# Implementation of Security Infrastructure on the Cloud

Date of Submission: 04-Aug-2021

Submitted By:

## SOI-2021-0031

| | |
|---|---|
| 190***** | MUHD WAFIYUDDIN BIN ABDUL RAHMAN |
| 190***** | QAMARRUL DANIAL B RAZALI |
| 190***** | NG ZI RUI |

School of Infocomm (SOI)
Republic Polytechnic

# - ACKNOWLEDGEMENTS

We would like to give special thanks to our supervisor Ms. Pauline and our evaluators, Mr. Zack and Mr. James for guiding us and giving the most support to ensure that we have successfully executed the project smoothly. With the help we have gotten, we have configured the infrastructure with fewer errors and if there is an error, we manage to troubleshoot together as a team.

Within the team, we would like to thank one another for supporting all of our works and always ensure that we work together to make this project a successful one. With communication, distributing the work evenly and many more, have made us learn more about this project and learning how to troubleshoot the various services such as phpmyadmin, Apache, and FTP server, instances, and aspects of AWS services. We could not proceed or execute the project smoothly without having each other back and supporting and helping one another to solve any problem in the project.

# TABLE OF CONTENTS

# - ABSTRACT

The project is about implementing security features inside the cloud infrastructure. A website and database are built from scratch before hosting on the cloud. Security testing is conducted to identify the vulnerabilities on the web and database server before mitigating security features. Some enhancements made include website codes with special security parameters, within the database and having security inside the cloud infrastructure using AWS Educate service. 2-factor authentication is also used to ensure changing of sessions constantly to prevent unauthorised access. After implementation, a monitoring system is also created to find abnormalities in the system for better troubleshooting purposes. With all criteria fulfilled, the project will ensure secure website browsing among users (website is HTTPS based using Cloudflare) based on the security features implemented. The final status of the project has been marked as completed and all of the security and AWS Cloud services are being used.

# 1 Introduction

Our project is called Qaziwa Project. It is a project which involves the Implementation of Security Infrastructure on the Cloud which aims to host a website onto the internet using AWS Educate account and implement some security features to ensure that the website and other services are secure.

Nowadays many companies have adapted infrastructure to the cloud to ensure the flexibility and the ease to deploy services onto the cloud to meet the agility of the business. The problem we want to solve is that many companies are still using physical servers that are hosted on site and this can cause a big amount of investment on the hardware, need a space or/and room to implement the servers on site, and may be susceptible to data loss during disaster situations such as earthquakes. As such cloud implementation is needed to resolve these types of situations. Like any other on-site server, cloud servers also need to implement some security features to ensure that to prevent attackers from attacking and exfiltrating the data from the cloud.

We are required to design and deploy cloud infrastructure to achieve similarities to the traditional 3 tier web application (Web Tier & Database Tier) using AWS cloud. Auto-scaling and other security features are important in this deployment and development.

# 2 Project Specification and Plan

Our project requires us to create a 2 tier Web-App and Database and have hosted it onto an AWS cloud VPC ensuring that it is facing the internet and the database server residing within the internal infrastructure. Other than that, we also need to create a Virtual Private Network to host these two services and use the AWS features such as Cloudwatch, IAM roles, and others to fully utilize the AWS services that they have offered to us.

For the creation of the Web application, The team has decided to go for an online book store that uses a SQL database to store and retrieve the necessary data. The database would store data such as username and password for the password, data of the books in the listing, individual orders of each user, etc.

The web and AWS platforms would be both configured with security implementations to prevent any vulnerable attacks. These security implementations would be based on security testing that would be done beforehand on a non-hardened version of the website, any vulnerabilities that could be exploited are then compiled into a list and become a checklist of issues that should be mitigated. As for the AWS cloud, due to the inability to perform attacks without getting our account disabled, we would perform our security configurations with reference to normal traffic attacks that attackers use and perform standard hardening configurations.

## 2.1 Project Overview

Due to the fast-paced technological world, more and more companies and organizations have already adapted their servers via the cloud. As such this project helps us to identify why hosting servers via the cloud is better than hosting servers on-site. Hosting servers on the cloud helps us to identify the possible future integration of on-site servers and cloud servers. In addition, using the cloud to host servers such as web servers and database servers provide more advantage and at a cheaper cost. With all of the features that AWS Cloud Computing has to offer, it provides countless API, Services, Products in which it is much easier to integrate with our own servers. By doing this project, we can upgrade ourselves and learn how IT administrators, programmers, software developers, and others host their own services on the cloud.

This project is trying to accomplish the integration of web service and AWS cloud together. This helps not hosting our website, but also helps in monitoring, adding in security features that help to protect the web server from any attacks or hackers. The project ensures that we implement defense-in-depth by using the AWS security features, our own WAF configuration in the instance, and even into our source code itself together with using the cloud features to ensure uptime, scalability, availability, and most important enhanced security features.

In this project, we have few major deliverables that we need to cover from the starting of the project to the end of the project. Those deliverables ensure that we utilize and provide security to the user and prevent any downtime of the server (scalability). The deliverables are shown below:

- Deploy a Virtual Private Cloud (VPC);
- Design and deploy a 2 tiers architecture on Cloud (Web-App Tier and DB Tier);
- Web-App tier uses apache web server and PHP application while Database tier uses MySQL. Use XAMPP software for the various tiers.
- Encrypt data in transit (using HTTPS) and data at rest;
- Set up the necessary security groups and IAM groups/users/roles;
- Deploy auto-scaling for Web-App tier;
- Develop a simple web service where read and write to the database are applicable;
- Backup and restoration of MySQL database using manual snapshot for Elastic Block Store (EBS); Store the EBS snapshot on AWS Simple Storage Service (S3).
- Use AWS CloudTrail to monitor activities in the VPC.

With all of the targeted deliverables, we can have a rough idea of how we are going to implement and deploy our web service and database server and even our monitoring of the traffic and the instance too.

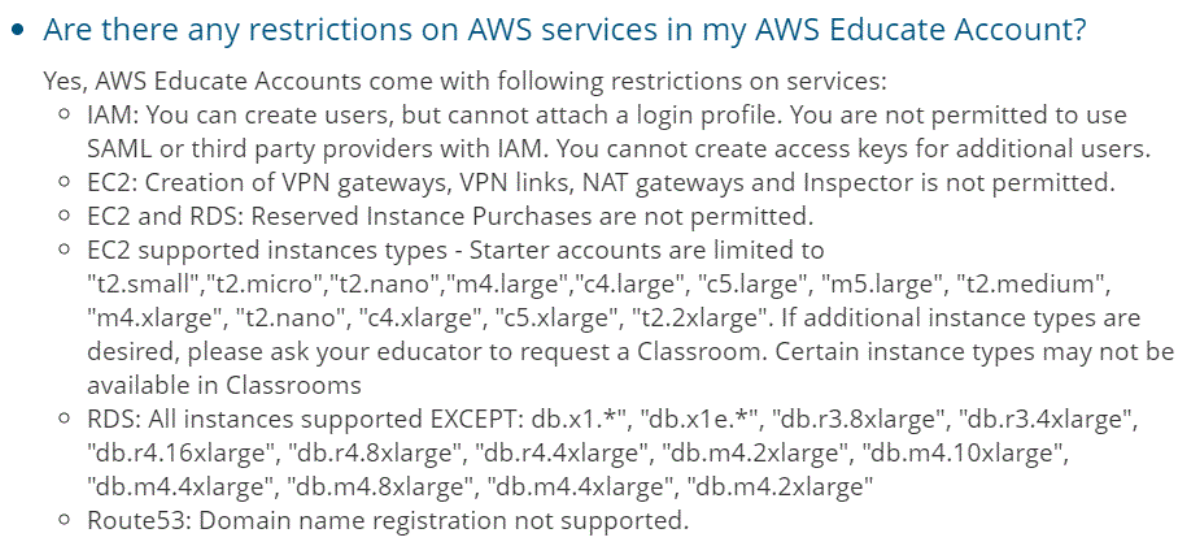
Skills Required:

1. AWS Cloud Services: - EC2 Instances - S3 Buckets - EBS Volumes - KMS -IAM - VPC - Route 53 - RDS - Security Hub - etc.

2. Networking
3. AWS CLI

The assumption of the deliverables and the project is that we need to distribute the workload and ensure that each one of us is confident in handling the workload. In addition, in order to make the end deliverable work, we need to follow step by step and follow the requirements of the deliverable before we can do something extra. The assumption that we have to start off doing the project is that we need to create a website and test it and after testing the website, we can prepare to integrate with the cloud services which is called an instance so that we can host our website from there. Firstly we need to think of the network infrastructure of the AWS network diagram and from there we can know how the configuration and the traffic will be about. Hence from there, we can start configuring all of the networks such as the VPC, subnets, and others before deploying the instance and hosting it on the internet. After hosting the web service and database, we then need to use auto-scaling and load balancer to ensure it provides scalability and availability to the users and adding some security features such as encryption of data at rest and data in transit (HTTPS). After all of that, we then need to set up cloud monitoring to monitor the traffic, CPU of the instance, and much more so that we can understand the traffic flow and analyze it. All of this must include some security so that the attacker will not have their own advantage and hack the web server or the database instance. All of the deliverables that are given to us are expected to finish and if there is quite some time, we can add some implementation to enhance the website and its security. This is our assumption that we feel it will make our end result work and meet all of the needed requirements.

Since we are using an AWS education account, there will be limited features in which we do not have the privileges or authority to proceed with the limited features. The limited features are shown below.



- Are there any restrictions on AWS services in my AWS Educate Account?

  Yes, AWS Educate Accounts come with following restrictions on services:
  - IAM: You can create users, but cannot attach a login profile. You are not permitted to use SAML or third party providers with IAM. You cannot create access keys for additional users.
  - EC2: Creation of VPN gateways, VPN links, NAT gateways and Inspector is not permitted.
  - EC2 and RDS: Reserved Instance Purchases are not permitted.
  - EC2 supported instances types - Starter accounts are limited to "t2.small","t2.micro","t2.nano","m4.large","c4.large", "c5.large", "m5.large", "t2.medium", "m4.xlarge", "t2.nano", "c4.xlarge", "c5.xlarge", "t2.2xlarge". If additional instance types are desired, please ask your educator to request a Classroom. Certain instance types may not be available in Classrooms
  - RDS: All instances supported EXCEPT: db.x1.*", "db.x1e.*", "db.r3.8xlarge", "db.r3.4xlarge", "db.r4.16xlarge", "db.r4.8xlarge", "db.r4.4xlarge", "db.m4.2xlarge", "db.m4.10xlarge", "db.m4.4xlarge", "db.m4.8xlarge", "db.m4.4xlarge", "db.m4.2xlarge"
  - Route53: Domain name registration not supported.

And here are the supported services that AWS education accounts have to offer.

<Insert the PDF after download><AWS_Educate_Starter_Account_Services_Supported.pdf>

As such we need to work our way around and try to work in what we have/what AWS education has for us so that we can set up our web service and the database with respect to using the AWS services too. Some of the limitations that will affect our deliverables are Security Hub, Certificate Manager (Import certificate to AWS Load Balancer) and IAM role.

## 2.2 Functional Requirements

The functions needed to have successfully been implemented in which services are being provided to the users to buy the books and even view books without any services being interrupted. The major function is providing goods and services of the user accessing the services that we provided using AWS Educate.

## 2.3 Project Plan

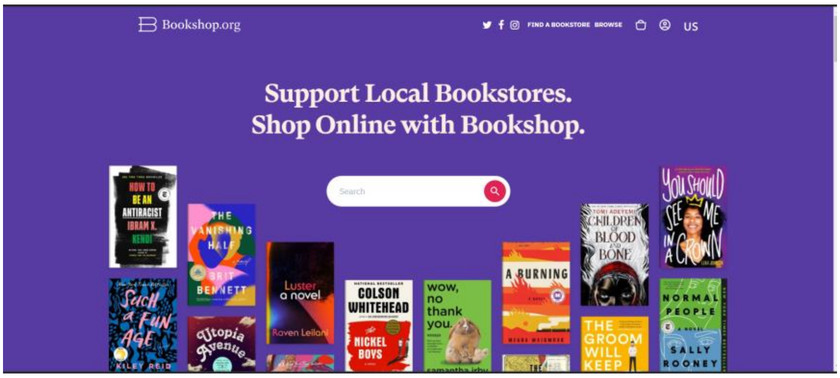
### 2.3.1 Project Details:

Students are required to design and deploy cloud infrastructure to achieve similarities to traditional 3 tier web applications i.e. web tier, application tier, and database tier. Students will be working using the AWS cloud. Auto-scaling and security of the application are important deployment factors.

Our starting of the project is to think and brainstorm for the website that we want to create. We start with the objective and the functionality of the website and how it can serve and help users who access the website. After brainstorming, we have come up with an idea for making a website that sells books and this is where the user can use it to buy books. Firstly we go to different websites that are available online and see the functionality, the books they are selling, and much more. The picture below shows the website that we have narrowed down:

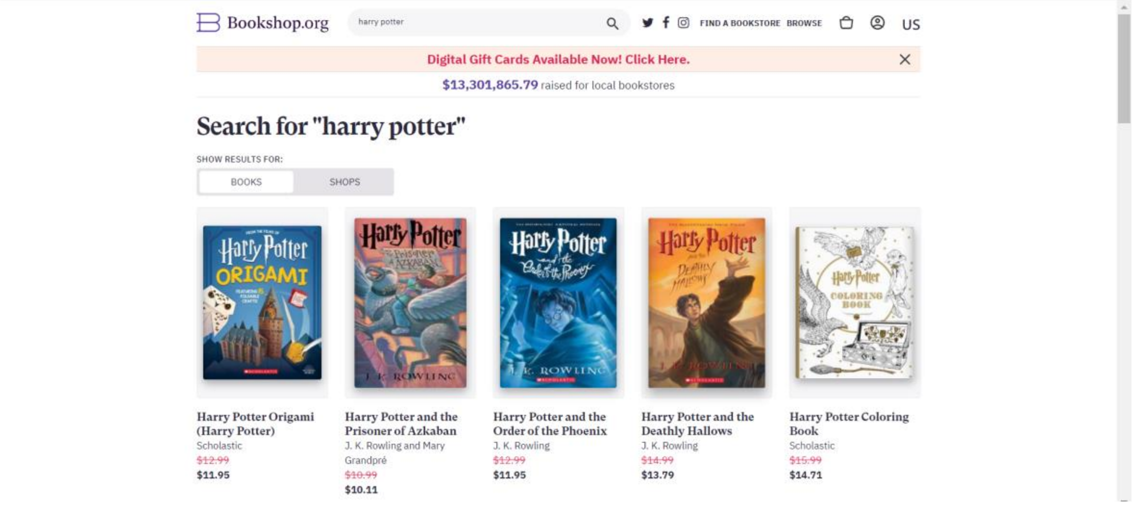
## 2.3.2 Design Planning:

Website main page view:



Support Local Bookstores.
Shop Online with Bookshop.

A website with books selling:



Digital Gift Cards Available Now! Click Here.

$13,301,865.79 raised for local bookstores

Search for "harry potter"

SHOW RESULTS FOR:

Harry Potter Origami (Harry Potter)
Scholastic
~~$12.99~~
$11.95

Harry Potter and the Prisoner of Azkaban
J. K. Rowling and Mary Grandpré
~~$10.99~~
$10.11

Harry Potter and the Order of the Phoenix
J. K. Rowling
~~$12.99~~
$11.95

Harry Potter and the Deathly Hallows
J. K. Rowling
~~$14.99~~
$13.79

Harry Potter Coloring Book
Scholastic
~~$15.99~~
$14.71

Within these first 3 weeks of the FYP timeline, it will be used for planning and therefore the project plan within this time period is subjected to a lot of change.

We also have implemented a Gantt Chart to see the time flow and planning of this project.

<Insert gantt chart here> <Project Planning Gantt Chart.mpp>

Task allocation (Web Server)

| No | Wafi | Zi Rui | Qamarrul |
|---|---|---|---|
| 1. | Website Planning | Website Planning | Database Planning |
| 2. | Finalised website design | Use-case diagram | ERD diagram |
| 3. | Listing Page | Login Page | Database Structure |
| 4. | Cart Page | Checkout page | Database Creation |
| 5. | Google authenticator | Registration Page | Table creation |
| 6. |  | Confirmed Payment Page |  |

Task allocation (Amazon Web Service/ Security Hardening)

| No | Wafi | Zi Rui | Qamarrul |
|---|---|---|---|
| 1. | Route Table | Database Implementation | Web Server Implementation |
| 2. | Subnet Creation | NAT Implementation | Snapshot Instances |
| 3. | Auto Scaling | Security Group | S3 bucket |
| 4. | Domain Name | Elastic IP | Load Balancing |
| 5. | Cloudflare | KMS | IAM Role |
| 6. | FTP Server | CloudTrail | CloudWatch |
| 7. | Internet Gateway | VPC (Virtual Private Cloud) Creation | Database (Rest) Encryption |

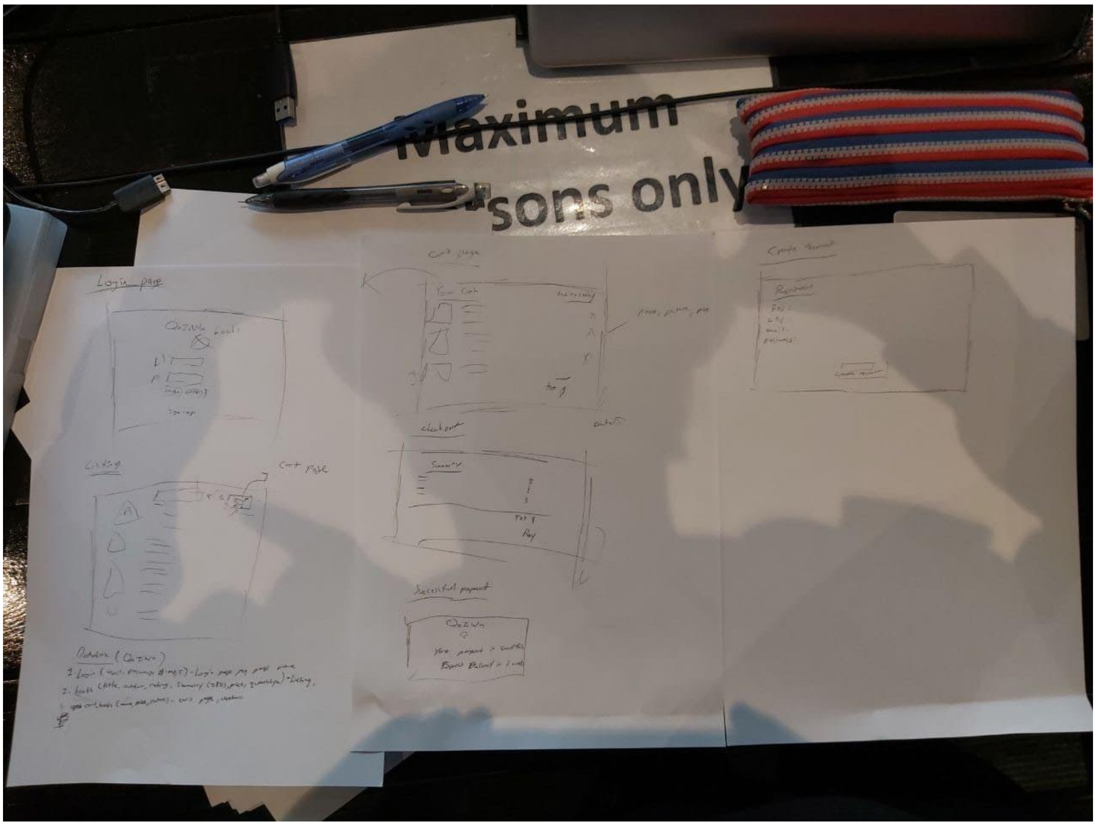
After searching on what a book website needs, we have created a list of the pages that are needed on the website.

1. Login page
   → Users would enter their login credentials
   → Having the admin functions
2. Listing page
   → Display all the books available & price & rating
   → For pictures save them into one folder and target it from there

→ List the inventory available smallest number has to be 0

3. Add to cart features  
   → Quantity left  
   → Book cost  
   → Total cost bottom of the page  
   → Payment type (nets, Mastercard, etc.)  
   → Having some sort of validation should not have values that are out of reality
4. Checkout page  
   → Payment is not of concern
5. Registration page  
   → Register users so that they can buy books

After knowing what pages are needed in the online book store, we now need to design how our website will look to the user. We have made a draft of the website design using pencil and paper.



This is not the final design, but the team has in mind a rough base and idea of how our website will look.

Since we have a rough idea of the website and the pages needed, now we need to plan on how our database will look. For the website, there will be users, display books, and add to cart features. So we have created a list of the tables that are needed for our websites.

1.User table

→ For all users that are within the website.

→ Normal user: First name, Last name, email & password

→ Guest user: No password or privileges

→ Admin user: Admin username and password

→ All of this will have their own unique primary key which we are going to call user_id

2.books table

→ For all books that are available to be sold

→ Title, author, summary, rating, price, quantity, pictures

→ A unique primary key ID which we are going to call title_id

3.cart_books table

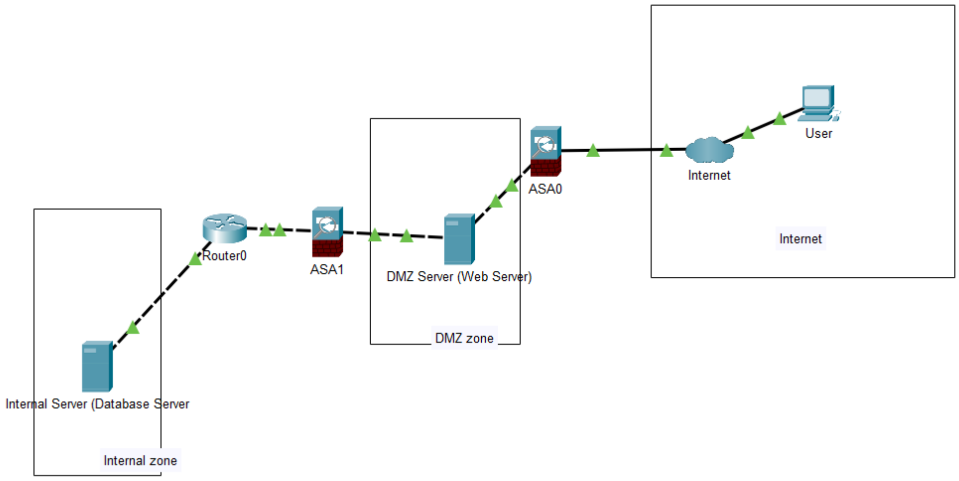
→ For users who have selected their books and proceed for payment.

→ Cart_title, cart_price, cart_picture

→ With a unique primary and foreign key

- Primary key: cart_id
- Foreign key: title_id & user_id

Since we know what to put for the database and web server, now we need to ensure that both database and web server should not be in the same instance. This is because if the web server is compromised, this can cause the database server to be compromised too. Therefore, to prevent this from happening, we need to ensure that the database server and the web server need to be in separate instances. Below is a network diagram of what our web and database server will be on.



As shown above, we are also planning to put a firewall to ensure that the security is being placed in. All of the configurations will be shown in the next few sections. The internet is able to access the web server and the web server can access the database server but the internet cannot access the database server which is the first step in planning the security features. This is our first planning on how our infrastructure will look before a security testing and configuration is being placed. This will then be implemented into the AWS and the network diagram will be changed based on the AWS network diagram. The above network infrastructure is the base of our execution and the actual implementation of the services in the AWS cloud.

For the planning software we will be using Xampp as a base of Web Service (Contain Apache & Database) → To be implemented in our localhost and if there is no other way, implement in AWS instances. To code will be transferred remotely.

### 2.3.3 Implementation planning

Since we know that we will be using the AWS cloud, an instance is needed in which to only access the instance via SSH, and putty software is the most suitable to be implemented to configure the instance.

After designing the website, implementation of the source code into the web instances will be initiated. By doing this we require a puTTY in which we SSH into the instance. Before SSH, ensure that the instance is deployed and ready to configure. After SSH install the apache server and test it.

For the database section, we have planned to use phpmyadmin, and since it is internal, a NAT is needed to ensure network connectivity to the internet. So we will be doing the same, SSH into the web instance and then using the web instance ssh to the database instance and from there start downloading the phpmyadmin. Before SSH, ensure that the instance is online and available and has connectivity between the web server and the internet.

Configuration on the AWS is what we need to ensure high availability by implementing auto-scaling and load balancer. In addition, we also need a VPC that includes subnets and an internet gateway to make the database instance and web instance to be in different zones. In addition, implementation of the data security at rest and transmit, adding permission in Group, user policies, and much more.

### 2.3.4 Test planning

The testing that we have planned to implement and then test. This means that after we have finished the website, we will be testing it in terms of its security, user validation, and much more. So, for every step, we will ensure testing of the services, project and much more will be added in. We have come out with a few of the testing before we can proceed to the next implementation. If the testing is successful, we can proceed and if it is not, we need to troubleshoot, mitigate and explore the possible ways to ensure that the testing is successful. Here is the testing that we have planned to do on our project implementation:

1. Security testing (e.g. Dos attack, Msfvenom, Metasploit, etc)
2. Connection testing (e.g. Ping, SSH)
3. Functional testing (e.g. Successful user login, add to cart, etc)
4. Load testing (e.g. DoS attack and ensure availability)
5. Performance testing (e.g. Fast uptime and reduce the amount of downtime)
6. Browser Compatibility testing (e.g. Able to work on IE, Chrome, Firefox, Puffin, etc)
7. Vulnerable testing (e.g. Nmap, Nessus, etc)

All of the testings will be done manually as a team and ensure that each of the tests is successful before we can proceed to the next one. Most of the testing is not necessarily needed to follow the baseline testing. All of the testings are as follows for each configuration that we did.

### 2.3.5 Security planning

Insecurity planning, we have planned to ensure that every security we test on the web server and the website, will ensure it will work and the hardening is being implemented. After we have developed the website and implemented it on the web server, there is a need to test the security to ensure that there are no exploits or vulnerabilities that could harm the webserver. As such, the security/attacking software we will be using to conduct an attack in a safe environment are listed below.

- Metasploit & MSFvenom
- Nmap
- Burp Suite

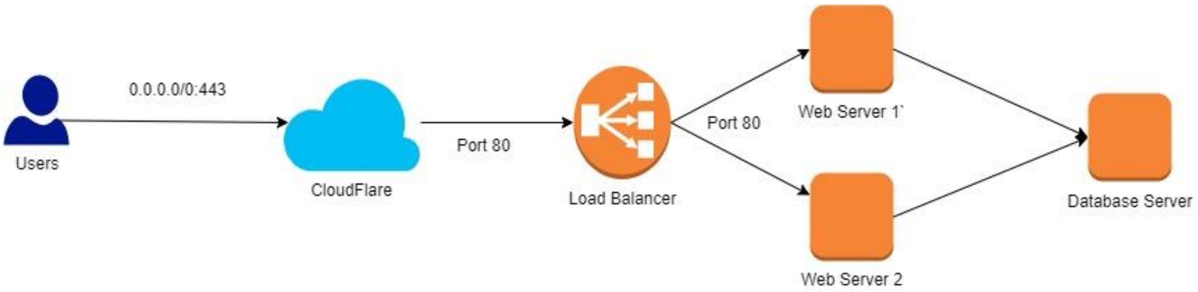
- OWASP
- Nessus scanning
- SQLmap
- Website attacks (RFI, LFI, Command Injection, etc)
- Pivoting
- Webshell (PHP script, shellshock)
- Invalid user validation
- Man-In-The-Middle attack

These are the few attacks we have planned to attack the server and from there we can narrow down if the server or the website is deemed vulnerable or not. We can use various attack software to conduct the attack and also we can implement attack techniques to see if there is a vulnerability within the server. After the designated attack we can also research and implement the security fixes which will help to ensure a more secure environment for the users accessing the website, and also for the servers in the cloud. After a security fix, we need to test its function and another testing so that we can ensure there are no loopholes in our services. All of the attacks will be recorded and the mitigation on how we fixed the vulnerabilities.

**-**

# - 3 System Design and Implementation

The main components used for our project are Qaziwa's website, a database that stores all books, and cloud infrastructure using AWS education. Since the final objective of our project is securing cloud infrastructure, we will need to create our own website and database before hosting it into the cloud. So to start off we have created a simple network diagram to show how the user will access our website.



From the above, the user has the ability to access Cloudflare via port 443. Cloudflare will help to ensure that the user will prevent DoS/DDoS attacks and give SSL encryption from the user to the Cloudflare service. But from Cloudflare, it will be on port 80 to the Load Balancer to the web server and port 3306 to the database server. Therefore, a configuration that accepts the IP address from the Cloudflare so that the user will not be able to access the load balancer directly and ensures that the Web Server accepts IP addresses from the load balancer. So this is to ensure that the user cannot be funny and access the load balancer and even the instance itself. Cloudflare offers a variety of protections such as SQL injection, Identity theft, DoS/DDoS protection and It even helps to improve the site performance and speed uploading time.

Since we have a rough idea of what configuration is needed, we have listed the summary of the configuration and implementation to our project.

1.Creation of the website

- Login Page, Register Page, Logoff Page, Listing Page (Guest), Listing Page (Registered User), Listing Page (Administrator), Shopping Cart Page, Payment Page, Successful Payment, Advertisement (Cart Page) and Verification Page

2.Creation of database tables

- Books, Users and Cart_Books tables

3.Creation of cloud network

- VPC, Subnets, Route Table (NAT instance & Internet Gateway) and Internet Gateway.

4.Creation of cloud and its available services.

- S3 Bucket, Security Group, Elastic IP, Instances (Web, Database, FTP & NAT), Instance Snapshot, Application Load Balancing, Target Groups, Auto Scaling, Amazon Machine Image, Launch Configuration in Auto Scaling, Volume & Key Pairs

5.Creation of Domain name

- Freenom Domain

6.Creation of CloudFlare

- Cloudflare SSL Certificate Configuration, Allowing Load Balancer to accept traffic from Cloudflare, Allowing Instance to accept traffic from Load Balancer, Implementation of SSL Certificate involving instance

7.Database Encryption

- Innodb encryption, Volume encryption

8.Creation of Monitoring system, Privileges and Encryption in AWS

- KMS, CloudWatch, CloudTrail, IAM

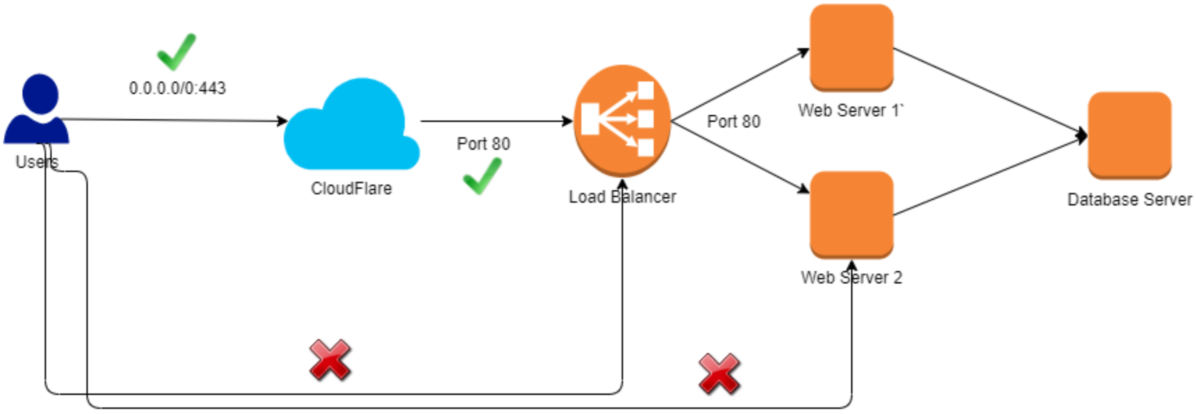
For full detailed steps and configuration, do refer to the document attachment below:
<Insert the Project Documentation file><Project Documentation.docx>

Since we have configured all of the services in AWS and available free website services, we now have an idea of what to test to check if everything runs smoothly before the end of the project.

3.1 System architecture

1.Cloudflare diagram

In our project, we have implemented a cloudflare to help in preventing DoS attacks and other security implementations that cloudflare has to offer. Cloudflare is the first line of defense against any attack in which it is important to ensure that it provides the defense against any web attacks and others before going through the second layer of defense.



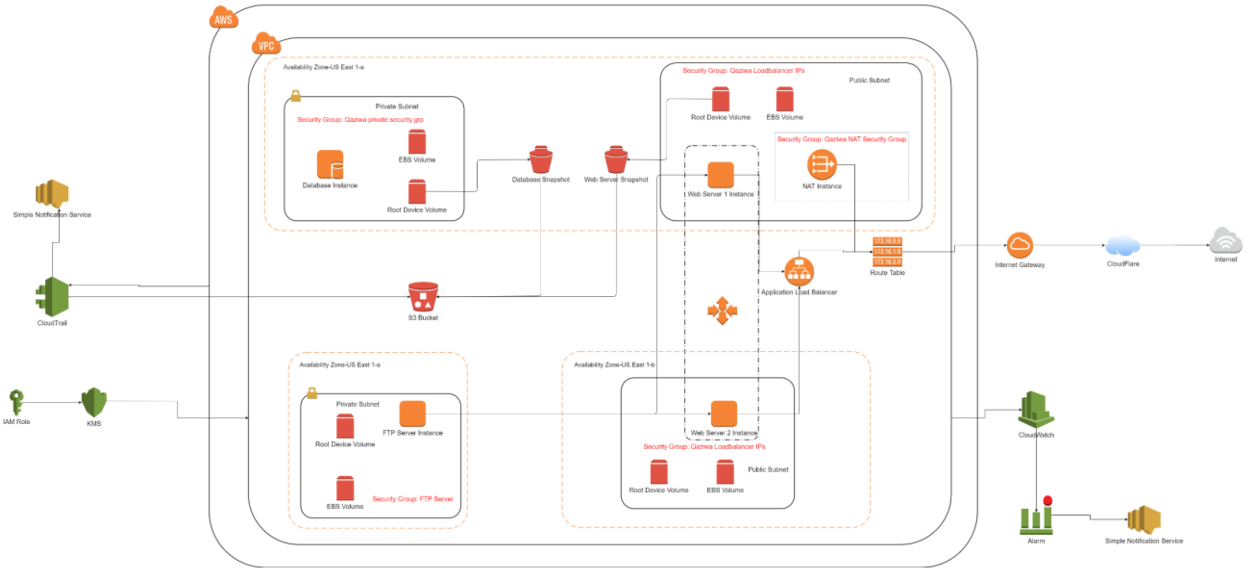
In the above diagram, we can see that the user is able to access the website via passing through the cloudflare to the LB and to the instance. But in a situation where the user wants to skip the cloudflare security to access the LB or the instance, it will not allow the user to do that. No matter what the user needs to access CloudFlare.

Situation:

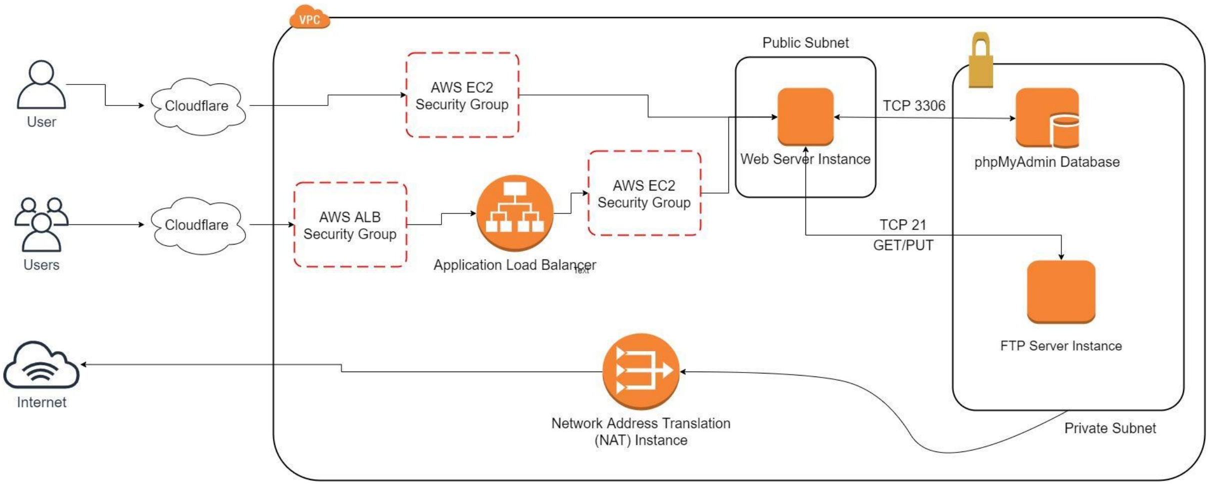
1. User >> CloudFlare >> LB >> Web server >> Website (Access Granted)
2. User >> LB >> Web server >> Website (Access Denied)
3. User >> Web Server >> Website (Access Denied)

The user is unable to access the database because it is in the private network.

## 2. AWS infrastructure diagram



## 3. User Access Structure



In the diagram above, we can see that the Web Server is connected to the FTP server and the database server instance in which for FTP server, the web server will get the images files from it and display the images in the website, and for the database server, it will display all of the records that we have stored in the 2-tier website.

In addition, both database and FTP server are in the internal network in which a NAT instance will help them to connect to the internet but the user on the internet will not be able to access the FTP and database instances.

So when the user wants to connect to the Web server, they need to access the cloudflare in which it is the first layer of protection. We have configured the security group in such a way that to access the website, the user needs to pass through the cloudflare and they cannot skip cloudflare security.

Scenario 1:

If there is one user who wants to access the instance, it will be loaded to the instance and from there the web server will connect to the FTP server instance and the Database instance.

Scenario 2:

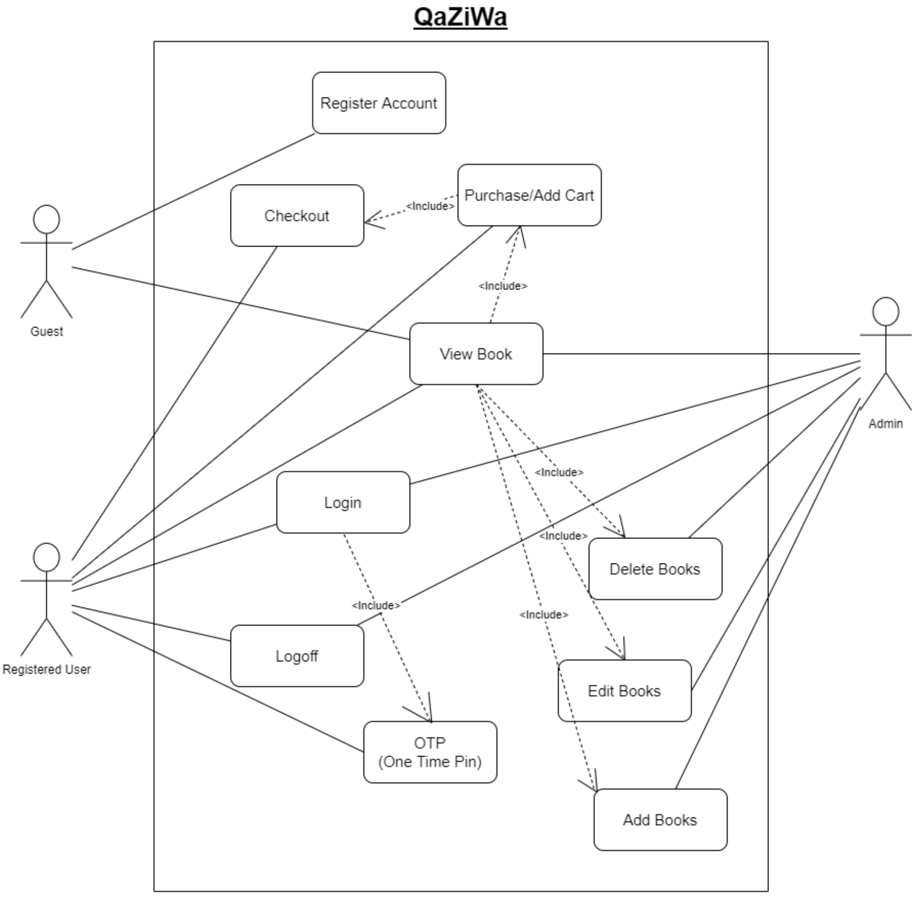
If there is a group of users who want to access the website, it will be handled by the ALB and from there the user will access the web server and retrieve the file from the FTP server and connect to the database instance.

## 2.4 Detailed System Design

Here we have put all of our system design for the project that we have made during the process of creating the project and implementing the features. We have the web case diagram, AWS infrastructure diagram, ERD diagram and many more shown below.

### 1.Use-Case Diagram

We will create a Use-Case Diagram to display the relationships between the different users using our website. There are mainly 3 types of users - guest, registered user and admin.



Admin → Have the full privilege to add, viewbooks, delete and edit books. (Most access)

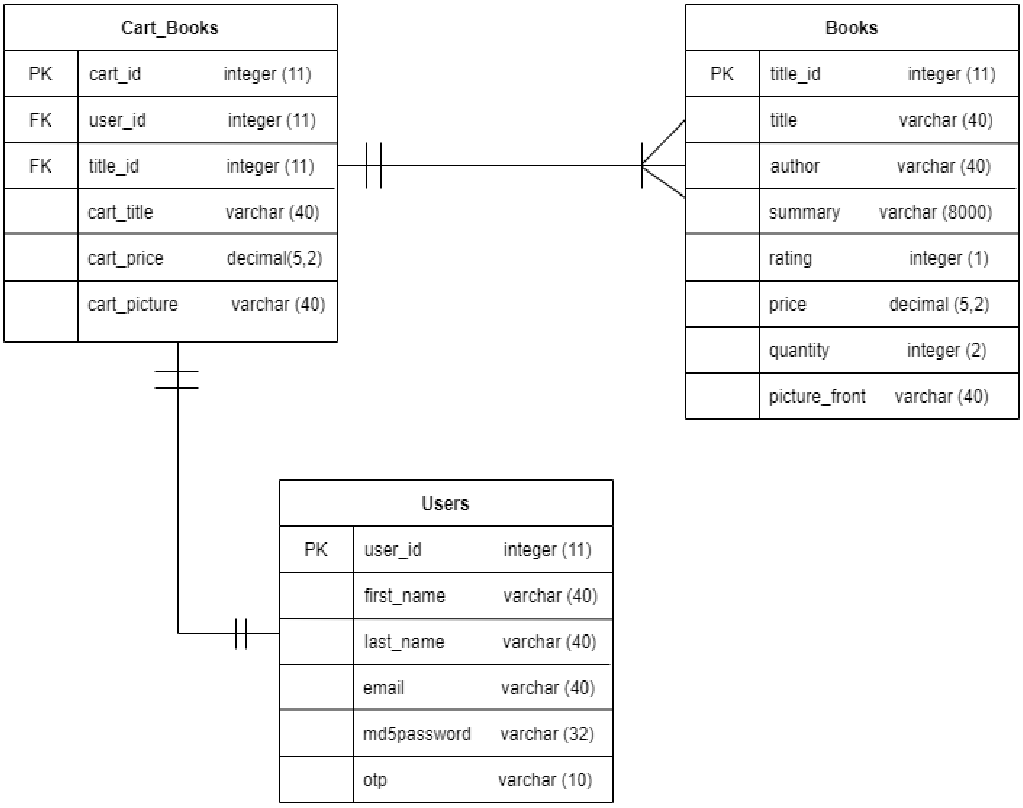
Registered users → Have few privileges to view books and add to the cart and the payment

Guest → Have the least amount of privileges to only view books. (Least access) → Can become a registered user

## 2. ERD Diagram

For our database, we have created 3 tables. The correlation between each table will be shown using an Entity Relationship Diagram (ERD) diagram.

The table “books” contains all books available where the parameter title_id is a unique identity (primary key). The table “cart_books” contains the books inside the shopping cart. A cart will be available for a user and vice versa. Once the purchase has been made and orders have been received, the cart will automatically be empty for a new set of transactions. The table “users” represents a registered user. Their credentials (first name, last name, email address, password) that were imputed during the creation of an account will be stored under this table. The password will be stored using the MD5 hash.



| Cart_Books | | |
|---|---|---|
| PK | cart_id | integer (11) |
| FK | user_id | integer (11) |
| FK | title_id | integer (11) |
| | cart_title | varchar (40) |
| | cart_price | decimal(5,2) |
| | cart_picture | varchar (40) |

| Books | | |
|---|---|---|
| PK | title_id | integer (11) |
| | title | varchar (40) |
| | author | varchar (40) |
| | summary | varchar (8000) |
| | rating | integer (1) |
| | price | decimal (5,2) |
| | quantity | integer (2) |
| | picture_front | varchar (40) |

| Users | | |
|---|---|---|
| PK | user_id | integer (11) |
| | first_name | varchar (40) |
| | last_name | varchar (40) |
| | email | varchar (40) |
| | md5password | varchar (32) |
| | otp | varchar (10) |

3 Tables (users, cart_books, books)

Relationship:

1 user → 1 cart, 1 cart → 1 user

1 cart → many books, many books → 1 cart to store

# 3 System Testing

All of the testing will be shown here. This System testing will cover various types of testing to ensure that we meet all of the testing requirements before we fully make it into production.

**Security testing**

As part of managing the website that it serves in the AWS cloud, we need to ensure that all of the vulnerabilities have been tested and fixed so that it reduces the chances of the attack and also provides services to the user's needs. As such before we deploy onto the internet using the Ubuntu web server, we need to test in an environment in which it is secure and also controlled. As such we have implemented a Kali Linux VM & A Ubuntu Web Server VM which is running in a controlled environment. All of the security and updates will be implemented in the environment in which it then will test again to see if the vulnerability is being fixed and ensure the website/database is working fine. The setup of the environment is shown below:

1. Kali Linux (Attacker >> Metasploit, Burp Suite and other attacking software has already been installed)
2. Ubuntu (Web Server & Database server >> Using apache2 [Web Server] & Phpmyadmin[Database server - Localhost])
3. Within the same network of 192.168.1.0/24 connected using Host-only as our network adapter.

As such below will cover the security tests that were performed on the QaZiWa website and other attacks which involve threats and vulnerability with our infrastructure. The attacks carried out would be recorded down here with screenshots to give a clear reference on the improvements and fixes needed to be implemented on the website. For this case, we have a web browser and other servers or end users that are vulnerable and post threats so therefore, a set up in our environment with multiple vulnerabilities to test and ensure that it is being fixed and how it can be fixed.

**Summary of all the attacks/implementation done:**

| Web Application Attacks | | |
|---|---|---|
| **Wafi** | **Zi Rui** | **Qamarrul** |

| | | |
|---|---|---|
| Session Hijacking | Unnecessary HTTP request | Brute Force Attack |
| Session Fixation | Trace HTTP Request | Kill All Session |
| SQL Injection | Command Injection Attack | Improper validation upload section |
| Cross-site Scripting | Separate Database and web server interfaces | File inclusion attack |
| DoS & DDoS | Exposed PHP Version | Showing of Apache Version and OS Identity |
| Invalid or Insufficient Condition attack | Redirect all traffic to pass through cloudflare | Web Server Directory Listing |
| Encrypted website (MITM) | Disable older version of TLS | Implementation of Mod security |

Test Specification ID : SOI-****-****-ST

Name of Tester : Wafi, Qamarrul, Zirui

Use Case ID : Web Server, Database Server & Website

Date of Test : June 19 2021

Description of Test : A security testing in ensuring all of the security vulnerabilities in the website, database & web server is being tested and all patches have been implemented.

| **S/No** | **Test Case** | **Expected Result** | **Vulnerable/Not Vulnerable** | **Remarks** |
|---|---|---|---|---|
| **Session Hijacking is a technique in getting in the user session and impersonating the user by login into the website using the user session cookie.** | | | | |
| 1. | Session Hijacking | An attacker has managed to hold the user session by using XSS script which | **Not Vulnerable** | > Reason for failure, a static session is being used together with XSS script. |

| | | | | |
|---|---|---|---|---|
| | | gets the session and stores it into the attacker database. (Pass) | | > Mitigation:<br>session_regenerate_id()<br>htmlspecialchars() |
| **Session fixation is where the attacker keeps login with the same session and the session will not change. For example if the attacker has an admin session, the attacker can use the same session over and over to get into the admin account.** | | | | |
| 2. | Session fixation | An attacker getting hold of the session but the session does not change upon login in (Pass) | **Not Vulnerable** | > Reason for failure, the session does not change upon login in<br>> Mitigation:<br>session_regenerate_id() |
| **SQL injection is a type of attack in which it communicates with the database ensuring an SQL script is being set to either retrieve the tables, drop tables and much more.** | | | | |
| 3. | SQL injection | An attacker send the SQL codes which communicate to the database server (Pass) | **Not Vulnerable** | > Reason for failure, no user validation to work on the sensitive symbols and PDO prepared statement<br>> Mitigation:<br>htmlspecialchars()<br>PDO prepared statement |
| **The attacker sends a <script> code into the web browser and from there the web browser will register it as a code in which it will run the code. This can cause redirection of the web, DoS, take in the user session and much more.** | | | | |
| 4. | Cross-site scripting | An attacker send a <script> | **Not Vulnerable** | > Reason for failure, there is no sanitisation<br>> Mitigation: |

| | | | | |
|---|---|---|---|---|
| | | into the website text inbox | | htmlspecialchars() |
| **The attacker aims to send a large amount of packet and also a request in which it will cause the server to hang and prevent it from providing services to the user.** | | | | |
| 5. | DoS & DDoS | An attacker send large amount of packets to the victim which cause denial of service (Pass) | **Not Vulnerable** | > Reason for failure, the server keeps take in the packets and does not have a timeout session.<br>> Mitigation:<br>TimeOut<br>MaxClients<br>KeepAliveTimeout<br>Implementation of Cloudflare |
| **Invalid or insufficient condition attack is where the attacker change the URL from user page to the admin page as an example without any validation to check if this user is a normal user or an admin.** | | | | |
| 6. | Invalid or insufficient condition attack | The attacker can use key in the URL and change the page from user to admin page (Pass) | **Not Vulnerable** | > Reason for failure, there is no validation in who have an admin right or user rights.<br>> Mitigation:<br>Use session tokens |
| **Man-in-the-Middle (MITM) attacks refer to an attacker sitting in the middle of the network and listening/seeing all of the packets that are being transferred. This is where if the user senda personal information such as password or other information, this cause the attacker to see the password and exfiltrate the data.** | | | | |
| 7. | Encrypted website (MITM) | Man-in-the middle attacker can sniff the packet in | **Not Vulnerable** | > Reason for failure, there is no encryption when transmitting data from user to the web server. |

| | | | | |
|---|---|---|---|---|
| | | plaintext and get all of the user data and information from it (Pass) | | > Mitigation:<br>HTTPS using Cloudflare |
| **Brute force attack refers to the attacker trying every possible combination to get the correct password and from there impersonate the user by login in with the password. This attacker can be used manually or automated depending on the password content of the user.** | | | | |
| 8. | Brute force attack | An attacker can try the possible combination to get into the user/admin account (Pass) | **Not Vulnerable** | > Reason for failure, there is a simple password and no OTP<br>> Mitigation:<br>Google Authenticator OTP<br>generateRandomSecret()<br>Strong password creation upon register |
| **Killing all of the session is the best method to prevent the attacker from gaining the session and using it to its advantage such as session fixation or session hijacking.** | | | | |
| 9. | Kill all session | Every session need to be killed upon logout of the session | - | Mitigation:<br>session_destroy() |
| **Improper validation of the file upload can cause a serious harm to the web server by the attacker can upload any folder and file in which it can contain malware or a script. Hence opening backdoor or exfiltrate the necessary data.** | | | | |
| 10. | Improper validation upload session | An attacker can upload a malware in which it can | **Not Vulnerable** | > Reason for failure, is that the programmer did not check the user upload and do a validation to it<br>> Mitigation: |

| | | | | |
|---|---|---|---|---|
| | | gain a backdoor session (Pass) | | Validate the user inputs and enable mod_security to scan for any malware, scripts in the file |
| **File inclusion attack is where the web server connects to the web server terminal, this can cause the attacker to send commands in which it can cause the web server to respond with the commands that the attacker have sent such as displaying password/user files and many more.** | | | | |
| 11. | File inclusion attack | An attack in which the attacker can display all of the web server files such as user files and many more (fail) | **Not Vulnerable** | It was found that this attack was not vulnerable in our web application |
| **This attacker can show the version and identity for the web server can cause the attacker to gain more knowledge during the reconnaissance phase and here he can do a search on the exploits and vulnerabilities that this server contains.** | | | | |
| 12. | Showing of Apache Version and OS Identity | An attacker which the attacker can gain information regarding on the server (Pass) | **Not Vulnerable** | > Reason for failure, The server signature was not off<br>> Mitigation: ServerSignature Off ServerTokens Prod |
| **Web Server directory listing is where the attacker can access the directory of the web server in which this can cause the attacker to know the structure of the web server system.** | | | | |
| 13. | Web server directory listing | The attacker can see the structure and the files | **Not Vulnerable** | > Reason for failure, the directory listing is enabled in the apache2 config file |

<table>
<tr><td></td><td></td><td>inside of the web server (Pass)</td><td></td><td>> Mitigation: Options -Indexes</td></tr>
<tr><td colspan="5"><b>Implementation of mod security which a Web Application firewall, helps to ensure that the traffic is secure and there is no suspicious traffic flowing through the web server. This is an implementation of defense in depth to ensure that the traffic and the connection to the web server is secure and all other than that will flag out an alert.</b></td></tr>
<tr><td>14.</td><td>Implementation of mod security</td><td>Mod Security enables us to flag out and restrict any access to the website.</td><td>-</td><td>Mitigation<br>> Mod_Security</td></tr>
<tr><td colspan="5"><b>We need to limit the HTTP request to ensure that there is not much traffic in the single web server. This can cause the web server to slow down and even DoS by having too many HTTP requests.</b></td></tr>
<tr><td>15.</td><td>Unnecessary HTTP request</td><td>Unlimited HTTP requests or a lot of HTTP requests can cause the web server to be slow and even hang. (Pass)</td><td><b>Not Vulnerable</b></td><td>> Mitigation: LimitRequestBody</td></tr>
<tr><td colspan="5"><b>Trace command can be used to exfiltrate the data of the user in which the attacker can impersonate as the in which to gain the data access.</b></td></tr>
<tr><td>16.</td><td>Trace HTTP request</td><td>Trace command use in cmd and be used to impersonate as</td><td><b>Not Vulnerable</b></td><td>> Reason for failure, is that the trace is by default is on<br>> Mitigation: TraceEnable off</td></tr>
</table>

<table>
<tr><td></td><td></td><td>the user and exfiltrate the data. (Pass0</td><td></td><td></td></tr>
<tr><td colspan="5"><b>Command injection is a type of attacker in which the website is communicating with the web server terminal and from there any of the commands entered will be directly connected to the terminal.</b></td></tr>
<tr><td>17.</td><td>Command injection attack</td><td>The attacker sends a code or command to the attacker in which to display the files and also gain backdoor connection. (Fail)</td><td><b>Not Vulnerable</b></td><td>It was found that this attack was not vulnerable in our web application</td></tr>
<tr><td colspan="5"><b>Separation of the server in which knowing which server is supposed to be in the internet and in the internal network is important as it can help in ensuring less changes of all of the data being exfiltrated and reduce the changes of pivoting.</b></td></tr>
<tr><td>18.</td><td>Separate database and web server interfaces</td><td>Seperate the network segment and ensure this can help to ensure the least chances of both servers to be hacked.</td><td>-</td><td>> Mitigation<br>Ensure the internet is facing the internet and the database server facing the internal and ensure they both are not using the same server.</td></tr>
<tr><td colspan="5"><b>Exposed PHP version is dangerous as an attacker can get the PHP version using a reconnaissance method in which it the attacker can do a research on the version to see if there are any exploits or vulnerabilities relating to the PHP version.</b></td></tr>
</table>

| | | | | |
|---|---|---|---|---|
| 19. | Exposed PHP version | Showing the PHP version can cause the attacker to know the version that the PHP is running on and from there finding the right exploits and vulnerabilities relating to the PHP version. (Pass) | **Not Vulnerable** | > Reason for failure, PHP config file of the php version is enabled.<br>> Mitigation: expose_php = Off |
| **Redirect all of the traffic to only allowing IP from cloudflare which means it does not allow the user to access the DNS of the load balancer directly in which it already skipped the first line of defense. We do not want that to happen therefore, editing of the security group to only allow access from only the IP address of the cloudflare is recommended.** | | | | |
| 20. | Redirect all traffic to through cloudflare | Due to the AWS limitation of importing certificates and implementing it to the load balancer, therefore we need to change the traffic to only cloudflare IP addresses so that the user cannot access the load balancer directly. (Pass) | **Not Vulnerable** | > Reason for failure, attacker can skip the cloudflare first level protection by accessing the DNS of the load balancer.<br>> Mitigation: Change the security group in AWS to only allow IP address from cloudflare. |

| | | | | |
|---|---|---|---|---|
| **Using an older TLS and SSL version can cause harm to the attacker and use the older version as an exploit.** | | | | |
| 21. | Disable older version of TLS | Older versions of TLS can be seen as vulnerable in which the attacker can use it as an exploit (Pass) | **Not Vulnerable** | > Reason for failure, the TLS/SSL versions setting is cloudflare enabled for all version.<br>> Mitigation<br>Edit the TLS setting in cloudflare to enable TLS 1.2 and above. |

<Attach security testing file here><Security Testing Documentation.docx>

**Functional Testing:**

For functionality we can test how the website is responding to the user when the user interacts with the website. As such we have come out with a summary list of testing we can use to test our website.

**Summary of the functional testing:**

- Admin functions
- User functions
- Guest functions
- Create user functions

Test Specification ID : SOI-****-****-FT

Name of Tester : Wafi, Qamarrul, Zirui

Use Case ID : Website functional testing

Date of Test : July 16 2021

Description of Test : A functional testing for users, guest, create user and admin page and ensuring all of the functions meet our expectations.

| S/No | Test Case | Expected Result | Pass/ Fail | Remarks |
|---|---|---|---|---|
| **User functions are able to display books and it content together with buy the books they need but unable to add, delete or edit books** | | | | |
| 1. | User functions | Able to display books and buy the books<br>Unable to buy the books | Pass | User functions are able to be processed successfully and able to buy the books |
| **Guest functions are able to display books and its content but unable to add, delete, edit, buy books.** | | | | |
| 2. | Guest functions | Able to display all of the books records<br>Unable to buy, edit, delete, add books | Pass | Guest functions are successfully processed and displayed to the guest user. |
| **Admin functions are able to display books and their content together with edit, add and delete the books but unable to buy the books** | | | | |
| 3. | Admin functions | Able to display books and add, edit and delta the books<br>Unable to buy the books | Pass | Admin functions successfully being processed. |

| | | | | |
|---|---|---|---|---|
| **Registered users are able to register their account to the book's website and they are able to login into the user which is part of the user functions listed above.** | | | | |
| 4. | Registered user | Able to registered their account it<br>Unable to login without a suitable account | Pass | Registering of the user is successfully being handled and added into the database. |

**Synthetic Monitoring Testing:**

Summary of the monitoring:

- Web Server
- Database
- Load Balancer
- Auto Scaling

Test Specification ID : SOI-****-****-SMT

Name of Tester : Wafi, Qamarrul, Zirui

Use Case ID : AWS Services

Date of Test : June 9 2021

Description of Test : Created dashboard are used to retrieve data of the web server and database running on the load balancer and individually

| S/No | Test Case | Expected Result | Pass/ Fail | Remarks |
|---|---|---|---|---|
| **CloudWatch monitoring abilities** | | | | |

| | | | | |
|---|---|---|---|---|
| 1. | The team used both their laptops and phones to enter the Qaziwa site. Team then checks the cloudwatch | · Pass | Pass | - The dashboard shows that the web server can handle 7 users (4 laptops 3 phones) simultaneously entering the Qaziwa domain.<br>- Not much load was used to handle 7 users |

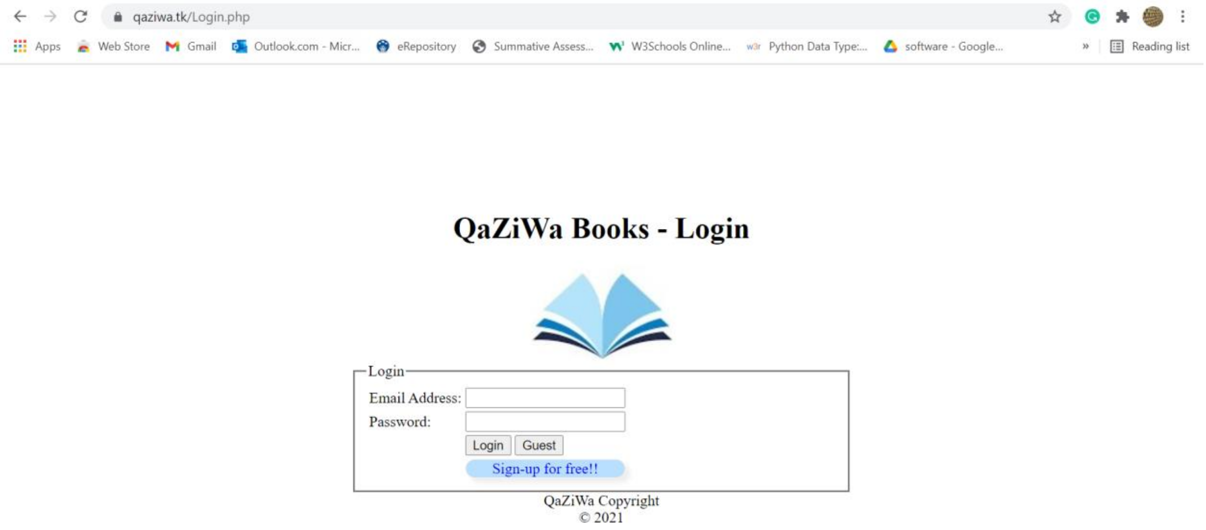
# 4 User and Technical Documentations

## 4.1 User Documentation/Guide/Manual

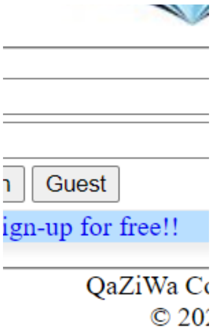
There will be 3 different users for our website - guests, registered users and admin. We will be covering the functionalities of our website as part of the user guide/documentation.

**Guest**

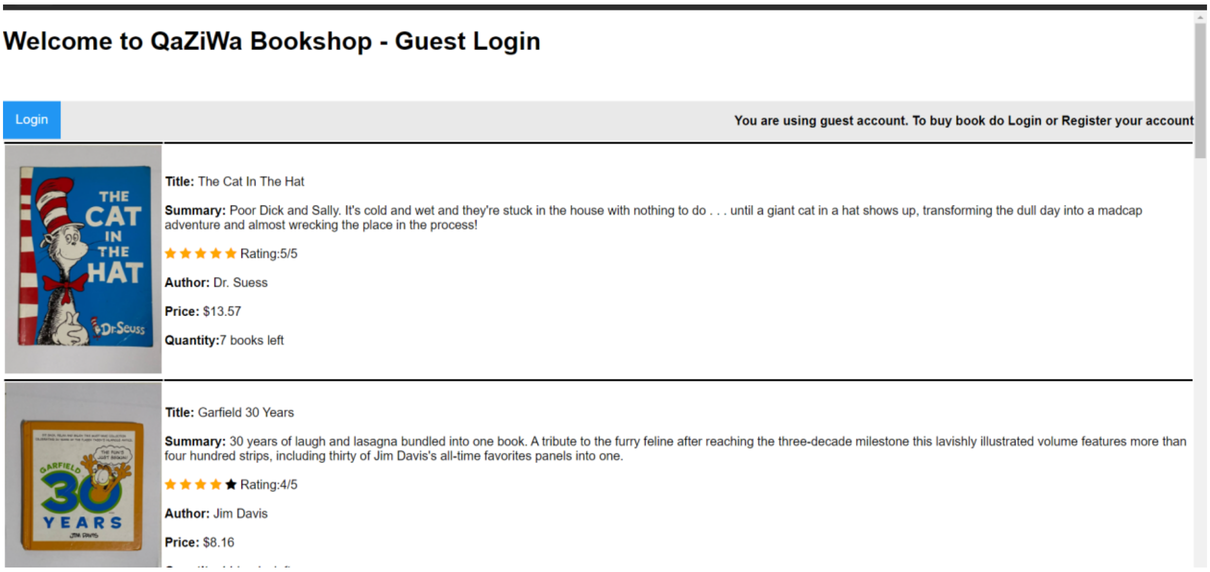
1. Users will access the website [www.qaziwa.tk](www.qaziwa.tk) from any browser.



2. For guest users, click on the "guest" button.



3. They will be directed to Qaziwa's bookstore listing page. However, only viewing of listed books is allowed.



Welcome to QaZiWa Bookshop - Guest Login

You are using guest account. To buy book do Login or Register your account

**Title:** The Cat In The Hat

**Summary:** Poor Dick and Sally. It's cold and wet and they're stuck in the house with nothing to do . . . until a giant cat in a hat shows up, transforming the dull day into a madcap adventure and almost wrecking the place in the process!

★★★★★ Rating:5/5

**Author:** Dr. Suess

**Price:** $13.57

**Quantity:**7 books left

**Title:** Garfield 30 Years

**Summary:** 30 years of laugh and lasagna bundled into one book. A tribute to the furry feline after reaching the three-decade milestone this lavishly illustrated volume features more than four hundred strips, including thirty of Jim Davis's all-time favorites panels into one.

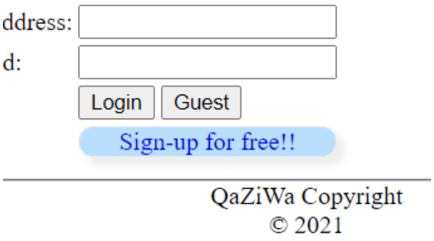
★★★★★ Rating:4/5

**Author:** Jim Davis

**Price:** $8.16

- **How to signup for an account**

1.Click on the "Sign-up for free!!" button.

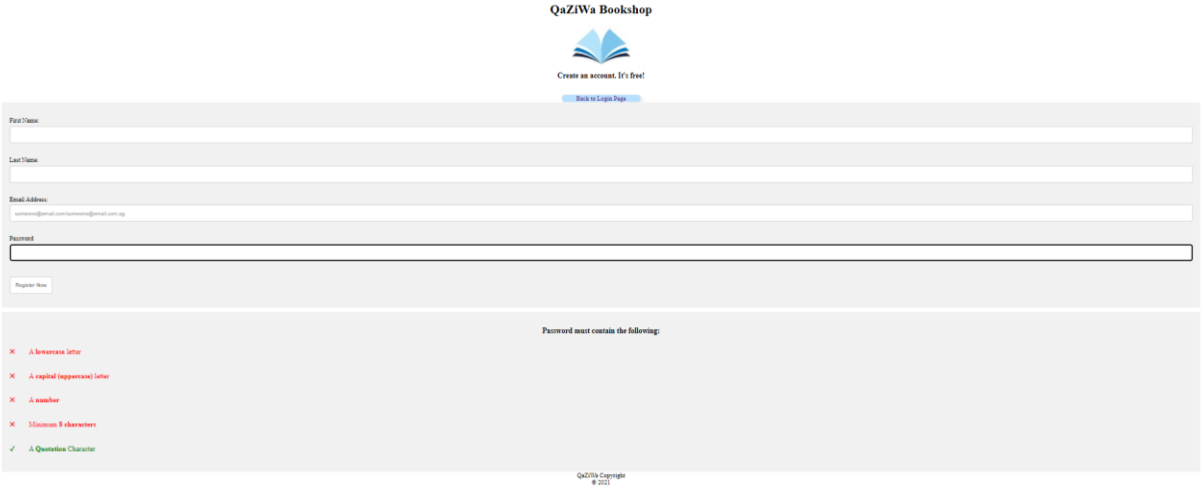


ddress:

d:

QaZiWa Copyright
© 2021

2.Enter the credentials. There is also a password policy enforced on every user account.



QaZiWa Bookshop

Create an account. It's free!

First Name:

Last Name:

Email Address:

someone@email.com/someone@email.com.sg

Password

Password must contain the following:

✕ A lowercase letter

✕ A capital (uppercase) letter

✕ A number

✕ Minimum 8 characters

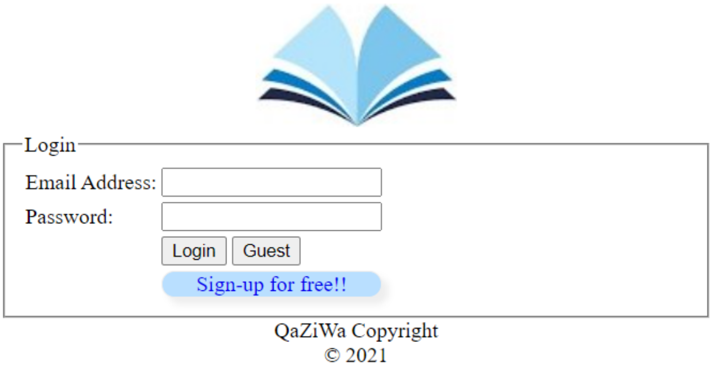
✓ A Quotation Character

QaZiWa Copyright
© 2021

**Registered users**

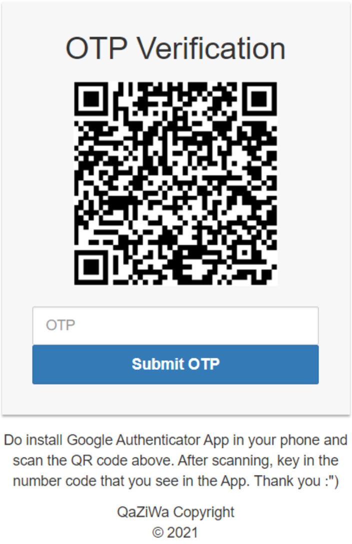
1.Users will access the website www.qaziwa.tk from any browser.

2.Since an account has been registered, users are required to log in using their email address and password used when registering an account. Click the "login" button or press "enter" from the keyboard after entering the credentials.



QaZiWa Copyright
© 2021

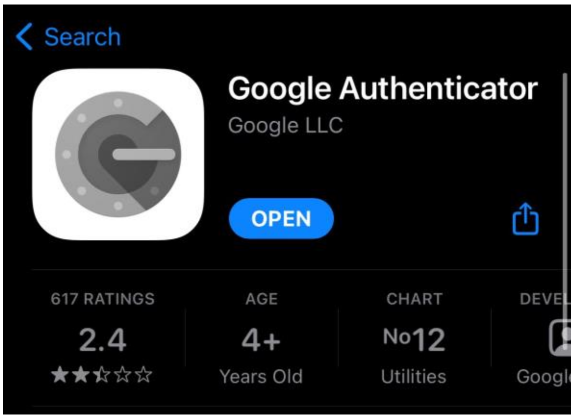
3.If the login is successful, users will be redirected to a verification page. On this page, a QR code will be shown. Users cannot scan this QR via their phone camera or normal QR scanner application. Instead, an authentication application called "Google Authenticator" is required for better security aspects.

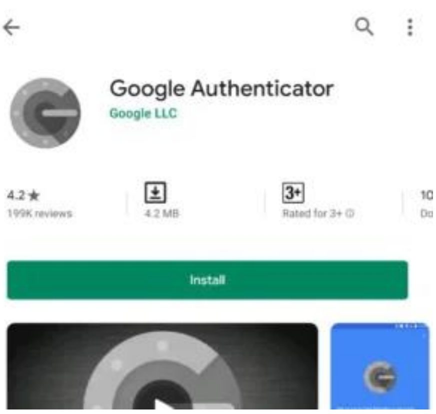


Do install Google Authenticator App in your phone and scan the QR code above. After scanning, key in the number code that you see in the App. Thank you :")

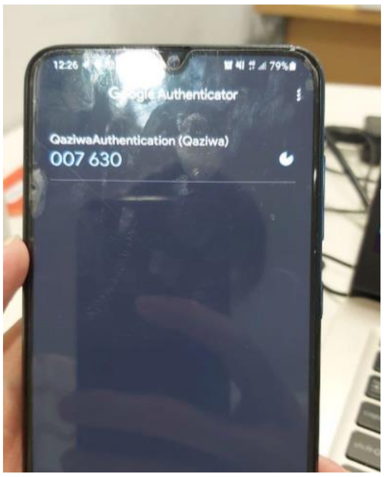
QaZiWa Copyright
© 2021

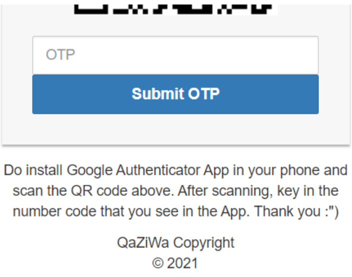
4.To scan this QR code, download/install this application from the AppStore or Google Play store.





5.After installation, open the application and scan the QR code that appeared on the screen. A 6 digit one-time password (OTP) will be given as shown below. Enter this OTP into the input box directly under the QR code. The OTP will change regularly at an interval of 20 seconds.



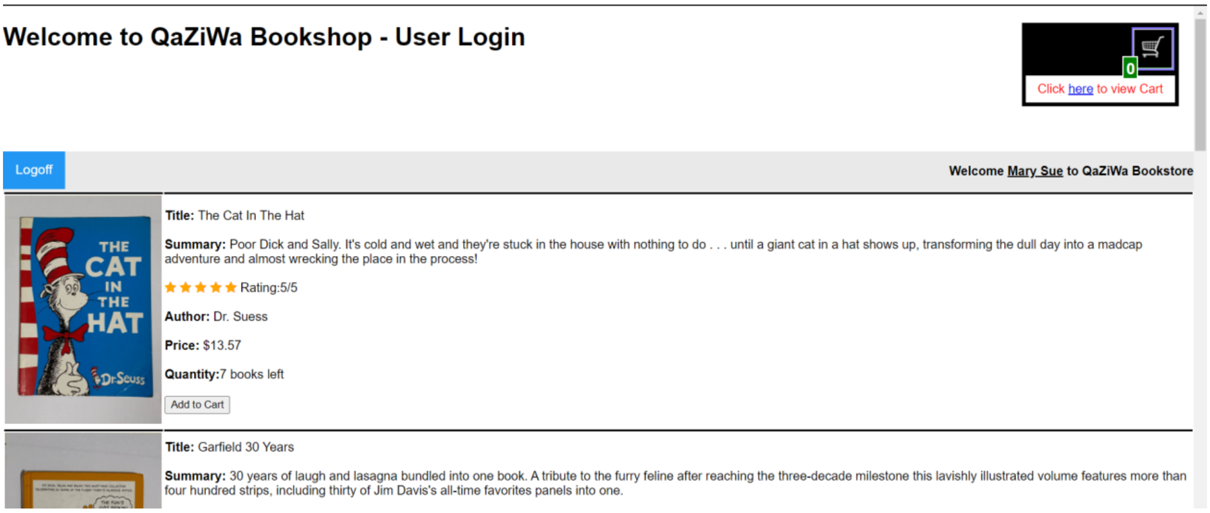

OTP

Submit OTP

Do install Google Authenticator App in your phone and scan the QR code above. After scanning, key in the number code that you see in the App. Thank you :")

QaZiWa Copyright
© 2021

6.Once successful, users will finally be able to browse the website.


Welcome to QaZiWa Bookshop - User Login

0
Click here to view Cart

Logoff

Welcome Mary Sue to QaZiWa Bookstore

**Title:** The Cat In The Hat

**Summary:** Poor Dick and Sally. It's cold and wet and they're stuck in the house with nothing to do . . . until a giant cat in a hat shows up, transforming the dull day into a madcap adventure and almost wrecking the place in the process!

★★★★★ Rating:5/5

**Author:** Dr. Suess

**Price:** $13.57

**Quantity:**7 books left

Add to Cart

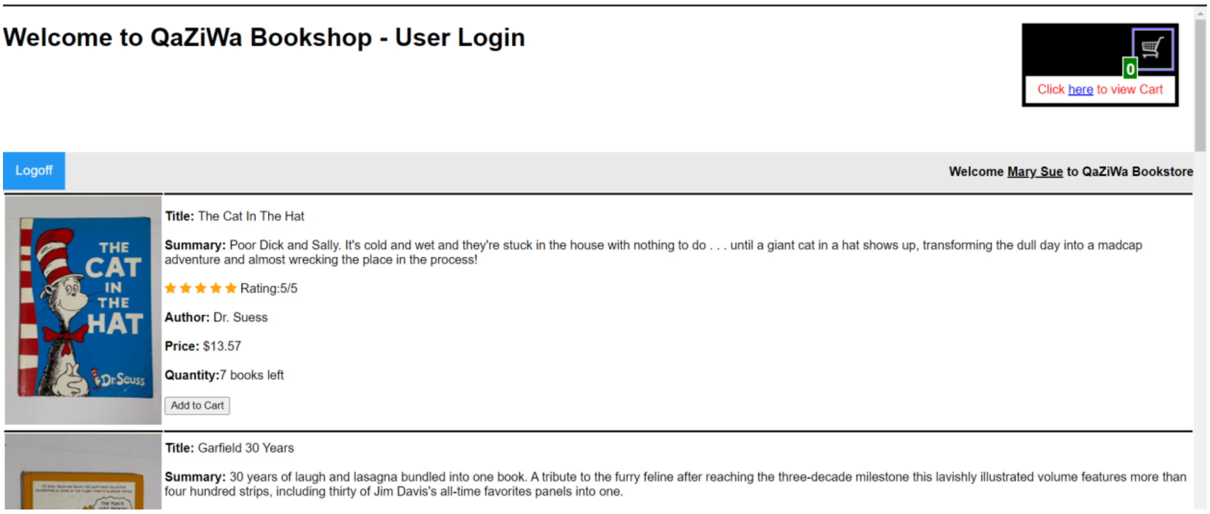
**Title:** Garfield 30 Years

**Summary:** 30 years of laugh and lasagna bundled into one book. A tribute to the furry feline after reaching the three-decade milestone this lavishly illustrated volume features more than four hundred strips, including thirty of Jim Davis's all-time favorites panels into one.

## - Functions inside website

This section will briefly show the different functions of the website

1. The listing page for the books. When you want to select a particular book, click on “add to cart” found beside the image of the book.



**Welcome to QaZiWa Bookshop - User Login**

Welcome Mary Sue to QaZiWa Bookstore

**Title:** The Cat In The Hat

**Summary:** Poor Dick and Sally. It's cold and wet and they're stuck in the house with nothing to do . . . until a giant cat in a hat shows up, transforming the dull day into a madcap adventure and almost wrecking the place in the process!

★★★★★ Rating:5/5

**Author:** Dr. Suess

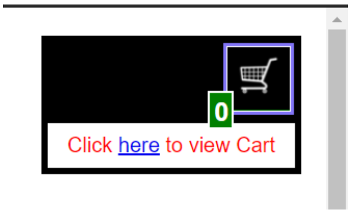
**Price:** $13.57

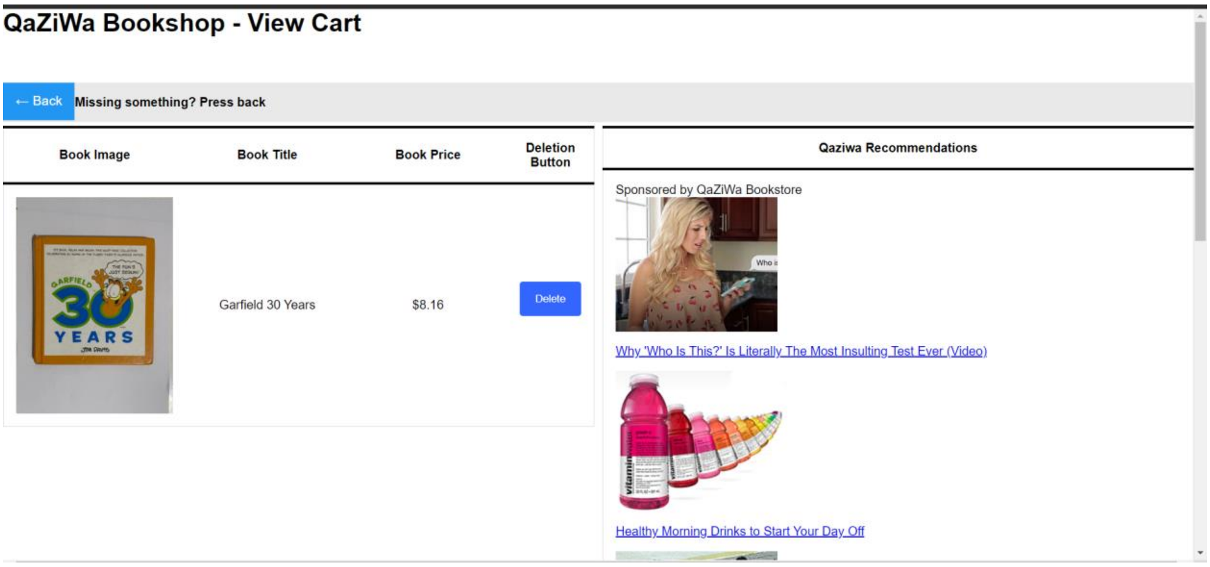
**Quantity:**7 books left

**Title:** Garfield 30 Years

**Summary:** 30 years of laugh and lasagna bundled into one book. A tribute to the furry feline after reaching the three-decade milestone this lavishly illustrated volume features more than four hundred strips, including thirty of Jim Davis's all-time favorites panels into one.

2. To view items inside the shopping cart, click “here” from the top right corner of the website. Books can still be removed from the cart before the purchase has been made.





**QaZiWa Bookshop - View Cart**

Missing something? Press back

| Book Image | Book Title | Book Price | Deletion Button |
|---|---|---|---|
| | Garfield 30 Years | $8.16 | Delete |

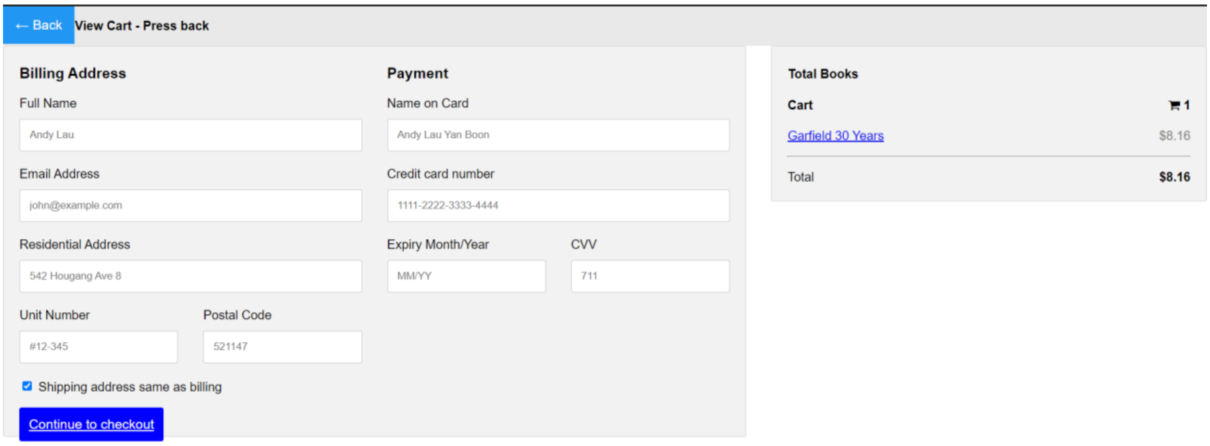
**Qaziwa Recommendations**

Sponsored by QaZiWa Bookstore

Why 'Who Is This?' Is Literally The Most Insulting Test Ever (Video)

Healthy Morning Drinks to Start Your Day Off

3.This is the payment page. All information is required to be filled in before payment can be made.



← Back View Cart - Press back

**Billing Address**

Full Name: Andy Lau

Email Address: john@example.com

Residential Address: 542 Hougang Ave 8

Unit Number: #12-345

Postal Code: 521147

☑ Shipping address same as billing

**Payment**

Name on Card: Andy Lau Yan Boon

Credit card number: 1111-2222-3333-4444

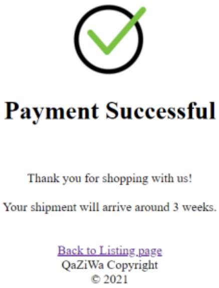
Expiry Month/Year: MM/YY

CVV: 711

**Total Books**

| | |
|---|---|
| Cart | 1 |
| Garfield 30 Years | $8.16 |
| Total | $8.16 |

4. When the payment is made successful, this page will appear as an indication.



**Payment Successful**

Thank you for shopping with us!

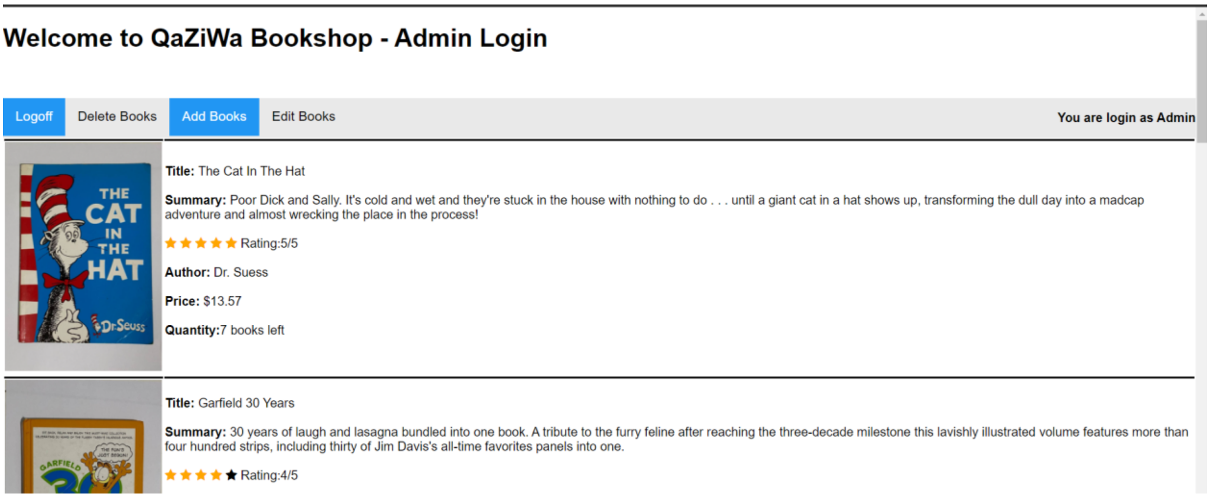
Your shipment will arrive around 3 weeks.

Back to Listing page
QaZiWa Copyright
© 2021

## 4.2 Technical Documentation (Installation guide/Manual)

**Administrator**

1.Users will access the website www.qaziwa.tk from any browser. Login into the website as usual with the registered email address and password.



Welcome to QaZiWa Bookshop - Admin Login

You are login as Admin

**Title:** The Cat In The Hat

**Summary:** Poor Dick and Sally. It's cold and wet and they're stuck in the house with nothing to do . . . until a giant cat in a hat shows up, transforming the dull day into a madcap adventure and almost wrecking the place in the process!

★★★★★ Rating:5/5

**Author:** Dr. Suess

**Price:** $13.57

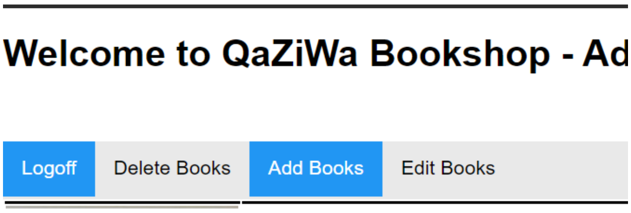
**Quantity:**7 books left

**Title:** Garfield 30 Years

**Summary:** 30 years of laugh and lasagna bundled into one book. A tribute to the furry feline after reaching the three-decade milestone this lavishly illustrated volume features more than four hundred strips, including thirty of Jim Davis's all-time favorites panels into one.

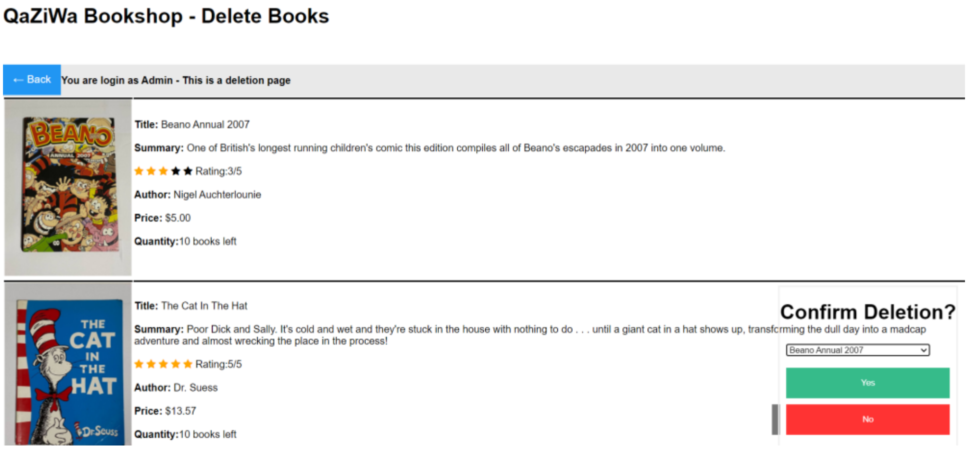
★★★★★ Rating:4/5

2.The administrator has all privileges - delete, add and edit. These actions can be achieved from the section right at the top of the website.



Welcome to QaZiWa Bookshop - Ad

3.Deleting of the books



QaZiWa Bookshop - Delete Books

You are login as Admin - This is a deletion page

**Title:** Beano Annual 2007

**Summary:** One of British's longest running children's comic this edition compiles all of Beano's escapades in 2007 into one volume.

★★★★★ Rating:3/5

**Author:** Nigel Auchterlounie

**Price:** $5.00

**Quantity:**10 books left

**Title:** The Cat In The Hat

**Summary:** Poor Dick and Sally. It's cold and wet and they're stuck in the house with nothing to do . . . until a giant cat in a hat shows up, transfcrming the dull day into a madcap adventure and almost wrecking the place in the process!

★★★★★ Rating:5/5

**Author:** Dr. Suess

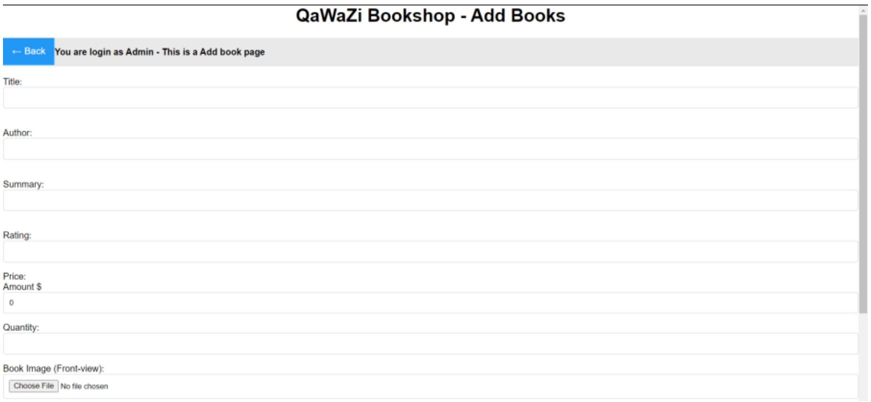
**Price:** $13.57

**Quantity:**10 books left

Confirm Deletion?

Beano Annual 2007

4. Adding of the books



QaWaZi Bookshop - Add Books

You are login as Admin - This is a Add book page

Title:

Author:

Summary:

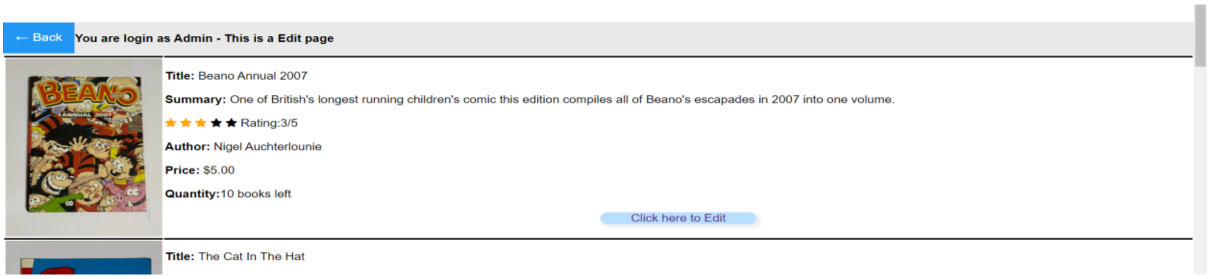
Rating:

Price:
Amount $
0

Quantity:

Book Image (Front-view):
No file chosen

5. Editing of the books



You are login as Admin - This is a Edit page

**Title:** Beano Annual 2007

**Summary:** One of British's longest running children's comic this edition compiles all of Beano's escapades in 2007 into one volume.

★★★★★ Rating:3/5

**Author:** Nigel Auchterlounie

**Price:** $5.00

**Quantity:**10 books left

**Title:** The Cat In The Hat

All amendments made by the administrator will be reflected inside the database.

# 5 Conclusions

The main takeaway will be the implementation of security infrastructure in a cloud environment. We have implemented 2-factor authentication (2FA) for our website using Google authenticator. Registered users will scan the QR code on the screen where an OTP will be displayed and change at an interval. With the limitation of importing SSL certification, our team managed to overcome it by using Cloudflare. This service will provide DOS/DDOS protection and many features apart from having an SSL certificate.

In addition, we have also learned how we can configure the instance using only the Command-line interface (CLI) as using a Graphic user interface (GUI) is neither possible nor feasible. As such we manage to do it with fewer struggles in which we have to work together as a team to solve, mitigate and complete the project on time.

Furthermore, learning and exploring various aspects of AWS education accounts have to offer us, and learning to configure them is also an accompaniment we have made. In the future, this knowledge can be used to configure the company server using a cloud-based approach.

To end off, once again, we would like to thank Ms. Pauline (our supervisor) for supporting us and giving us the resources in ensuring the completion of the project and its deliverables without having major issues in configuring and troubleshooting. Lastly, I would like to thank the assessors Mr. James and Mr. Zack for suggesting the possible upgrades in our project in which to go further in learning and understand the various concepts on why we need to implement them.

Together as a team we can achieve greater heights and ensure we have each other’s back.

# - References

1.The Training Help. (2018, May 30). How to install PHP+MySQL+phpmyadmin in EC2 AWS server. Retrieved from https://www.youtube.com/watch?v=WqUnwWhCdoI&ab_channel=TheTrainingHelp

2.Valaxy Technologies. (2017, Aug 8). Create custom VPC with public, private subnets, internet gateways, security groups, route tables. Retrieved from https://www.youtube.com/watch?v=NHozohphtEg&list=LL&index=15&t=1279s

3.Stephane Maarek. (2020, Mar 12). AWS Elastic Load Balancing Tutorial. Retrieved from https://www.youtube.com/watch?v=OGEZn50iUtE&ab_channel=StephaneMaarek

4.Stack Instance. (2016, Nov 5). How to Create a AWS EC2 Snapshot and Restore from Snapshot. Retrieved from https://www.youtube.com/watch?v=W89C_OqlOwE&ab_channel=StackInstance

5.Amazon Web Services. (2014, Nov 13). Getting Started with AWS Key Management 6Service. Retrieved from https://www.youtube.com/watch?v=-5MPXHvKDnc&t=480s

6.Amazon Web Service. (2013, Jan 15). Getting Started with AWS Identity and Access Management. Retrieved from https://www.youtube.com/watch?v=ySl1gdH_7bY

7.Pawan Kumar. (2017, May 20). Connect and Handle Files in FTP Server using PHP. Retrieved from https://unitedwebsoft.in/blog/connect-handle-files-ftp-server-using-php/

8.Aaron kili. (2017, Feb 24). How to install and Configure FTP Server in Ubuntu. Retrieved from https://www.tecmint.com/install-ftp-server-in-ubuntu/

9.Enable write permissions Ubuntu Server in var/www/image directory. (Feb 2012). Retrieved from https://stackoverflow.com/questions/9181254/enabling-write-permissions-ubuntu-server-in-var-www-image-directory/9181838

10. freenom, A Name for Everyone. Retrieved from https://www.freenom.com/en/index.html?lang=en

11. Sticky sessions for your Application Load Balancer. (2021). Retrieved from https://docs.aws.amazon.com/elasticloadbalancing/latest/application/sticky-sessions.html

12. Manjot Singh. (2017, Jun 28). MySQL encryption at rest. Retrieved from https://www.percona.com/blog/2017/06/28/mysql-encryption-rest-part-2-innodb/

13. Percona Server 8.0. (2017) Verify Encryption for Tables, Tablespaces, and Schemas. Retrieved from https://www.percona.com/doc/percona-server/8.0/security/verifying-encryption.html

14. MySQL. (2021). 15.13 InnoDB Data-at-Rest Encryption. Retrieved from https://dev.mysql.com/doc/refman/8.0/en/innodb-data-encryption.html

15. Encryption of stored data in MySQL 8. (2020, Aug 21). Retrieved from https://www.sqlsplus.com/encryption-of-stored-data-in-mysql-8/#encryption-takes-place-at-the-table-space-level

16. rodrigosilvaesilva. (2019, June 29). ERROR 3185 (HY000): Can't find master key from keyring, please check in server log if keyring plugin is loaded and initialized successfully. Retrieved from https://rodrigosilvaesilva.com.br/index.php/2019/06/29/error-3185-hy000-cant-find-master-key-from-keyring-please-check-in-the-server-log-if-a-keyring-plugin-is-loaded-and-initialized-successfully/

17. Mitchell Anicas. (2014, November 25). How To Install an SSL Certificate from a Commercial Certificate Authority. Retrieved from

https://www.digitalocean.com/community/tutorials/how-to-install-an-ssl-certificate-from-a-commercial-certificate-authority#install-certificate-on-web-server

18. idroot. (2020, November 24). How To Install ModSecurity Apache on Ubuntu 20.04 LTS. Retrieved from https://idroot.us/install-modsecurity-apache-ubuntu-20-04/

19. Tarunika Shrivastava. (2016, November 14). 13 Apache Web Server Security and Hardening Tips. Retrieved from https://www.tecmint.com/apache-security-tips/

20. Domain Dossier. Retrieved from https://centralops.net/co/DomainDossier.aspx

21. Rande. (April 23 2018). Google Authenticator Code. Retrieved from https://github.com/sonata-project/GoogleAuthenticator/blob/2.x/sample/example.php

22. learnWebCoding. (March 24 2019). Two Factor Authentication using Google Authenticator in PHP. Retrieved from https://www.youtube.com/watch?v=RERhlLu6auc&ab_channel=learnWebCoding

23. webbiedave. (January 30 2011). Remotely connecting to a MySQL database. Retrieved from https://stackoverflow.com/questions/4683554/remotely-connecting-to-a-mysql-database

24. A2Hosting. How to connect to MySQL using PHP. Retrieved from https://www.a2hosting.sg/kb/developer-corner/mysql/connect-to-mysql-using-php

25. Paul T. (August 30 2012). How do I use two submit buttons, and differentiate between which one was used to submit the form? Retrieved from https://stackoverflow.com/questions/11929471/how-do-i-use-two-submit-buttons-and-differentiate-between-which-one-was-used-to

26. W3Schools. SQL Injection. Retrieved from https://www.w3schools.com/sql/sql_injection.asp

27. Amazon Web Service. Cryptography concepts. Retrieved from https://docs.aws.amazon.com/crypto/latest/userguide/cryptography-concepts.html

28. itsourcecode. (October 29 2019) How to create a Secure Login Page. Retrieved from https://itsourcecode.com/free-projects/php-project/secure-login-page-using-phpmysql/

# -	Appendices

<Insert web source code in zip file><Qaziwa Web Source Code folder>

<Insert sql file in zip file><Qaziwa.sql>

# - Project Poster

<Insert project poster file><Project Poster.pptx>