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

**HỆ THỐNG PHÁT HIỆN VÀ**

**PHÒNG CHỐNG XÂM NHẬP**

**LỚP: PROJ215879\_22\_1\_16CLC**

**Môn: Đồ án Công nghệ thông tin  
NHÓM: 1  
GVHD: Huỳnh Nguyên Chính**

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

**DANH SÁCH NHÓM**

HỌC KÌ I, NĂM HỌC: 2022-2023

**Tên đề tài: Hệ thống phát hiện và phòng chống xâm nhập**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Mức độ đóng góp |
| 1 | Võ Trần Bảo Nguyên | 20110138 | 100% |
| 2 | Huỳnh Hồ Thọ Tỷ | 20110597 | 100% |

**Nhận xét của giáo 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 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)*

*Tiếp theo chúng em muốn gửi một lời cảm ơn chân thành đến thầy Huỳnh Xuan 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 4](#_Toc121131722)

[1. Phần mở đầu: 4](#_Toc121131723)

[1.1. Tính cấp thiết của đề tài: 4](#_Toc121131724)

[1.2. Đối tượng nghiên cứu: 4](#_Toc121131725)

[1.3. Phạm vi nghiên cứu: 4](#_Toc121131726)

[1.4. Kết quả dự kiến đạt được: 4](#_Toc121131727)

[2. Đặc tả: 5](#_Toc121131728)

[2.1. Mục tiêu đề tài: 5](#_Toc121131729)

[2.2. Cơ sở lý thuyết: 5](#_Toc121131730)

[2.2.1. Tổng quan về AWS Lambda: 5](#_Toc121131731)

[2.2.2. Tổng quan về AWS DynamoDB: 5](#_Toc121131732)

[3. Thực hiện đồ án: 6](#_Toc121131733)

[3.1. Phần Database: 6](#_Toc121131734)

[3.2. Phần Back-End: 6](#_Toc121131735)

[3.3. Phần Front-End: 7](#_Toc121131736)

[3.4. Hướng dẫn cài đặt và chạy dự án trên Local: 9](#_Toc121131737)

[4. Đánh giá và nhận xét: 10](#_Toc121131738)

[4.1. Phần Back-end và Database: 10](#_Toc121131739)

[4.2. Phần Front-End: 10](#_Toc121131740)

[5. Tài liệu tham khảo: 11](#_Toc121131741)

**Danh mục các hình**

[Hình 1.1 Sơ đồ các thành phần trong hệ thống 10](#_Toc120692060)

[Hình 1.2 Kiến trúc của hệ thống IDS 11](#_Toc120692061)

[Hình 1.3 Kiến trúc triển khai hệ thống 12](file:///C:\Users\Admin\Documents\Hoctap\2022_HK1\DACNTT\Nhom1-IDS&IPS.docx#_Toc120692062)

[Hình 3.1 Sơ đồ các máy ảo sử dụng trong dự án 15](file:///C:\Users\Admin\Documents\Hoctap\2022_HK1\DACNTT\Nhom1-IDS&IPS.docx#_Toc120692063)

[Hình 4.1 Màn hình tùy chọn tải file ISO cho máy pfSense 17](#_Toc120692064)

[Hình 4.2 Thiết lập hệ thống mạng cho các máy ảo trên VMware 18](#_Toc120692065)

[Hình 4.3 Màn hình cài đặt máy ảo trên Vmware 19](file:///C:\Users\Admin\Documents\Hoctap\2022_HK1\DACNTT\Nhom1-IDS&IPS.docx#_Toc120692066)

[Hình 4.4 Màn hình tùy chọn file ISO 20](file:///C:\Users\Admin\Documents\Hoctap\2022_HK1\DACNTT\Nhom1-IDS&IPS.docx#_Toc120692067)

[Hình 4.5 Giao diện cài đặt các thiết bị trên máy pfSense 21](#_Toc120692068)

[Hình 4.6 Giao diện máy pfSense sau khi cài đặt với 2 card mạng 22](#_Toc120692069)

[Hình 4.7 Màn hình đăng nhập trên GUI của pfSense 23](#_Toc120692070)

[Hình 4.8 Giao diện Dashboard của pfSense 23](file:///C:\Users\Admin\Documents\Hoctap\2022_HK1\DACNTT\Nhom1-IDS&IPS.docx#_Toc120692071)

[Hình 4.9 Cấu trúc rule của IDS/IPS 24](#_Toc120692072)

[Hình 4.10 Giao diện Global Settings của Snort 25](#_Toc120692073)

[Hình 4.11 Giao diện phần Oinkcode trên trang web của snort 26](#_Toc120692074)

[Hình 4.12 Phần cuối của giao diện Global Settings 26](#_Toc120692075)

[Hình 4.13 Giao diện phần Updates dùng để Update tải lên các bộ Rules 27](#_Toc120692076)

[Hình 4.14 Giao diện thêm Interface ở phần Snort Interface 27](#_Toc120692077)

[Hình 4.15 Giao diện của máy pfSense (Với 2 địa chỉ mạng mà ta đã cài đặt từ trước) 28](#_Toc120692078)

[Hình 4.16 Giao diện Snort Interface sau khi thêm một interface mới 28](#_Toc120692079)

[Hình 4.17 Giao diện thiết lập interface ở phần WAN Categories 29](#_Toc120692080)

[Hình 4.18 Snort status của interface khi chưa được khởi động 29](#_Toc120692081)

[Hình 5.1 Giao diện tạo custom rule cho Snort trên GUI 31](#_Toc120692082)

[Hình 5.2 Màn hình terminal của máy Ubuntu thực hiện ping 31](#_Toc120692083)

[Hình 5.3 Danh sách các thông báo phát hiện thành công các hoạt động sử dụng giao thức ICMP 32](#_Toc120692084)

[Hình 5.4 Giao diện cài đặt Block ở phần WAN Setting 33](#_Toc120692085)

[Hình 5.5 Màn hình ping trước khi từ máy pfSense đến máy Kali 33](#_Toc120692086)

[Hình 5.6 Nơi chứa thư mục DdoS-Ripper dùng để thực hiện tấn công DoS 34](#_Toc120692087)

[Hình 5.7 Màn hình Terminal khi đang trong quá trình thực hiện tấn công DoS 35](#_Toc120692088)

[Hình 5.8 Danh sách một số Alert thu được sau khi cuộc tấn công bắt đầu 35](#_Toc120692089)

[Hình 5.9 Giao diện hiển thị các IP đã bị chặn và các chi tiết các Alert liên quan 36](#_Toc120692090)

[Hình 5.10 Ping thất bại sau khi đã tấn công và Snort đã thực hiện chặn địa chỉ IP 36](#_Toc120692091)

# 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:

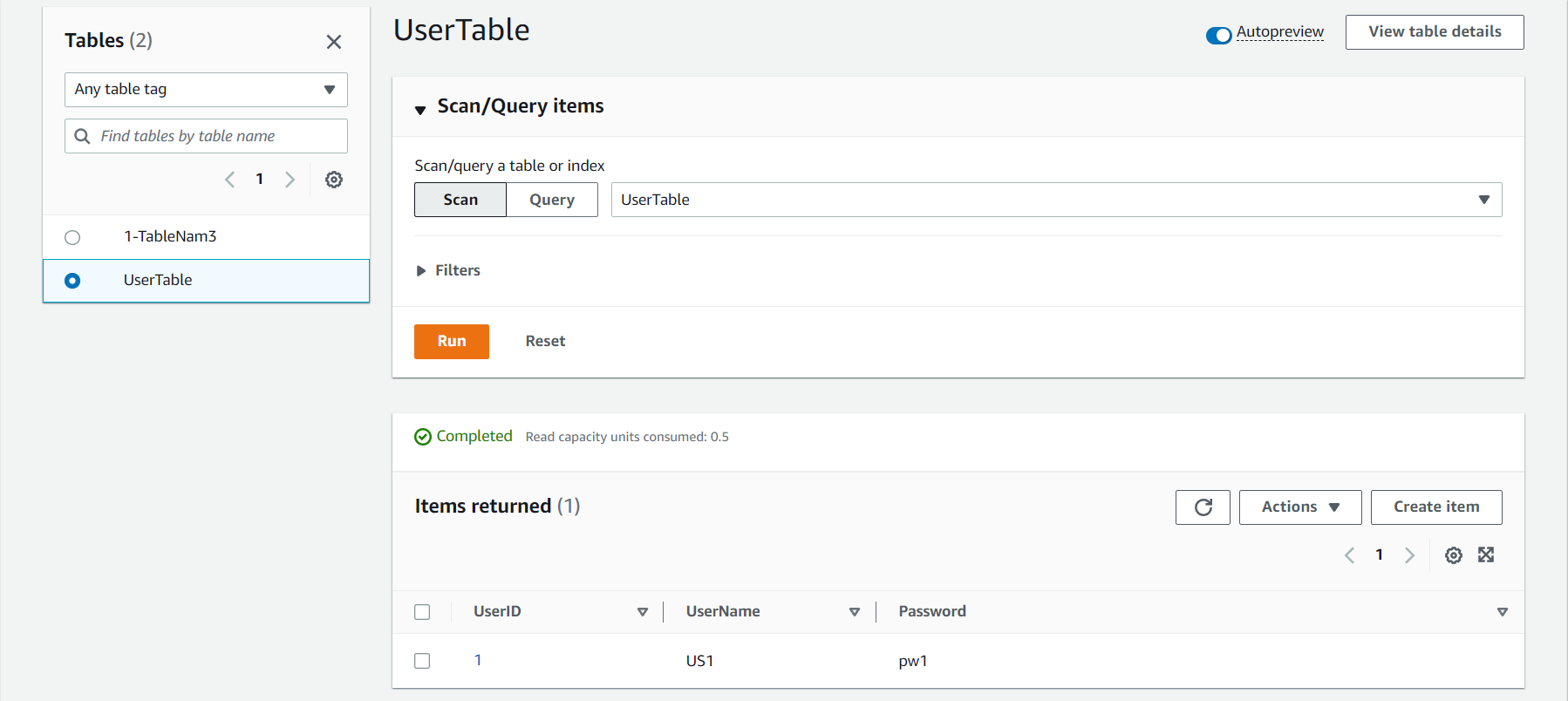
AWS DynamoDB là một dạng cơ sở dữ liệu phi quan hệ (NoSQL) được lưu trữ ngay trên môi trường của AWS. Dịch vụ này cung cấp cho người dùng một Database duy nhất và người dùng có thể thông quan Database đó tạo ra những bảng (Table) để lưu trữ các dữ liệu của bản thân. 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

# Thực hiện đồ án:

## Phần Database:

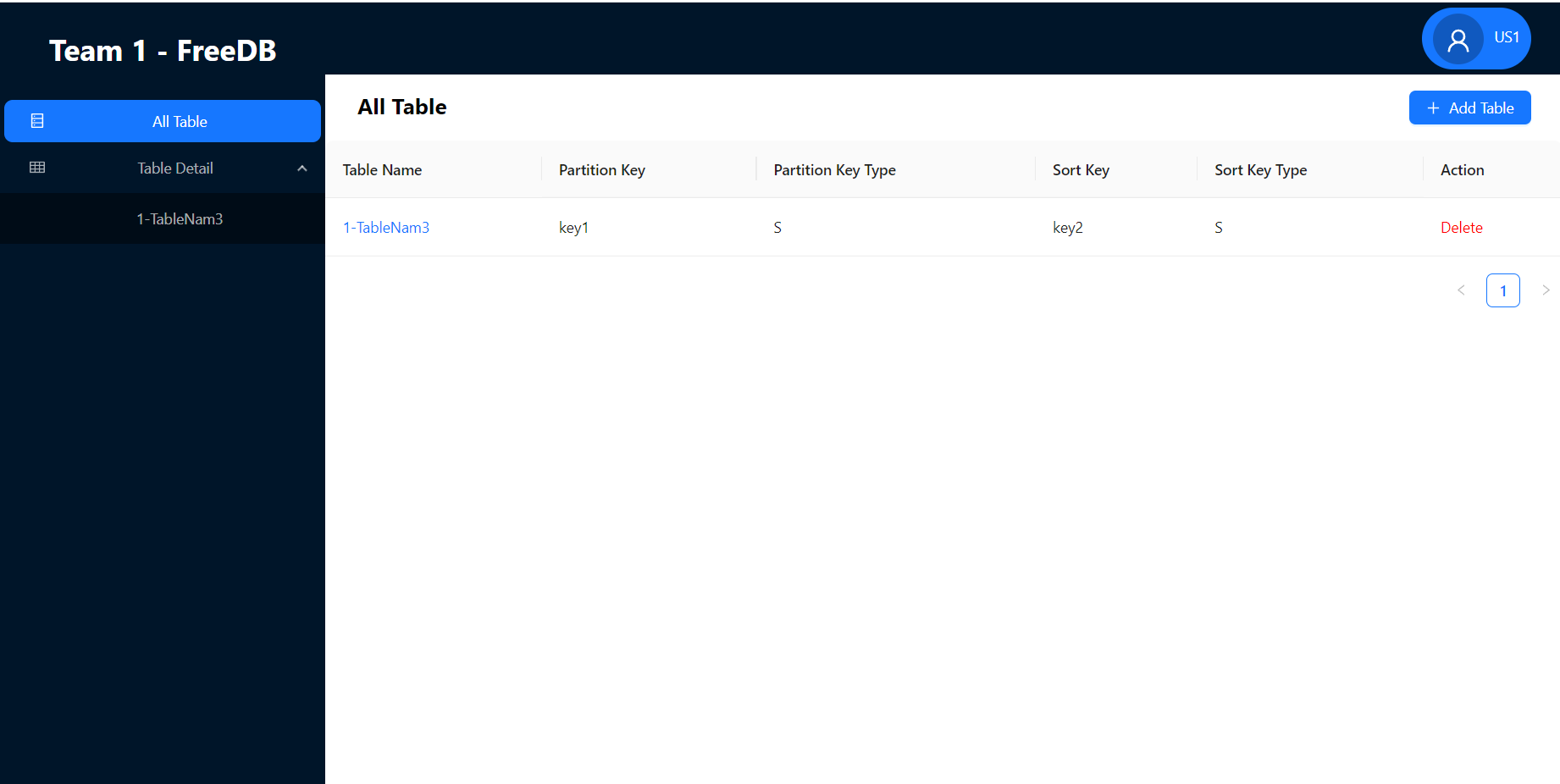
Database sẽ được tạo sẵn một bảng có tên là UserTable với khóa chính là UserID với kiểu dữ liệu là Number, khóa phụ là UserName với kiểu dữ liệu là String. Sau đó ta tạo một Record với User ID là **1**, UserName là **US1** và thêm một thuộc tính với kiểu là String 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.



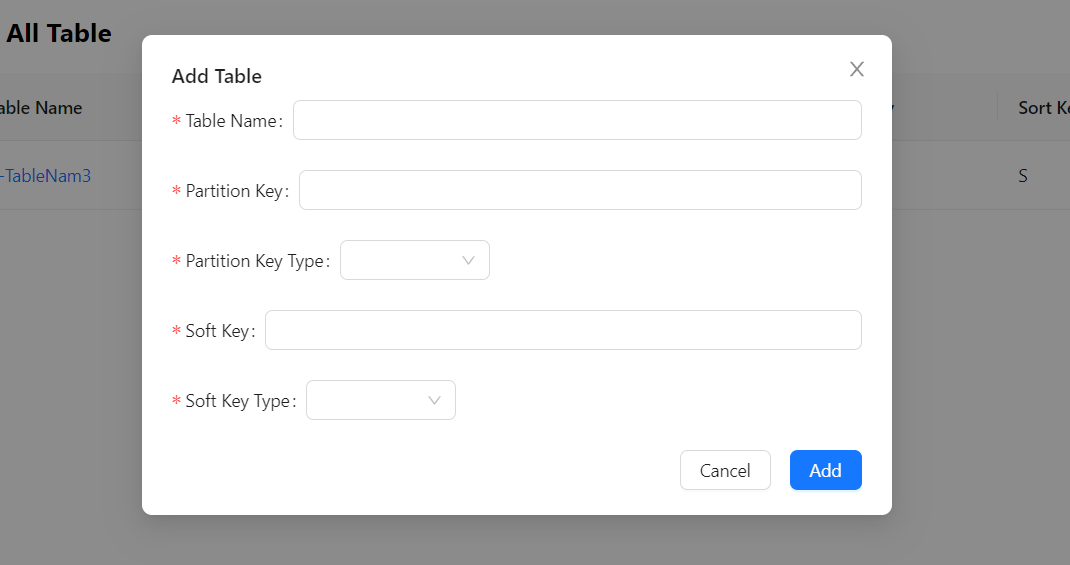
## Phần Back-End:

## Phần Front-End:

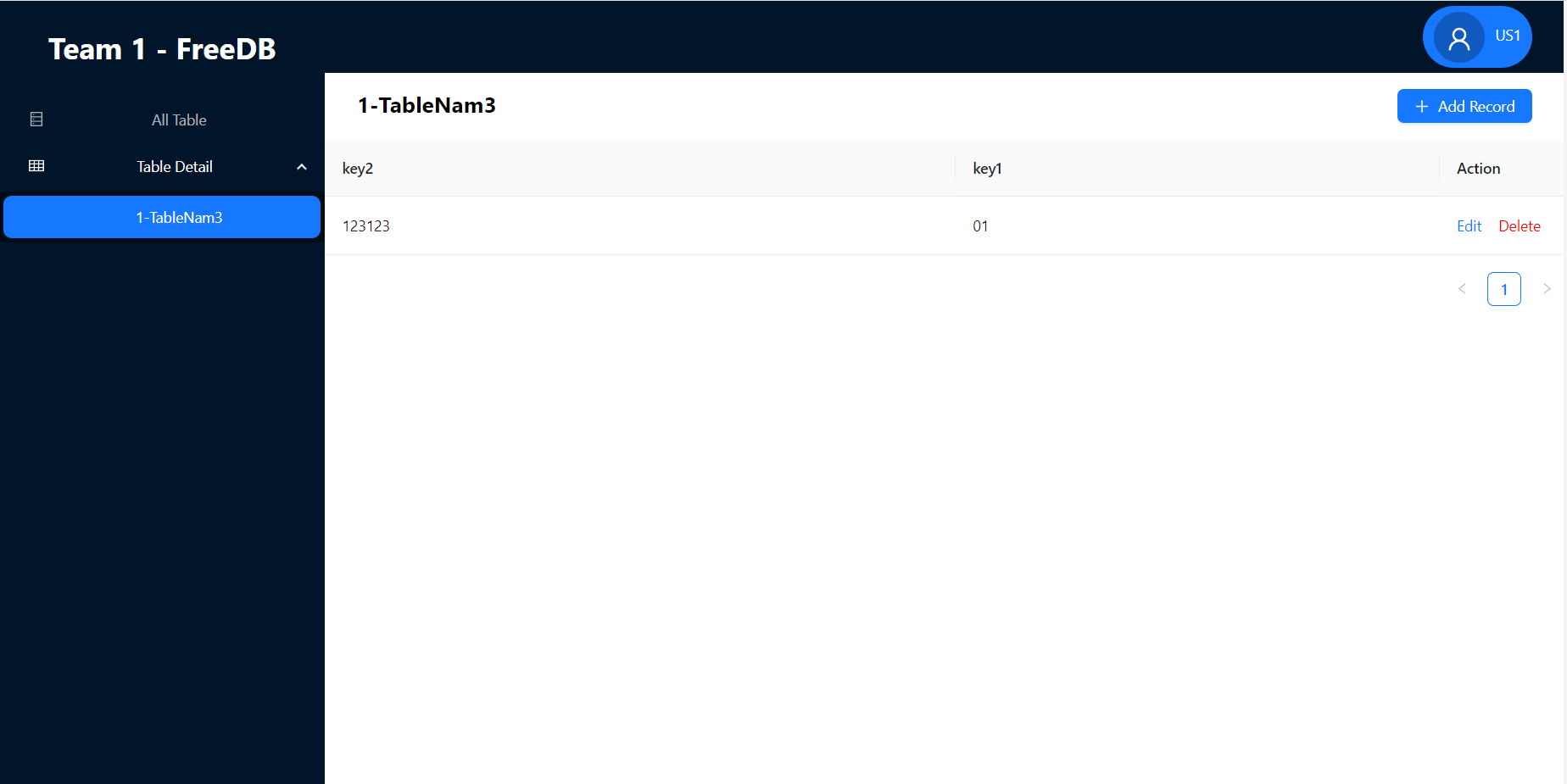
Phần Front-End sẽ được tạo và viết bằng Ngôn ngử Javascript và sử dụng thư viện chính là ReactJS để thực hiện. Ngoài ra thư viện Ant-Design cũng được sử dụng đễ hộ trợ tạo phần UI nhanh chóng và dễ dàng hơn



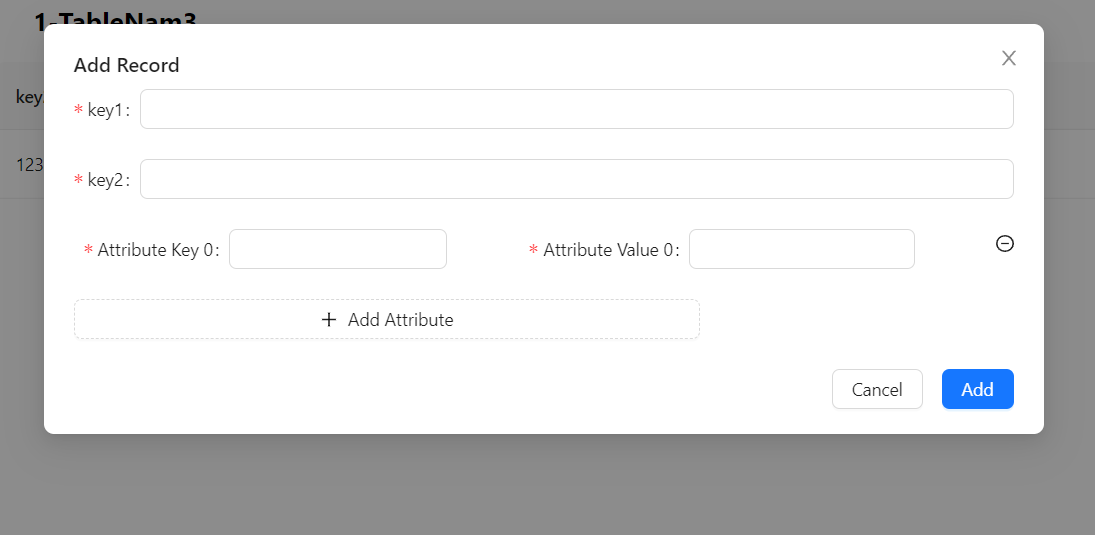
Giao diện trên chính là giao diện cho phép xem tất cả các bảng thuộc về User trên với tên 2 khóa đã tạo và kiểu dữ liệu của khóa và chức năng xóa bảng



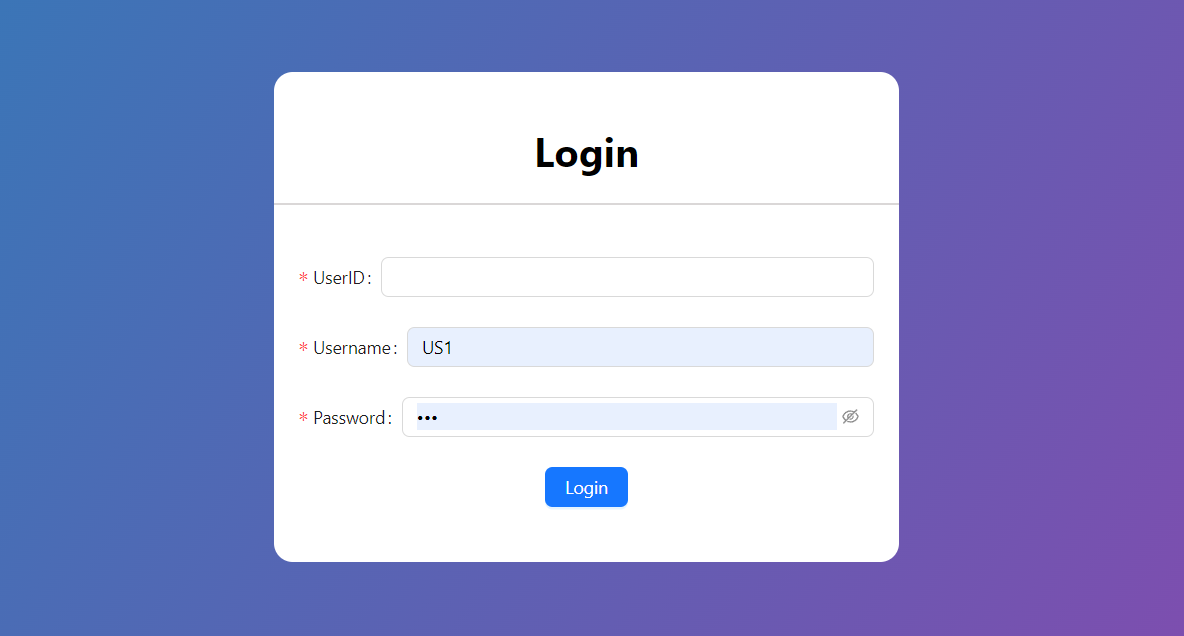
Ngoài ra khi nhấn vào nút Add table ta sẽ có một modal hiện ra cho phép tạo thêm một bảng mới



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

# Đánh giá và nhận xét:

## Phần Back-end và Database:

## Phần Front-End:

Hiện tại trang web vẫn còn thiếu xót khá nhiều về mặt giao diện như là vẫn chưa có giao diện phần đăng kí tài khoản, phần đăng nhập vẫn chưa hiện được thông báo khi đăng nhập sai yêu cầu, các phần chức năng bên trong đều còn thiếu nhiều giao diện tải trang vì vậy 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>
* 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>