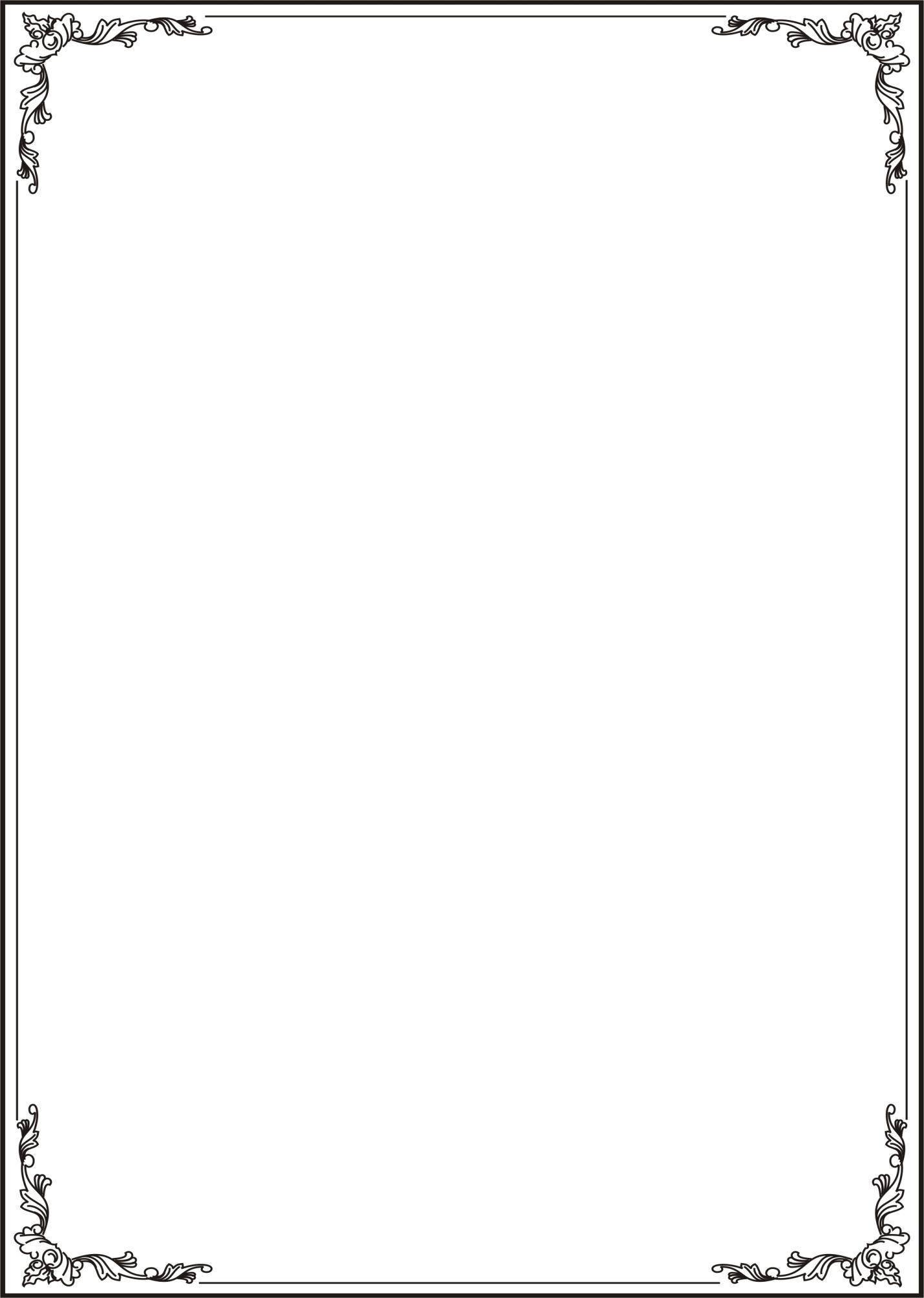
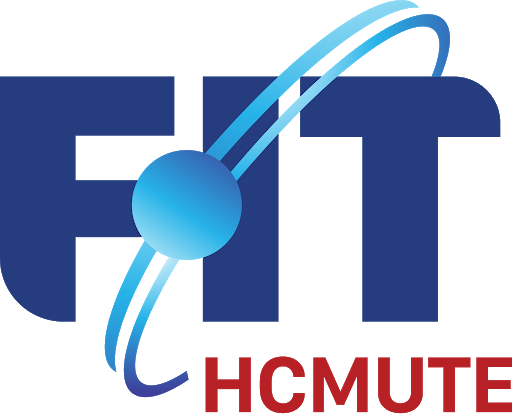
**.**

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

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN ĐIỆN TOÁN ĐÁM MÂY**

-----🙞🙜🕮🙞🙜-----

****

****



****

**KHÓA 2019-2023**

| **ĐH SƯ PHẠM KỸ THUẬT TP. HCM**  **KHOA CNTT**  \*\*\*\*\*\*\* | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh Phúc**  \*\*\*\*\*\*\* |
| --- | --- |

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1: **Lý Quốc Dũng** MSSV 1: **19133015**

Họ và tên Sinh viên 2: **Nguyễn Huỳnh Minh Trung** MSSV 2: **19133060**

Họ và tên Sinh viên 2: **Bùi Thị Ngân Tuyền** MSSV 2: **19133066**

Ngành: **Kỹ thuật dữ liệu**

Tên đề tài: **Xây dựng ứng dụng trên AWS cho phép tạo database và cung cấp API để thêm, xóa sửa trên database**

Họ và tên Giáo viên hướng dẫn: **TS. Huỳnh Xuân Phụng**

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

| *Tp. Hồ Chí Minh, ngày tháng năm 2022* | | |
| --- | --- | --- |
|  | Giáo viên hướng dẫn  *(Ký & ghi rõ họ tên)* |

# 

# **LỜI CẢM ƠN**

*Lời đầu tiên nhóm xin phép được gửi lời cảm ơn chân thành và sâu sắc nhất đến với Khoa Công Nghệ Thông Tin – 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 và thực hiện đề tài này.*

*Bên cạnh đó nhóm chúng em xin gửi đến thầy Huỳnh Xuân Phụng lời cảm ơn sâu sắc nhất. Trải qua một quá trình dài học tập và thực hiện đề tài trong thời gian qua. Thầy đã tận tâm chỉ bảo nhiệt tình nhóm chúng em trong suốt quá trình từ lúc bắt đầu cũng như kết thúc đề tài này.*

*Với sự hướng dẫn nhiệt tình, giảng dạy tận tình đầy đủ kiến thức của thầy Huỳnh Xuân Phụng, chúng em đã học tập và hiểu được những kiến thức cơ bản về AWS (Amazon Wed Service).*

*Tuy nhiên lượng kiến thức là vô tận và với khả năng hạn hẹp chúng em đã rất cố gắng để hoàn thành một cách tốt nhất. Chính vì vậy việc xảy ra những thiếu sót là điều khó có thể tránh khỏi. Chúng em hi vọng nhận được sự góp ý tận tình của quý thầy (cô) qua đó chúng em có thể rút ra được bài học kinh nghiệm và hoàn thiện và cải thiện nâng cấp lại sản phẩm của mình một cách tốt nhất có thể.*

*Chúng em xin chân thành cảm ơn!*

***Nhóm thực hiện***



# **MỤC LỤC**

[**LỜI CẢM ƠN**](#_heading=h.30j0zll) **3**

[**MỤC LỤC**](#_heading=h.1fob9te) **4**

[**DANH MỤC CÁC HÌNH**](#_heading=h.3znysh7) **6**

[**PHẦN MỞ ĐẦU**](#_heading=h.2et92p0) **6**

[**1. Tính cấp thiết của đề tài**](#_heading=h.tyjcwt) **7**

[**2. Đối tượng nghiên cứu**](#_heading=h.3dy6vkm) **7**

[**3. Phạm vi nghiên cứu**](#_heading=h.1t3h5sf) **7**

[**4. Kết quả dự kiến đạt được**](#_heading=h.4d34og8) **7**

[**PHẦN NỘI DUNG**](#_heading=h.ztmddar5imb) **8**

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**](#_heading=h.17dp8vu) **8**

[**Tổng quan về Lambda**](#_heading=h.3rdcrjn) **8**

[**Các tính năng của Lambda**](#_heading=h.p70bzuuhegf) **8**

[**. Các dịch vụ liên quan đến Lambda**](#_heading=h.z6lufxy5z92x) **9**

[**Tổng quan về DynamoDB**](#_heading=h.ohmqw8sh96in) **10**

[**. Các tính năng của DynamoDB**](#_heading=h.tnk5a3epviyu) **11**

[**CHƯƠNG 2: THỰC HIỆN**](#_heading=h.26in1rg) **12**

[**2.1 Cài đặt**](#_heading=h.lnxbz9) **12**

[*2.1.1 Cài đặt Function Lamda-*](#_heading=h.35nkun2) *12*

[**CreateTable: Function dùng để tạo bảng trong DynamoDB**](#_heading=h.2yq0y55w0un0) **12**

[**2.1.2 Cài đặt Simple Queue Service:**](#_heading=h.6xpehqxgtfom) **15**

[*2.1.3 Setup Web (PHP):*](#_heading=h.44sinio) *15*

[**2.2 Triển khai Project:**](#_heading=h.2jxsxqh) **19**

[**2.2.1. Local:**](#_heading=h.8lbev8y0o0w) **19**

[**2.2.2. EC2:**](#_heading=h.fwmdfafau0sz) **21**

[**2.3 Thiết kế DynamoDB minh họa cho việc tương tác:**](#_heading=h.z337ya) **21**

[**Danh sách các bảng**](#_heading=h.3whwml4) **21**

[**PHẦN KẾT LUẬN**](#_heading=h.vryggtmqd972) **22**

[**Kết quả đạt được**](#_heading=h.2bn6wsx) **22**

[*1.1. Kiến thức tìm hiểu được*](#_heading=h.qsh70q) *22*

[*1.2. Chương trình đã làm được*](#_heading=h.3as4poj) *22*

[**Ưu điểm**](#_heading=h.1pxezwc) **22**

[**Nhược điểm**](#_heading=h.49x2ik5) **23**

[**Hướng phát triển**](#_heading=h.2p2csry) **23**

[**TÀI LIỆU THAM KHẢO**](#_heading=h.147n2zr) **24**

# **DANH MỤC CÁC HÌNH**

[**Hình 1: CreateTable** 1](#_heading=h.2grqrue)2

[**Hình 2: DeleteTable** 1](#_heading=h.vx1227)2

[**Hình 3: CreateItem** 1](#_heading=h.3fwokq0)3

[**Hình 4: UpdateItem** 1](#_heading=h.3fwokq0)4

[**Hình 5: DeleteItem** 1](#_heading=h.3fwokq0)4

[**Hình 6: Tạo SQS** 1](#_heading=h.3fwokq0)5

[**Hình 7: Source Code Web** 1](#_heading=h.3fwokq0)5

[**Hình 8: Hàm Call API** 1](#_heading=h.3fwokq0)6

[**Hình 9: Giao diện danh sách bảng** 1](#_heading=h.3fwokq0)7

[**Hình 10: Giao diện Thêm Bảng** 1](#_heading=h.3fwokq0)8

[**Hình 11: Giao diện danh sách Items** 1](#_heading=h.3fwokq0)8

[**Hình 12: Giao diện thêm/sửa bảng**](#_heading=h.3fwokq0) 19

[**Hình 13: Config SQS vào Source code PHP**](#_heading=h.3fwokq0) 20

[**Hình 14: Config API vào Source code PHP**](#_heading=h.3fwokq0) 20

[**Hình 15: Danh sách các bảng DynamoDB**](#_heading=h.3fwokq0) 21

[**Hình 16: Danh sách Items**](#_heading=h.3fwokq0) 22

# **PHẦN MỞ ĐẦU**

## **1. Tính cấp thiết của đề tài**

Trong thời đại hiện nay Database hay cơ sở dữ liệu là một yếu tố không thể thiếu của ngành công nghệ thông tin. Nó đóng vai trong quan trọng nắm giữ mọi thông tin, dữ liệu mà các lập trình viên cần có để xây dựng hệ thống của mình.

## **2. 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.

## **3. 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.

## **4. Kết quả dự kiến đạt được**

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

# 

# **PHẦN NỘI DUNG**

# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

## **Tổng quan về Lambda**

Lambda là một dịch vụ máy tính cho phép bạn chạy mã mà không cần cung cấp hoặc quản lý máy chủ. Lambda chạy mã của bạn trên cơ sở hạ tầng máy tính có tính khả dụng cao và thực hiện tất cả việc quản lý các tài nguyên máy tính, bao gồm bảo trì máy chủ và hệ điều hành, cung cấp dung lượng và tự động mở rộng quy mô, theo dõi mã và ghi nhật ký. Với Lambda, bạn có thể chạy mã cho hầu hết mọi loại ứng dụng hoặc dịch vụ phụ trợ.

Lambda chỉ chạy chức năng khi cần thiết và tự động mở rộng quy mô, từ một vài yêu cầu mỗi ngày đến hàng nghìn yêu cầu mỗi giây. Chỉ trả tiền cho thời gian tính toán sử dụng - không tính phí khi mã không chạy.

Có thể gọi các hàm Lambda bằng cách sử dụng Lambda API hoặc Lambda có thể chạy các hàm để phản hồi các sự kiện từ các dịch vụ AWS khác. Sử dụng Lambda để:

* Xây dựng trình kích hoạt xử lý dữ liệu cho các dịch vụ AWS như Amazon Simple Storage Service (Amazon S3) và Amazon DynamoDB.
* Xử lý dữ liệu phát trực tuyến được lưu trữ trong Amazon Kinesis.
* Tạo chương trình phụ trợ của riêng bạn hoạt động ở quy mô, hiệu suất và bảo mật AWS..

## **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:

* Đồng thời và kiểm soát tỷ lệ

Các điều khiển đồng thời và điều chỉnh tỷ lệ như giới hạn đồng thời và đồng thời được cung cấp cung cấp khả năng kiểm soát chi tiết đối với việc mở rộng quy mô và khả năng đáp ứng của các ứng dụng sản xuất.

* Các hàm được xác định là hình ảnh vùng chứa

Sử dụng công cụ hình ảnh vùng chứa, quy trình làm việc và phụ thuộc ưa thích để xây dựng, kiểm tra và triển khai các chức năng của Lambda.

* Ký mã

Việc ký mã cho Lambda cung cấp các kiểm soát tin cậy và tính toàn vẹn cho phép xác minh rằng chỉ mã không thay đổi mà các nhà phát triển được chấp thuận đã xuất bản mới được triển khai trong các chức năng của Lambda.

* Phần mở rộng Lambda

Sử dụng các phần mở rộng Lambda để tăng cường các chức năng Lambda. Ví dụ: sử dụng các tiện ích mở rộng để dễ dàng tích hợp Lambda với các công cụ yêu thích để theo dõi, quan sát, bảo mật và quản trị.

* Bản thiết kế chức năng

Bản thiết kế chức năng cung cấp mã mẫu cho biết cách sử dụng Lambda với các dịch vụ AWS khác hoặc các ứng dụng của bên thứ ba. Bản thiết kế bao gồm mã mẫu và cài đặt trước cấu hình chức năng cho thời gian chạy Node.js và Python.

* Truy cập cơ sở dữ liệu

Một proxy cơ sở dữ liệu quản lý một nhóm các kết nối cơ sở dữ liệu và chuyển tiếp các truy vấn từ một hàm. Điều này cho phép một chức năng đạt đến mức đồng thời cao mà không làm cạn kiệt các kết nối cơ sở dữ liệu.

* Quyền truy cập hệ thống tệp

Có thể định cấu hình một chức năng để gắn hệ thống tệp Amazon Elastic File System (Amazon EFS) vào một thư mục cục bộ. Với Amazon EFS, mã chức năng có thể truy cập và sửa đổi các tài nguyên được chia sẻ một cách an toàn và có tính đồng thời cao.

## **. 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:

* Sử dụng API Gateway để cung cấp một cổng an toàn và có thể mở rộng cho các API web định tuyến các yêu cầu HTTP đến các hàm Lambda.
* Đối với các dịch vụ tạo hàng đợi hoặc luồng dữ liệu (chẳng hạn như DynamoDB và Kinesis), Lambda thăm dò hàng đợi hoặc luồng dữ liệu từ dịch vụ và gọi hàm để xử lý dữ liệu đã nhận.
* 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.

## **Tổng quan về DynamoDB**

Amazon DynamoDB là một dịch vụ cơ sở dữ liệu NoSQL được quản lý hoàn toàn, cung cấp hiệu suất nhanh và có thể dự đoán được với khả năng mở rộng liền mạch. DynamoDB giúp giảm bớt gánh nặng quản trị của việc vận hành và mở rộng cơ sở dữ liệu phân tán không phải lo lắng về việc cung cấp phần cứng, thiết lập và cấu hình, sao chép, vá lỗi phần mềm hoặc mở rộng cụm. DynamoDB cũng cung cấp mã hóa ở trạng thái nghỉ, giúp loại bỏ gánh nặng hoạt động và sự phức tạp liên quan đến việc bảo vệ dữ liệu nhạy cảm.

Với DynamoDB, có thể tạo các bảng cơ sở dữ liệu có thể lưu trữ và truy xuất bất kỳ lượng dữ liệu nào cũng như phục vụ bất kỳ mức lưu lượng yêu cầu nào. Có thể tăng hoặc giảm quy mô công suất thông qua bảng mà không có thời gian chết hoặc giảm hiệu suất. Có thể sử dụng AWS Management Console để theo dõi việc sử dụng tài nguyên và các chỉ số hiệu suất.

Có thể tạo bản sao lưu theo yêu cầu và bật khôi phục tại thời điểm cho các bảng Amazon DynamoDB. Khôi phục theo thời gian giúp bảo vệ bảng khỏi các thao tác ghi hoặc xóa ngẫu nhiên. Với khôi phục theo thời gian, có thể khôi phục bảng về bất kỳ thời điểm nào trong 35 ngày.

DynamoDB cho phép tự động xóa các mục đã hết hạn khỏi bảng để giúp giảm mức sử dụng bộ nhớ và chi phí lưu trữ dữ liệu không còn phù hợp.

## **. Các tính năng của DnamoDB**

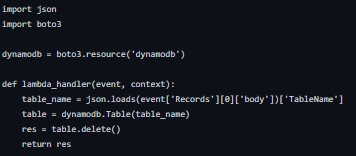
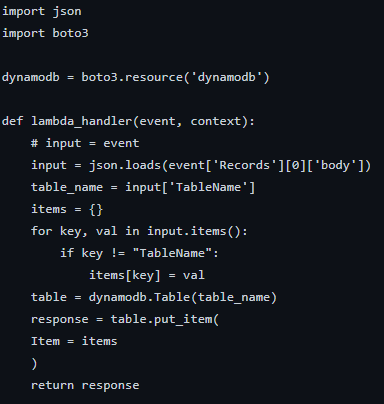
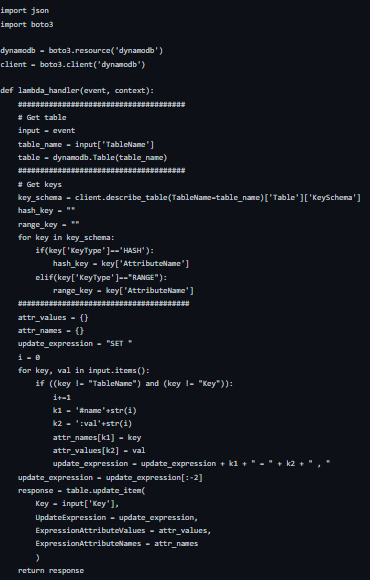
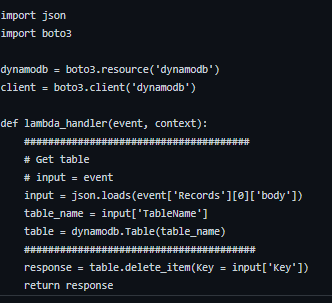
* AWS Glue Elastic Views hỗ trợ DynamoDB dưới dạng một nguồn để kết hợp và sao chép liên tục dữ liệu trên nhiều cơ sở dữ liệu sát thời gian thực.
* Sử dụng PartiQL, một loại ngôn ngữ truy vấn tương thích SQL để truy vấn, chèn, cập nhật và xóa dữ liệu bảng trong DynamoDB.
* Sử dụng Amazon Kinesis Data Streams để nắm bắt các thay đổi cấp độ mục trong các bảng DynamoDB của bạn.
* Khôi phục các bảng DynamoDB nhanh hơn nữa.
* Công cụ tính giá AWS hiện đã hỗ trợ DynamoDB.
* 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: THỰC HIỆN**

## **2.1 Cài đặt**

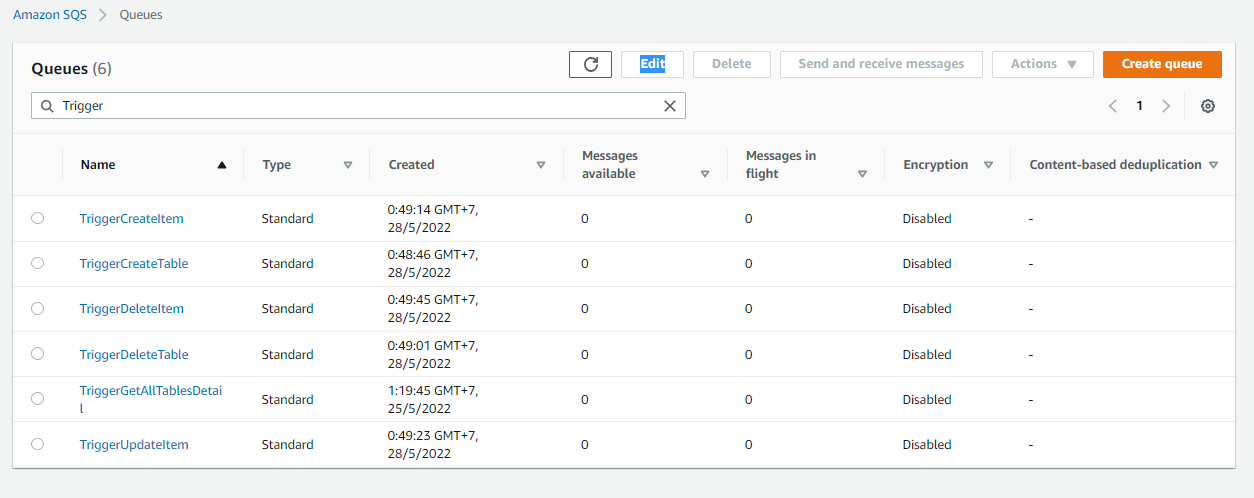
### *2.1.1 Cài đặt Function Lamda*-

#### CreateTable: Function dùng để tạo bảng trong DynamoDB

* DeleteTable: Function dùng để xóa bảng trong DynamoDB 
* CreateItem: Function dùng để tạo item trong DynamoDB
* UpdateItem: Function dùng để cập nhật các item trong DynamoDB 
* DeleteItem: Function dùng để xóa các item trong DynamoDB

### *2.1.2 Cài đặt Simple Queue Service:*

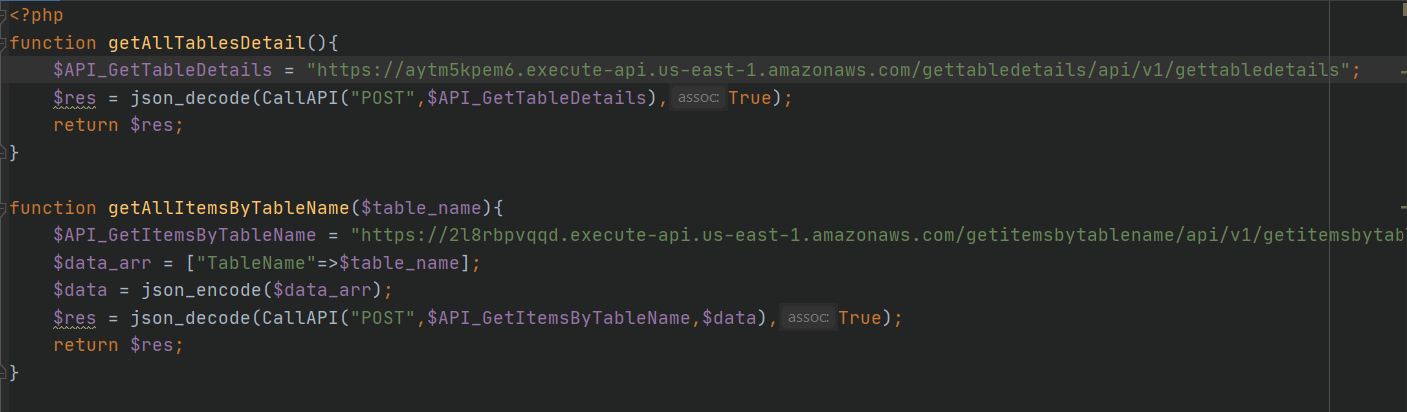
Lần Lượt tạo 5 SQS và kết nối đến các Function Lamda đã tạo:

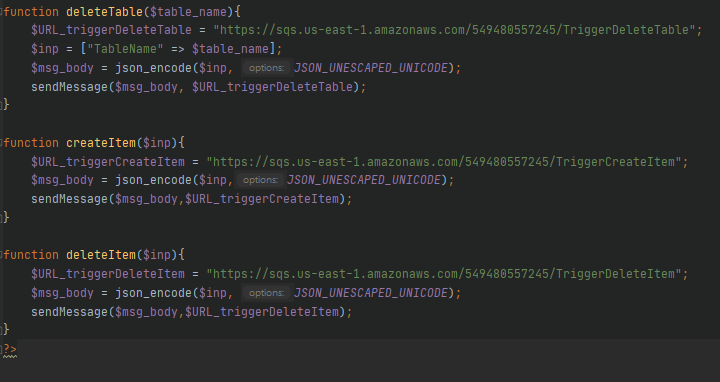
* TriggerCreateItem: trigger đến Lamda tên CreateItem
* TriggerCreateTable: trigger đến Lamda tên CreateTable
* TriggerDeleteItem: trigger đến Lamda tên DeleteItem
* TriggerDeleteTable: trigger đến Lamda tên DeleteTable
* TriggerUpdateItem: trigger đến Lamda tên UpdateItem
* 

### *2.1.3 Setup Web (PHP):*

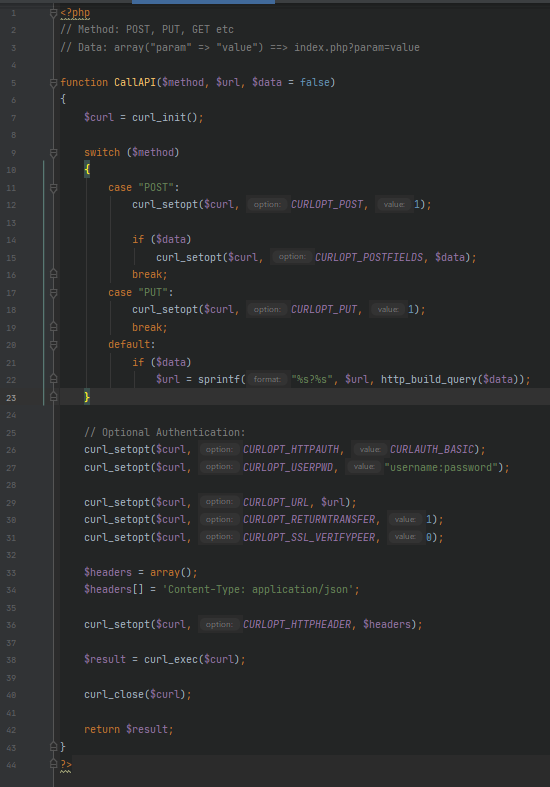
* Source Code:

**Tất cả các hàm sử dụng API để tương tác với Dynamodb**



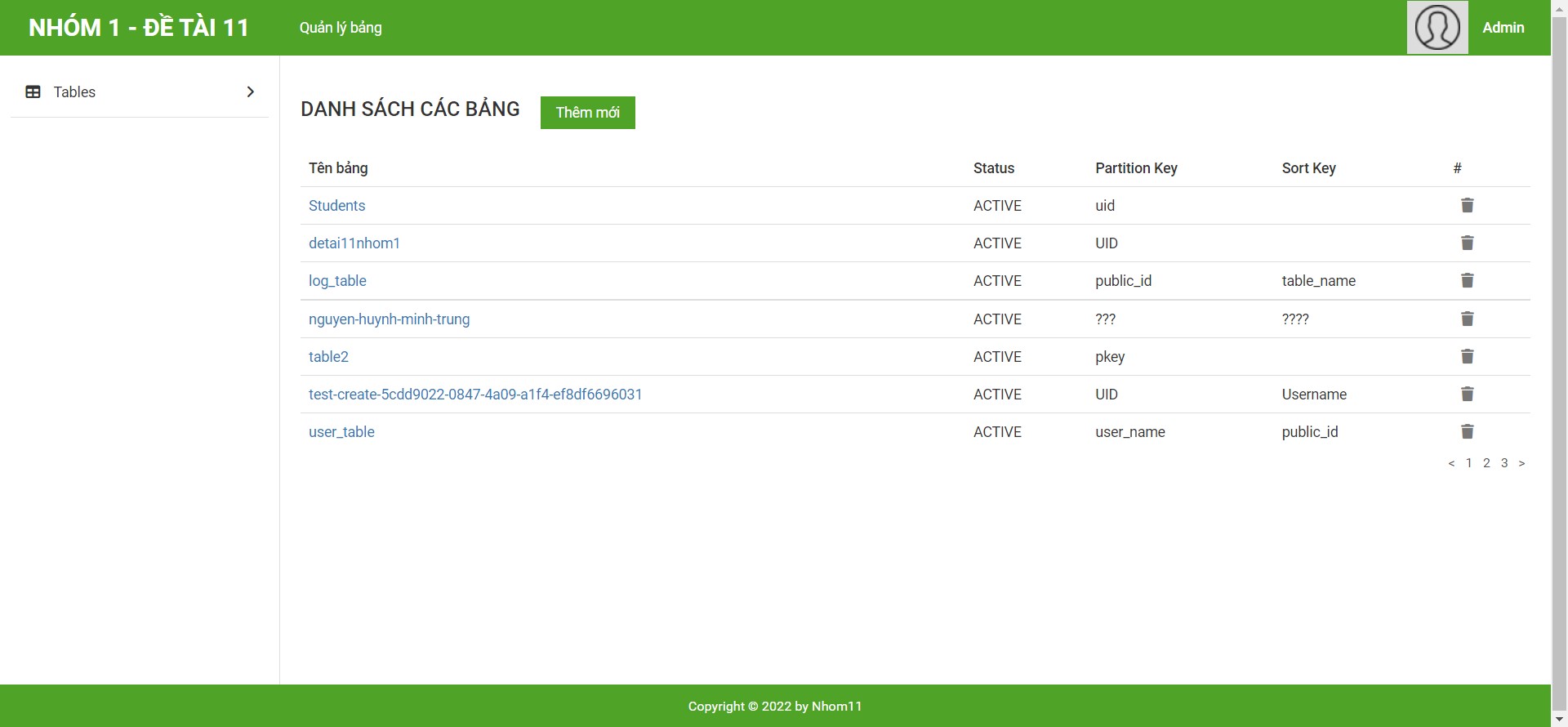


**Hàm Call API cho phép gửi request đến một API và nhận kết quả trả về của API đó**

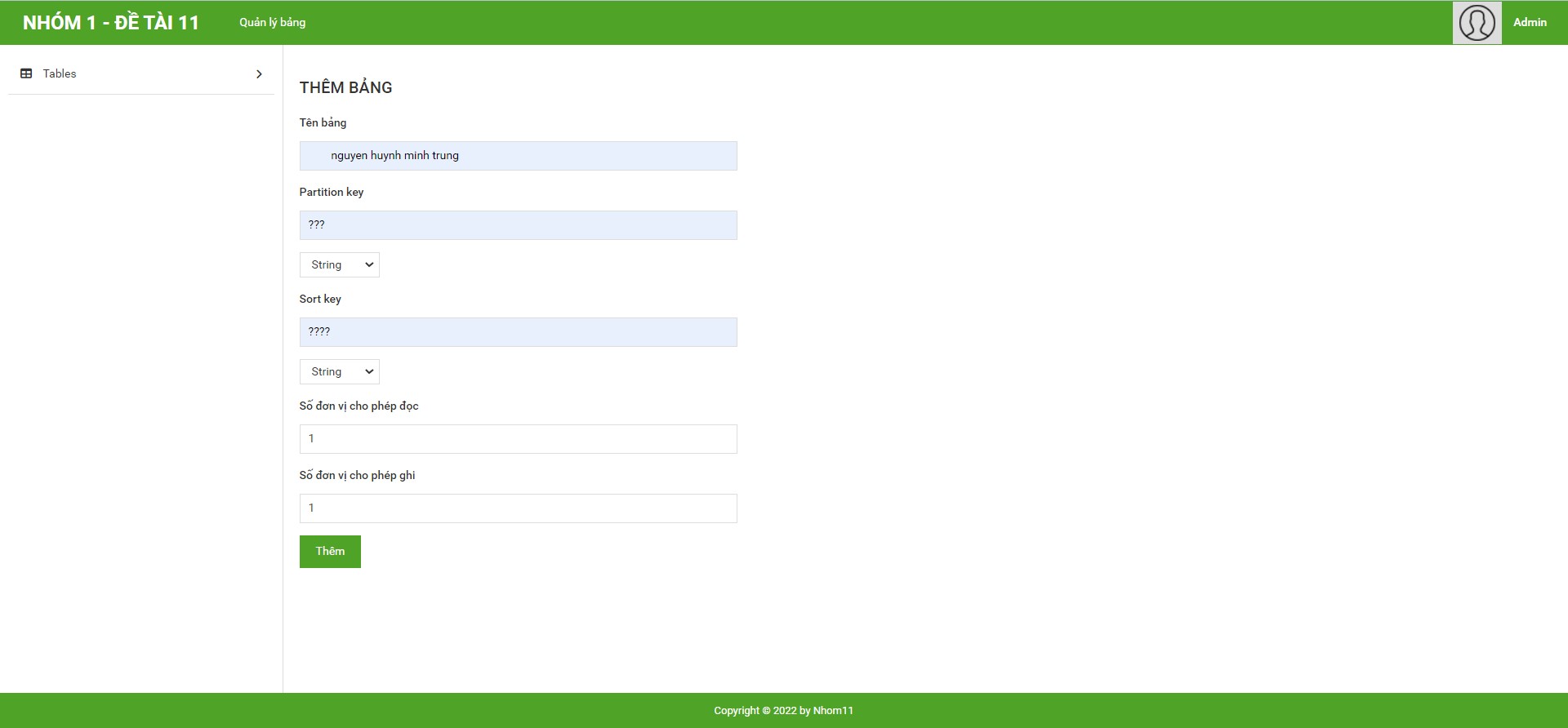
****

* Giao Diện:

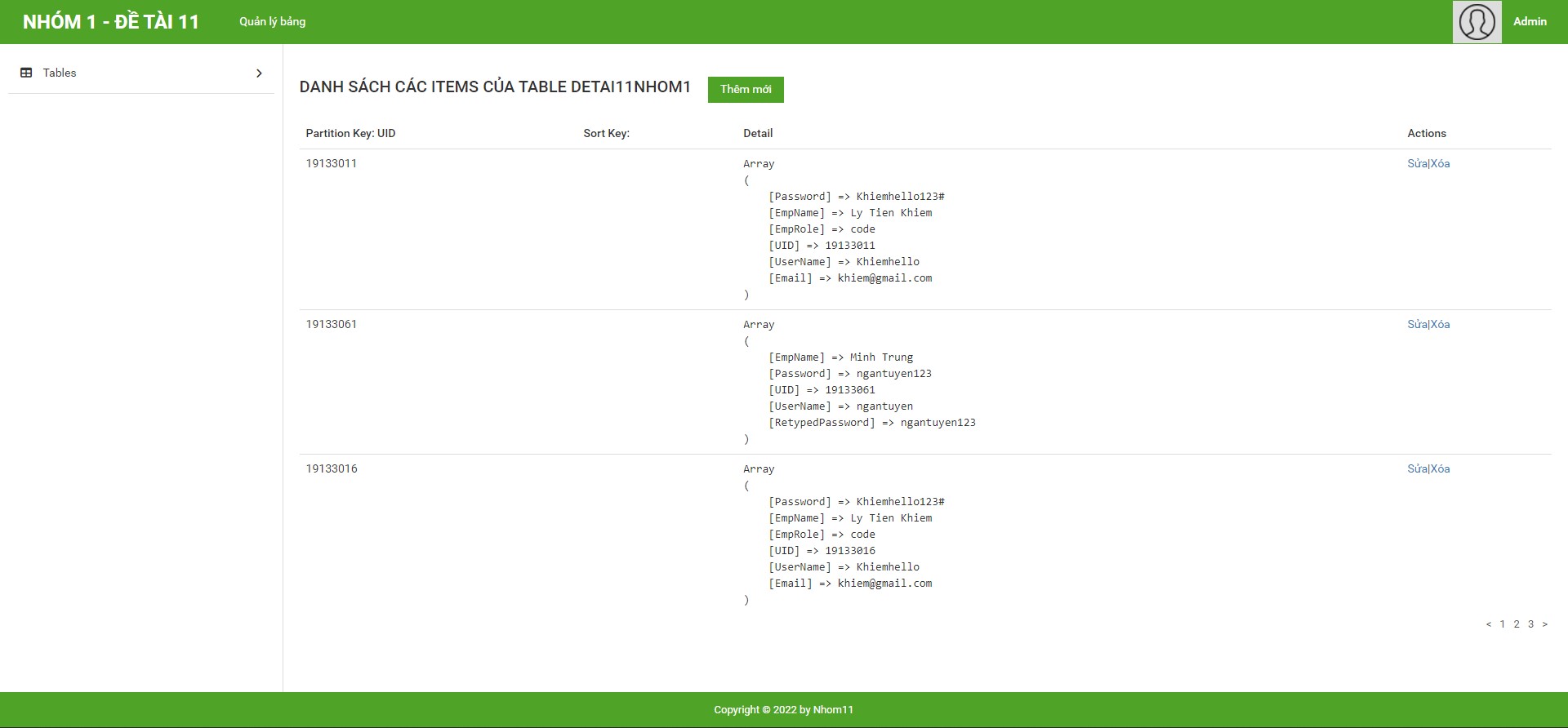
Giao diện hiển thị danh sách bảng



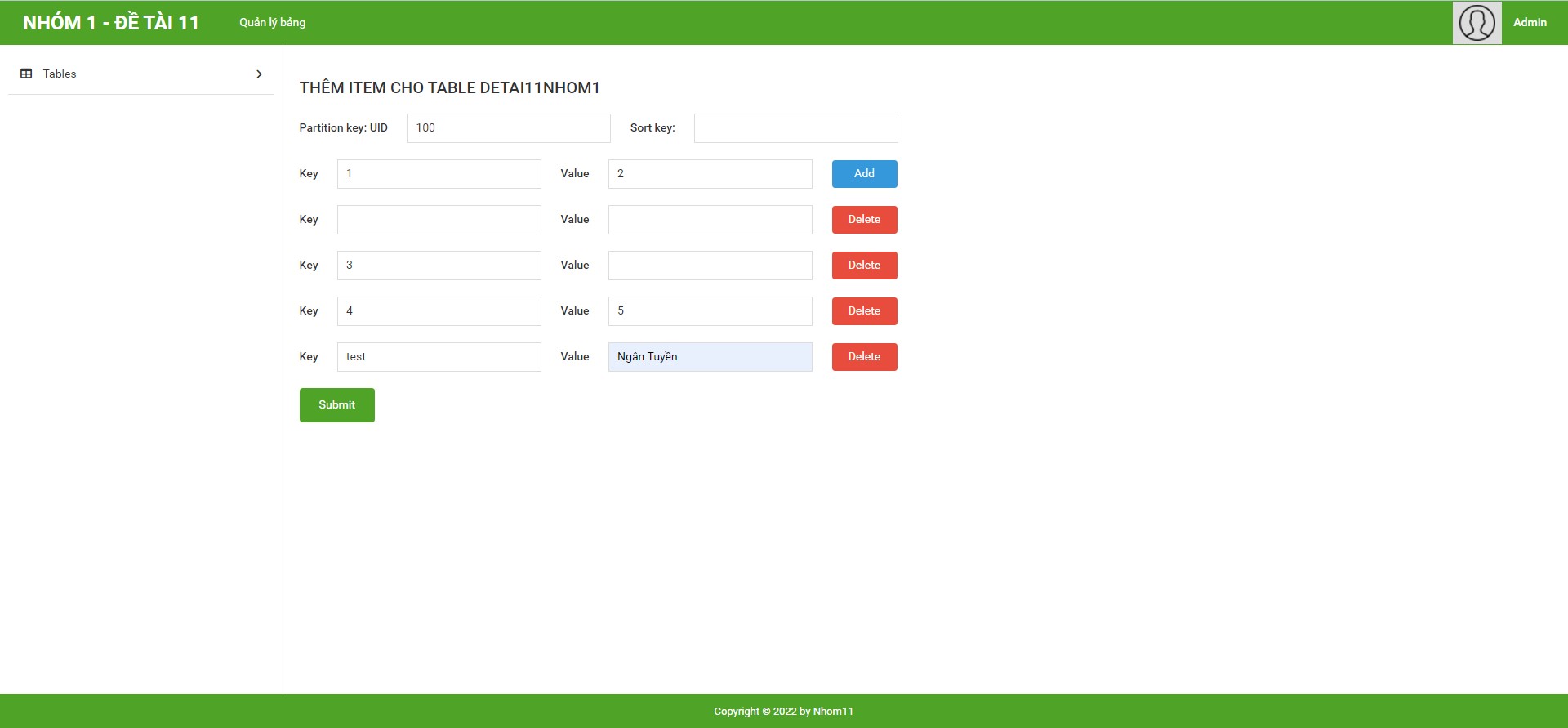
Giao diện thêm bảng



Giao diện hiển thị danh sách các Items trong bảng



Giao diện thêm/Sửa Item của bảng



## **2.2 Triển khai Project:**

### *2.2.1. Local:*

-Link video hướng dẫn: <https://youtu.be/L9aZ38QmrSo>

Bước 1: Chuẩn bị môi trường

Vào LearnLab--> start lab --> Aws Detail --> AWS CLI --> show copy bảng credentials

Vào thư mục C:\Users\tên máy\ .aws\credentials

Past AWS CLI vừa copy file credentials

Tải và cài đặt XAMPP và Git

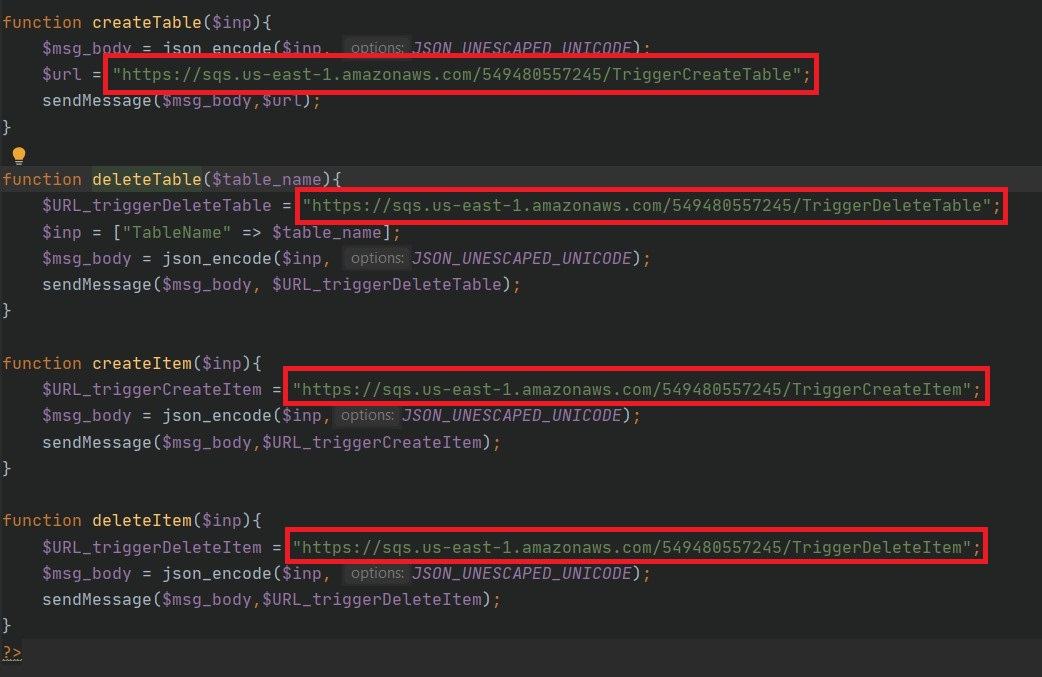
Bước 2: Cấu hình lại đồ án

git clone https://github.com/trungnguyenhuynhminh46/ProjectDetai11Nhom1v2 vào thư mục htdocs của XAMPP

**Tạo API bằng SQS:**

Vào AWS Tạo các SQS và Lamda theo file Lamda.txt

Chú ý Copy URL các SQS vừa tạo và past vào file lib/tables.php trong thư mục project để thay thế các URL của SQS

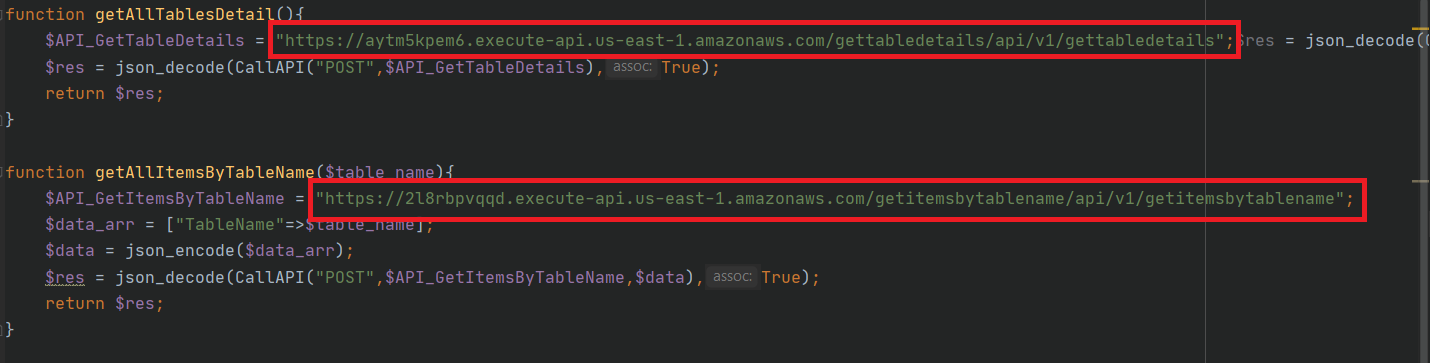


**Tạo API bằng API Gateway**

Vào AWS tạo các Lamda function theo API.txt

Làm theo video sau để tạo API cho 2 function vừa tạo: https://youtu.be/vKmmZ9Wp2-s

Sửa đường dẫn API tương tự SQS



Bước 3: Chạy

Mở XAMPP và start Apache

mở trình duyệt http://localhost:'port-apache'/ProjectDetai11Nhom1

### *2.2.2. EC2:*

* Link video hướng dẫn: <https://youtu.be/LqgAbx-NY0w>
* Các Bước thực hiện

Tạo EC2 với hệ điều hành window, mở port http và alltraffic

Tạo Elastic IP kết nối đến EC2

Connect EC2 bằng remote Desktop Connection

Trên window của EC2 vào Window Security --> Firewall & network protection --> Windows Defender Firewall Properties --> tag Public Profile --> Allow Inbound connections --> Apply--> ok

Dowload XAMPP và GIT

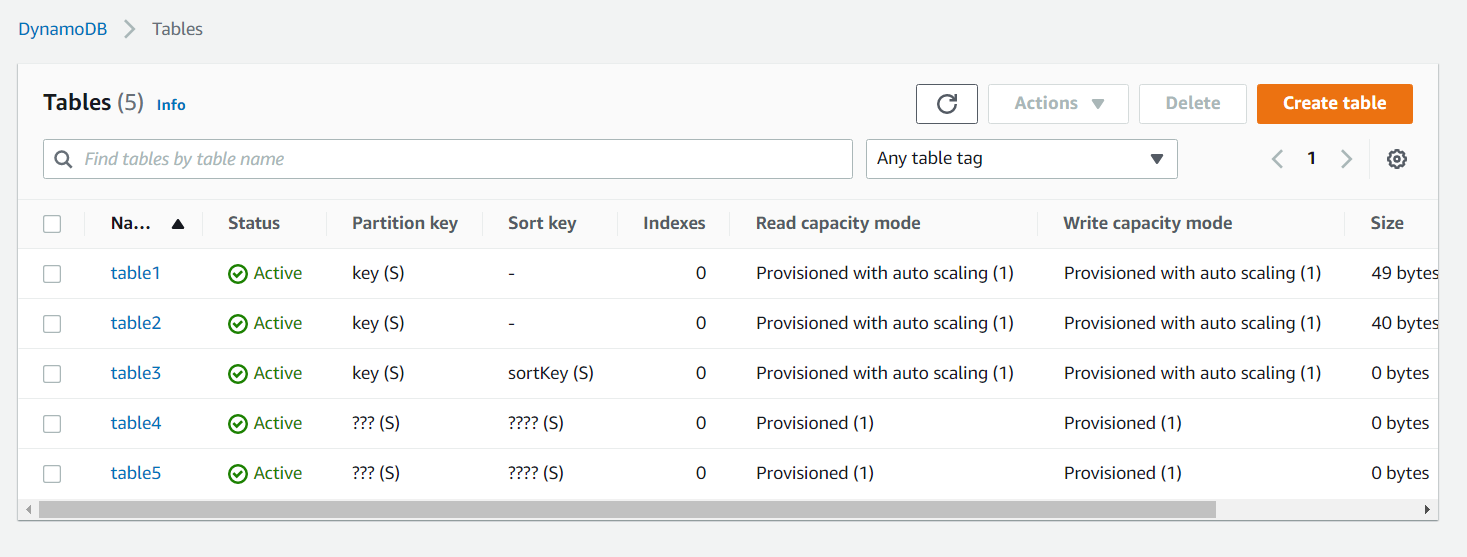
Mở git bash trong thư mục htdocs của XAMPP vừa tải về và git clone <https://github.com/trungnguyenhuynhminh46/ProjectDetai11Nhom1.git>

Copy .aws của local máy tính và past vào c/user/Administrator của máy window EC2

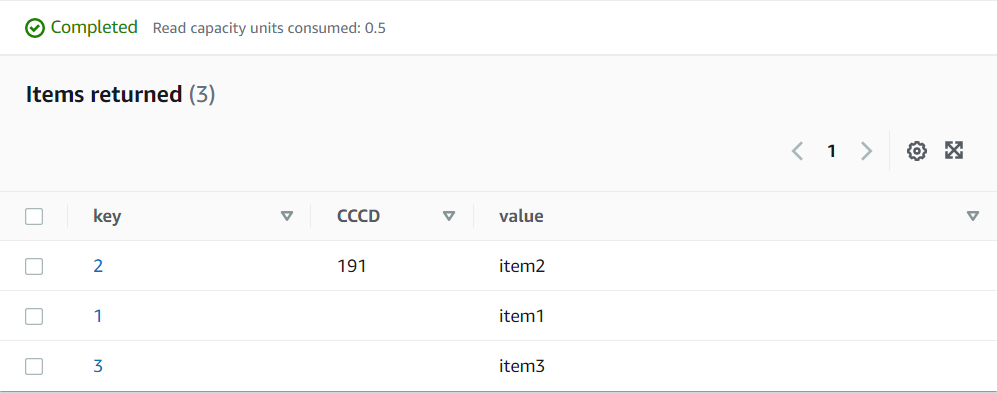
Mở XAMPP start apache và mở trình duyệt vào http://<ip máy EC2>/ProjectDetai11Nhom1 => Deploy thành công

## **2.3 Thiết kế DynamoDB minh họa cho việc tương tác:**

# Danh sách các bảng



Danh sách Items của một bảng (table1)



# **PHẦN KẾT LUẬN**

## **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:

### *1.1. Kiến thức tìm hiểu được*

Nắm bắt được các kiến thức cũng như những cách hoạt động của các dịch vụ Lamda Function, SQS, API, DynamoDB 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.

### *1.2. Chương trình đã làm được*

Xây dựng hoàn chỉnh một ứng dụng tương tác với database (DynamoDB) bằng code PHP, Html, Css và các dịch vụ của AWS: SQS, 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.

## **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.

## **Nhược điểm**

* Khó Setup.
* Các Function Lamda được theo một mẫu sẵn nên cần phải dành thời gian tìm hiểu.
* Tốn phí vì sử dụng dịch vụ của AWS và phụ thuộc phần lớn vào AWS trong vấn đề chạy code.

## **Hướng phát triển**

* Cải thiện giao diện

# **TÀI LIỆU THAM KHẢO**

[1]. What is Amazon Simple Queue Service?

Link: https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/welcome.html

[2]. AWS Lambda The Ultimate Guide

Link: https://www.serverless.com/aws-lambda

[3]. What is DynamoDB?

Link: https://www.dynamodbguide.com/what-is-dynamo-db/

[4]. *Amazon API Gateway*

Link: https://aws.amazon.com/api-gateway/

[5]. *Using AWS Lambda with Amazon API Gateway*

Link: https://docs.aws.amazon.com/lambda/latest/dg/services-apigateway.html

[6]. *PHP Code Samples for Amazon API Gateway*

Link: https://docs.aws.amazon.com/code-samples/latest/catalog/code-catalog-php-example\_code-apigateway.html