ và quản lý dữ liệu dưới dạng cấu trúc cây (Tree Structure) dưới dạng JSON.
* Nhờ vậy việc nhận diện và truy cập realtime của Firebase Realtime sẽ trở nên mạnh mẽ và tốc độ tốt hơn so với Cloud Firestore được đề cập ở trên.

1. **Firebase Storage**

* Firebase Storage là giải pháp lưu trữ file an toàn và dễ mở rộng cho các ứng dụng, cung cấp bởi nền tảng Firebase. Dịch vụ này hỗ trợ việc tải lên và tải xuống ảnh, video, tài liệu một cách dễ dàng.
* Firebase Storage được tích hợp sâu với các dịch vụ Firebase khác như Authentication và Firestore, giúp quá trình phát triển ứng dụng trở nên đơn giản hơn. Người dùng có thể truy cập và sử dụng Firebase Storage một cách dễ dàng nhờ việc đã hỗ trợ các thư viện từ phía google ở nhiều nền tảng khác nhau giúp việc upload và download dữ liệu một cách nhanh chóng và dễ dàng.

1. **Firebase Function**

* Firebase Function là dịch vụ serverless FaaS nhằm cho phép triển khai các chức năng backend tùy chỉnh phù hợp với hệ thống.
* Các function có thể được kích hoạt bởi các sự kiện như thay đổi dữ liệu trong Firestore, quá trình xác thực người dùng hoặc yêu cầu HTTP.
* Nhờ trong hệ sinh thái Firebase mà Firebase Function có thể xử lí các vấn đề liên quan đến Firebase Authentication hay Firestore thông qua phương function reaction của Firebase Function để xử lí các trigger của Authentication hay Firestore.

1. **Firebase Hosting**

* Dịch vụ lưu trữ và cung cấp nội dung web của Firebase là một giải pháp hiệu quả, cho phép triển khai các trang web tĩnh và động một cách nhanh chóng và an toàn.
* Dịch vụ này tích hợp sâu với các dịch vụ Firebase khác như Functions và Storage, tạo ra một hệ sinh thái liền mạch cho các ứng dụng.
* Firebase Hostring này cũng hỗ trợ tự động triển khai từ các công cụ phát triển như Git, CI/CD khi đẩy ứng dụng lên trên Git. Đồng thời với Firebase Hosting cung cấp các tính năng bảo mật và quản lý domain tích hợp, đảm bảo an toàn và dễ dàng quản lý cho các ứng dụng.

1. **Firebase Emulators**

* Với việc sử dụng 1 dịch vụ Serverless như firebase thì đi kèm theo nó sẽ phát sinh khá nhiều chi phí nếu sử dụng trên server của Firebase nên vì vậy Google đã cung cấp Firebase Emulator là một máy ảo của Firebase được khởi chạy ở dưới máy local của người sử dụng trong quá trình xây dựng ứng dụng để tránh mất quá nhiều chi phí phát triển sản phẩm như triển khai chúng trên server thực tế của Firebase. Với Firebase Emulator người dùng có thể khởi chạy máy ảo nhằm mô phỏng service của Firebase nhưng không tốn chi phí phát sinh nhờ đó giúp việc phát triển ứng dụng có thể tối ưu hóa kinh phí hơn cho đội ngũ phát triển.

1. **Firebase Admin SDK**

* Firebase cung cấp SDK dành riêng cho phía server hoạt động độc lập có thể truy cập vô việc quản lý và kiểm soát ứng dụng hoạt động firebase ở phía backend. Đây là một phần quan trọng khi bạn cần quyền truy cập mở rộng để thực hiện các tác vụ mà Firebase Client SDK không cung cấp, như quản lý người dùng hoặc xử lý logic nghiệp vụ phức tạp.
* Server backend có thể truy cập và quản lý các dữ liệu, action bên trên firebase và toàn quyền truy cập xử lí mà không bị giới hạn tính năng ở Firebase Client SDK.
* Có thể kiểm soát quyền truy cập và thay đổi hệ thống ở phía server của phía người dùng.

1. **Firebase Client SDK**

* Firebase Client SDK là thư viện dành riêng cho phía client (frontend), được thiết kế để tích hợp Firebase vào ứng dụng của người dùng.Nhờ thư viện này giúp nhà phát triển dễ dàng kết nối và sử dụng các dịch vụ Firebase như cơ sở dữ liệu, lưu trữ, xác thực, và thông báo mà không cần backend riêng biệt.
* Firebase Client SDK cung cấp việc sử dụng ở trên nhiều nền tảng khác nhau như Web, Android, IOS, Cross-Platform mà nhờ đó việc xây dựng ứng dụng trở nên dễ dàng hơn cho lập trình viên khi có thể chủ động làm việc với Firebase thay vì phải code một server Backend truyền thống một cách riêng biệt.
* Firebase Client SDK cung cấp các chức năng cơ bản cho việc xây dựng ứng một cách dễ dàng như việc truy cập cơ sở dữ liệu, quản lý xác thực phiên người dùng, tương tác với các cloud storage.

Chính vì những service trên mà Firebase trở thành phương pháp truy cập hướng serverless phù hợp mà nhóm lựa chọn để có thể xây dựng ứng dụng thực tiễn với serverless

## Chính sách chi phí sử dụng của Firebase

* Với việc là nền tảng dịch vụ serverless nên đi kèm theo đó là chi phí sử dụng mà người dùng phải trả khi dùng các dịch vụ của Firebase. Tuy nhiên với Firebase, Google cung cấp hai loại plan mà người dùng có thể sử dụng tùy vô nhu cầu sử dụng của người dùng gôm: **Spark plan** (Không tốn phí) **và Blaze plan** (Tốn phí)
  + **Spark plan:** Người dùng sẽ không tốn phí dịch vụ khi sử dụng nhưng đồng thời việc truy cập và sử dụng sẽ chỉ nằm ở một giới hạn nhất định nào đó.
  + **Blaze plan:** Người dùng sẽ tốn phí dịch vụ dựa vô số lượng truy cập sử dụng của ứng dụng mà sẽ tính phí dựa trên đó.

**Bảng chi phí sử dụng của Firebase:**

A blue screen shot of a blue screen

Description automatically generated

A screenshot of a computer

Description automatically generated

*Firebase Authentication Pricing*

*A screenshot of a computer

Description automatically generated*

*Cloud Firestore Pricing*

*A screenshot of a computer

Description automatically generated*

*Firebase Cloud Function Pricing*

*A screenshot of a computer

Description automatically generated*

*Firebase Cloud Storage Pricing*

*A screenshot of a computer

Description automatically generated*

*Firebase App Hosting Pricing*

Thông qua việc tiếp cận bảng giá trên giúp nhóm có thể ước chừng và nhắm trong việc tính toán các chi phí khi xây dựng ứng dụng thông qua Firebase Service mà không mất quá nhiều chi phí không cần thiết.

## Chức năng ứng dụng demo

Ứng dụng demo là một hệ thống quản lý công việc, có các chức năng cơ bản như

* Đăng nhập / Đăng ký qua email: Cho phép người dùng đăng ký tài khoản bằng email và xác thực tài khoản trước khi đăng ký. Sử dụng Firebase authentication trong nhóm các service của firebase để thực hiện chức năng authentication cho ứng dụng.
* CRUD (Create, Read, Update, Delete) Boards: Cho phép người dùng có thể tạo các bảng quản lý công việc đồng thời có thể thay đổi, thêm xóa các thành viên hoạt động trong bảng và có thể xóa bảng nếu cần thiết
* CRUD (Create, Read, Update, Delete) và realtime update cho List: List là danh sách các công việc bên trong bảng và phân ra các quá trình khác nhau trong Board như list “Đang làm”, list “ Đã hoàn thành” và cho phép người dùng có thể thêm, xóa, sửa các danh sách đồng thời các thành viên bên trong board có thể nhìn thấy sự thay đổi đó một cách tức thời.
* CRUD (Create, Read, Update, Delete) và realtime update cho Task: Task là công việc và người dùng có thể thêm xóa sửa hoặc chuyển sang các list khác nhau phù thuộc vào trạng thái của của task đó và cho phép assign các task đó cho 1 thành viên trong bảng đó, đồng thời hiển thị cập nhật sự thay đổi đó một cách tức thời thời cho các thành viên.
* CRUD (Create, Read, Update, Delete) và realtime update cho Comment: Người dùng có thể comment hoặc góp ý các thông tin cho từng task đó.
* UI: Giao diện được tối ưu để phản hồi nhanh và thân thiện với người dùng.
* Xử lý lỗi: Hệ thống sẽ xử lý và ghi nhận các lỗi có thể xảy ra trong quá trình sử dụng.

Ứng dụng các service serverless của Firebase trong việc xây dựng ứng dụng:

* Đối với việc đăng nhập và xác thực người dùng thì nhóm sẽ ứng dụng Firebase Authentication kết hợp với Firebase Function để tạo ra function xử lí đăng nhập phù hợp cho người dùng.
* Về tính năng realtime và cập nhật nhanh chóng các task trong ứng dụng cũng như quản lý nó thì nhóm sử dụng Firebase Firestore dù việc cập nhật có thể không nhanh bằng Firebase realtime nhưng với Firestore thì dễ dàng quản lý dữ liệu người dùng hơn.
* Việc lưu trữ dữ liệu thì nhóm cũng sử dụng Firestore để lưu trữ phù hợp cho ứng dụng này.
* Để có thể test, dev mà không bị ảnh hưởng nhiều chi phí thì nhóm sử dụng Firebase Emulators để có thể mô phỏng các dịch vụ của Firebase trên chính máy local nhằm gia tăng hiệu suất cũng như giảm chi phí người dùng khi không cần deploy các function mới lên firebase liên tục.

## Xây dựng ứng dụng demo

## Thiết kế cơ sở dữ liệu

Xây dựng hệ thống dữ liệu trên firebase firestore NoSQL.

Với việc hệ thống cơ sở dữ liệu Firestore được xây dựng theo cơ chế phân lớp Collection, Doc và sub Collection nên việc xây dựng cơ sở dữ liệu ở đây sẽ khác đôi chút so với lại 1 số cơ sở dữ liệu NoSQL khác như mongoDb.

1. **Users Collection**

{

"uid": "string",

"displayName": "string",

"email": "string",

"avatarURL": "string",

"boards": ["boardId1", "boardId2", ...]

"createdAt": "timestamp",

"updatedAt": "timestamp"

}

1. **Board Collection**

{

"boardId": "string", // Unique ID

"name": "string",

"description": "string",

"ownerId": "uid", // ID of the owner

"members": ["uid1", "uid2", ...], // List of user IDs part of the board

"createdAt": "timestamp",

"updatedAt": "timestamp"

}

1. **Lists SubCollection (Sub Collection của Board)**

{

"listId": "string",

"name": "string",

"position": "number", // Vị trí của list trong board

"createdAt": "timestamp",

"updatedAt": "timestamp"

}

1. **Card SubCollection (Sub Collection của List)**

{

"cardId": "string",

"title": "string",

"description": "string",

"assignedTo": ["uid1", "uid2", ...], // List of user IDs assigned to this card

"labels": ["label1", "label2", ...], // Optional labels for the card

"dueDate": "timestamp",

"position": "number", // Vị trí của card trong list

"attachments": ["attachmentId1", "attachmentId2", ...], // List of attachment IDs

"comments": ["commentId1", "commentId2", ...], // List of comment IDs

"createdAt": "timestamp",

"updatedAt": "timestamp"

}

1. **Comment SubCollection (Sub Collection của Card)**

{

"commentId": "string",

"authorId": "uid", // ID of the user who made the comment

"content": "string",

"createdAt": "timestamp",

"updatedAt": "timestamp"

}

1. **Attachment SubCollection (Sub Collection của Card)**

{

"attachmentId": "string",

"fileName": "string",

"fileURL": "string", // URL dẫn đến file trên Firebase Storage hoặc một dịch vụ lưu trữ khác

"uploadedBy": "uid", // ID của người upload

"uploadedAt": "timestamp"

}

Mối quan hệ dữ các collection cũng như các đối tượng:

- **Users** và **Boards**: Quan hệ n-n

- **Boards** và **Lists**: Quan hệ 1-n

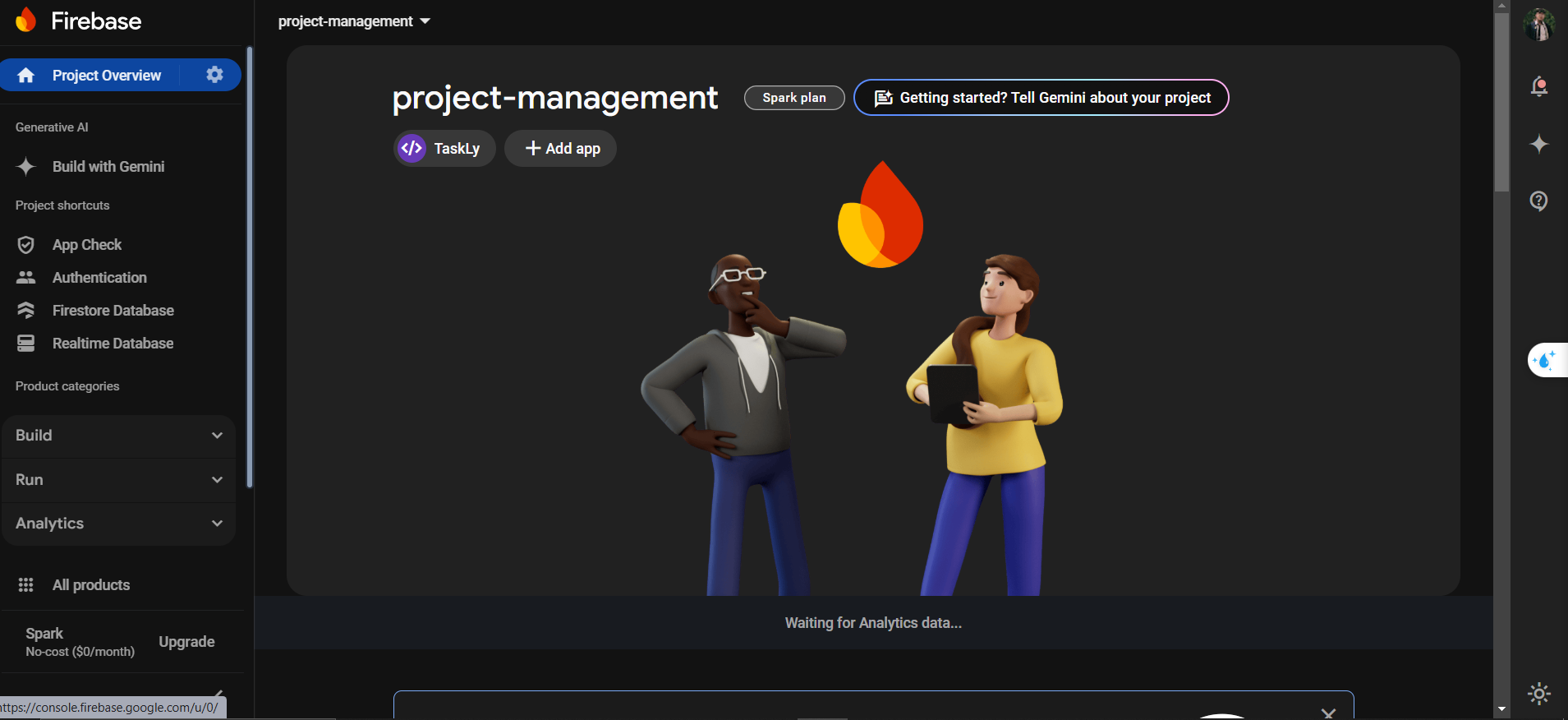
- **Lists** và **Cards**: Quan hệ 1-n

- **Cards** và **Comments**: Quan hệ 1-n

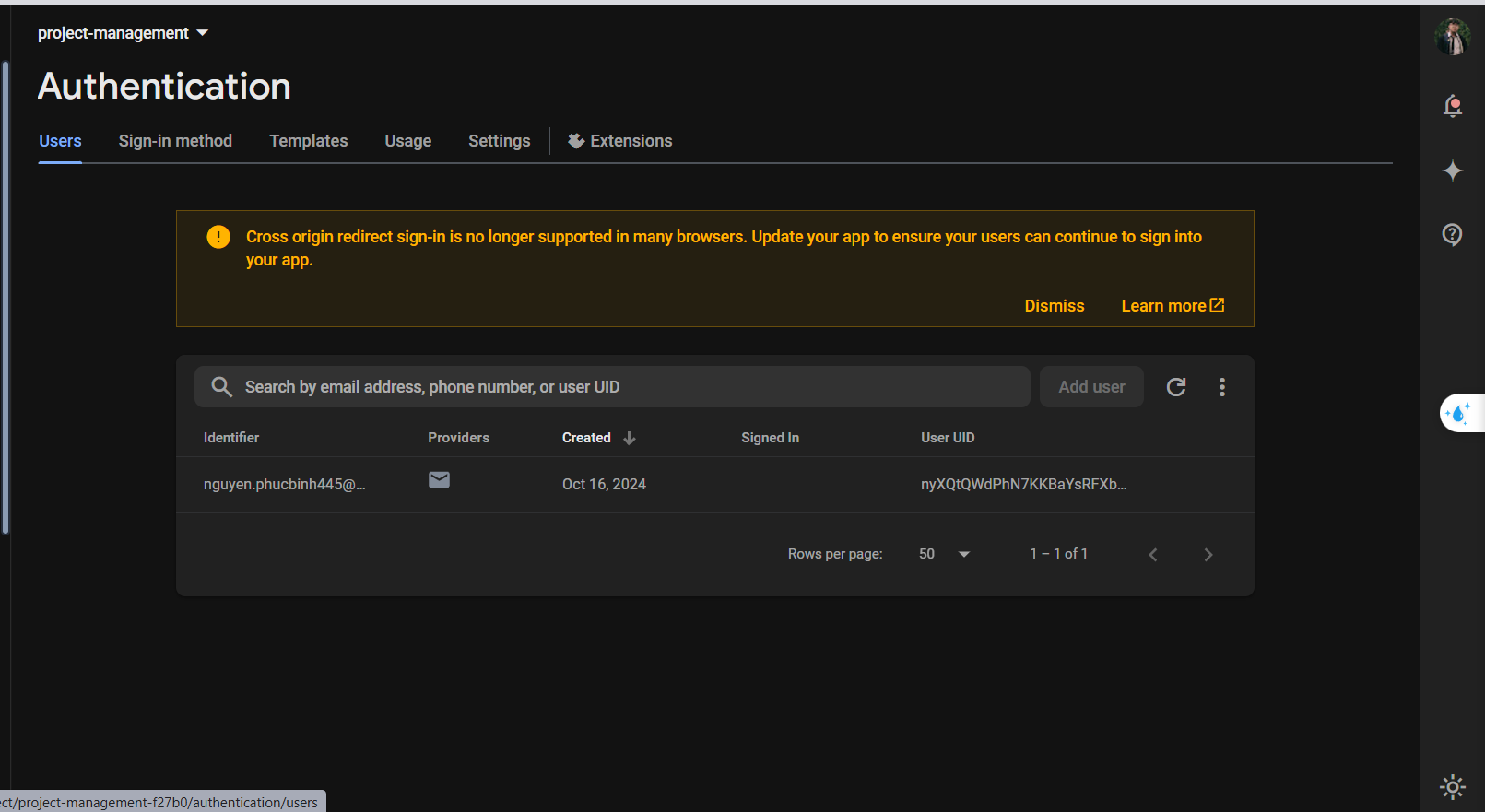
- **Cards** và **Attachments**: Quan hệ 1-n

## Cấu hình ứng dụng Firebase

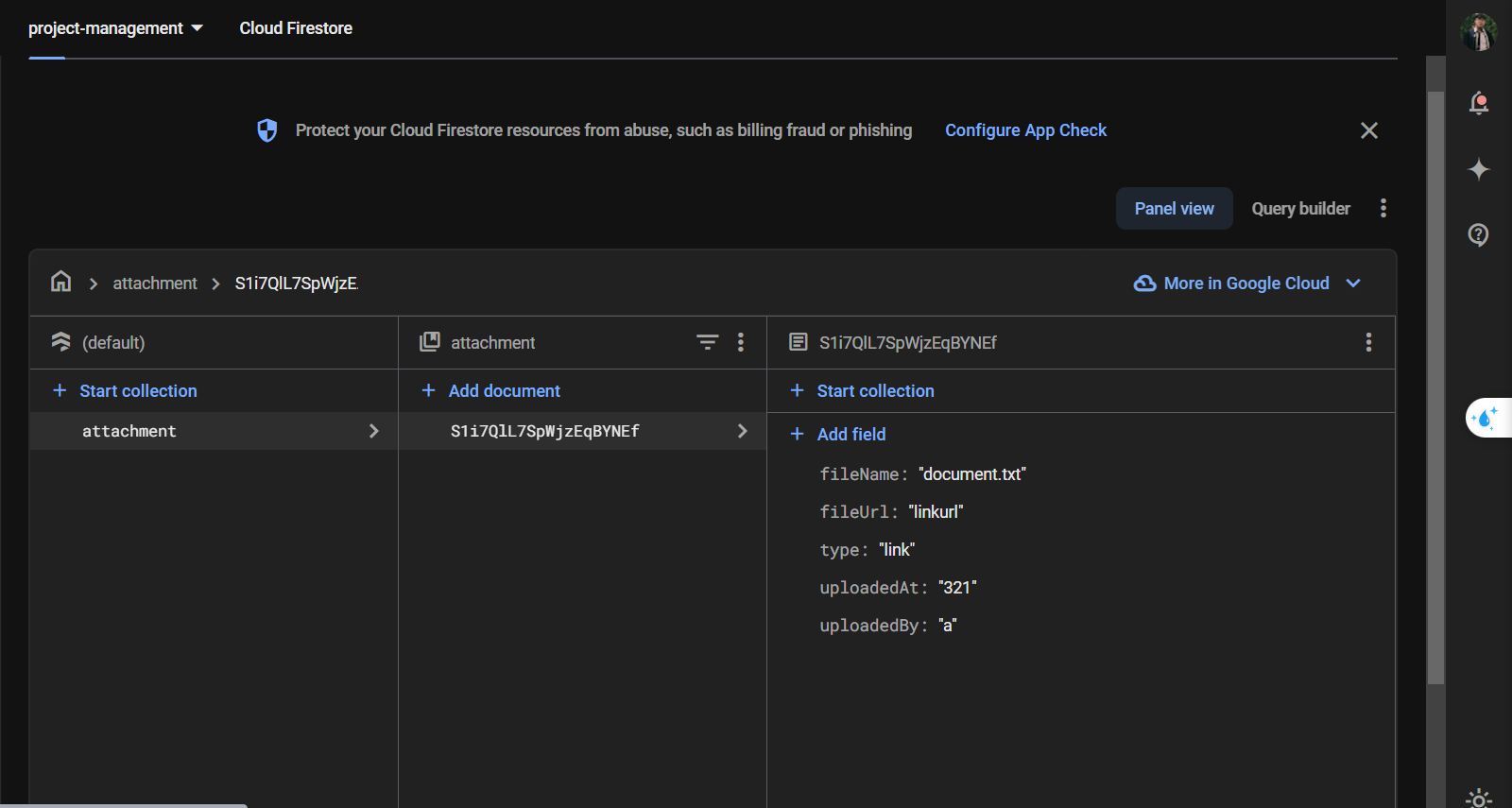
* Vô trang Firebase và tạo dự án bao gồm các việc đặt tên dự án chọn vùng của Cloud và Pricing Plan dựa trên nhu cầu sử dụng của người dùng



* Khởi tạo các Service sử dụng trong ứng dụng bao gồm Firebase authentication, Firestore, Storage,



*Firebase Authentication Service*

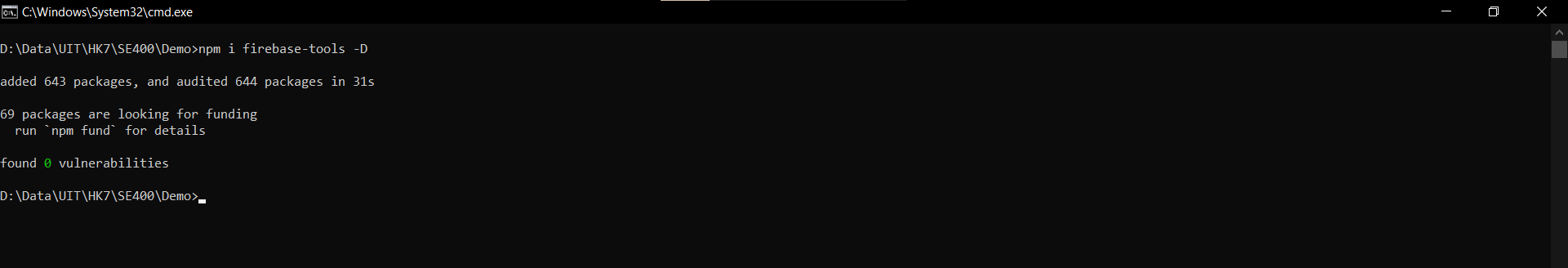
**

*Firestore Service*

Qua đó ta có thể sử dụng các service của Firebase thông qua các service này.

## Kết nối và khởi tạo các Firebase dưới dạng local

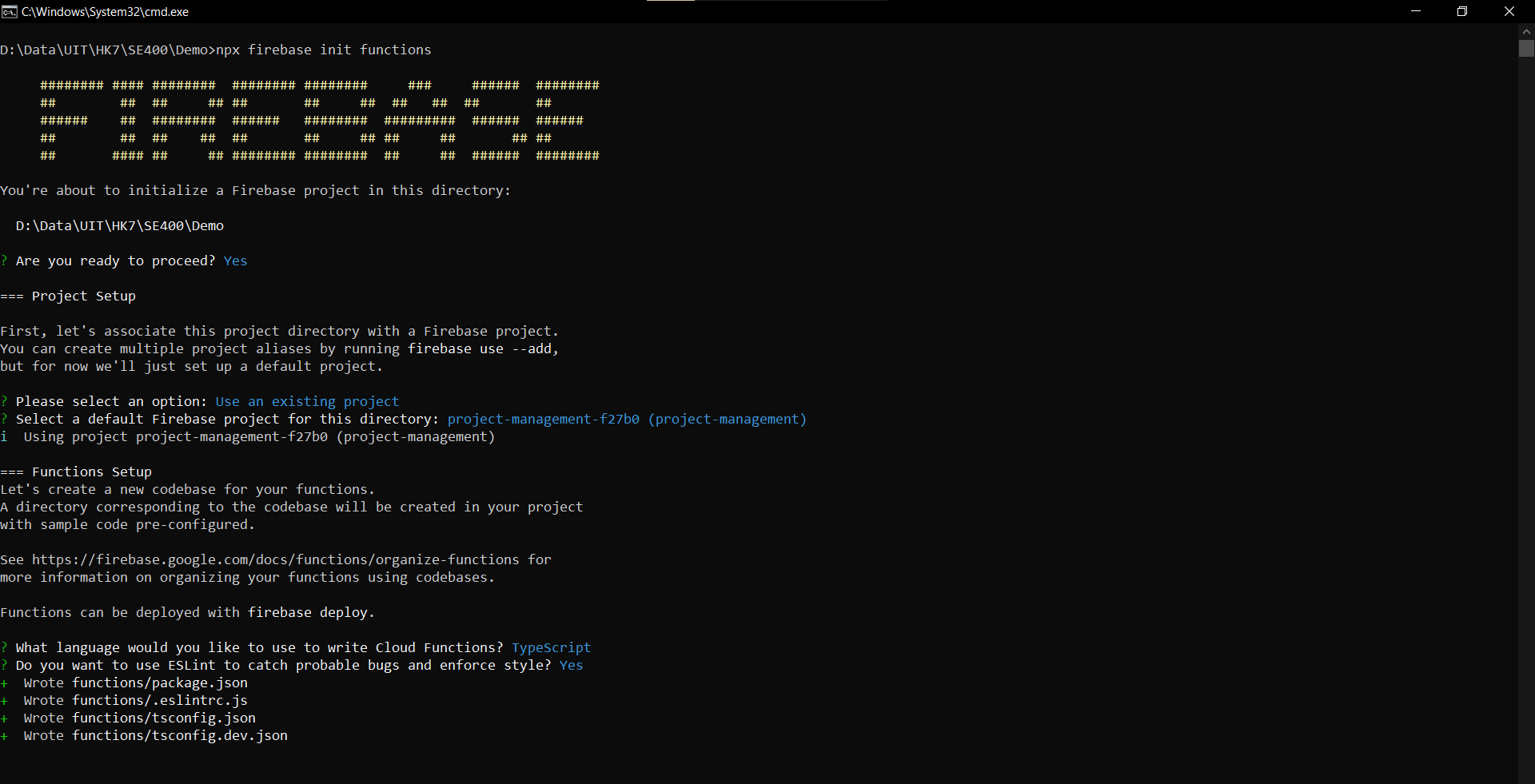
1. Tải Firebase-CLI về máy local thông qua câu lệnh: npm i Firebase-tools -D



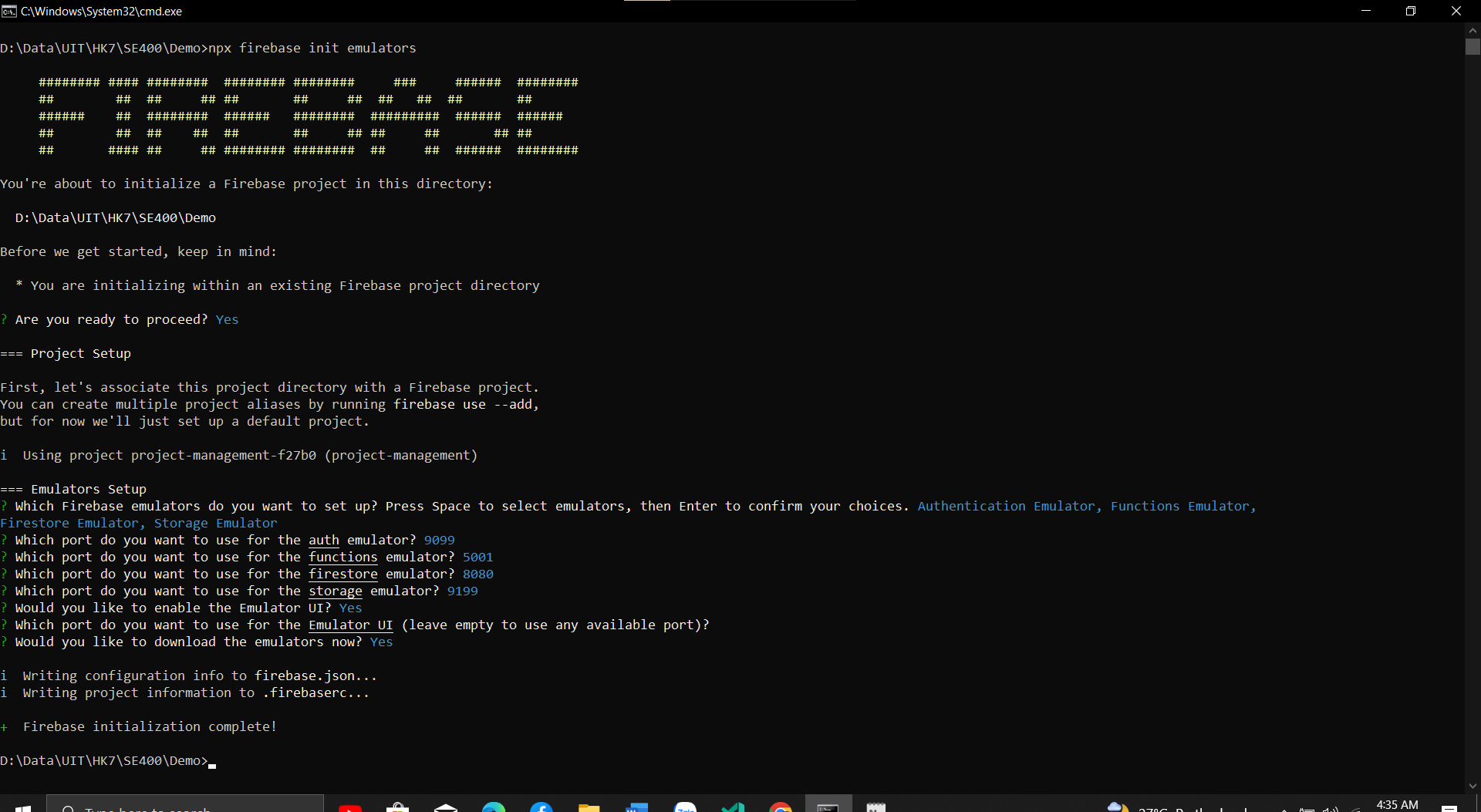
1. Kết nối với Firebase-CLI trên server bằng cách đăng nhập thông qua login tài khoản Firebase



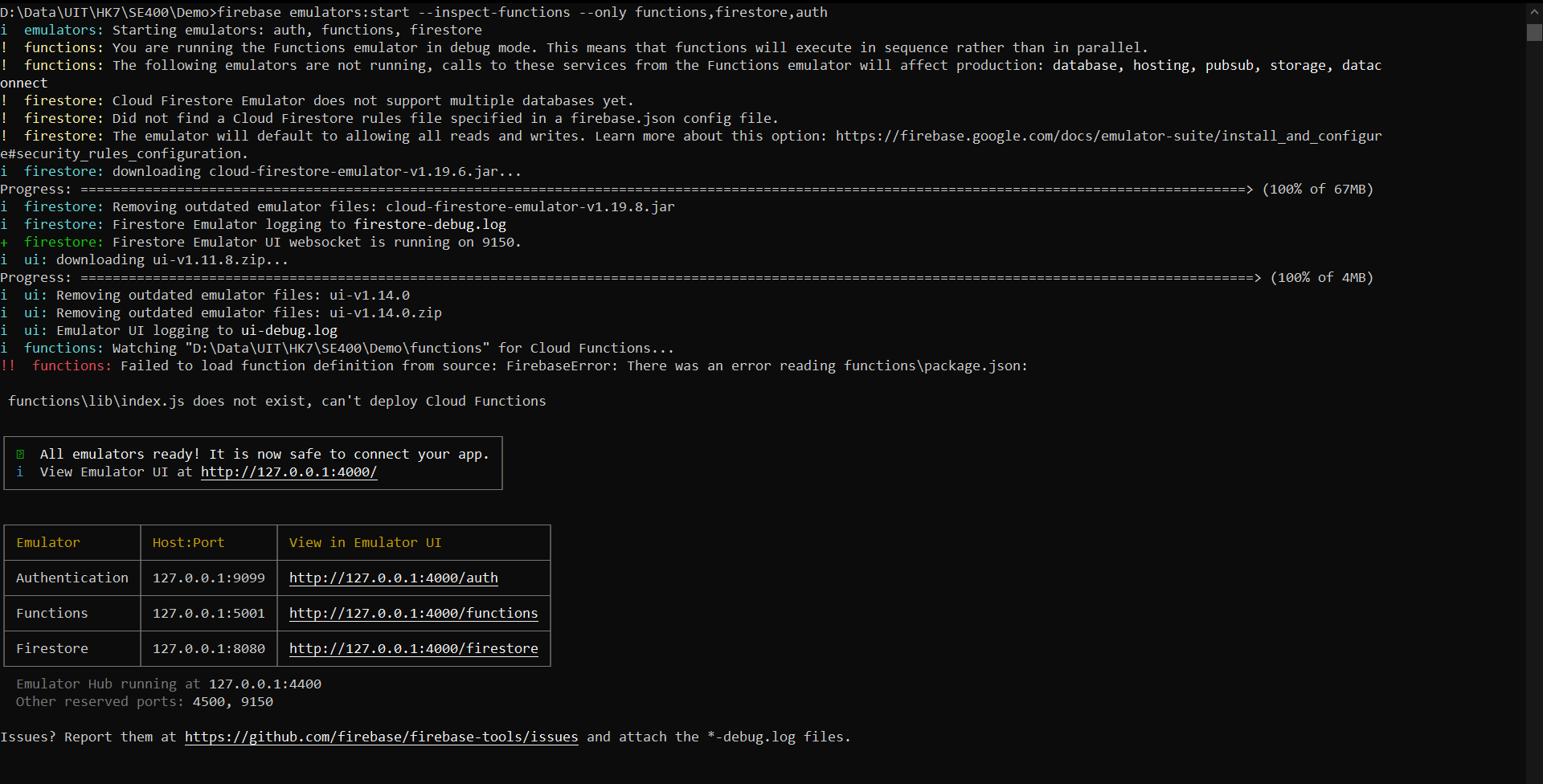
1. Tạo project function ở phía local và kết nối chúng với project đã có trên Firebase



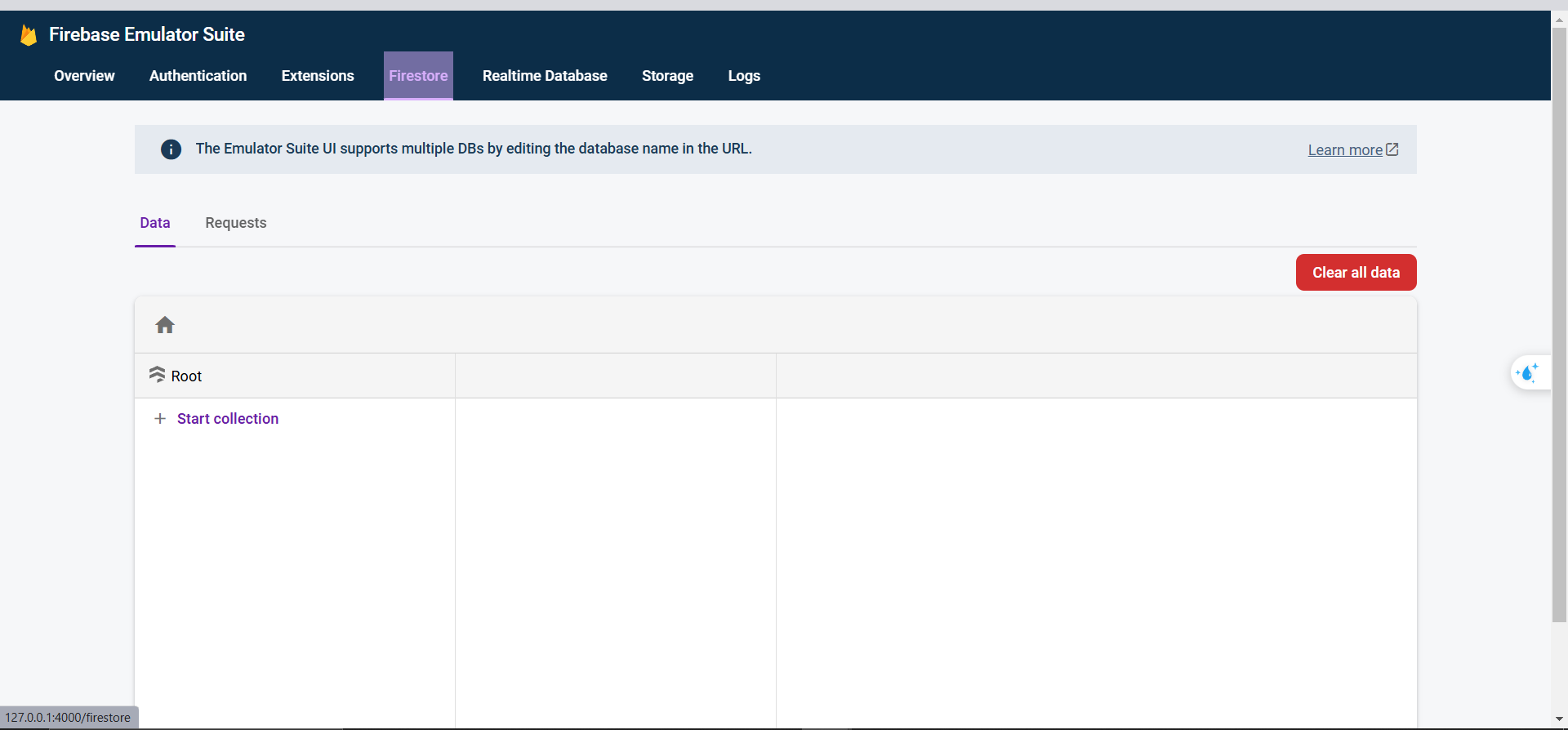
1. Dựa trên Firebase Emulators và tạo các máy ảo cho các service Authentication, Firestore, Firebase Function

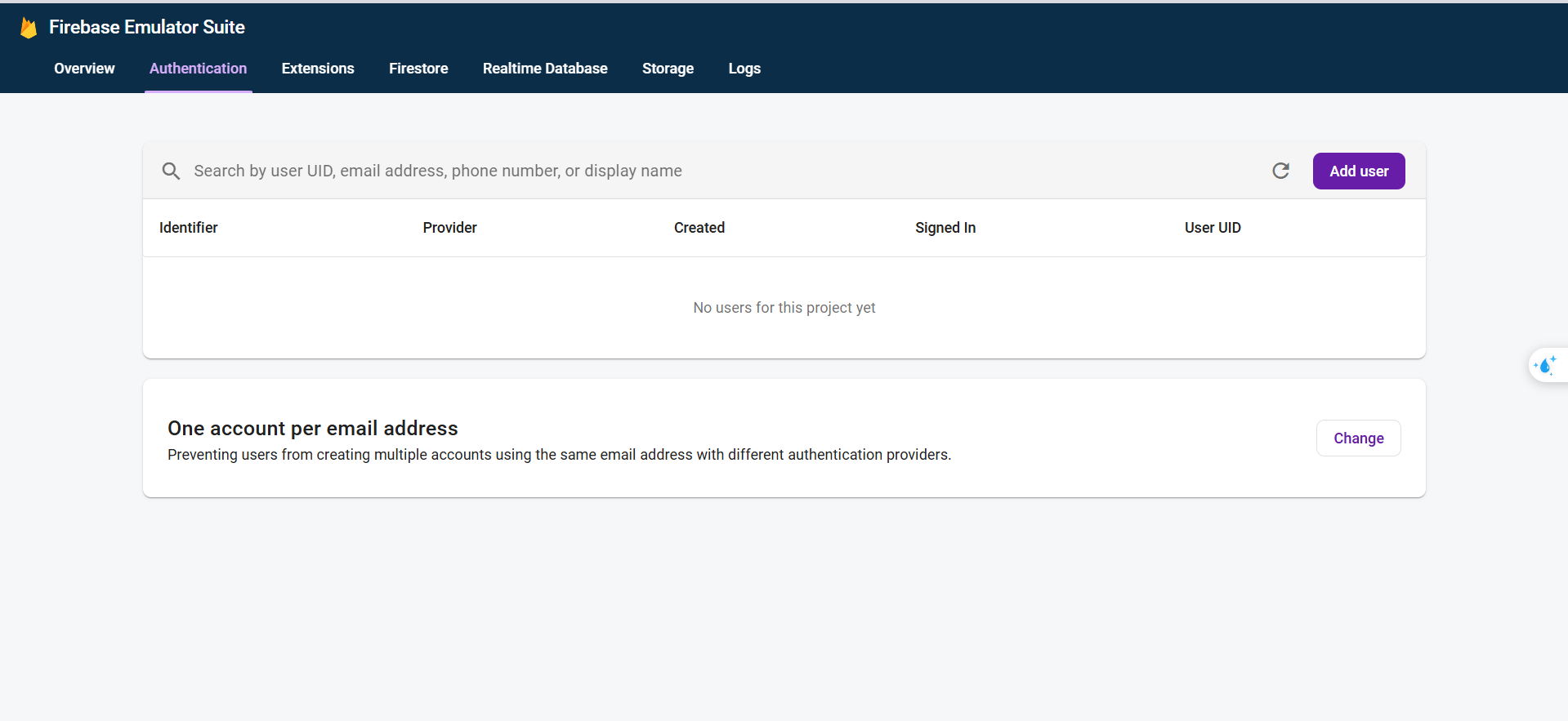


1. Ta có thể khởi chạy các service bằng lệnh: firebase emulators:start --inspect-functions --only functions,firestore,auth



1. Sau đó ta có thể tương tác và sử dụng các Firebase dưới dạng local mà không cần tương tác trực tiếp với server chính.





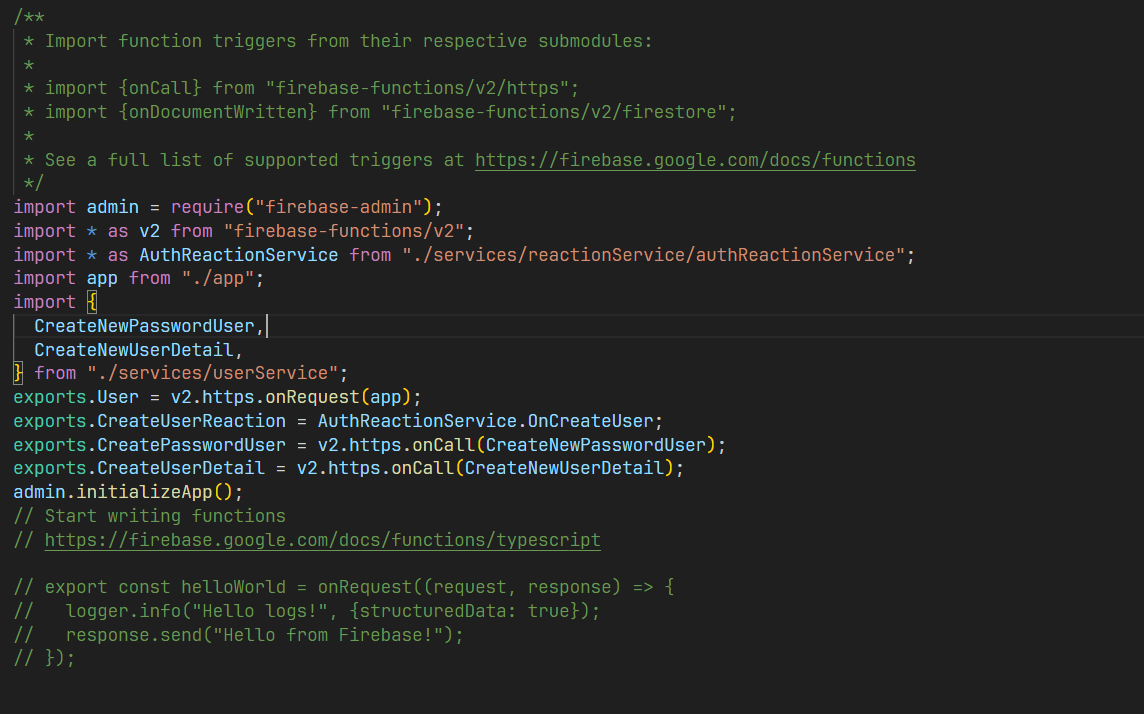
## Tạo tương tác giữa Client và Firebase

1. **Code các function serverless thông qua Firebase Function**

Đối với các Function của Firebase sẽ tồn tại 2 loại Function là HttpRequest Function và Reaction Function:

* HttpRequest Function là loại function mà người dùng chỉ cần tương tác thông qua request response như cách truyền thống tùy nhiên việc quản lý và mở rộng khi call các function đều được Firebase hỗ trợ thay vì tạo một server truyền thống
* Reaction Function là loại Function trigger mà khi người dùng tương tác với một trong các service của Firebase thì function này sẽ làm việc nếu như trường hợp 1 trigger nào đó phù hợp với yêu cầu diễn ra: Ví dụ khi ta up ảnh lên Firebase Storage thì Reaction Function sẽ chỉnh sửa các hình ảnh đó ở một độ phân giải phù hợp.

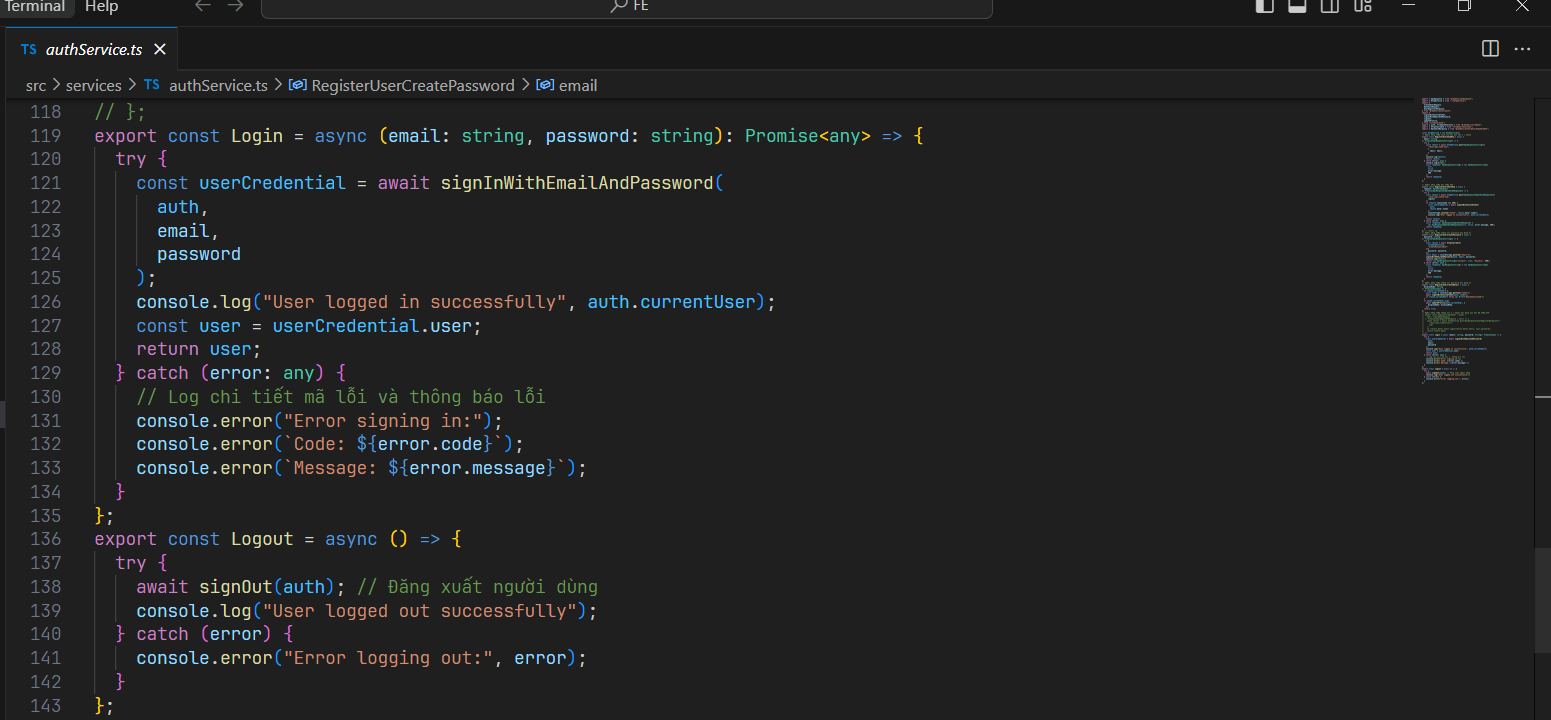
Với Firebase Function ta chỉ cần export các function cần thiết và phù hợp sau đó Firebase Function sẽ tự setup và build nó thay vì cần deploy như server truyền thống.



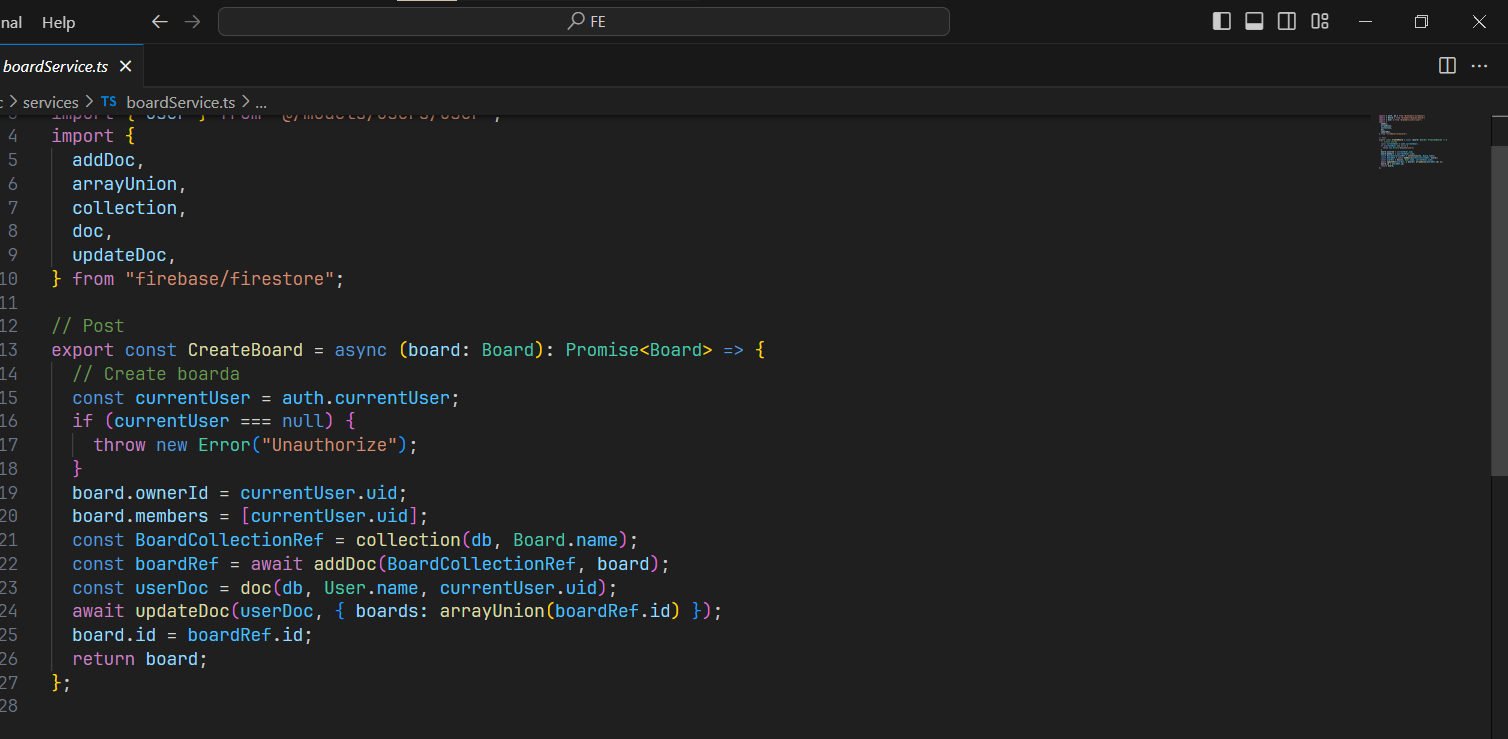
1. **Web Client trực tiếp tương tác với Firebase Service mà không cần thông qua server**

* Khi làm việc với Serverless Service thì người dùng ở phía web client không cần tương tác với phía backend mà có thể tương tác và thay đổi dữ liệu trực tiếp thông qua Firebase

Ở đây với các service ở phía front end có thể thực hiện các việc truy vấn hoặc đăng nhập dựa trên các hàm đã được hỗ trợ tốt từ phía Firebase mà ta không cần implement lại .



*Tương tác với Firebase Authentication*



*Tương tác dữ liệu trực tiếp với Firestore*

**Ưu điểm:**

* **Tương tác realtime:** Việc tương tác trực tiếp với dữ liệu của Firebase giúp cho phía client nhận biết sự thay đổi dữ liệu một cách nhanh nhất và thay đổi dữ liệu một cách nhanh chóng.
* **Triển khai ứng dụng nhanh chóng:** Có thể triển khai một cách nhanh chóng vì nhờ việc truy cập trực tiếp và sử dụng các service khác của firebase từ phía client giúp loại bỏ nhu cầu viết và duy trì backend server. Điều này giảm đáng kể thời gian phát triển và chi phí bảo trì server.
* **Quản lý và mở rộng quy mô:** Do hoạt động dưới dạng Serverless Service nên ta không cần quan tâm đến việc mở rộng quy mô ứng dụng dù ứng dụng có mở rộng ra như nào nữa.

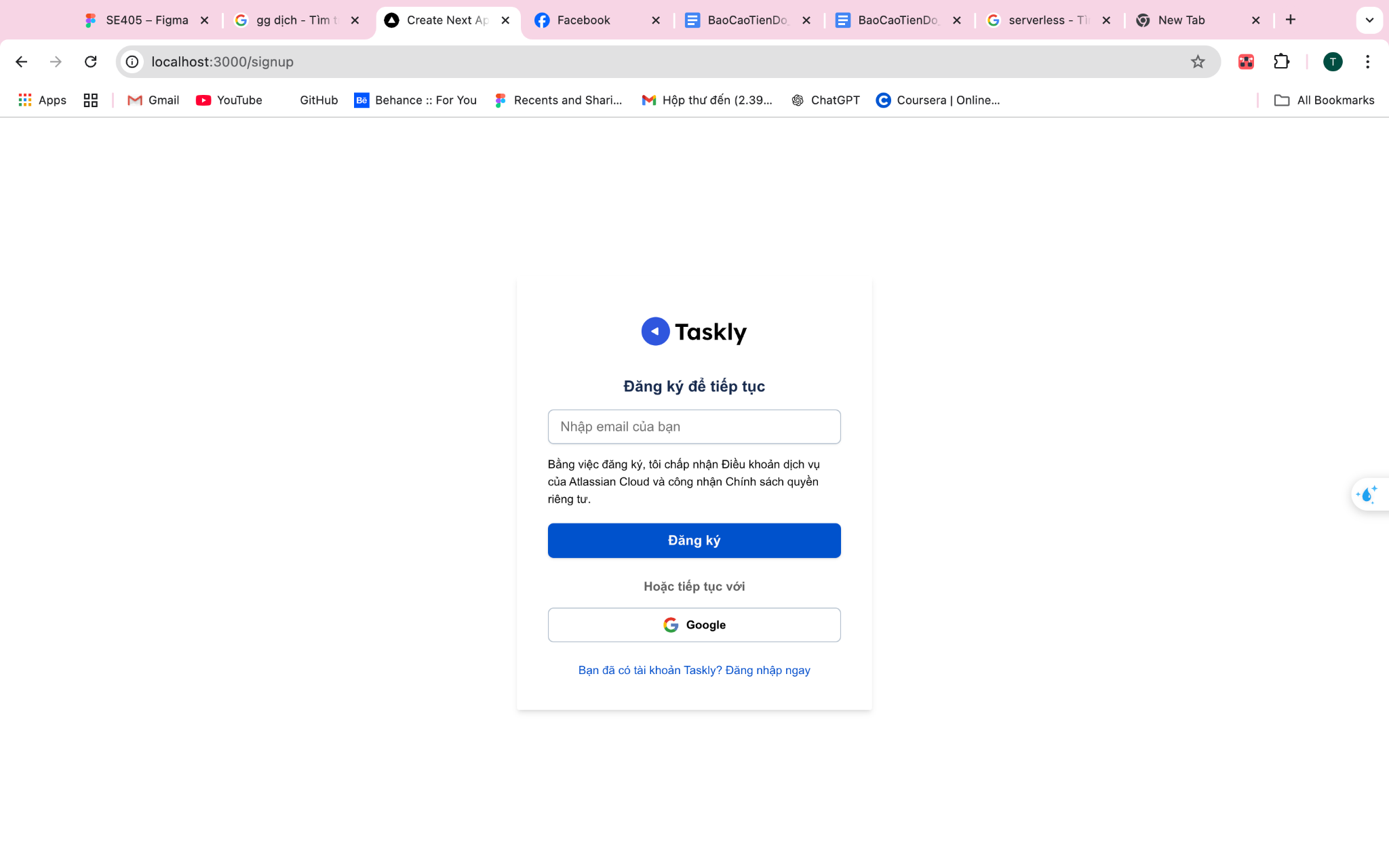
**Nhược điểm:**

* **Về bảo mật:** Việc tương tác trực tiếp từ client với cơ sở dữ liệu có thể làm lộ một số cấu trúc hoặc thông tin về database. Làm gây ra vấn đề bảo mật cho hệ thống nếu như không được quản lý một cách chặt chẽ.
* **Việc tái sử dụng kém:** Tuy việc xây dựng với firebase có thể triển khai ứng dụng một cách nhanh chóng nhưng nếu ta mở rộng ứng dụng trên các nền tảng khác nhau như web, android app, ios app thì ta sẽ tốn thời gian implement lại các phương thức từ đó gây tốn nhiều chi phí.

## Giao diện Người dùng

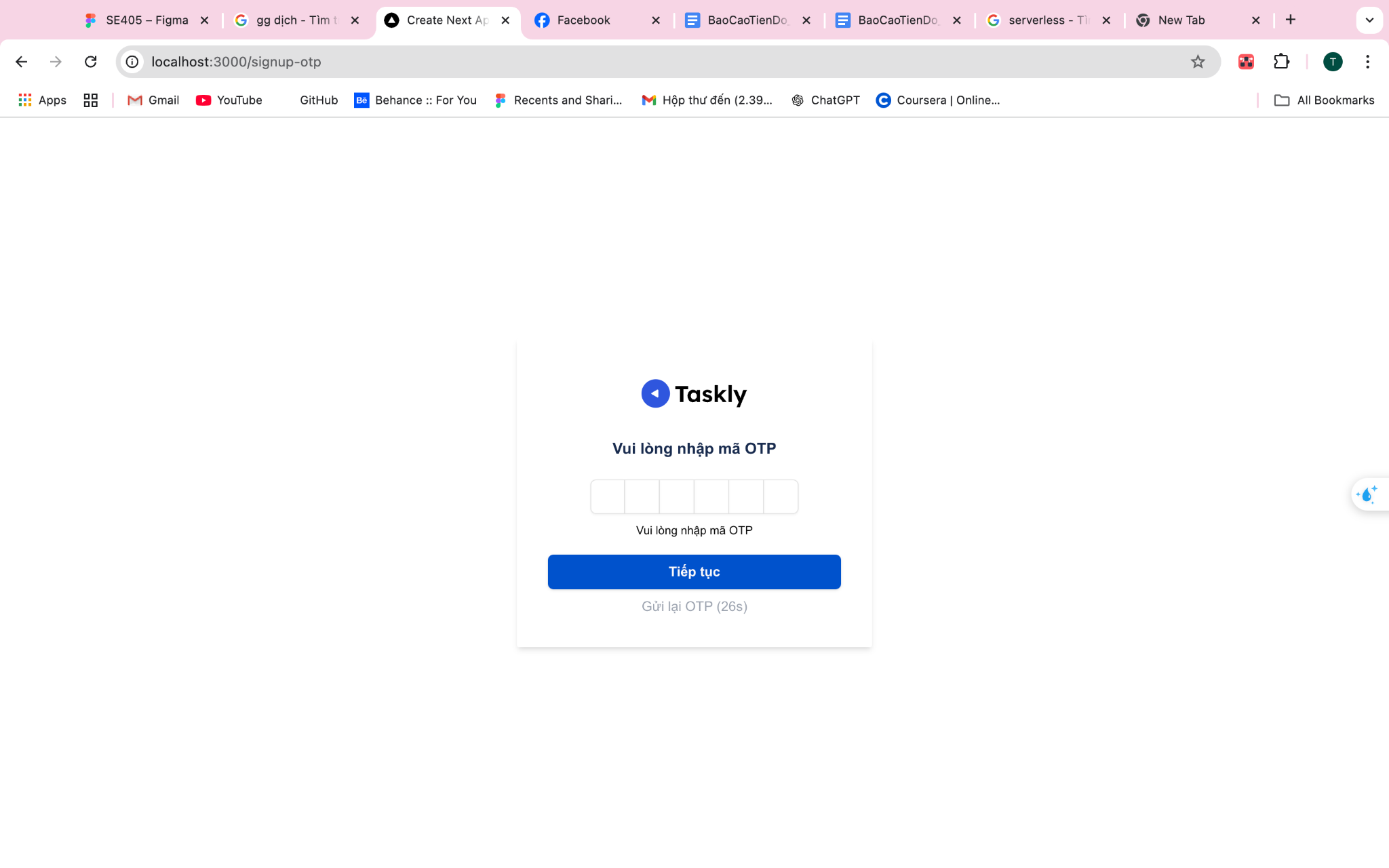
* + 1. **Tạo account**

Bạn có thể nhập email hoặc chọn đăng nhập bằng Google

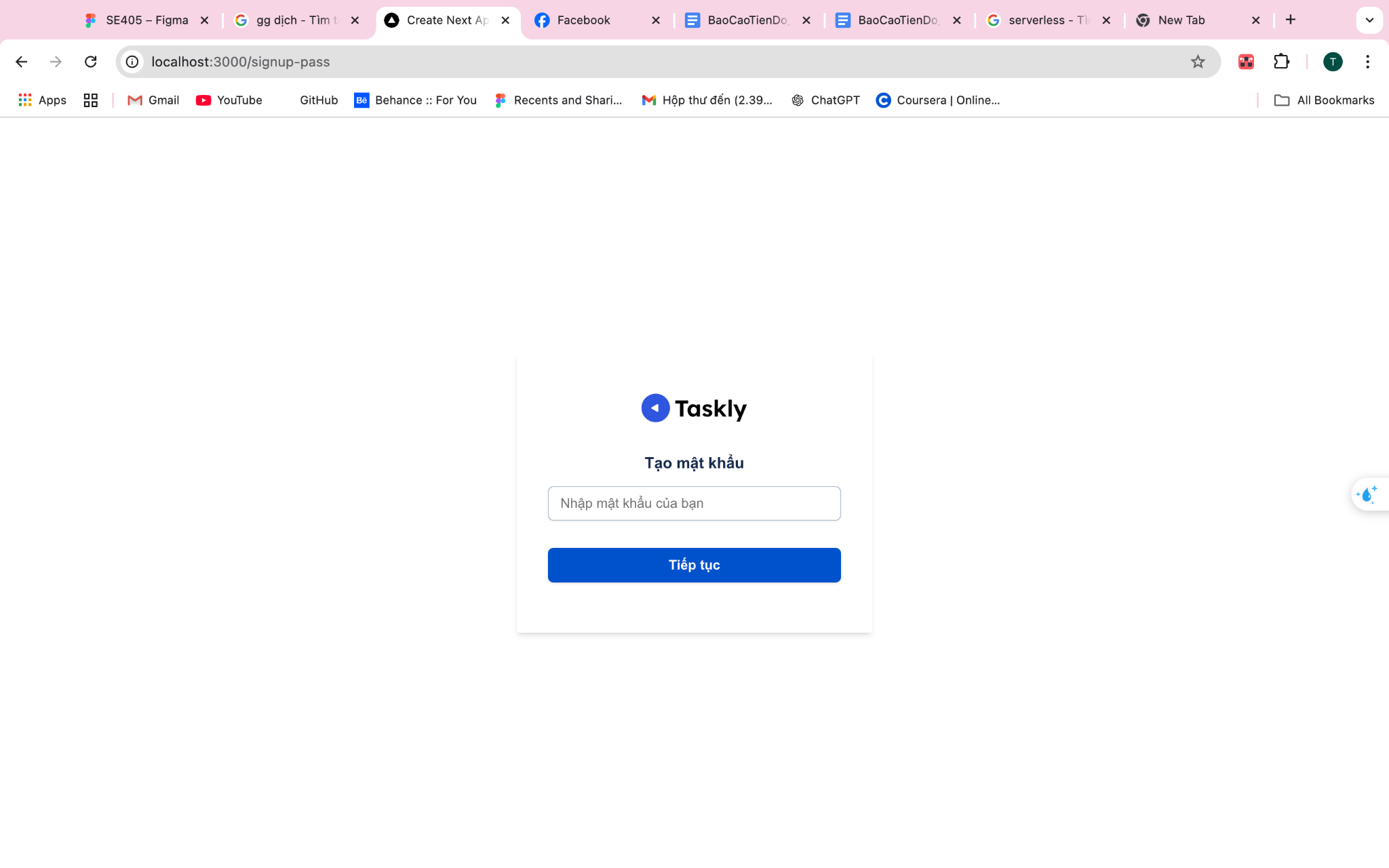


* + 1. **Nhập mã OTP**

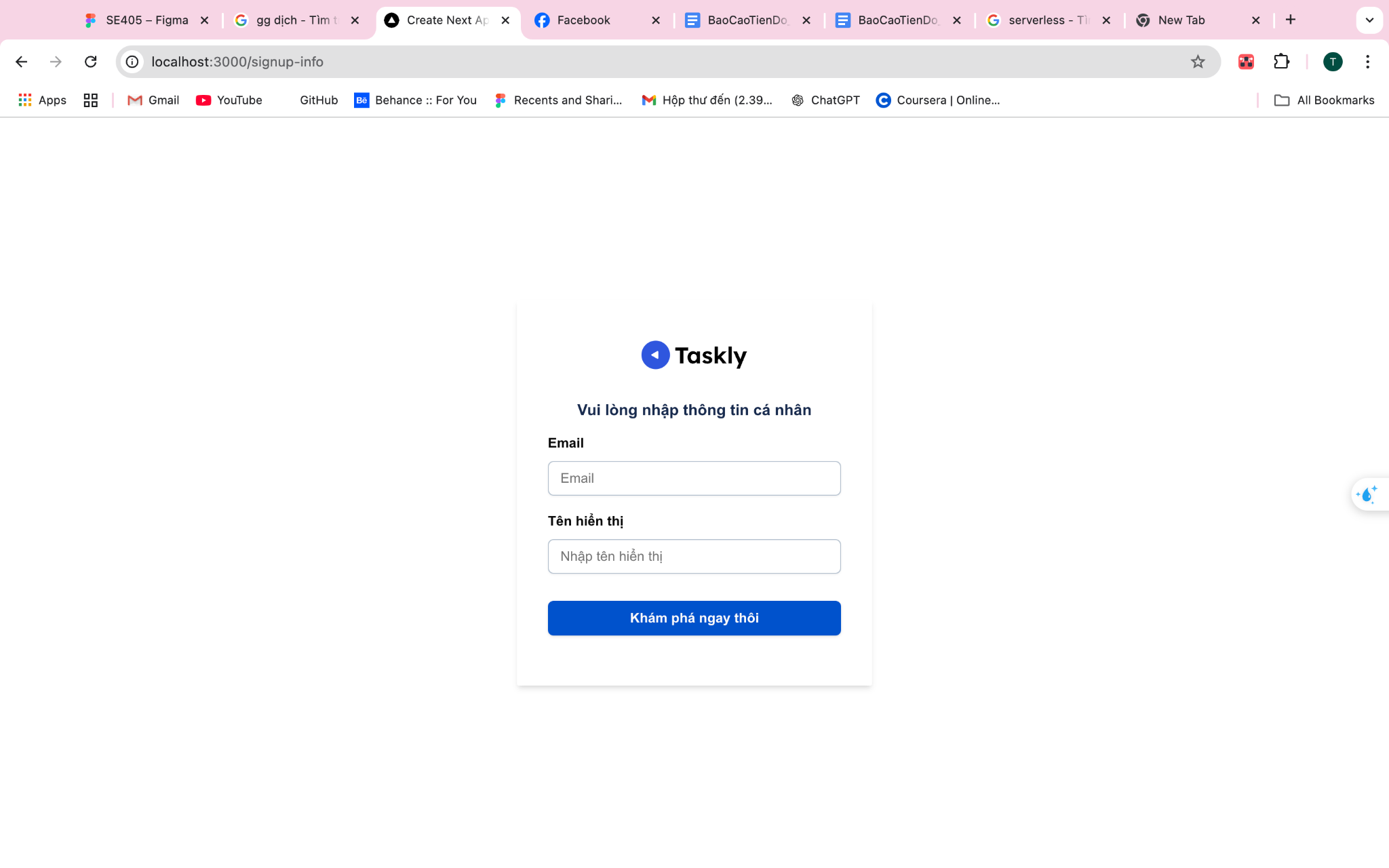
Nhập mã OTP để xác thực email



* + 1. **Tạo mật khẩu**

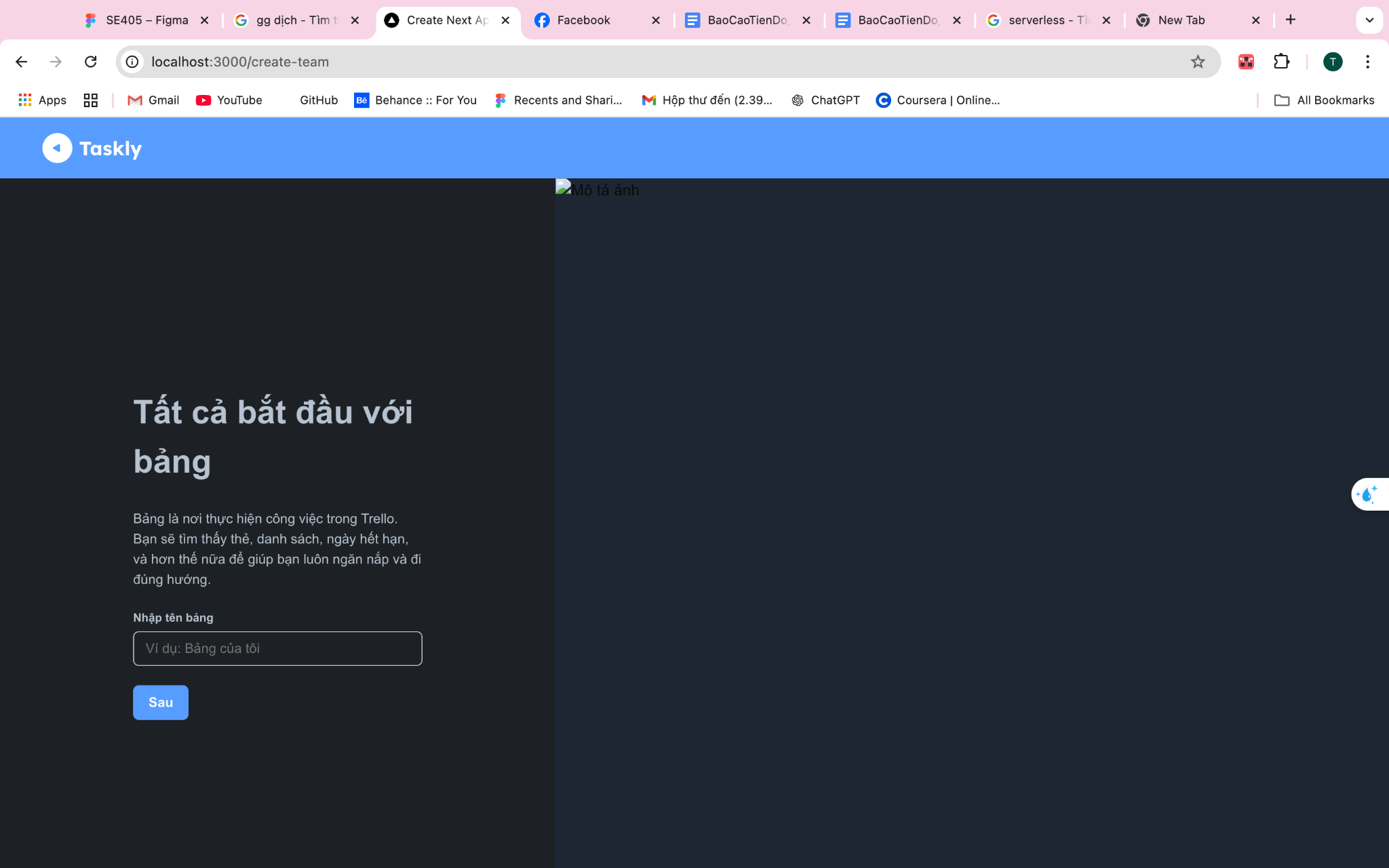


* + 1. **Nhập thông tin chi tiết**

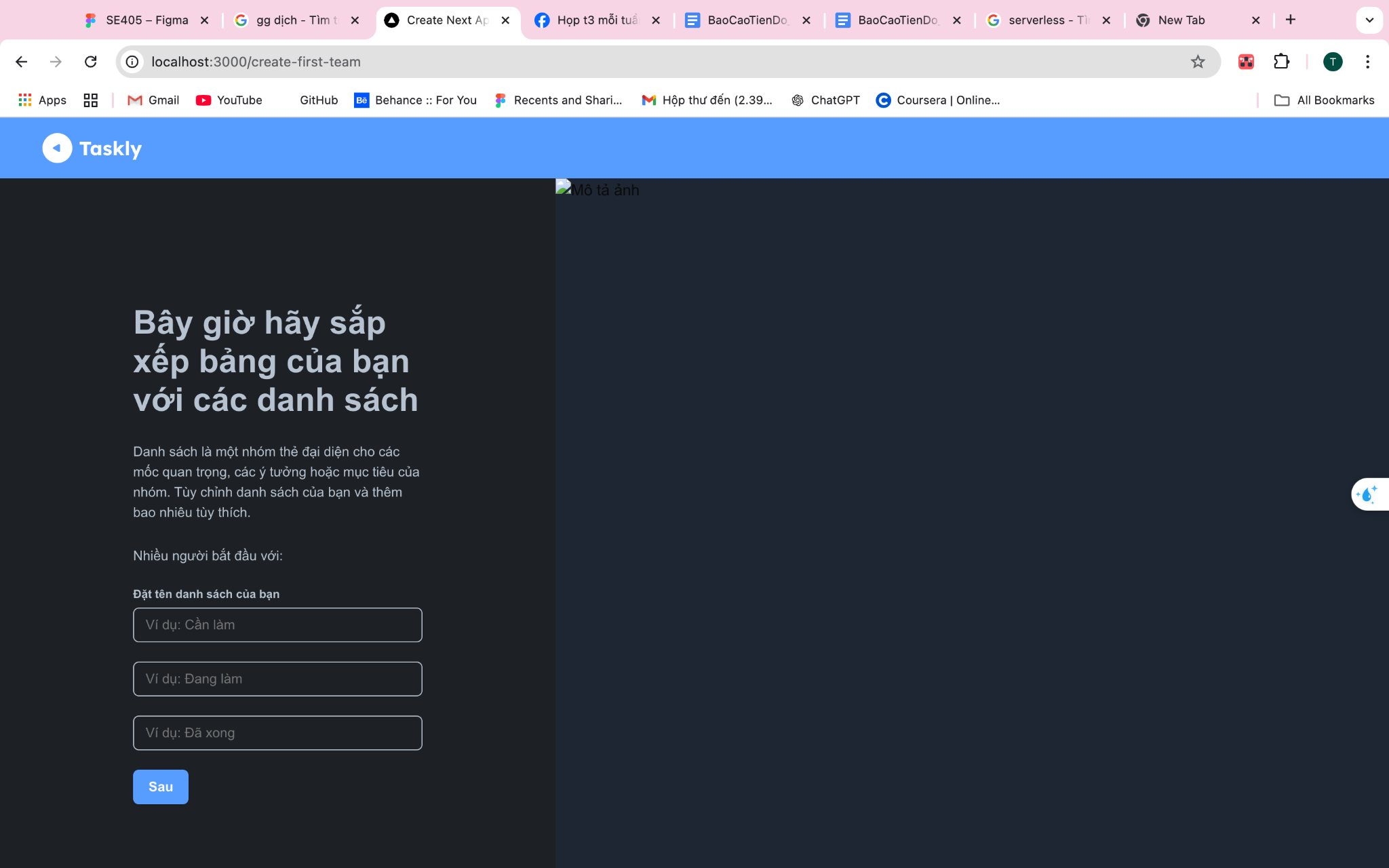


* + 1. **Tạo bảng**

Để bắt đầu sử dụng, bạn hãy tạo bảng



* + 1. **Đặt tên và tạo các task**



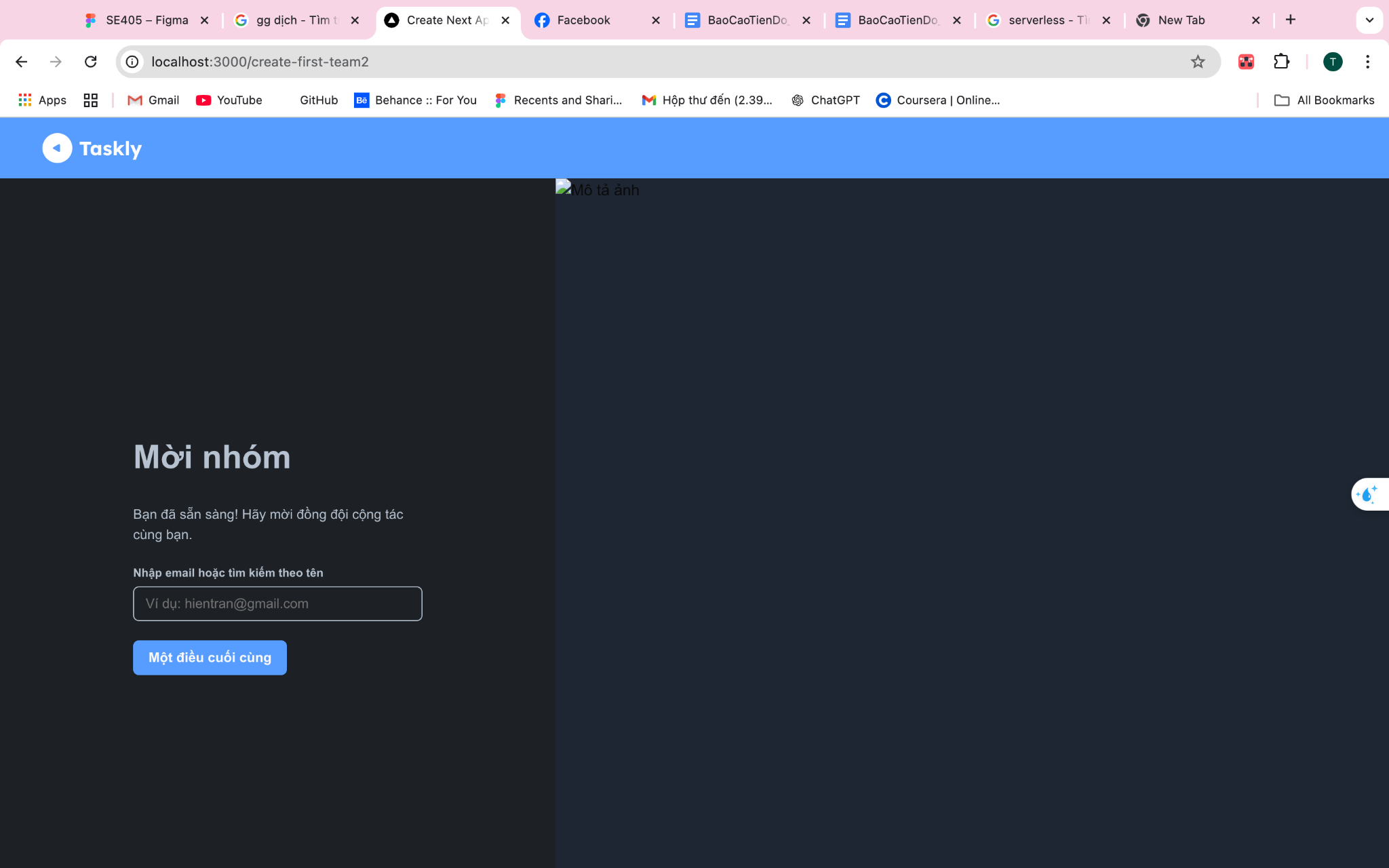
* + 1. **Đặt tên và tạo các thẻ trong card task**

Ví dụ như: SEM-12 code home page



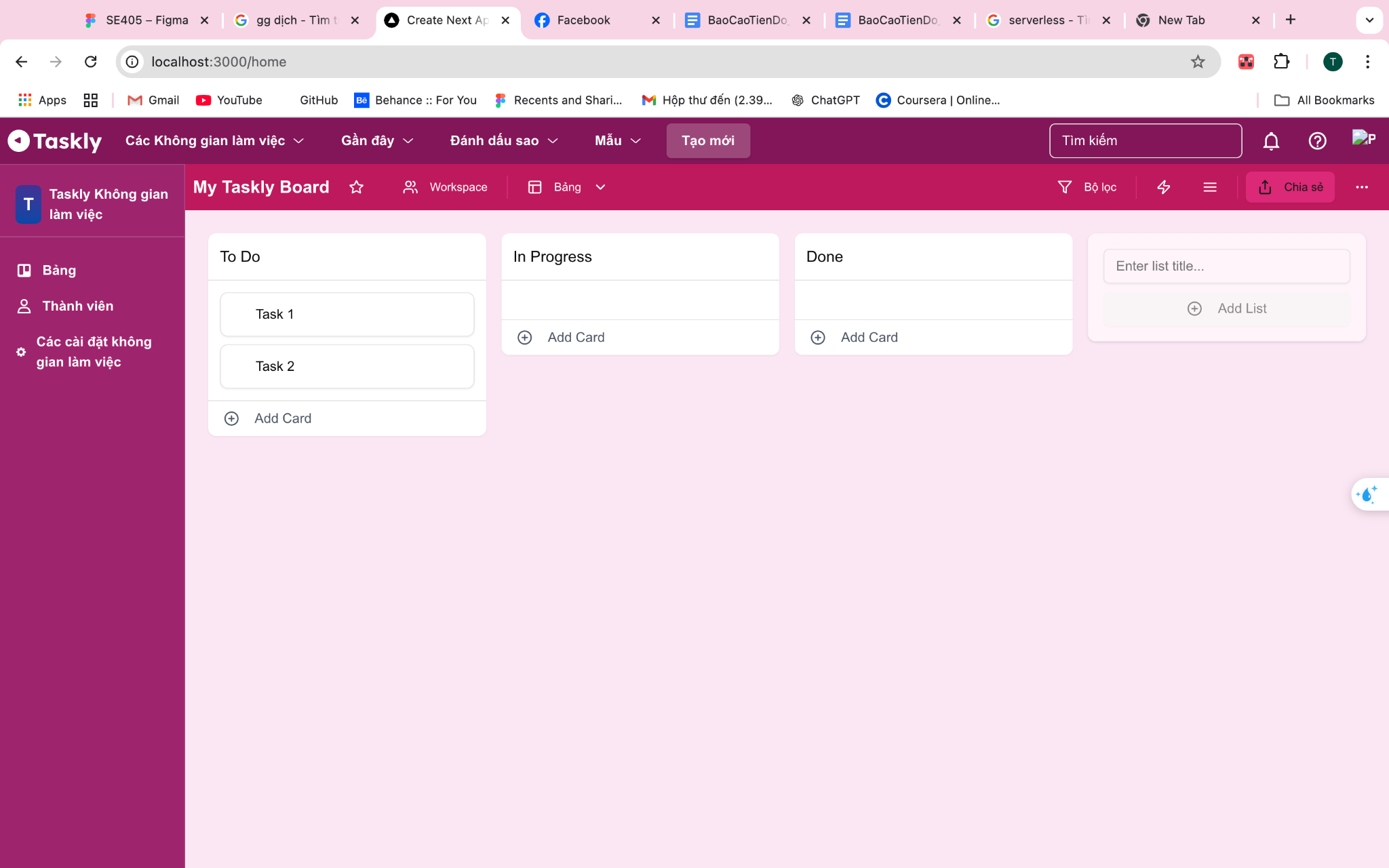
* + 1. **Mời nhóm người dùng**

Bạn có thể mời bạn bè join vào group Taskly để phân chia công việc



* + 1. **Trang home page**

Tại đây, chúng ta có thể create task, xoá task, chuyển vị trí task), chúng ta cũng có thể làm như vậy với các cột



## 

## Đánh giá việc triển khai ứng dụng web trên việc sử dụng dịch vụ Firebase Service

**Thuận lợi:**

* Việc triển khai ứng dụng với Firebase ban đầu khá ổn trong việc sử dụng và implement các phương thức mà nhóm mong muốn.
* Việc sử dụng các phương thức Firebase cung cấp dễ dàng và dễ dàng xử lí các tác vụ cơ bản.
* Việc sử dụng Firestore giúp việc phát hiện và xử lí realtime các đối tượng một cách dễ dàng hơn.
* Nhờ có Firebase Authentication mà việc kiểm tra và xác thực user và lưu trữ thông tin đăng nhập đã được Firebase cung cấp hỗ trợ nên việc xử lí các luồng kiểm tra Authentication một cách dễ dàng.

**Khó khăn:**

* Có thể gây tốn chi phí ít nhiều khi deploy và sử dụng các firebase function ở trên server chính của firebase tuy nhiên hiện tại nhóm chỉ triển khai ở mức local và chạy emulator nên vẫn chưa ảnh hưởng lắm.
* Do cấu trúc CSDL của Firestore khá khác với các kiểu cấu trúc CSDL mà nhóm đã quen sử dụng nên khi tiếp xúc với cấu trúc mới vẫn chưa tìm ra cách truy vấn nào có tốc độ thực thi tốt nhất cho hệ thống.
* Còn phát sinh một số lỗi kết nối khi kết nối từ client đến phía firebase emulators trong việc thực thi và tương tác với máy ảo.

## Kết luận

# 

## Kết quả đạt được

* Việc triển khai ứng dụng còn khá chậm so với kế hoạch dự tính ban đầu tuy nhiên vẫn đang ở tiến độ hợp lý.
* Tìm hiểu và nắm cách thức hoạt động của Firebase từ đó có thể triển khai ứng dụng chạy trên Firebase và kết nối với ứng dụng web của nhóm.
* Xây dựng và triển khai web dựa trên sự phát triển của Firebase

## 

## Bài học kinh nghiệm

* Từ việc tìm hiểu và ứng dụng Firebase Service Serverless vào trong ứng dụng thực tế nhóm nhận ra việc xây dựng ứng dụng của Firebase giúp dễ dàng và nhanh chóng xây dựng ứng dụng tuy nhiên chỉ phù hợp với các ứng dụng vừa và nhỏ vì các service của Firebase cung cấp các tính năng cơ bản, truy vấn và xử lí dữ liệu ở một mức nào đó.
* Cách tốt nhất để ứng dụng Firebase Service là chỉ nên ứng dụng ở một hoặc một số tính năng nào đó cần việc hỗ trợ đặc thù từ các service của Firebase ở mức độ cơ bản và dễ dàng.

## 

## Hướng phát triển

* Ứng dụng các tính năng khác của Firebase như FCM để thông báo cũng như nhận thông báo các task trong quá trình hoạt động.
* Sử dụng Firebase Analytics để theo dõi hiệu suất task thông qua việc sử dụng Firebase Analytics theo dõi hành vi người dùng, từ đó giúp nhóm phát triển cải thiện trải nghiệm quản lý công việc. Chúng ta có thể theo dõi các sự kiện như khi người dùng tạo task mới, hoàn thành task, hoặc thay đổi trạng thái task. Điều này sẽ giúp chúng ta hiểu rõ hơn về cách người dùng tương tác với ứng dụng và tối ưu hóa tính năng.TÀI LIỆU THAM KHẢO

1. **Firebase Documentation. "Build with Firebase." Firebase.** <https://firebase.google.com/docs/build>.
2. **Talbot, Mike. "Creating a Serverless Application with Firebase." Dev.to, 26 tháng 10, 2020.** <https://dev.to/miketalbot/creating-a-serverless-application-with-firebase-5d31>.
3. **"Tìm hiểu về Serverless (FaaS) - Deploy Functions lên Firebase Cloud Functions." Viblo, 5 tháng 5, 2021**. <https://viblo.asia/p/tim-hieu-ve-serverless-faas-deploy-functions-len-firebase-cloud-functions-yMnKMw0El7P>.
4. **Cloudflare. "Backend-as-a-Service (BaaS)." Cloudflare, 2024.** <https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/>.
5. **Cloudflare. "Serverless vs PaaS." Cloudflare, 2024.** <https://www.cloudflare.com/learning/serverless/glossary/serverless-vs-paas/>.

[Link github](https://github.com/leesoonduck3009/seminar-cnpm)