 **BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT TP.HCM  
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**ĐỀ TÀI 11:**

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

**Lớp: CLCO332779\_22\_1\_05CLC**

**Môn: Điện toán đám mây  
Nhóm: 46  
GVHD: Huỳnh Xuân Phụng**

***Thành phố Hồ Chí Minh, Tháng 12 năm 2022***

**DANH SÁCH NHÓM**

HỌC KÌ I, NĂM HỌC: 2022-2023

**Đề tài 11:** xây dựng ứng dụng trên aws cho phép tạo database và cung cấp api để thêm, xóa sửa trên database.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Mức độ đóng góp |
| 1 | Võ Trần Bảo Nguyên | 20110138 | 100% |
| 2 | Trần Văn Dân | 20110451 | 100% |

**Nhận xét của giảng viên:**

*Ngày ........tháng........năm.......*

Giáo viên chấm điểm

LỜI CẢM ƠN

*“Lời đầu tiên cho phép chúng em xin được gửi lời cảm ơn đến Khoa Đào tạo Chất lượng cao và thầy Nguyễn Đăng Quang đã mở ra các môn học này cũng như tìm giáo viên phù hợp để giảng dạy và mang lại cho chúng em nhiều kiến thức bổ ích và liên quan đến các dịch vụ Điện toán đám mây (Cloud Computing)*

*Tiếp theo chúng em muốn gửi một lời cảm ơn chân thành đến thầy Huỳnh Xuân Phụng đã dẫn dắt và chỉ dẫn chúng em rất nhiệt tình trong thời gian qua. Thầy đã chỉ dạy cũng như mang lại những kiến thức bổ ích về các dịch vụ cũng như các chức năng của các dịch vụ AWS (Amazon Web Services). Ngoài ra vào thời điểm cuối kì thầy cũng đã cho nhóm em một cơ hội tiếp cận và xây dựng một dự án thực tế dựa trên hai dịch vụ đó là AWS DynamoDB và AWS Lambda để tạo ra một trang web cho phép tạo Database cũng như cung cấp API để Thêm, Xóa, Sửa trên Database.*

*Trong thời gian làm dự án trên chúng em đã nhận được sự giúp đỡ cũng như chỉ dẫn khá nhiệt tình từ thầy và thông qua đó chúng em có thể phát triển và hoàn thành tốt dự án cuối kì của môn này. Chúng em thực sự chân thành cảm ơn thầy vì những sự hỗ trợ, giúp đỡ nhiệt tình trên.”*

*Chúng em xin chân thành cảm ơn*

**Nhóm thực hiện**

Võ Trần Bảo Nguyên 20110138

Trần Văn Dân 20110451

Mục lục

[LỜI CẢM ƠN 3](#_Toc121579650)

[1. Phần mở đầu: 2](#_Toc121579651)

[1.1. Tính cấp thiết của đề tài: 2](#_Toc121579652)

[1.2. Đối tượng nghiên cứu: 2](#_Toc121579653)

[1.3. Phạm vi nghiên cứu: 2](#_Toc121579654)

[1.4. Kết quả dự kiến đạt được: 2](#_Toc121579655)

[2. Đặc tả: 3](#_Toc121579656)

[2.1. Mục tiêu đề tài: 3](#_Toc121579657)

[2.2. Cơ sở lý thuyết: 3](#_Toc121579658)

[2.2.1. Tổng quan về AWS Lambda: 3](#_Toc121579659)

[2.2.2. Tổng quan về AWS DynamoDB: 3](#_Toc121579660)

[2.2.3. Tổng quan về Amazon S3 4](#_Toc121579661)

[3. Thực hiện đồ án: 4](#_Toc121579662)

[3.1. Phần Database: 4](#_Toc121579663)

[3.2. Phần Back-End: 5](#_Toc121579664)

[3.3. Phần Front-End: 16](#_Toc121579665)

[3.4. Hướng dẫn cài đặt và chạy dự án trên Local: 18](#_Toc121579666)

[3.5. Hướng dẫn cài đặt và chạy dự án lên trên AWS: 19](#_Toc121579667)

[4. Đánh giá và nhận xét: 19](#_Toc121579668)

[4.1. Phần Back-end và Database: 19](#_Toc121579669)

[4.2. Phần Front-End: 19](#_Toc121579670)

[5. Tài liệu tham khảo: 20](#_Toc121579671)

# Phần mở đầu:

## Tính cấp thiết của đề tài:

Hiện nay nhu cầu sử dụng cơ sở dữ liệu đang dần tăng lên và trở thành một phần không thể thiếu trong công việc cũng như trong việc học tập của các sinh viên. Đặc biệt, với các ưu điểm như đơn giản dễ sử dụng, không cần phải thiết lập quan hệ khóa giữa các bảng mỗi khi cài đặt,… của loại cơ sở dữ liệu không quan hệ (NoSQL) thì đây là một lựa chọn khá được nhiều học sinh sinh viên ưu chuộng. Dựa trên những điều kiện trên mà hiện nay các loại cơ sở dữ liệu NoSQL luôn được lập trình và có thể sử dụng hoàn toàn trên Mạng.

## Đối tượng nghiên cứu:

Trong đề tài này, đối tượng được nghiên cứu cụ thể là DynamoDB và Lambda, hai công cụ này sẽ đóng vai trò là phần Database và phần Back-End của web sử dụng API do Lambda tạo ra để thực hiện quản lý Database trên DynamoDB

## Phạm vi nghiên cứu:

Đề tài này chỉ yếu tập trung vào phần sử dụng Lambda để viết các hàm (Function) xử lý trên Database của DynamoDB, đồng thời tạo ra các API để phục vụ và sử dụng cho việc, thêm xóa sửa dữ liệu trực tiếp trên web hoặc có thể cho người dùng sử dụng

## Kết quả dự kiến đạt được:

* Cài đặt được gia diện Front-end cho web
* Tạo được các Lambda Function để xử lý các nghiệp vụ trên DynamoDB
* Phân quyền cho người dùng với các nghiệp vụ liên quan tới Database
* Cung cấp được API cho người dùng có thể trực tiếp thêm xóa sửa các dữ liệu của họ mà không cần phải lên trực tiếp web

# Đặc tả:

## Mục tiêu đề tài:

Sử dụng được các dịch vụ AWS Lambda và dịch vụ AWS DynamoDB để tạo ra được phần Database và phần Back-End cho web với mục tiêu là sử dụng các API để thực hiện các nghiệp vụ tạo, thêm, xóa, sửa cho các Table và các Record của table trong Database. Ngoài ra, còn sử dụng được công nghệ ReactJS để tạo nên phần giao diện Front-End đẹp, cho người dùng có thể dễ dàng thao tác, thực hiện các tác vụ cần thiết ngay trên Web

## Cơ sở lý thuyết:

### Tổng quan về AWS Lambda:

AWS Lambda là một dịch vụ được cung cấp và quản lý bởi đội ngũ AWS, dịch vụ này cung cấp cho người dùng khả năng tạo các Function, các Function này sẽ được chạy trên các ngôn ngữ mặc định mà AWS đã đưa ra, người dùng có thể chọn một ngôn ngữ và thực hiện viết và đưa vào các doạn code của chính mình, các đoạn code này có thể ảnh hưởng trực tiếp hoặc gián tiếp đến các dịch vụ của AWS tùy theo mục đích và sẽ được thực hiện khi Function được gọi.

Ngoài ra AWS Lambda còn 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 Lamda có thể dễ dàng sử dụng để xây dựng các ứng dụng cùng với các dịch vụ khác của Lamda như S3, DynamoDB,... với các hiệu suất và độ bảo mật cao

### Tổng quan về AWS DynamoDB:

AWS DynamoDB là một dạng cơ sở dữ liệu phi quan hệ (NoSQL) được lưu trữ ngay trên môi trường của AWS. Dịch vụ này cung cấp cho người dùng một Database duy nhất và người dùng có thể thông quan Database đó tạo ra những bảng (Table) để lưu trữ các dữ liệu của bản thân. Các bảng khi tạo sẽ chỉ yêu cầu 1 khóa chính và một khóa phụ (Tùy chọn) để tạo ra các bảng, và khi thêm các dữ liệu (Record) thì người dùng có thể sử dụng các khóa đã tạo để tạo ở trên để tạo dữ liệu và có thể thêm các thuộc tính mới tùy vào nhu cầu sử dụng

Ngoài việc cung cấp một Database để lưu trữ dữ liệu, AWS DynamoDB còn cung cấp cho người dùng hai công cụ để truy vấn dữ liệu là Query và Scan. Công cụ Scan sẽ giúp cho người dùng quét được toàn bộ các dữ liệu đã tạo trong một bảng và công cụ query sẽ giúp cho người dùng lấy ra được các dữ liệu theo một số yêu cầu nhất định

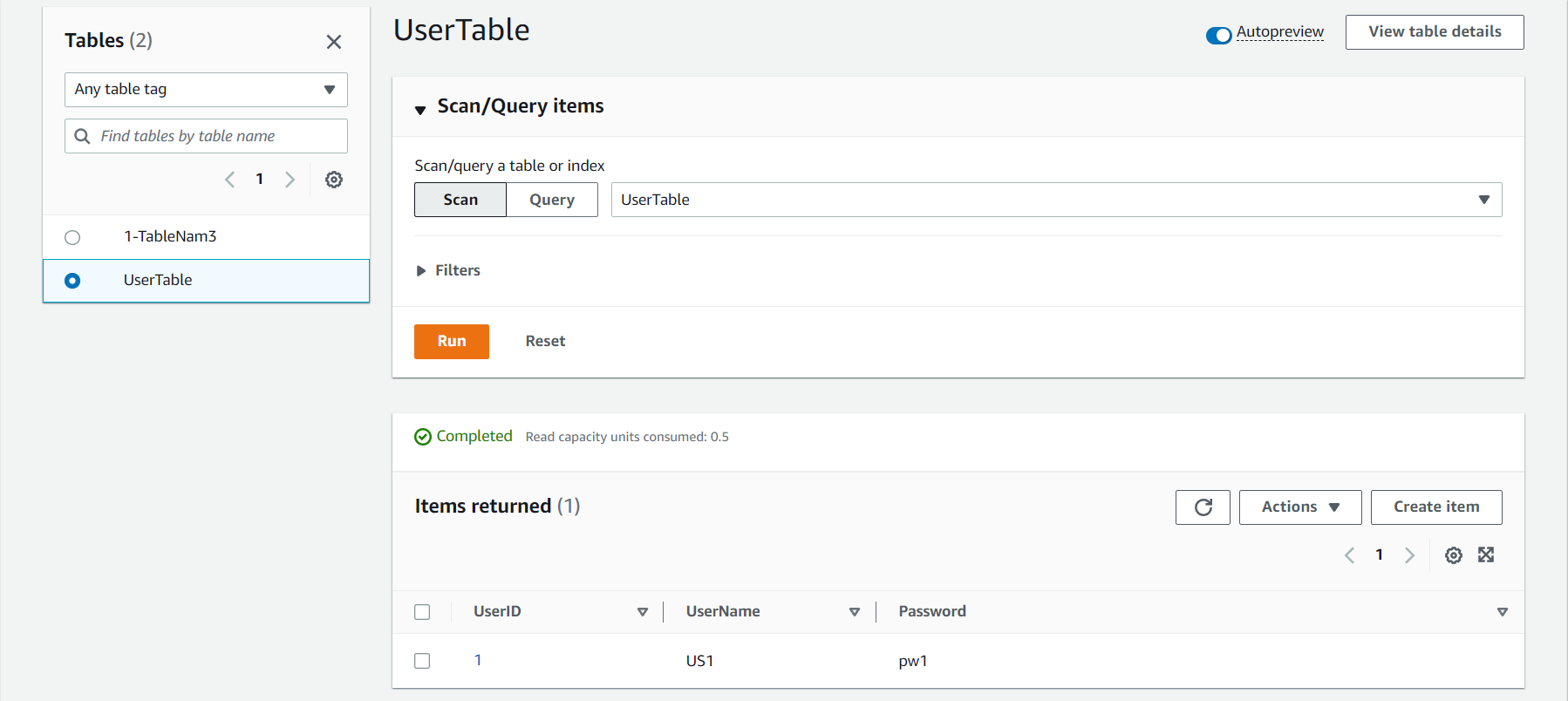
### Tổng quan về Amazon S3

Amazon S3 là dịch vụ lưu trữ đối tượng được xây dựng để lưu trữ và truy xuất bất kỳ lượng dữ liệu nào từ bất cứ nơi nào. Đây là dịch vụ lưu trữ đơn giản có độ bền, độ sẵn có, hiệu suất, tính bảo mật dẫn đầu ngành và khả năng thay đổi quy mô gần như không giới hạn với chi phí cực kỳ thấp.

# Thực hiện đồ án:

## Phần Database:

Database sẽ được tạo sẵn một bảng có tên là UserTable với khóa chính là UserID với kiểu dữ liệu là Number, khóa phụ là UserName với kiểu dữ liệu là String. Sau đó ta tạo một Record với User ID là **1**, UserName là **US1** và thêm một thuộc tính với kiểu là String tên là Password với dữ liệu tạo ra là **pw1.** Database này sẽ hỗ trợ cho việc test phân quyền và sử dụng trang web sau này.



## Phần Back-End:

Phần Back-End sử dụng dịch vụ Lambda của AWS, cụ thể là dùng dịch vụ Lambda URL viết bằng ngôn ngữ Javascrip.

Các API cung cấp các công cụ hỗ trợ thêm sửa xóa dữ liệu trên DynamoDB của AWS.

Source code của các chức năng: (chú ý: một số bảng ban đầu tên ‘UserManger’ sau đó đã đổi tên thành ‘UserTable’)

Chức năng tạo table: CreateTable

Text

Description automatically generated

Chức năng lấy ra các bảng từ bằng id của user: GetTableByUserID

Text

Description automatically generated

Chức năng xóa bảng: DeleteTable

Graphical user interface, text, application

Description automatically generated

Chức năng thêm hoặc sửa Item: AddOrUpdateItem

Text

Description automatically generated

Chức năng lấy toàn bộ Item trong Table: GetAllItemByTableName

Graphical user interface, text, application

Description automatically generated

Chức năng xóa Item: DeleteItem

Text

Description automatically generated with low confidence

Chức năng đăng nhập: Login

Text

Description automatically generated with low confidence

Chức năng đăng ký: Register

Text

Description automatically generated

Tham khảo các body của request đến API và cài đặt các API với link sau: <https://github.com/votranbaonguyen/DTDM_Detai11.git>

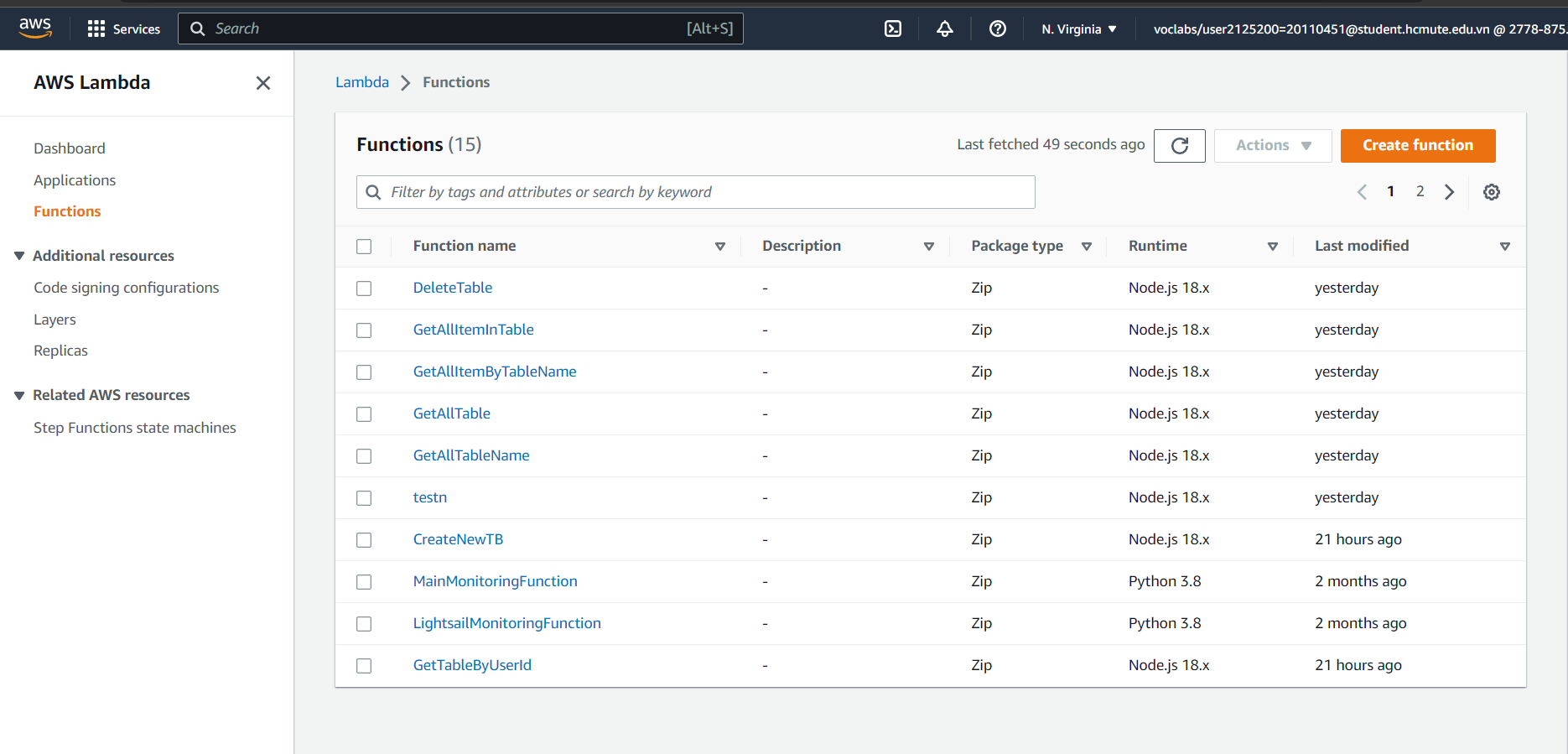
Body của các request gửi đến API:

Graphical user interface, text

Description automatically generated

Cách cài đặt các chức năng trên AWS:

Trên AWS chọn **Search**, nhập “**Lambda**” rồi chọn chức năng **Lambda** ta được như hình sau:



Sau đó, chọn **Create function**, ta bắt đầu tạo funtion:

-Chọn **Author from scratch**

-Nhập **Function name**

-**Runtime** chọn **Node.js 18.x**

-Ở **Architecture** chọn **x86\_64**

-Mở **Change default execution role**

-Chọn **Use an existing role**

-Ở **Existing role** chọn **LabRole**

-Mở **Advanced Settings**

-Chọn **Enable function URL**

-Ở **Auth type** chọn **None**

-Chọn **Configure cross-origin resource sharing (CORS)**

-Cuối cùng chọn **Create function**

Ta thu được:

Graphical user interface, application

Description automatically generated

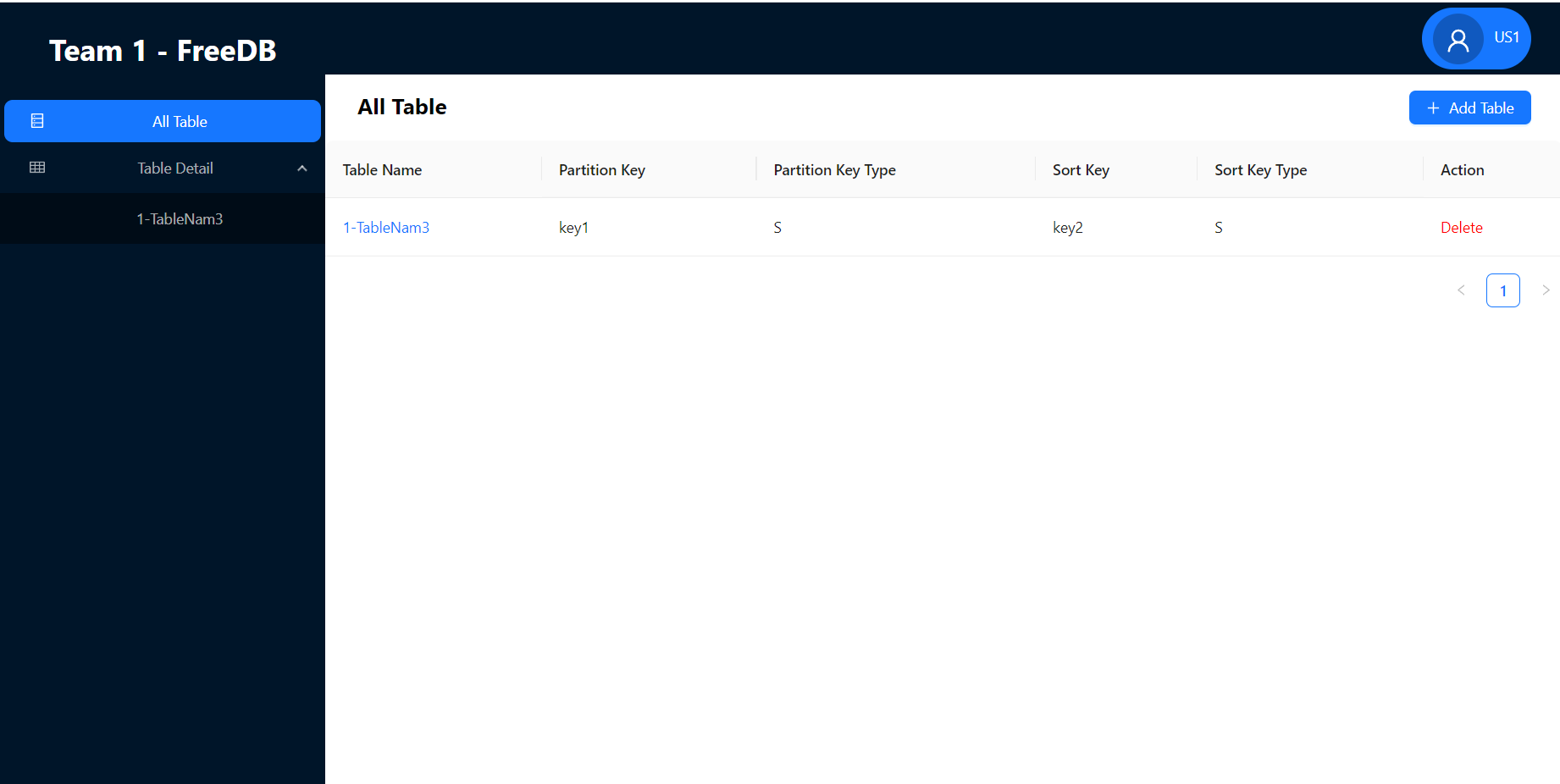
Ở phần **Code**, ta chọn file **index.mjs**, và paste các **source code** của chức năng ta muốn vào. Lưu ý, các **REGION** hoặc **DEFAULT\_REGION** trong **source code** ta thay bằng Region bạn mong muốn, hiện tại là **"us-east-1"**

Graphical user interface, application, table, Teams

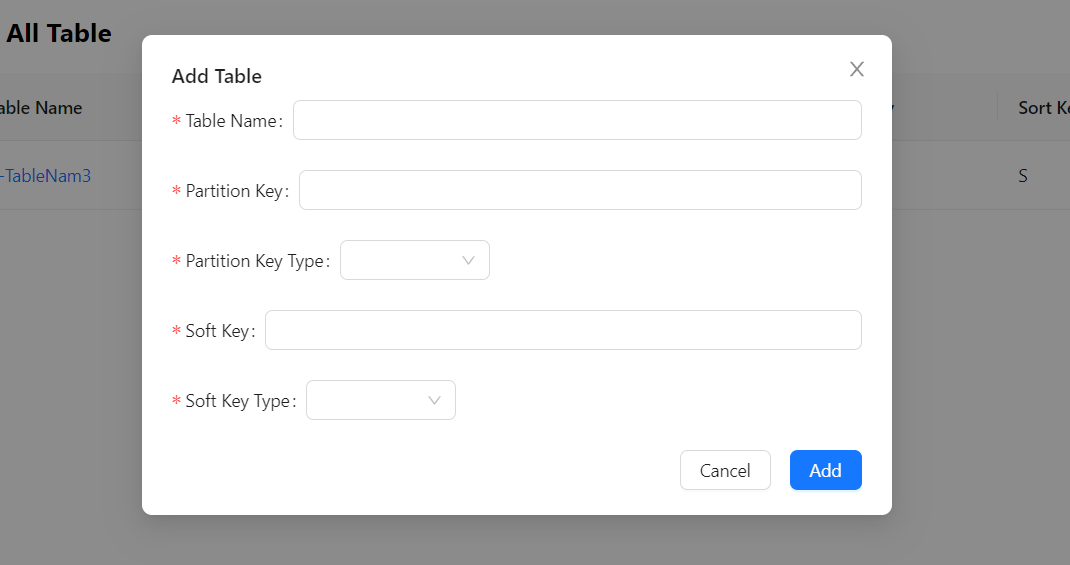
Description automatically generated

## Phần Front-End:

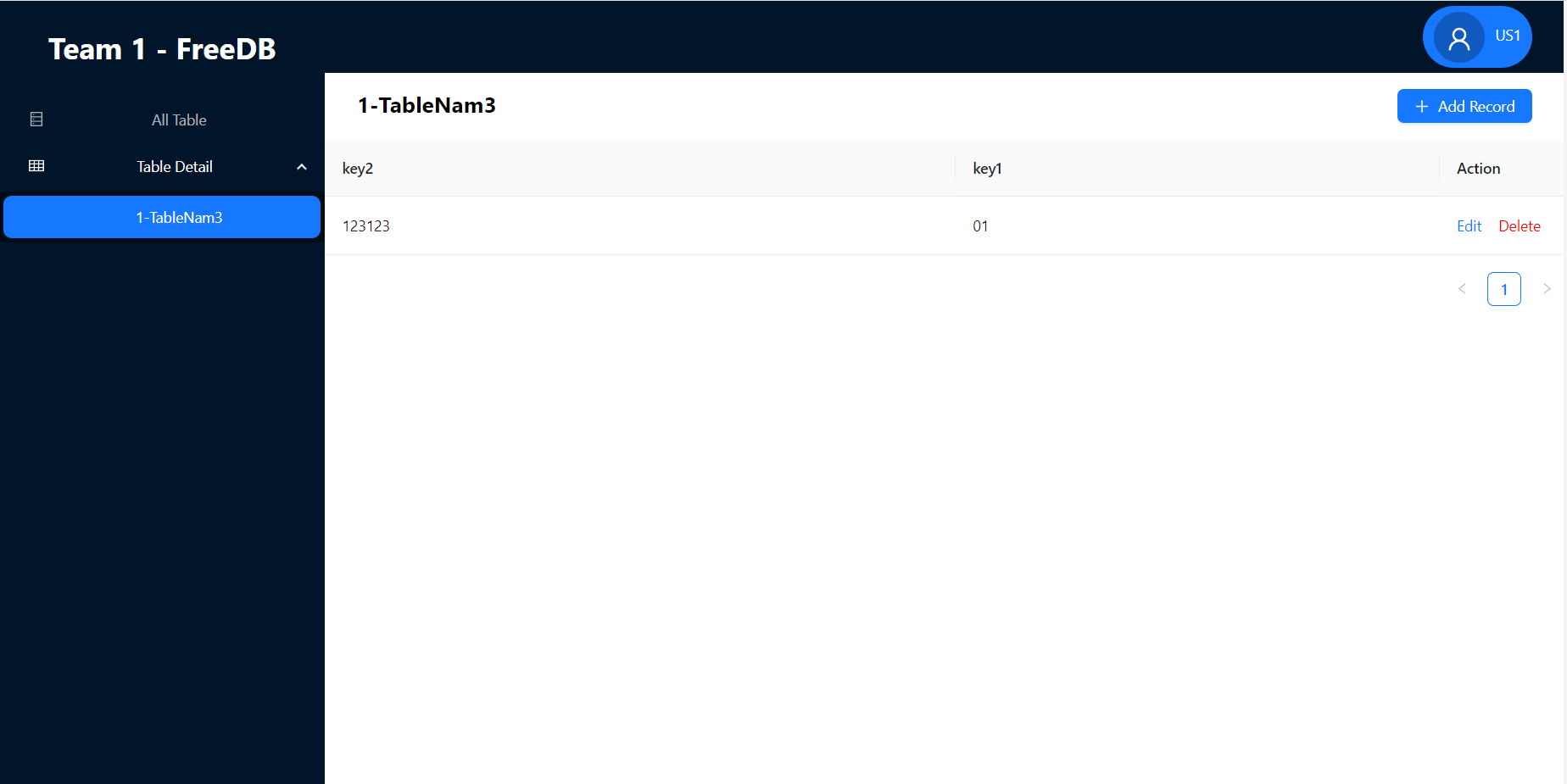
Phần Front-End sẽ được tạo và viết bằng Ngôn ngử Javascript và sử dụng thư viện chính là ReactJS để thực hiện. Ngoài ra thư viện Ant-Design cũng được sử dụng đễ hộ trợ tạo phần UI nhanh chóng và dễ dàng hơn



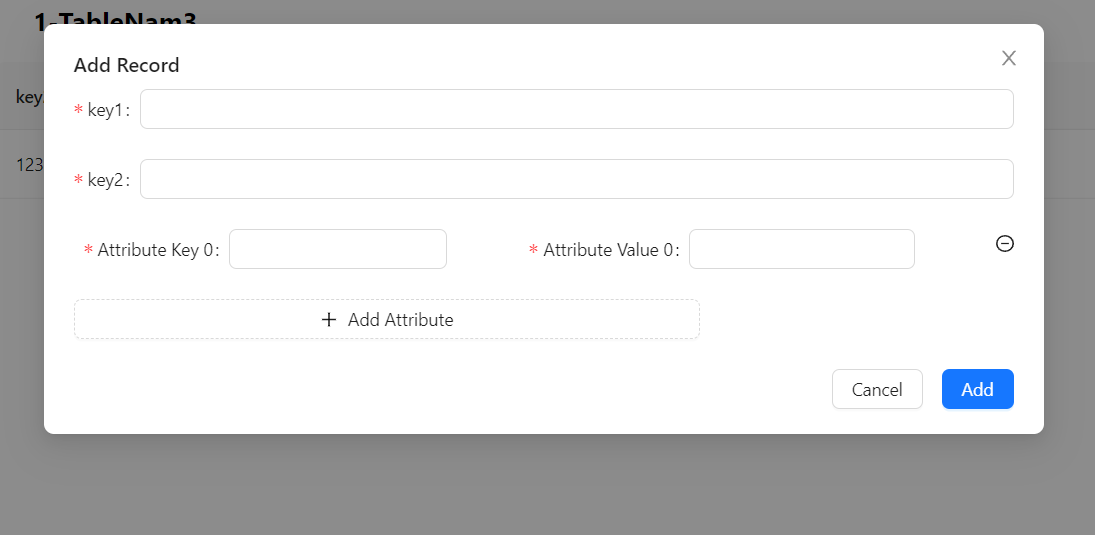
Giao diện trên chính là giao diện cho phép xem tất cả các bảng thuộc về User trên với tên 2 khóa đã tạo và kiểu dữ liệu của khóa và chức năng xóa bảng



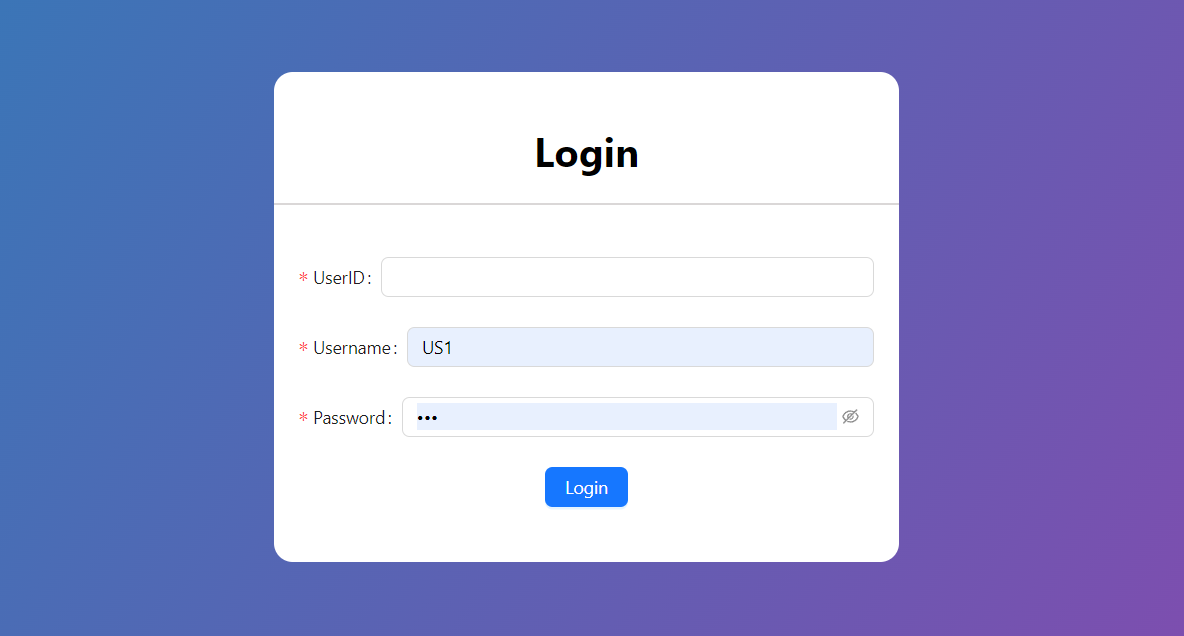
Ngoài ra khi nhấn vào nút Add table ta sẽ có một modal hiện ra cho phép tạo thêm một bảng mới



Giao diện phần chi tiết của bảng (Table), ở đây chúng ta có thể xem các Record đã tạo cũng như có thể tùy chọn sửa xóa các Record.



Ngoài ra chúng ta còn có một giao diện modal thêm Record khi nhấn vào nút Add Record, giao diện này ban đầu sẽ chỉ hiện hai khóa mà ta đã tạo sẵn trước và nếu ta muốn tạo thêm các thuộc tính mới cho Record chúng ta có thể nhấn vào nút Add Attribute, thì ta sẽ có thêm nơi để thêm các thuộc tính mong muốn.



Trên là giao diện đăng nhập, ở đây được thiết kế khá đơn giản với 3 trường cần nhập là UserID, UserName và Password

## Hướng dẫn cài đặt và chạy dự án trên Local:

* Bước 1: Sử dụng link: https://github.com/votranbaonguyen/DTDM\_Detai11.git để clone đồ án về máy
* Bước 2: Cài đặt NodeJS vào máy tính nếu chưa có, có thể tham khảo link sau: <https://nodejs.org/en/> (Lưu ý: Luôn ưu tiên tải và cài đặt bản LTS)
* Bước 3: Thực hiện tạo và thiết lập code cho các Function ở trên Lambda theo hướng dẫn và nội dung ở phần 3.2 (Toàn bộ source code của các function nằm trong thư mục *backend/lambda\_nodejs*)
* Bước 4: Vào thư mục frontend/src/services/ mở file api.js lên bằng các phần mềm chuyên dụng để viết code, ở đây ta thực hiện dán các Function Url đã tạo ở bước trên tương ứng với tên của mỗi Function vào



* Bước 5: thực hiện mở terminal tại thư mục frontend và chạy các dòng lệnh sau:

+ npm i (Dùng để cài đặt node\_module và các thư viện cần thiết)

+ npm start (Dùng để chạy web ở local)

* Bước 6: Nếu web không tự động mở lên ta có thể dùng đường dẫn <http://localhost:3000/> để mở trang web. Giao diện login sẽ được bật ra, ở đây ta dùng dữ liệu đã tạo ở phần 3.1 với UserID là **1**, UserName là **US1** và Password là **pw1** để đăng nhập. Sau khi đăng nhập thành công ta có thể sử dụng các chức năng đã nói ở phần 3.3

## Hướng dẫn cài đặt và chạy dự án lên trên AWS:

-Bước chuẩn bị: chạy được dự án trên **Local**

-Bước 1: Ở phần 3.4, sau khi chạy “npm i”, thay vì chạy “npm start”, ta chạy lệnh “npm run build” . Sau khi chạy xong, ta vào thư mục frontend/build, đó là bản build ta dùng để deploy lên AWS (dùng dịch vụ Amazon S3).

-Bước 2: Tạo S3 bucket

**+**Trên AWS chọn **Search**, nhập “**Amazon S3**” rồi chọn chức năng **S3** ta được như hình sau:

Graphical user interface, application

Description automatically generated

+Chọn **Create bucket**

+Nhập **Bucket name**

+Ở **AWS Region** chọn **US East (N.Verginia) us-east-1** hoặc **vùng của bạn.**

+**Object Ownership** giữ nguyên **ACLs disabled (recommended)**

+Ở **Block Public Access settings for this bucket**, bỏ chọn **Block all public access**, rồi chọn vào **I acknowledge that the current settings might result in this bucket and the objects within becoming public**.

+Ở **Bucket Versioning**, giữ nguyên mặc định

+Ở **Default encryption**, giữ nguyên mặc định

+Cuối cùng, chọn vào **Create bucket**

-Bước 3: Cấu hình bucket và deploy

+Ở **Amazon S3**, chọn **Buckets** rồi chọn vào bucket ta vừa tạo ở trên.

+Chọn vào thẻ **Permissions**, kéo xuống **Bucket policy**, chọn **Edit.** Trong dự án ta thấy 1 file **bucketpolicy.txt**, **copy** nội dung file đó dán vào đây, thay **example-bucket** bằng tên bucket chúng ta vừa tạo rồi chọn **Save changes.**

+Chọn vào thẻ **Properties**, kéo xuống **Static website hosting**, chọn **Edit**. Ta chọn **Enable**. Ở **Hosting type** chọn **Host a static website**. Ở **Index document** nhập **index.html**, cuối cùng chọn **Save changes.**

+Ở phần **Bucket website endpoint**, ta lưu lại url để truy cập website.

+Chọn vào thẻ **Objects** chọn **Upload**, kéo những file trong thư mục frontend/build thả vào đây xong chọn **Upload**. Đợi Upload thành công ta sử dụng url ở trên để truy cập vào website.

# Đánh giá và nhận xét:

## Phần Back-end và Database:

Hiện tại phần Back-end và Database vẫn còn một vài thiếu xót, nhưng về mặt chức năng gần như là đầy đủ tuy nhimột số chức năng trả về kết quả chưa thực sự tối ưu cho front-end.

Bỏ qua những thiếu xót thì phần Back-end và Database cơ bản đáp ứng được việc cho phép tạo database và thêm, xóa sửa trên database theo yêu cầu của đề tài.

Hi vọng trong tương lai sẽ cải thiện và tối ưu được những điểm yếu trên.

## Phần Front-End:

Hiện tại trang web vẫn còn thiếu xót về mặt giao diện, các thao tác trên trang web cũng không thực sự quá mượt mà, ngoài ra trang web vẫn còn tồn tại một số Bug vẫn chưa khắc phục được. Khi tạo bảng hoặc tạo thêm Record cũng chỉ mặc định cho kiểu dữ liệu luôn là String mà vẫn chưa có nhiều lựa chọn kiểu dữ liệu mới

Bên cạnh những điểm còn thiếu xót của trang web thì trang web vẫn đáp ứng khá đầy đủ các chức năng như đã dự định như thêm xóa các Table và thêm xóa sửa cho các Record trong bảng, ngoài ra web còn tận dụng và sử dụng tốt được các API (Function Url) của Lambda để thực hiện các tác vụ trên

Hi vọng trong tương lai Phần giao diện sẽ được cải thiện hơn ở những điểm yếu nêu trên và phát huy được hết các chức năng cần thiết cho một trang web cung cấp khả năng tạo Database và cung cấp API cho việc thêm xóa sửa.

# Tài liệu tham khảo:

* Project tham khảo của nhóm thực hiện đề tài trước: <https://github.com/trungnguyenhuynhminh46/ProjectDetai11Nhom1v2>
* Tài liệu tham khảo về các định nghĩa của những dịch vụ được sử dụng: <https://aws.amazon.com/>
* aws-doc-sdk-examples (Hỗ trợ viết các Lambda Function xử lý trên DynamoDB bằng ngôn ngữ JS): <https://github.com/awsdocs/aws-doc-sdk-examples/tree/main/javascriptv3/example_code/dynamodb/src>