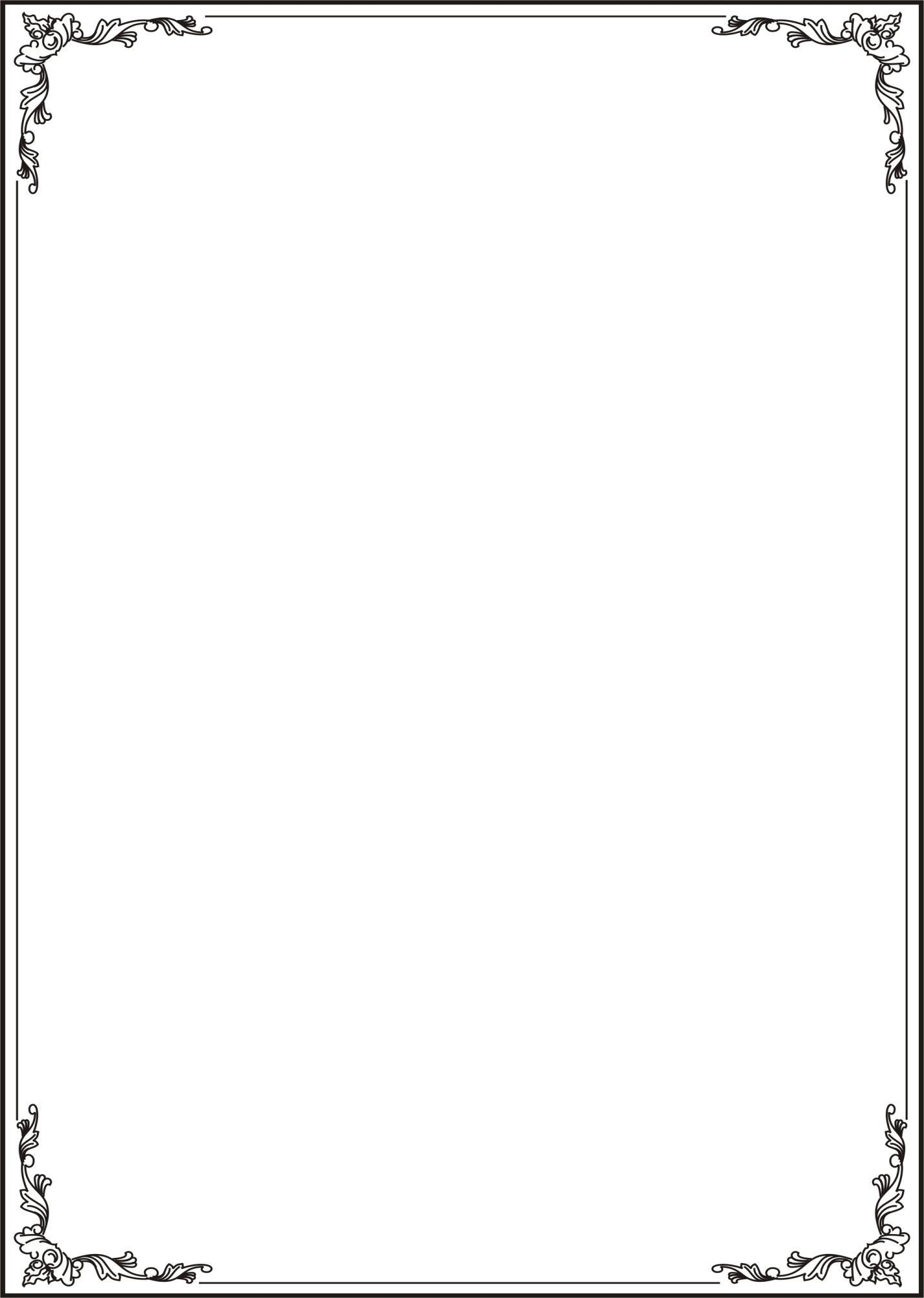
**.**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**BỘ MÔN ĐIỆN TOÁN ĐÁM MÂY**

-----🙞🙜🕮🙞🙜-----

****

**XÂY DỰNG ỨNG DỤNG TRÊN AWS CHO PHÉP TẠO DATABASE VÀ CUNG CẤP API ĐỂ THÊM, XÓA, SỬA TRÊN DATABASE**

Khóa 2019-2022

**GVHD: TS. HUỲNH XUÂN PHỤNG**

**SVTH:**

**NGUYỄN KIM HẠNH 20110468**

**ĐẶNG MINH CHÂU 20110445**

**ĐẶNG QUỐC DUY 20110060**

Khóa 2019 -2023

**TP.HCM 11/2022**

**MỤC LỤC**

[LỜI CẢM ƠN 4](#_Toc122202983)

[MỞ ĐẦU 5](#_Toc122202984)

[**1.** **Tính cấp thiết của đề tài** 5](#_Toc122202985)

[**2.** **Đối tượng nghiên cứu** 5](#_Toc122202986)

[**3.** **Phạm vi nghiên cứu** 5](#_Toc122202987)

[**4.** **Kết quả dự kiến** 5](#_Toc122202988)

[NỘI DUNG 6](#_Toc122202989)

[Chương 1: Cơ sở lý thuyết 6](#_Toc122202990)

[**1.** **Lambda** 6](#_Toc122202991)

[**1.1.** **Tổng quan về Lambda** 6](#_Toc122202992)

[**1.2.** **Các tính năng của Lambda** 6](#_Toc122202993)

[**1.3.** **Các dịch vụ liên quan đến Lambda** 7](#_Toc122202994)

[**2.** **DynamoDB** 7](#_Toc122202995)

[**2.1.** **Tổng quan về DynamoDB** 7](#_Toc122202996)

[**2.2.** **Các tính năng liên quan đến DynamoDB** 8](#_Toc122202997)

[Chương 2: Xây dựng ứng dụng 9](#_Toc122202998)

[**1.** **Giới thiệu website** 9](#_Toc122202999)

[**2.** **Giao diện** 9](#_Toc122203000)

[**3.** **Cài đặt** 12](#_Toc122203001)

[**3. 1.** **Cài đặt AWS S3** 12](#_Toc122203002)

[**3. 2.** **Cài đặt Lambda** 15](#_Toc122203003)

[**3. 3.** **Cài đặt Amazon API Gateway** 17](#_Toc122203004)

[**3. 4.** **Cài đặt DynamoDB** 20](#_Toc122203005)

[**3. 5.** **Cài đặt CloudFront** 20](#_Toc122203006)

[**4.** **Các chức năng thực hiện được** 21](#_Toc122203007)

[**3. 1.** **Xác thực tài khoản** 21](#_Toc122203008)

[**3. 2.** **Quản lý table** 24](#_Toc122203009)

[**3. 3.** **Quản lý table items** 25](#_Toc122203010)

[KẾT LUẬN 27](#_Toc122203011)

[**1.** **Kết quả đạt được:** 27](#_Toc122203012)

[**2.** **Ưu điểm:** 27](#_Toc122203013)

[**3.** **Nhược điểm** 27](#_Toc122203014)

[**4.** **Hướng phát triển:** 27](#_Toc122203015)

# **LỜI CẢM ƠN**

Lời đầu tiên nhóm xin phép được gửi lời cảm ơn chân thành đến Khoa Đào Tạo Chất Lượng Cao – Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh đã tạo điều kiện cho nhóm chúng em được học tập, phát triển nền tảng kiến thức sâu sắc để có thể thực hiện đề tài này.

Bên cạnh đó chúng em xin gửi lời cảm ơn đến thầy Huỳnh Xuân Phụng đã chỉ dạy những kiến thức bổ ích của bộ môn Điện Toán Đám Mây và hướng dẫn chúng em thực hiện đề tài Xây Dựng Ứng Dụng Trên AWS Cho Phép Tạo Database Và Cung Cấp API Để Thêm, Xóa, Sửa Trên Database.

Chúng em xin cảm ơn thầy vì thời gian qua đã hướng dẫn và chỉ dạy chúng em nhiệt tình giúp chúng em nắm được những kiến thức bổ ích của môn học này.

Tuy nhiên, trong quá trình tìm hiểu đề tài, chúng em có thể còn nhiều thiếu sót nên mong thầy có thể thông cảm và góp ý để chúng em có thể hoàn thiện đề tài hơn.

Chúng em xin chân thành cảm ơn!

# **MỞ ĐẦU**

1. **Tính cấp thiết của đề tài**

Cơ sở dữ liệu (database) nói chung và cơ sở dữ liệu không quan hệ (noSQL database) nói riêng là một thành phần không thể thiếu trong việc tạo nên các phần mềm ứng dụng hoặc website có như cầu lưu trữ thông tin có thể hoặc không do người dùng nhập vào. Nhờ chúng mà dữ liệu với những insight quan trọng có thể được lưu trữ một cách dễ dàng. Vì vậy mà việc tạo nên một hệ thống có khả năng làm cho việc tương tác đến cơ sở dữ liệu của các lập trình viên trở nên dễ dàng hơn là một việc cấp thiết cần phải được thực hiện.

1. **Đối tượng nghiên cứu**

Đối với đề tài này, đối tượng nghiên cứu là Lambda, DynamoDB. Đồng thời kết hợp với các kiến thức về Front-end và Back-end để tạo một trang web cụ thể.

1. **Phạm vi nghiên cứu**

Đề tài này chủ yếu tập trung vào việc kết hợp sử dụng Lambda và DynamoDB nhằm thực hiện các câu truy vấn để thêm, sửa, xóa dữ liệu trên database.

1. **Kết quả dự kiến**

* Cài đặt các giao diện
* Cài đặt phía backend
* Thêm, xóa các table trong DynamoDB
* Thêm, sửa, xóa các item trong DynamoDB

# **NỘI DUNG**

# **Chương 1: Cơ sở lý thuyết**

1. **Lambda**
   1. **Tổng quan về Lambda**

AWS Lambda là nơi để cho lập trình viên upload các function code của mình dưới dạng các ngôn ngữ được hỗ trợ, và AWS Lambda sẽ giúp người lập trình chạy các đoạn code đó dựa trên các thư viện có sẵn của AWS, đồng thời AWS Lambda sẽ lưu lại các tiến trình thay đổi của function và lưu vào CloudWatch.

AWS Lambda được hỗ trợ và quản lý bởi đội ngũ AWS nên các function sẽ thực hiện code của lập trình viên khi cần, và có khả năng tự động mở rộng, từ một vài request đến hàng trăm ngàn request trên giây. Nhờ vào các ưu điểm này mà Lambda có thể dễ dàng sử dụng để xây dựng các ứng dụng cùng với các dịch vụ khác của Lambda như S3, DynamoDB,... với các hiệu suất và độ bảo mật cao.

* 1. **Các tính năng của Lambda**

Các tính năng chính sau đây giúp bạn phát triển các ứng dụng Lambda có khả năng mở rộng, bảo mật và dễ dàng mở rộng:

**Khả năng mở rộng**

AWS Lambda cho phép lập trình viên dễ dàng kết hợp với các dịch vụ khác như S3, DyanmoDB,.... vì vậy có thể dễ dàng kích hoạt các Lambda Function, khi có các luồng dữ liệu hoặc tài nguyên có sự thay đổi giúp lập trình viên hiểu được các tiến trình của các Request

**Xây dựng Back-end**

AWS Lambda cho phép lập trình viên có thể tạo các backend cho ứng dụng bằng cách sử dụng các API Lambda hoặc API endpoint của Amazon API Gateway. Lambda sẽ tiến thành xử lý các event thay vì cung cấp các event lên client, nhờ đó giúp giảm tiêu hao các tài nguyên, dễ dàng bảo trì cũng như cập nhật hơn.

**Sử dụng code riêng**

AWS Lambda hỗ trợ đa dạng các ngôn ngữ cũng như cung cấp đầy đủ các API Runtime, ngoài ra AWS còn hỗ trợ lập trình viên có thể sử dụng bất kỳ thư viên, kể cả các thư viện thuần, đa dạng các framework, lập trình viên cũng có thể đóng gói các source code dưới dạng các class của Lambda, dễ dàng chia sẻ cũng như quản lý các code trên nhiều Function khác nhau.

**Các quản lý được tự động hoàn toàn**

Lambda được quản lý hoàn toàn bởi các đội ngũ AWS nên có độ sẵn sàng và khả năng chịu lỗi cao, ngoài ra lập trình viên sẽ không cần cập nhật các bản vá cũng như cập nhật dung lượng, bổ sung máy chủ mới. AWS sẽ thực hiện tất cả các hoạt động bảo trì, vá lỗi, cung cấp chức năng lưu trữ nhật ký của các quá trình Lambda thông qua Amazon CloudWatch.

* 1. **Các dịch vụ liên quan đến Lambda**

Lambda tích hợp với các dịch vụ AWS khác để gọi các hàm dựa trên các sự kiện được chỉ định:

+ API Gateway cung cấp các đường dẫn có tính bảo mật cao, có thể mở rộng với các API web định tuyến đến các request HTTP và kích hoạt đến các Lambda Function.

+ Đối với các dịch vụ tạo hàng đợi hoặc các luồng dữ liệu (chẳng hạn như DynamoDB và Kinesis), Lambda sẽ xem xét các hàng đợi hoặc luồng dữ liệu từ dịch vụ và gọi Function để xử lý dữ liệu của các hàng đợi hoặc luồng dữ liệu.

+ Xác định các sự kiện Amazon S3 gọi một hàm Lambda để xử lý các đối tượng Amazon S3, chẳng hạn như khi một đối tượng được tạo hoặc xóa.

+ Sử dụng chức năng Lambda để xử lý các thông báo Amazon SQS messages hoặc Amazon Simple Notification Service (Amazon SNS).

+ Sử dụng chức năng từng bước AWS để kết nối các chức năng Lambda với nhau thành quy trình làm việc không máy chủ được gọi là máy trạng thái.

1. **DynamoDB**
   1. **Tổng quan về DynamoDB**

DynamoDB là một cơ sở dữ liệu NoSQL hỗ trợ các mô hình dữ liệu khóa-giá trị và tài liệu. Các nhà phát triển có thể dùng DynamoDB để xây dựng các ứng dụng hiện đại phi máy chủ. Những ứng dụng này có thể bắt đầu hoạt động trên quy mô nhỏ và mở rộng ra quy mô toàn cầu để hỗ trợ nhiều petabyte dữ liệu cũng như hàng triệu yêu cầu đọc và ghi mỗi giây. DynamoDB được thiết kế để chạy các ứng dụng hiệu năng cao trên quy mô internet. Đó là các ứng dụng sẽ khiến cơ sở dữ liệu quan hệ truyền thống phải hoạt động quá mức.

DynamoDB được quản lý đầy đủ hỗ trợ cả mô hình dữ liệu dưới dạng tài liệu và cặp value-key.

Mô hình dữ liệu rất linh hoạt và hiệu suất cao phù hợp cho nhiều loại ứng dụng, nền tảng khác nhau.

* 1. **Các tính năng liên quan đến DynamoDB**

AWS Glue Views cho phép các lập trình viên tạo nên các giao diện trực quan cho phép dễ dàng tìm kiếm, kết hợp, nhân bản giữa nhiều kho dữ liệu khác nhau theo thời gian thực mà không cần phải tự viết code tùy biến, dữ liệu tổng hợp có thể được chuyển đến các dịch vụ khác như: Redshift, S3, Aura … phục vụ nhiều mục đích khác nhau.

PartiSQL là một ngôn ngữ truy vấn tương thích với SQL, có thể liên kết với nhiều cơ sở dữ liệu với nhiều định dạng dữ liệu khác nhau như dữ liệu có cấu trúc, dữ liệu bán cấu trúc, dữ liệu dạng lưới. Ngôn ngữ truy vấn này được áp dụng trong nhiều dịch vụ của AWS, trong đó có Dynamodb.

Amazon Kinesis được sử dụng để tổng hợp và xử lý các luồng dữ liệu lớn theo thời gian thực. Các dữ liệu này được lưu trong các data record và có thể được sử dụng cho nhiều mục đích khác nhau: tạo thông báo, tạo các bảng báo cáo, các bảng giá có giá trị thay đổi theo thời gian,…

Công cụ tính giá tiền AWS dùng dể tính tiền các dịch vụ hoặc các tài nguyên đã sử dụng trên AWS.

Xuất dữ liệu từ DynamoDB sang Dịch vụ lưu trữ đơn giản của Amazon (Amazon S3) và sử dụng các dịch vụ AWS khác như Amazon Athena để phân tích dữ liệu của bạn cũng như trích xuất thông tin chuyên sâu có ý nghĩa.

# **Chương 2: Xây dựng ứng dụng**

1. **Giới thiệu website**

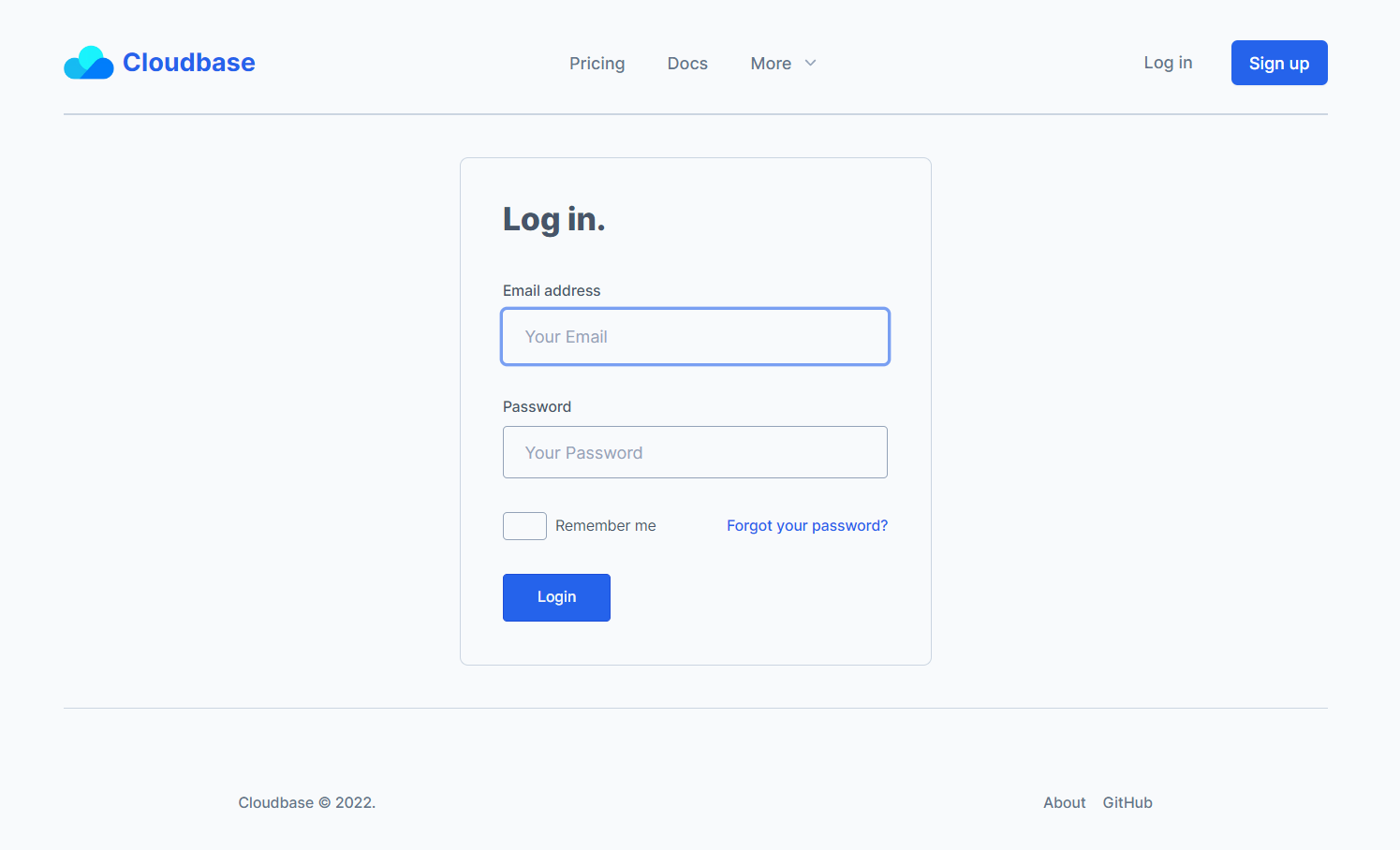
Thiết kế trang web chạy trên dịch vụ S3, cụ thể là chức năng Static Website Hosting có trong mỗi Bucket.

Server-less website, hoạt động theo nguyên lý REST APIs và dịch vụ Lambda đóng vai trò serverless backend có nhiệm vụ xử lý các request được gửi đến. Việc dựng ứng dụng serverless cho phép bạn tập trung vào phần mã ứng dụng thay vì phải lo lắng cho việc quản lý và vận hành cơ sở hạ tầng. Bạn sẽ không phải để tâm suy tính về việc cung cấp hay cấu hình máy chủ do AWS đã xử lý toàn bộ phần việc này cho bạn.

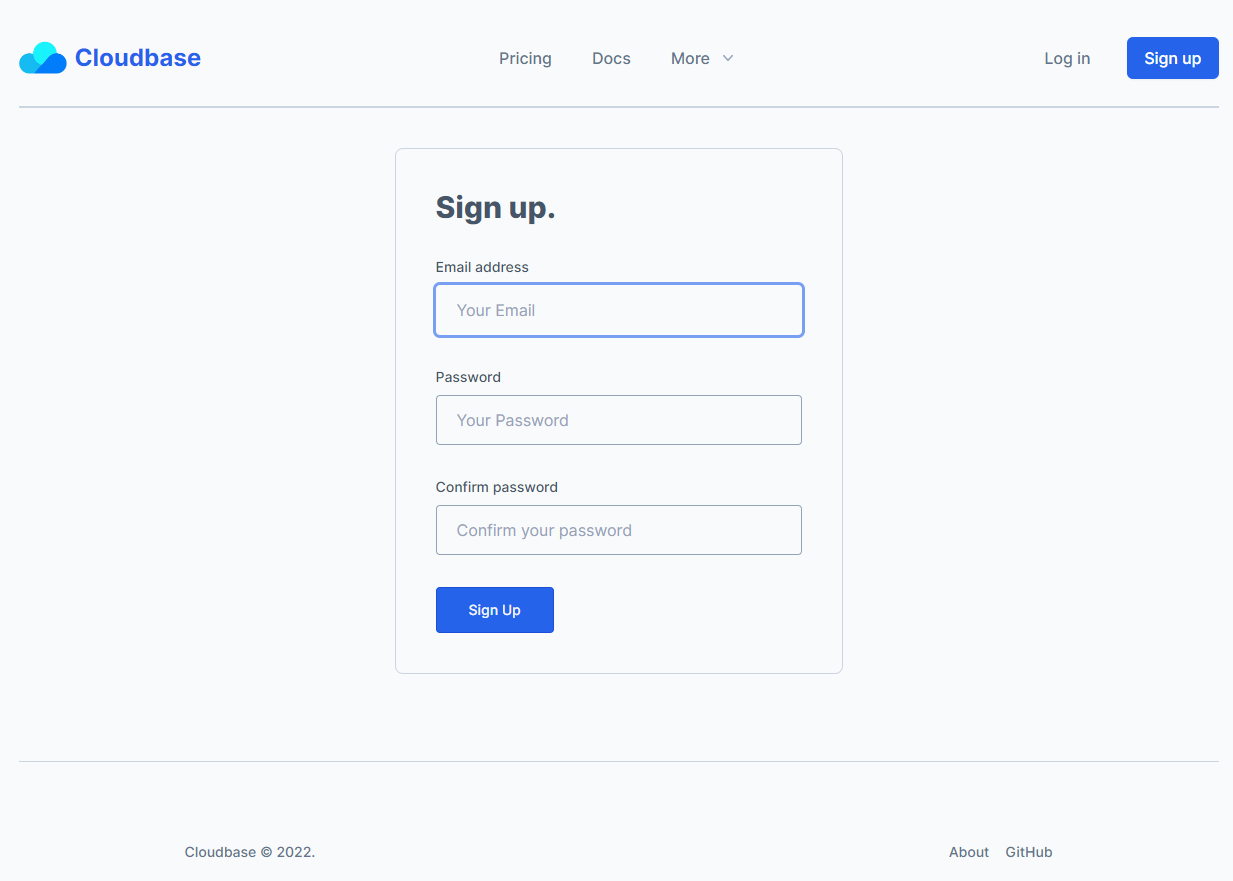
Cung cấp dịch vụ thao tác cơ sở dữ liệu trên DynamoDB, bao gồm các chức năng: đăng ký, đăng nhập, đọc/thêm/xóa/sửa các table, đọc/thêm/xóa/sửa các item trong mỗi table. Ngoài ra, còn cung cấp RESTful APIs cho từng chức năng trên để bên thứ ba sử dụng.

1. **Giao diện**

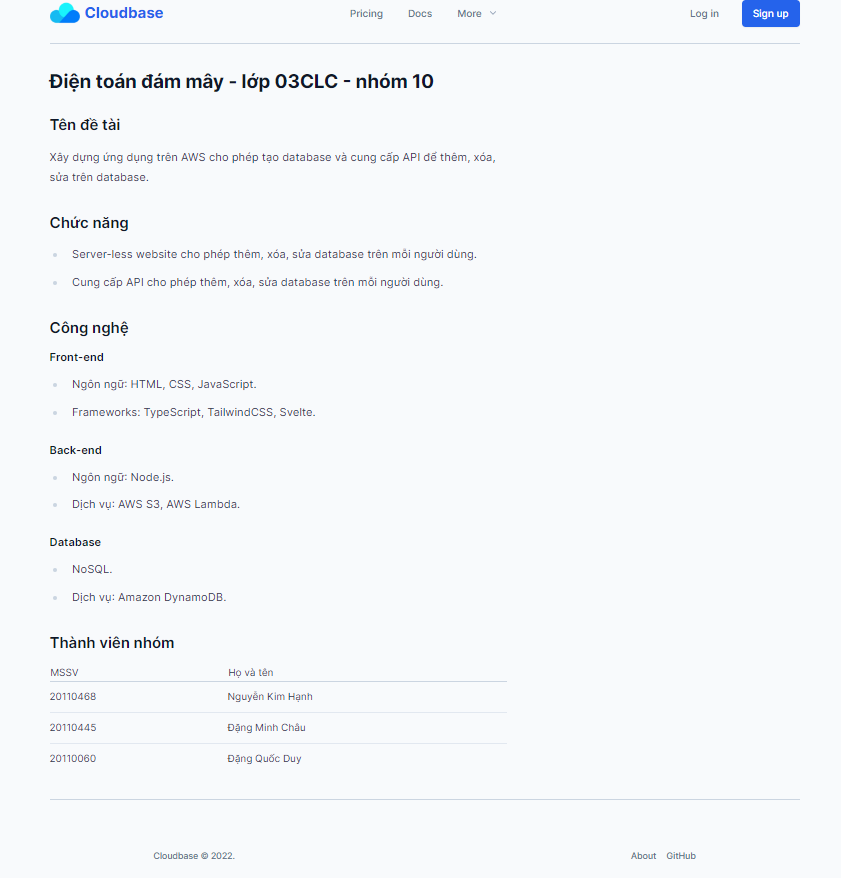
Trang đăng nhập



Trang đăng ký



Trang about



Trang Docs

Graphical user interface, application

Description automatically generated

Trang table

Graphical user interface, text, application, email

Description automatically generated

Trang item

Graphical user interface, table

Description automatically generated

1. **Cài đặt**
   1. **Cài đặt AWS S3**

Dịch vụ AWS S3 trong đề tài được sử dụng để lưu trữ website tĩnh. Các bước cài đặt AWS S3 diễn ra như sau:

B1: Tạo một AWS S3 Bucket với chế độ truy cập công khai.

Text, application

Description automatically generated

B2: Chỉnh sửa Bucket policy để cho phép truy cập tất cả tài nguyên được lưu trữ trong bucket.

Graphical user interface, text, application

Description automatically generated

B3: Thiết lập Static Website Hosting với cấu hình dưới đây.

Graphical user interface, text, application, email

Description automatically generated

B4: Các bước thiết lập hoàn tất, tiến hành tải lên các tập tin để thử truy cập tại địa chỉ được in dưới đây.

Graphical user interface, text, application, email

Description automatically generated

* 1. **Cài đặt Lambda**

Số lượng hàm Lambda cần tạo tương ứng với mỗi thư mục con của src/functions trong cấu trúc thư mục của project Lambda tại <https://github.com/hcmute-clco03clc-group10/lambda>.

Các bước tạo một hàm trên Lambda diễn ra như sau:

B1: Ấn nút “Create Function” và điền thông tin tương tự như hình dưới.

Graphical user interface, text, application

Description automatically generated

B2: Sử dụng các scripts đã được viết trong package.json để phiên dịch index.ts (TypeScript) của hàm tương ứng trong source code Lambda để tạo ra index.js (JavaScript). Sau đó, tải index.js lên hàm Lambda vừa tạo.

Graphical user interface, text, application, email

Description automatically generated

B3: Với mỗi hàm Lambda vừa tạo, cần thiết lập biến môi trường để phục vụ việc xác thực. Trong mục “Configuration”, thêm hai biến môi trường với giá trị là chuỗi base64 ngẫu nhiên bất kỳ.

Graphical user interface, text, application, email

Description automatically generated

* 1. **Cài đặt Amazon API Gateway**

Các bước cài đặt Amazon API Gateway diễn ra như sau:

B1: Tạo một REST API với các thông số sau.

Graphical user interface, text, application, email

Description automatically generated

B2: Thêm ANY method vào resource gốc của API với các tùy chọn dưới, để tích hợp Lambda Proxy vào toàn bộ resources con mà ta sẽ tạo sắp tới.

Graphical user interface, text, application, email

Description automatically generated

B3: Tạo các resource và method tương ứng với các hàm ở phía Lambda.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

B4: Sau khi hoàn tất thiết lập cho toàn bộ resources, tiến hành deploy API.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

B5: Sử dụng địa chỉ URL được cung cấp để sử dụng REST API.

Text

Description automatically generated with medium confidence

* 1. **Cài đặt DynamoDB**

Ở dịch vụ DynamoDB, ta cần tạo sẵn các tables mà mã nguồn phía Lambda sử dụng (cần đọc source code lambda). Ví dụ dưới đây là các thông số cho một table trong số đó, table “users”. Các tùy chọn còn lại được giữ nguyên.

Graphical user interface, text, application, email

Description automatically generated

* 1. **Cài đặt CloudFront**

Dịch vụ CloudFront giúp ta phân phối static website trên S3 nhanh chóng hơn. Thực hiện theo các bước sau để cài đặt dịch vụ CloudFront.

B1: Chọn “Create distribution” và nhập các thông số theo hình dưới.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

B2: Ấn nút “Create distribution” để tiến hành tạo distribution.

B3: Chờ đợi distribution chuyển trạng thái sẵn sàng, sau đó truy cập “Distribution domain name” để kiểm tra.

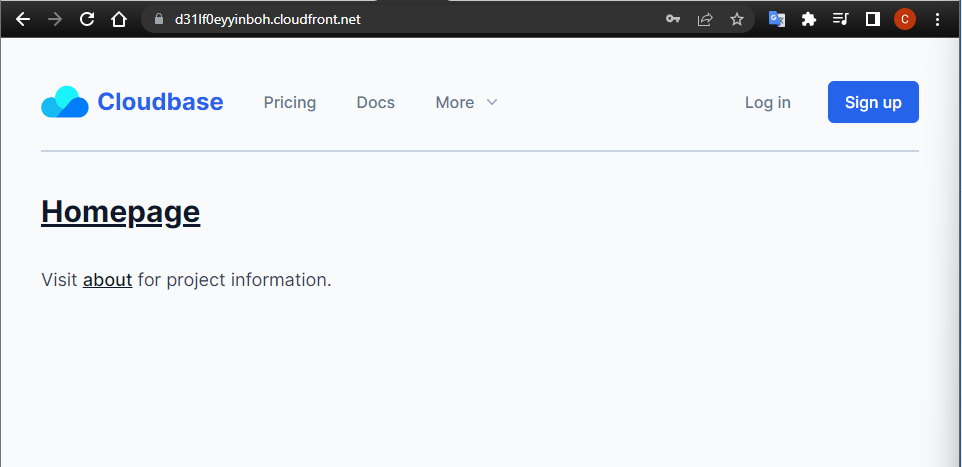
Graphical user interface, text, application, email

Description automatically generated

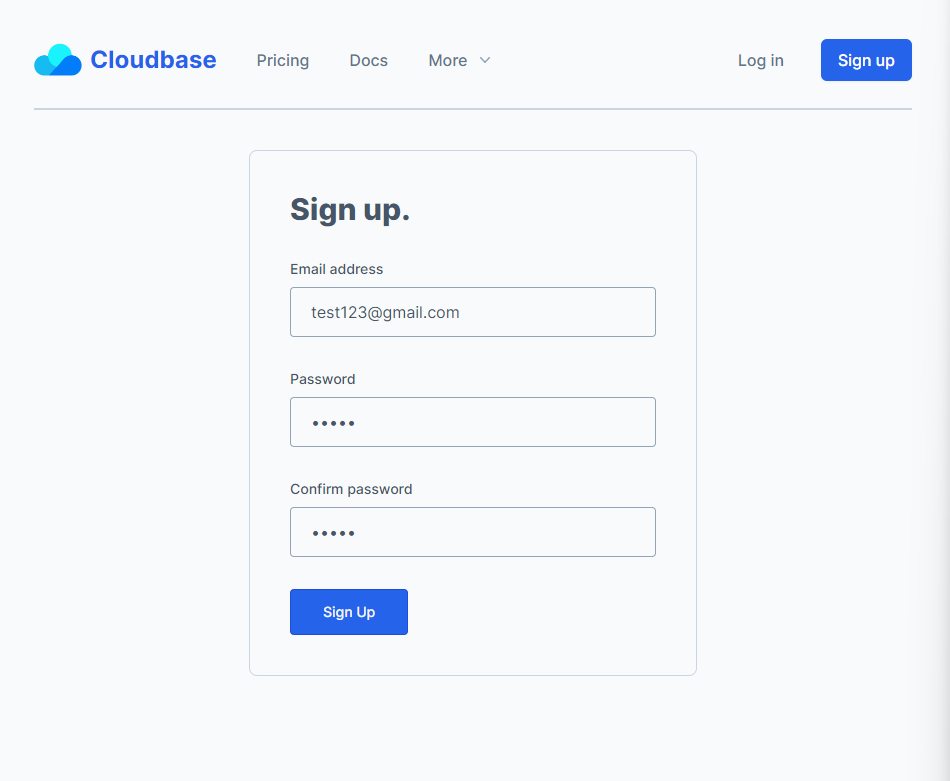
1. **Các chức năng thực hiện được**
   1. **Xác thực tài khoản**

Để sử dụng các chức năng của trang web, cần phải xác thực tài khoản. Các bước xác thực như sau:

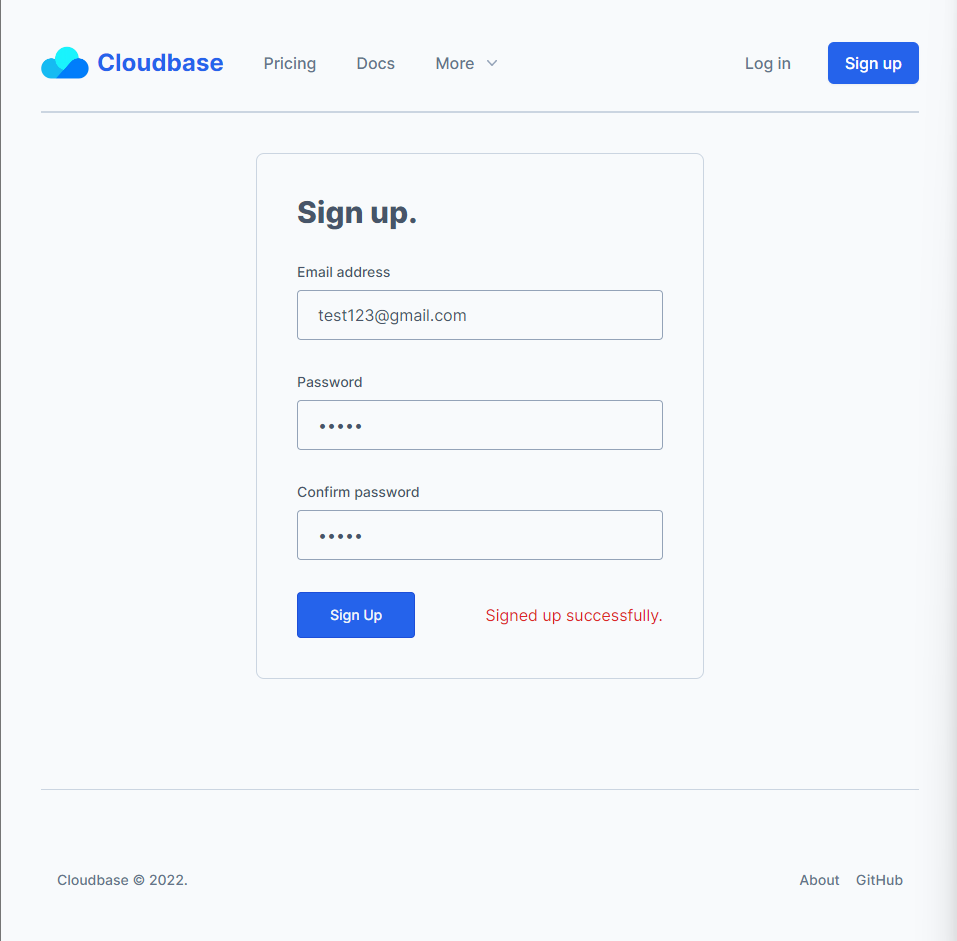
1. Nếu chưa có tài khoản, ở giao diện trang chọn “Sign up”:



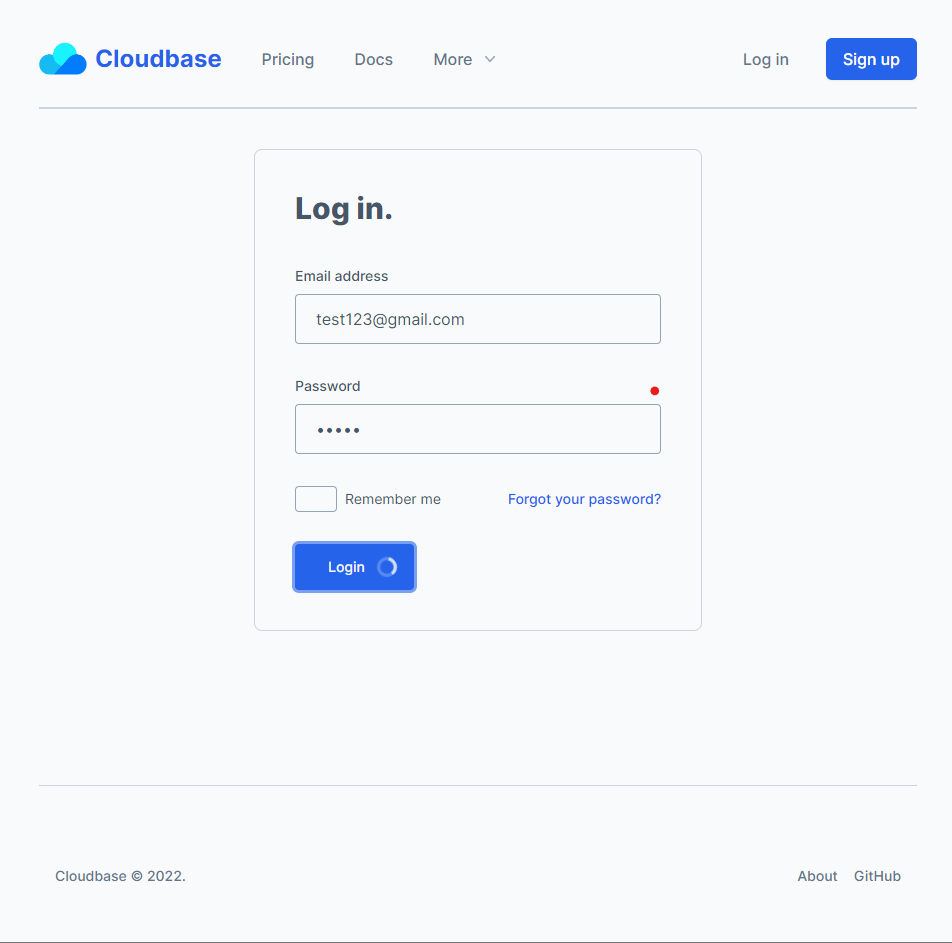
1. Tiếp đó điền thông tin “Email” và “Password” và chọn Sign Up:



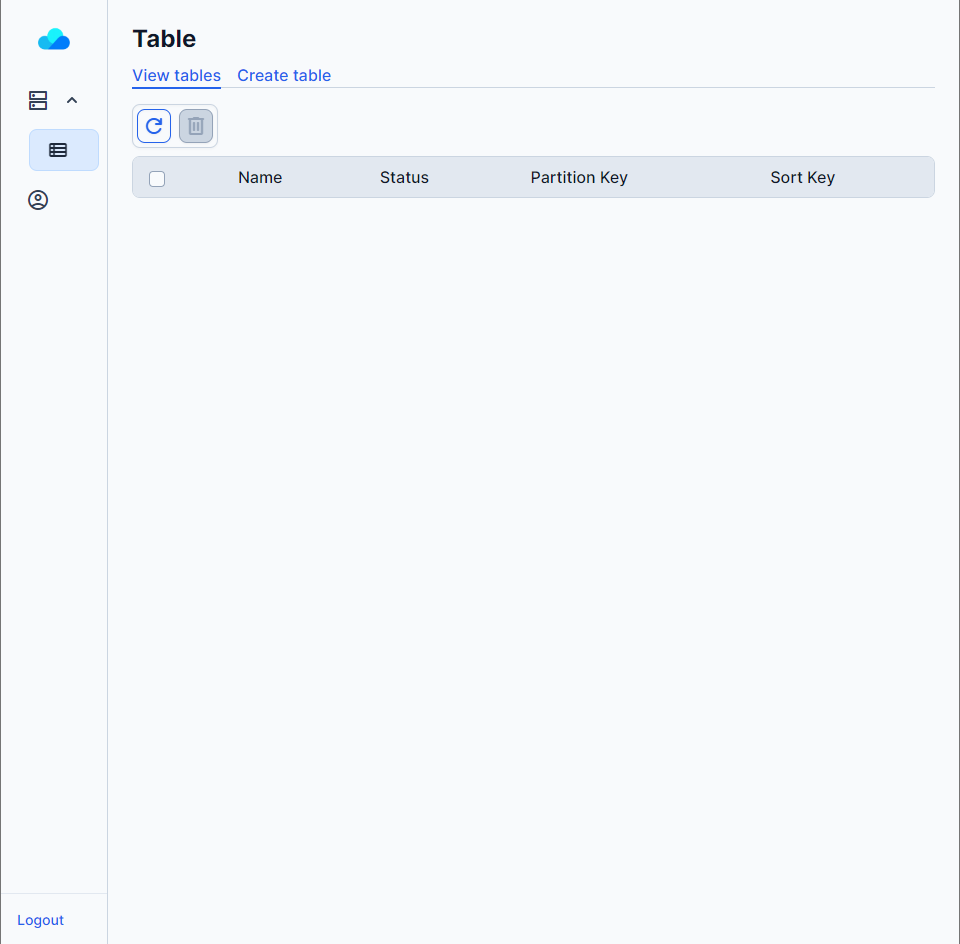
1. Sau khi có thông báo “Signed up successfully” có nghĩa là đã đăng ký thành công:



1. Vào trang “Login” và điền thông tin “Email”, “Password” tài khoản vừa tạo và chọn “Login”:



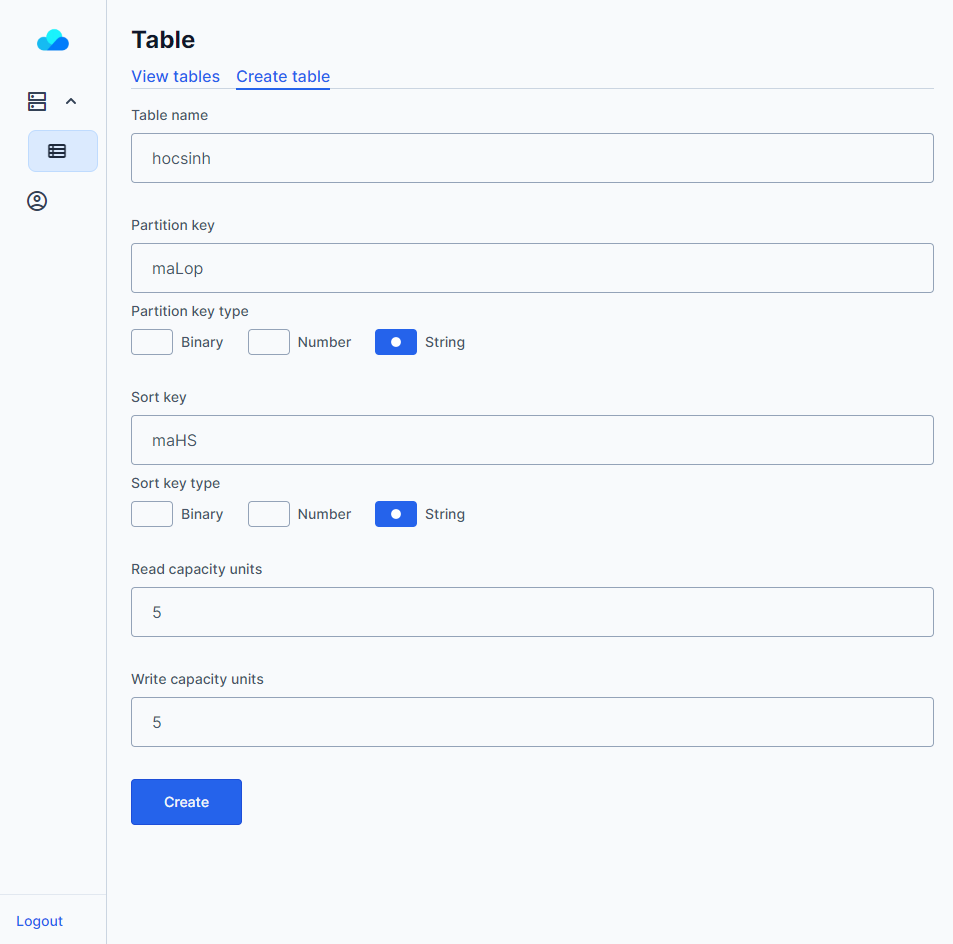
1. Sau khi đăng nhập thành công, web có giao diện như sau:



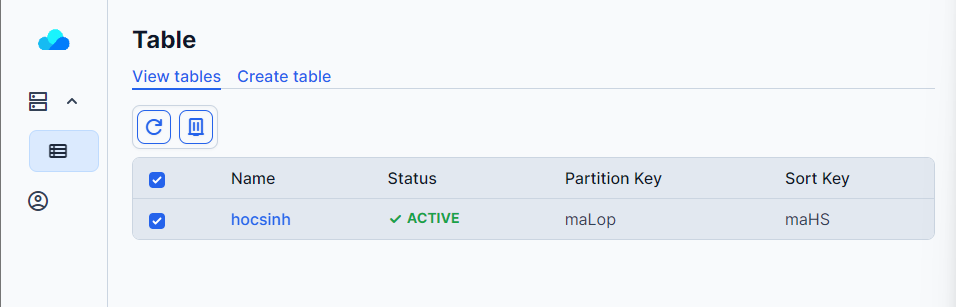
* 1. **Quản lý table**

Quản lý table có 2 chức năng cơ bản là tạo table và xoá table. Chi tiết như sau:

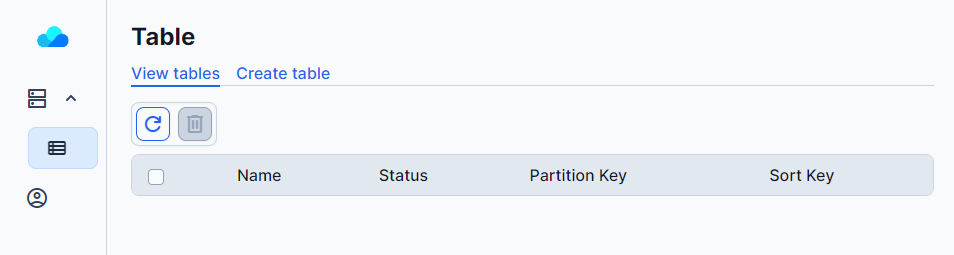
1. Nếu chưa có table, ở giao diện sau khi đăng nhập chọn “Create table” và điền các thông tin sau đó chọn “Create”:



1. Ở tab “View tables”, tables vừa tạo được hiển thị tại đây. Để xoá table, chọn table và chọn biểu tượng “thùng rác” để xoá:



1. Table đã được xoá:



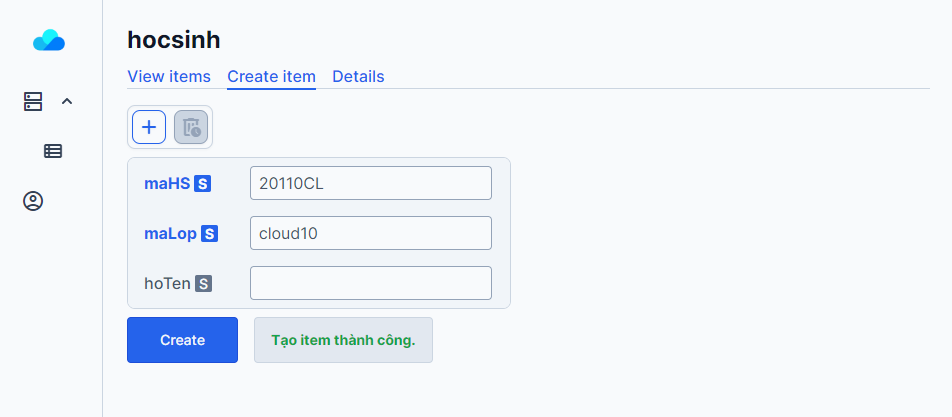
* 1. **Quản lý table items**

Quản lý items gồm các chức năng như thêm item, chỉnh sửa item và xoá item. Chi tiết như sau:

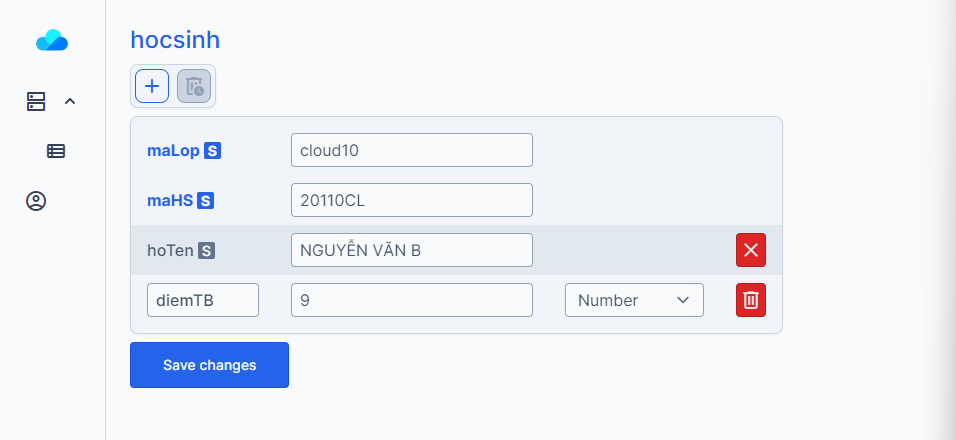
1. Để vào giao diện quản lý items, ở giao diện quản lý table chọn tên table cần quản lý item của nó:



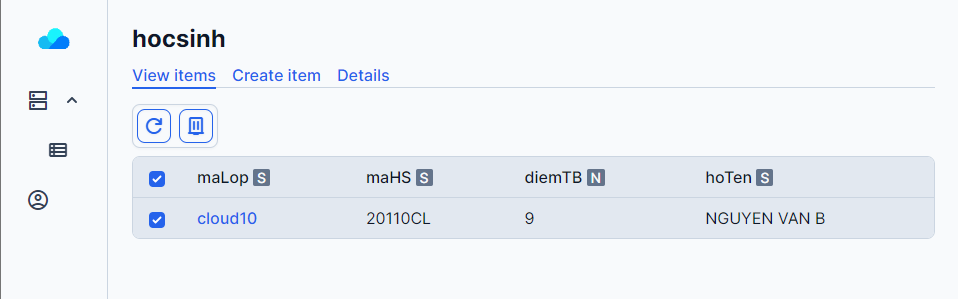
1. Chọn “Create item” để thêm item mới vào table (có thể chọn button “+” để thêm attributes mới), và chọn “Create”:



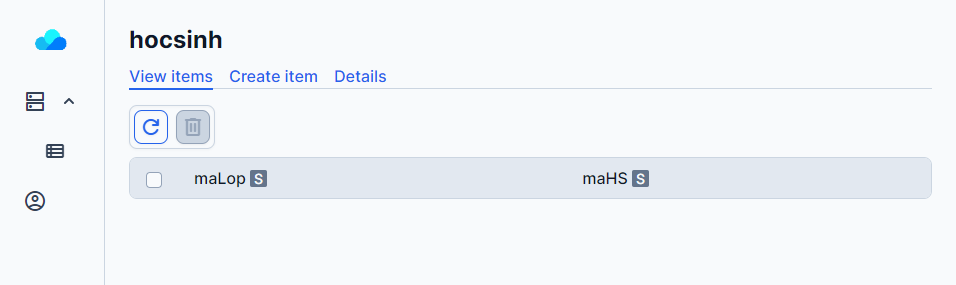
1. Ở giao diện “View items”, item mới được hiển thị, nếu muốn sửa item thì nhấp vào tên Patitional key để vào giao diện chỉnh sửa. Sau khi chỉnh sửa chọn “Save changes” để lưu những thay đổi:



1. Ở giao diện “View items”, item mới đã được cập nhật. Tuy nhiên muốn xoá item đó, chọn item đó và chọn biểu tượng “thùng rác” để xoá:



1. Item đã được xoá khỏi table:



# **KẾT LUẬN**

1. **Kết quả đạt được:**

Sau một thời gian nghiên cứu và thực hiện đề tài “***Xây dụng ứng dụng trên AWS cho phép tạo database và cung cấp API để thêm, sửa, xóa trên Database***”, nhóm chúng em đã đạt được những kết quả như sau:

* Nắm vững các kiến thức quan trọng cũng như những cách hoạt động của các dịch vụ Lamda Function, API, DynamoDB, S3, CloudFont và áp dụng kiến thức để thiết kế và xây dựng một mô hình tương tác với database.
* Nắm bắt được mô hình hoạt động, cách các luồng Data được di chuyển, thiết kế và xây dựng ứng dụng để tương tác, thực hiện tích hợp API vào ứng dụng để tạo tương tác với Database.
* Xây dựng hoàn chỉnh một ứng dụng tương tác với database (DynamoDB) bằng các dịch vụ của AWS: Lamda Function (Python), API, DynamoDB với các chức năng cơ bản như:

+ Thêm, xóa, đọc các table trong DynamoDB.

+ Thêm, sửa, xóa, đọc các Item trong Table của DynamoDB.

1. **Ưu điểm:**

* Ứng dụng dễ tương tác, thân thiện với người sử dụng.
* Luồng xử lý dữ liệu rõ ràng vì được xử lý thông qua các API.
* Lưu trữ cũng như xử lý được lượng dữ liệu lớn nhanh chóng do cơ sở dữ liệu được quản lý bởi AWS.

1. **Nhược điểm**

* Phần giao diện còn đơn giản.
* Cần phải cài đặt nhiều dịch vụ trên AWS.
* Ngoài các kiểu dữ liệu String, Number, Boolean, các kiểu dữ liệu còn lại như String set, Number set, Map, List,… chưa được hỗ trợ.

1. **Hướng phát triển:**

* Tìm hiểu thêm kiến thức về các dịch vụ AWS, để sử dụng dịch vụ tiện lợi và áp dụng chúng nhiều trong hành trang nghề nghiệp trong tương lai.
* Nâng cao tính thẩm mỹ cho giao diện.