mingjun wei

2202588F

P01

CTEC REPORT

Contents

[CRUD Functions 3](#_Toc143295505)

[Usage of AWS Lambda 3](#_Toc143295506)

[Users 4](#_Toc143295507)

[Get All Users 5](#_Toc143295508)

[Get Specific User 5](#_Toc143295509)

[Add New User 6](#_Toc143295510)

[Edit User Info 7](#_Toc143295511)

[Delete User 8](#_Toc143295512)

[User Login 9](#_Toc143295513)

[Configuration 10](#_Toc143295514)

[Anime List 11](#_Toc143295515)

[Get All in Anime List 12](#_Toc143295516)

[Get Specific Anime 13](#_Toc143295517)

[Configuration 14](#_Toc143295518)

[AI features 15](#_Toc143295519)

[AWS Translate 15](#_Toc143295520)

[What this is 15](#_Toc143295521)

[How does it work? 15](#_Toc143295522)

[Rationale behind it 15](#_Toc143295523)

[How does it value add to the website? 15](#_Toc143295524)

[Configuration 16](#_Toc143295525)

[Security features 17](#_Toc143295526)

[Password encryption 17](#_Toc143295527)

[What this is 17](#_Toc143295528)

[How does it work? 17](#_Toc143295529)

[Rationale behind it 17](#_Toc143295530)

[How does it make the website more secure? 17](#_Toc143295531)

[Identity and Access Management (IAM) 18](#_Toc143295532)

[What this is 18](#_Toc143295533)

[How does it work? 18](#_Toc143295534)

[Rationale behind it 18](#_Toc143295535)

[How does it make the website more secure? 18](#_Toc143295536)

[Additional AWS functions used (website deployment) 19](#_Toc143295537)

[Amplify 19](#_Toc143295538)

[What this is 19](#_Toc143295539)

[Deployment 19](#_Toc143295540)

[Role on the website 19](#_Toc143295541)

[RDS 20](#_Toc143295542)

[What this is 20](#_Toc143295543)

[Deployment 20](#_Toc143295544)

[Role in the website 20](#_Toc143295545)

[DynamoDB 21](#_Toc143295546)

[What this is 21](#_Toc143295547)

[Deployment 21](#_Toc143295548)

[Role on the website 21](#_Toc143295549)

[S3 bucket 21](#_Toc143295550)

[What this is 21](#_Toc143295551)

[Deployment 21](#_Toc143295552)

[Role on the website 21](#_Toc143295553)

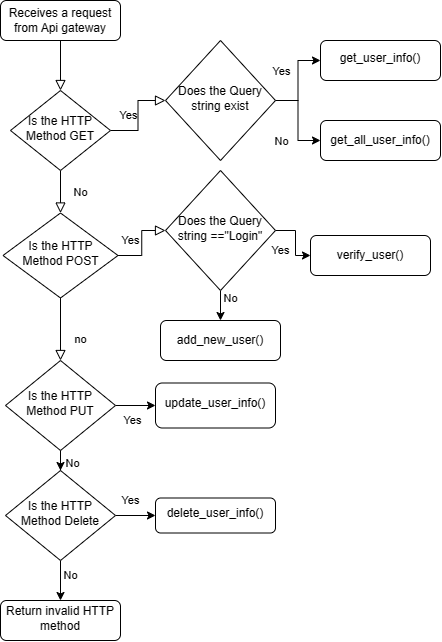
# CRUD Functions

## A logo with white lines on an orange square Description automatically generatedUsage of AWS Lambda

AWS Lambda is a compute service that let users run code without a server. This is vital to a serverless deployment like this project. Lambda is at the core of every serverless deployment. It is used as the connecting layer between a HTTP request and a respond. All AWS service can be activated and used from Lambda. Being one of the most used functions in the whole of AWS. The obvious choice would be to use this for the website’s main form of communication with its backend and that is the choice for this project.

## Users

Above is the code for the function decision tree for the Lambda function of UserDB. It decides what type of request it is by the HTTP Method and uses Query String Parameters for any additional function that use the same HTTP method. Below is the same decision tree but visualised in a flow chart for easier understanding of the process it is bordered in green.

The decision tree has a decision node for every HTTP Method that is being used and when required, Query String Parameters are also used to further differentiate between the functions.

A screen shot of a computer program

Description automatically generatedIf the HTTP request is not in any of the decision nodes, it will be returned with an error message stating that there is something wrong with the request.

### Get All Users

#### What this is

This is the function called when the HTTP Method is GET. And it returns all the existing users within the database back as a table.

#### How does it work?

It works by getting all the rows from the RDS SQL table and for each row. Each row’s column is then assigned a header before it is passed off as a response in the body.

#### Rationale behind it

This function’s use case if not by the end user by rather by an administrator that would like to have the added functionality of getting all the users for easier management, makes exportation of data to a new platform much easier.

### Get Specific User

#### What this is

This is the function called when the HTTP Method is GET and there is a Query string parameter called “head”. It returns the user that have the matching user is that is supplied with the Query string parameter.

#### How does it work?

It works by getting the row where the user id is the same as the one supplied in the Query String Parameter and for that row. Its column is then assigned a header before it is passed off as a response in the body as a table. If the user’s id does not exist in the SQL table, an error 404 is returned with the message “Error, user not found”.

#### Rationale behind it

In the website, this function is called and used when the user displays their profile. By just getting the row that is required, lesser bandwidth is used and that increases the responsiveness of the website.

#### How does it value-add to the website?

It value adds to the website as it allows the user to see their own personal information that they have set and able to adjust it accordingly by just changing the values that need changing. It also value adds to the website by improving the user experience. By increasing the responsiveness of the website, it makes the website easier to use, and for the user to navigate around.

### Add New User

#### A screen shot of a computer program Description automatically generatedWhat this is

This is the function called when the HTTP Method is POST and there is no Query string parameter. It adds the input payload from the website’s request into the SQL table (shown in picture with green border), and also create another table that is called anilist(shown in picture with blue border) It either returns status 200 where the user is added successfully for status 500 where there is an error processing the request.

#### How does it work?

A computer screen shot of a program code

Description automatically generatedIt works by taking the payload’s body and assigning variables to the different values that is supplied by the body. It then inserts a new row into the userINFO table using the values that are supplied by the body.

The code for the paragraph above is shown in picture with green border.

Then the create\_anime\_list is called, after the row is inserted into the SQL table. The code for the function is shown in the picture with blue border.

The function first gets the user\_id from the userINFO table with the email that is passed from the previous function. This is done as the user\_id is automatically given. And since the row was just made, it is not possible to pass the user\_id to the function straight. Thus, the email is passed and then the user\_id is found through the email after the row is made, the email is used as the email is also an unique identifier. The anilist table is then updated and a row’s value is set to the user\_id and the email. The error handling is done through try and except. With each except having a explicit error message indicating where went wrong.

#### Rationale behind it

In the website, some core functions like translate only works with an account as it requires users to have a preference for the translation language. They can also only keep track of the anime’s that they have looked at with an account. Thus creating an account is necessary.

#### How does it value-add to the website?

Having an account to the website makes it easier to deliver the right information to the right users, with users setting their language preferences in the sign up screen, it makes it more seamless for the translating process as users can just see the translations without any input on their end.

### Edit User Info

#### What this is

This is the function called when the HTTP Method is PUT. It finds the user\_id and then updates the SQL table with the new values. It returns status 200 where the user is added successfully for status 500 where there is an error processing the request.

#### How does it work?

A screen shot of a computer program

Description automatically generatedIt works by taking the payload’s body and assigning variables to the different values that is supplied by the body. It then updates the row in userINFO table where the matching user\_id is found using the values that are supplied by the body.

The code for the paragraph above is shown in picture with blue border.

The comparison of values isn’t done in the lambda but is done in the websites script. The script compares the values inputted by the user with the existing information that is saved in the website, if there is a difference, the new user input value will be taken. The script uses if else statements to compare the values. The code for this is shown in the picture with the green border.

#### Rationale behind it

In the website, some core functions like translate only works with an account as it requires users to have a preference for the translation language. This creates a problem, what if the user wants to change the preference for the translation language? Maybe they learnt a new language and want to put it to the test, maybe they set the wrong language in the first place. The same concept goes for other things like the user’s email address, profile picture and password. Thus, having an editable user profile is needed.

#### How does it value-add to the website?

It gives the user more flexibility, as the user now does not have to rely on getting everything for their account right on the first time or even wating to change the settings down the road. It will all be possible. This makes the user experience more user friendly. Makes changing settings easier for the users as they do not have to make a new account just to change the information.

### Delete User

#### A screen shot of a computer Description automatically generatedWhat this is

This is the function called when the HTTP Method is DELETE. It finds the row with the matching user\_id and then deletes the row from the SQL table It returns status 200 where the user is added successfully, for errors it will fall back to the decision tree’s error message.

#### How does it work?

It works by taking the payload’s body and assigns the user\_id to a variable in the lambda function. It then deleted the row in the SQL table where the user\_id is matching using the values that are supplied by the body.

#### Rationale behind it

In the website, although without an account, the users will be losing out on a lot of functionality, Ultimately, it is the user’s decision to stop using the service and leave, and them being able to delete the unused account is an additional layer of privacy protection as it prevents the possibility of a data breach on a service they no longer user to affect them. Not only does this improves the user experience, it also gains the user’s trust as they have confidence that even if they stop using the service, their information can be deleted from the website.

### A screen shot of a computer code Description automatically generatedUser Login

#### What this is

This is the function called when the HTTP Method is POST and the Query string parameter exists. It adds the input payload from the website’s request and verify if the user exists in the database.

#### A computer screen shot of code Description automatically generatedHow does it work?

It works by taking the payload’s body and assigning variables to the different values that is supplied by the body. The values supplied are email and password.

It then selects the password from the SQL table where the email is matching the one supplied.

If the email is not found in any of the rows, an error 404 with the message “user not found” will be returned.

If the password is in correct, a value of “FAKE USER” will be returned denying the user of logging in.

The comparison of the values are done in the Lambda function it first hashes the password passed then compares the hashed password to the password stored inside the SQL table this allows for additional security as the actual password never leaves to the Lambda and is allowed returned to the website only the results are returned to the website

The code for the paragraph above is shown in picture with blue border.

The picture with the green border shows the function that is used to hash the password whenever a password is involved this is used when comparing passwords creating passwords and updating passwords.

#### Rationale behind it

In the website we deemed the log in function as a necessity and thus there needed to be a user verification function it verifies the identity of the user based on the stored information in the SQL table.

#### How does it value-add to the website?

This improves the security of the website as every user’s log in is being verified is insurance that the users are seeing what they're supposed to see with content they are made for them.

### Configuration

A screenshot of a computer

Description automatically generatedA close-up of a computer screen

Description automatically generatedThis Lambda function is used together with API gateway. The Lambda function itself is configured with python 3.8 as the language and it runs on x86. It runs in x86 as it is the most widely used architecture, it has the widest compatibility ensuring that no matter what is used, there is a compatible version of it.

A screenshot of a computer

Description automatically generatedThe execution role is set to the default LabRole that was made; this is the default role that is used in the learner lab account that allows the lambda function to work with other resources effectively.

The API gateway is set to ANY to allow all the HTTP methods. This allows all the requests to go through this one gateway trigger, making the maintenance of the function easier as there will only be one thing that have to be checked through.

## A computer screen shot of text Description automatically generatedAnime List

Beside on the left is the code for the decision tree for the Lambda Function AnimeDBFun. It decides which function to invoke by the HTTP Request Method.

Below boxed in green is the same decision tree but formatted into a flow chart for better visualisation.

The tree first checks if the HTTP Request is valid by comparing if the requested HTTP method is contained in the set array. If it is then the tree continues, if not the tree passes an error.

A screenshot of a computer screen

Description automatically generated

### A screen shot of a computer code Description automatically generatedA screenshot of a computer program Description automatically generatedGet All in Anime List

#### What this is

This is the function called when the HTTP Method is GET, This returns all of the animes inside the anime list, it is returned as a table.

#### How does it work?

The code for the lambda function is shown in the picture bordered by green, it shows that the function returns the entire table after scanning it.

The code for the script for the website is shown in the picture bordered by blue, it shows how the received information is then transformed into a grid with a for loop scanning through every iteration of the result.

#### Rationale behind it

The minimal code on the lambda function side, reduces the latency for processing and in this case it is simply not needed as the expected result is just all of the anime’s inside the database and not a part of it. Thus there is no need for any additional processing apart from displaying it in a presentable manner.

### Get Specific Anime

#### What this is

A screen shot of a computer program

Description automatically generatedThis is a function that displays specific anime that the user have clicked on. Instead of using the Lambda function, it is ran on the website itself using Javascript.

**How does it work?**

This works by getting the specific anime that the user clicked on using the onclick function, it saves the anime into sessionStorage with the key of “selectedAnime”. The code for this is shown in the picture boxed in blue.

This works together with the function displayAnimeDetails, the code for which is boxed in green.

The code shows how the selected anime is stored in sessionStorage and then read from sessionStorage.

#### Rationale behind it

This is done as there is already a copy of the dataset on the website itself, so I felt that there was not a need to put in another HTTP request to get something that the website already have. Thus I just saved the selection to session storage and read from session storage.

#### How does it value-add to the website?

This improves the responsiveness of the website as reading the data locally is much faster than wating for a response from the lambda function. Thus this improves the user experience by speeding up the loading speed of the website.

### A screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedConfiguration

A screenshot of a computer

Description automatically generatedThis Lambda function is used together with API gateway. The Lambda function itself is configured with python 3.8 as the language and it runs on x86. It runs in x86 as it is the most widely used architecture, it has the widest compatibility ensuring that no matter what is used, there is a compatible version of it.

The execution role is set to the default LabRole that was made; this is the default role that is used in the learner lab account that allows the lambda function to work with other resources effectively.

The API gateway is set to ANY to allow all the HTTP methods. This allows all the requests to go through this one gateway trigger, making the maintenance of the function easier as there will only be one thing that have to be checked through.

# AI features

## AWS Translate

### What this is

This is a text translation service that uses advanced machine learning technologies to provide high-quality translation on demand. This is used on the website to translate anime details into the desired language.

### A screen shot of a computer code Description automatically generatedHow does it work?

It works in 2 parts the first is the lambda function on it own, this function handles the HTTP request, it takes the body of the request and turn it into a string and the targeted language based on their different key names.

This information is then passed to the function called trans where the actual translation happens. It automatically detects the source language and translate it into the targeted language. The result of it is then returned as TranslatedText.

The users enter in their preferred result language when they sign up and that can be changed when they edit their profile.

### Rationale behind it

A screenshot of a video game

Description automatically generatedA reason why this website is created is to let anime fans all around the world to come together and have a space for them to discuss and talk about their common interests. By far the largest challenge is the language barrier, how does one ensure that someone from France can understand what a person from Korea is trying to say. The same goes for anime sources, most anime sources are translated into English only, the world’s largest anime dataset provider, My Anime List is only available in English. To ensure that any one around the world can understand and have unrestricted access to information about anime, I have decided to implement this feature.

### A computer screen shot of a code Description automatically generatedHow does it value add to the website?

This basically eliminates the language barrier that someone has when trying to look for information about an anime.

It makes the user experience seamless without discrimination of the user’s region or language that they are familiar with.

Thus, this will entice more users to join the website, making this function a valuable addition to the website.

### Configuration

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

This Lambda function is used together with API gateway. The Lambda function itself is configured with python 3.8 as the language and it runs on x86. It runs in x86 as it is the most widely used architecture, it has the widest compatibility ensuring that no matter what is used, there is a compatible version of it.

The execution role is set to a new role called roleLambda that was made; this a role created by me to allow this specific lambda function to have the specific permission needed to run the function without problem.

The API gateway is set to Post to allow only POST of the HTTP methods. This restricts unwanted access from making use of this service, securing it so that only the rightful users are able to access this function.

# Security features

## Password encryption

A screenshot of a computer

Description automatically generated

### What this is

This is a way of adding security to the website at a relatively low cost. It basically encrypts the password that is saved by the table so that even in the case of a data breach the password is still save.

### A screen shot of a computer code Description automatically generatedHow does it work?

It works by passing the password string through a hashing process. This is shown on the left in the picture boxed in green.

The password string is encoded into bytes and a new Hash object is created and it encodes the bytes inti hashes. The 2nd last line converts the hashed sting into hex that can be read and used.

In the table above, you can see that in the first 2 row the password is not encrypted and can be read easily, however the rest of the row have their passwords encrypted. And even if I have access to the table I do not actually know what their password is.

### Rationale behind it

This is important to the website as it acts as a privacy protection measure for the users. It reduces the risk of identity theft, fraud, and other cybercrimes that can harm the users and damage the reputation of the website. Furthermore, encrypting the passwords also demonstrates that the website cares about the security and trust of its users and follows the best practices in password management. This can increase the user satisfaction and loyalty and attract more potential users to join the website.

### How does it make the website more secure?

Password encryption makes the website more secure by transforming the passwords into data that cannot be converted back to the original passwords. This means that even if someone manages to access the database and steal the hashed passwords, they cannot use them to log in as the users or crack them easily. Hashing also prevents attackers from using common techniques such as dictionary attacks, rainbow table attacks, or brute force attacks to guess the passwords.

## Identity and Access Management (IAM)

### What this is

A screenshot of a web page

Description automatically generatedA screenshot of a phone

Description automatically generatedIAM is a web service that helps you securely control access to AWS resources. It enables management of users, groups, role and permissions. Allowing only a specific role to interact with a specific service decentralize the control flow and make it more resilient to attacks.

### How does it work?

As shown on the right, this IAM works by creating roles and users that have specific permissions. And these services can only use these permissions.

### Rationale behind it

IAM is important for the security and efficiency of AWS resources. By using IAM, I can ensure that only authorized users and services can access the resources they need, and that they have the minimum permissions required to perform their tasks. This reduces the risk of unauthorized access, data breaches, or misuse of resources.

### How does it make the website more secure?

Using IAM it allows for a specific role to interact with a specific service this decentralise the control flow and make it more resilient to attacks as during an attack if only one part or one service is compromised you still ensure that the other services are up and running and are not compromised. Thus, making the website more secure by making it more resilient to attacks.

# Additional AWS functions used (website deployment)

## Amplify

### What this is

For this project I am using Amplify hosting. Amplify Hosting provides a git-based workflow for hosting full-stack serverless web apps with continuous deployment.

### Deployment

1. number one users click on host app
2. Users choose GitHub
3. A screenshot of a computer

   Description automatically generatedAdd repository branch to the already uploaded source code
4. Users den choose the brunch that the source code is in
5. The app is ready to be deployed

### A screenshot of a computer Description automatically generatedRole on the website

This hosts the website and is an alternative to s3 bucket hosting websites. This is simpler than s3 hosting because there is not a need to re upload the files every single time however the users need to be competent in GitHub knowledge and version control in order to not overwrite or delete data

## RDS

### What this is

RDS is a distributed relational database service that runs in the cloud this is a SQL method to store and manage data it also helps in relational database management where tables in the same database are linked to each other

### A screenshot of a computer Description automatically generatedDeployment

1. click on create database
2. Select standard create
3. Under engine options pick my SQL
4. Under templates pick development slash test
5. Under availability and durability pick single DB instance as we are trying to save money
6. auto generate a password to ensure security
7. allocate enough storage in this case 20 GB will be enough
8. Under connectivity select do not connect to a easy to compute resource
9. A screenshot of a computer

   Description automatically generatedA screenshot of a computer screen

   Description automatically generatedturn on monitoring

### Role in the website

this is used to store the data of the user information as a table this is chosen because user data is usually a fixed set of columns that are not really changing an RDS makes it easy for interrelated tables for example the user information table is actually linked to the user list table although the user list table is not used in this project.

## DynamoDB

### What this is

Animal DP is a fully managed no SQL database that provides fast and reliable performance with seamless scalability in the scope of the project this is used to store anime data.

### A screenshot of a computer Description automatically generatedDeployment

1. Create a new table
2. Under table name put in an anime DB
3. Partition key will be anime ID
4. Leave as Default table settings

### Role on the website

this is used to start the data off the anime as a table this is chosen because the data for anime can change from time to time and the columns should be able to adjust freely without changing to previous the values tasks are no SQL deployment is the better approach and with data sets that are very large like anime collection Dynamo DB can actually be faster improving the websites responsibility.

## S3 bucket

### What this is

A screenshot of a computer

Description automatically generatedThis is a simple storage service that can be used to store any data in our scenario we are storing the user profiles profile picture this has high data reliability and availability which ensures that the data that we store inside will not be lost.

### Deployment

1. Click on create bucket
2. Name the bucket user profile pic
3. Enable ACL this allows the Lambda function to assess create and modify or even delete existing files
4. Disable block all public access
5. disable bucket versioning

### A screenshot of a bucket policy Description automatically generatedRole on the website

The S3 bucket is used to host the user's profile pictures the user's profile pictures are hosted here for easy access as the pictures can be saved inside the SQL table as a link and then the link can be shown on the website.