SECURE TEMPORARY CLOUD STORAGE FOR FILE TRANSFER BASED ON ACCESS RIGHTS

ERIC HENG CHEE SENG

SESSION 2018/2019

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY

MULTIMEDIA UNIVERSITY

MARCH 2019

SECURE TEMPORARY CLOUD STORAGE FOR FILE TRANSFER BASED ON ACCESS RIGHTS

BY

ERIC HENG CHEE SENG

SESSION 2018/2019

THE PROJECT REPORT IS PREPARED FOR

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY
MULTIMEDIA UNIVERSITY
IN PARTIAL FULFILLMENT
FOR

BACHELOR OF INFORMATION TECHNOLOGY
B.I.T. (HONS) SECURITY TECHNOLOGY

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY

MULTIMEDIA UNIVERSITY

MARCH 2019

DECLARATION

I hereby declare that the work have been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Eric Heng Chee Seng

Faculty of Information Science & Technology
Multimedia University

Date: 11 February 2019

ACKNOWLEDGMENT

First and foremost, I would like to take this opportunity to thank everyone who has contributed to the successful completion of this project. I would like to express a big thanks to my supervisor Mr Khoh Wee How for his advice and guidance throughout the development of this project. As the project was ongoing, he has given me suggestion and explanations that were very helpful for proceeding to the rest of the project.

Besides that, I would like to thank my parents and family members who have given me the support and confidence I need throughout my whole journey. Their understanding and advice have helped me overcome obstacles that were difficult to get through. They were also very helpful in advising me on the knowledge that they know about to help me further understand my work. On the other hand, I am also thankful for all my coursemates who have for their suggestions and feedback in aiding my project. Their support was also very helpful in my journey to complete my work.

Last but not least, I would like to thank Multimedia University, my university, for giving me the chance to challenge myself and for the opportunity to expand my knowledge with this project. The facilities and resources were very helpful in the process of compiling information for my project. In conclusion, this project has opened my eyes and mind to a whole new perspective towards the IT industry, and I have a higher level of respect and admiration towards the people working in the field of technology. This project taught me more about the workings of the industry and how much of a challenge it is to be able to create something that will benefit the society, and for that, it was an honour and privilege for me.

ABSTRACT

File transfer is a very common system that everyone is using in their everyday life. The issues most of us are facing are security and the usability of the system. There are multiple ways the data can be compromised during transmission which could lead to the loss of the data's confidentiality and integrity. Besides that, most of the file transferring services have does not encrypt their files and data which could lead to multiple problems. The main purpose of this project is to ease the way people transfer their files and data across the network without worrying about files being compromised by attackers. The encryption used to convert the plaintext to ciphertext is using AES algorithm and is performed at the back-end of the system. This algorithm is used to encrypt the files and decrypt the files. This system has built-in functionalities for the key generation for the encryption and the keys are stored in a secure server. Therefore, users do not have to worry about the key generation, key management, key distribution and key storage.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS/ SYMBOLS

AES	Advanced Encryption Standard
AJAX	Asynchronous JavaScript And XML
AWS	Amazon Web Services
CSS	Cascading Style Sheet
DVD	Digital Video Disc
DOM	Document Object Model
ERD	Entity Relationship Diagram
FIPS	Federal Information Processing Standards
FTP	File Transfer Protocol
GB	Gigabyte
HTML	Hypertext Markup Language
HTTPS	Secure Hypertext Transfer Protocol
JS	JavaScript
JSON	JavaScript Object Notation
MySQL	My Structured Query Language
NIST	National Institute of Standard and Technology
SDLC	System Development Life Cycle
S3	Simple Storage Service
UML	Unified Modelling Language
URL	Universal Resource Locator
USB	Universal Serial Bus

CHAPTER 1

INTRODUCTION

1.1 Overview

File transfer is a transmission of computer files between two computing endpoints through a communication channel of any sorts. The channels are through physical transmission such as wire or through a logical transmission such as computer networking. This is a practice of distributing and transferring digital contents such as documents, multimedia (audio, video, images) or e-books. File transfer is widely used in commercial usage, organisational usage and business usage. This plays a very big role in our current generation.

In this project, a system developed built mainly to solve the way an organisation transfer their files among each other. This was built with security in mind and every file transfer into the system is secure as it is encrypted with Advanced Encryption Standard (AES) 256-bits. The communication between the system and the server is secured with Hypertext Transfer Protocol (HTTPS). Besides that, all digital files and data transferred are all stored in a cloud server using Amazon Web Server (AWS). On the other hand, access control is also developed to have selective access and restriction of access to the files. Therefore, uploaders have the function to select who is allowed access to download the uploaded files.

In terms of security, every file is encrypted with AES a symmetric key which was previously known as Rijndael and was developed by two Belgian cryptographers, Vincent Rijmen and Joan Daemen. The NIST (2001) stated the FIPS-approved cryptographic algorithm named Advanced Encryption Standard (AES) can be used to protect electronic data. Therefore, it is highly encouraged to apply this cryptographic protection to protect the confidentiality and secrecy of the organisation files. Besides cryptographic protection, HTTPS is built into the system to provide protection on privacy and integrity for the data. This bidirectional encryption protects the data exchanged like user information while in transit from attacks like eavesdropping or tampering of the communication. With this practice, assurance is provided without worrying attackers in the network.

The cloud storage is a logical pool used to store all the digital files and data uploaded by the user. The cloud storage holds all the resources of the system and is responsible for providing the correct resources to the authorised users. All encrypted files are to be stored here in the server and all the database data are also stored here in the cloud. This cloud system is developed by Amazon and is called Amazon Web Server.

Access rights or access control is the selective access or restriction of who or whom has access to the files or resources uploaded into cloud storage. This function is built to limit the number of users that have access to the resources and only limit to the authorised users. Users are to select who is authorised to download the file when they are uploading the file as this is part of the process of the system.

1.2 Problem Statement

In this era of technology, file transfer plays a very big role in communication among different organisation parties. Traditionally, people used to use physical devices such as USB memory stick and DVD to create a copy of the file to transfer from PC to PC manually. This makes the transferring of a file very troublesome as it is required to do one by one and it also consumes a lot of time and effort.

Besides that, malware is also pledged a huge problem when transfer files using physical devices. This risk can occur when the sender’s physical device is connected to another PC and that PC is infected with malware such as a virus. Because of this, it has a high probability of damaging the physical device and this malware can be spread when it is connected to other computers when transferring the file. Furthermore, the malware infested in the memory stick could damage the confidentiality and integrity of the file. Because of this, this file transfer method is not suitable for files with sensitive and confidential information.

Other than that, confidentiality should be also a factor for file transfer. Confidentiality preservers the secrecy of the file’s content. If the memory stick is lost or stolen, the content of the file will be revealed, and it will no longer be a secret. Furthermore, some people may use third-party web services to perform file transfer. These services have a high possibility of not encrypting the content of the file or uploading a file without AES implemented in the web service. This could lead to multiple types of attack such as the Man-in-the-Middle attack.

In conclusion, using a physical memory stick and or unsecured third-party web services are not recommended for the transference of files as it can lead to multiple type risks.

1.3 Project Objectives

The main purposes of this project are to solve and replace the way people transfer their files from one to another in the most efficient way possible. The objectives of this project are

- To replace the traditional methods of transferring files
- To develop a cloud storage to transfer a file from among computers
- To able to upload a file and download files from the cloud storage
- To able to delete a file when all receivers have downloaded the file
- To be able to encrypt uploaded files and decrypt the downloaded files
- To ease the transfer of files

1.4 Project Scope

This project is built for organisations to transfer their files among each other. Any companies that would require a fast, secure and efficient way to transfer their files across the internet. This system is built for whatever size of the company, from small organisations to large organisations. The requirement to host this service is for the employees to have access to a computer and internet access. The limitation of this service is that it is not for commercial usage meaning it is used only for company usage.

1.5 Report Organisation

The whole report of this project consists of a number of chapters which are an introduction, literature review, methodology, implementation plan, implementation process, testing & evaluation and conclusion.

In the introduction, it gives a general overview of the whole project. The problem statement is produced to showcase the current problems that this project will be able to solve. The objectives of this project have been stated and the scope has been declared.

In the literature review, 3 existing systems in our market which are similar to this project are studied. Each of these systems performs the main function but they have different built-in features to support their main function.

In methodology, it describes the SLDC and how it is implied to the project in every phase of the SDLC. The tools, programming languages and software required for this project is stated.

In the implementation, several UML diagrams are drawn to make it easier to understand the flow of the system and how the system operates. The diagrams used are use case diagram, flowchart, sequence diagram and ERD diagram.

In the implementation process, the code of the system is described and explain and the system is explained by screenshots from the UI of the system

In testing and evaluation, it shows an explanation and description of how the test is conducted on the system.

Lastly, this chapter covers concludes this project. It also covers the limitation of the system and any possible future enhancements that could be integrated into the system.

CHAPTER 2

LITERATURE REVIEW

2.1 SendGB

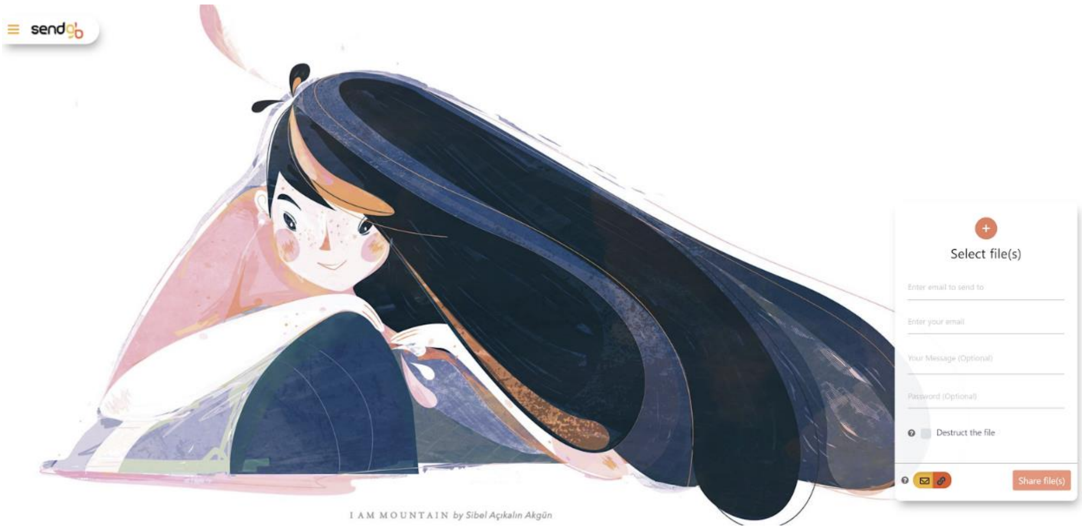


Figure 2.1.1: SendGB home page

Sending a large file with SendGB acts the main function of this web service. SendGB is made to be easy to use by just adding files and start sharing the files without and registration. Users can download the large attachments uploaded to the cloud via emails or links generated and users don't require any software and application to use this service. The transfer rate of files is very fast as SendGB provides a multi-thread and high bandwidth cloud servers. All files transfer over an encrypted line, therefore, making this service secure. Besides that, using SendGB is completely free and users are not required to register or sign in to upload and download files.

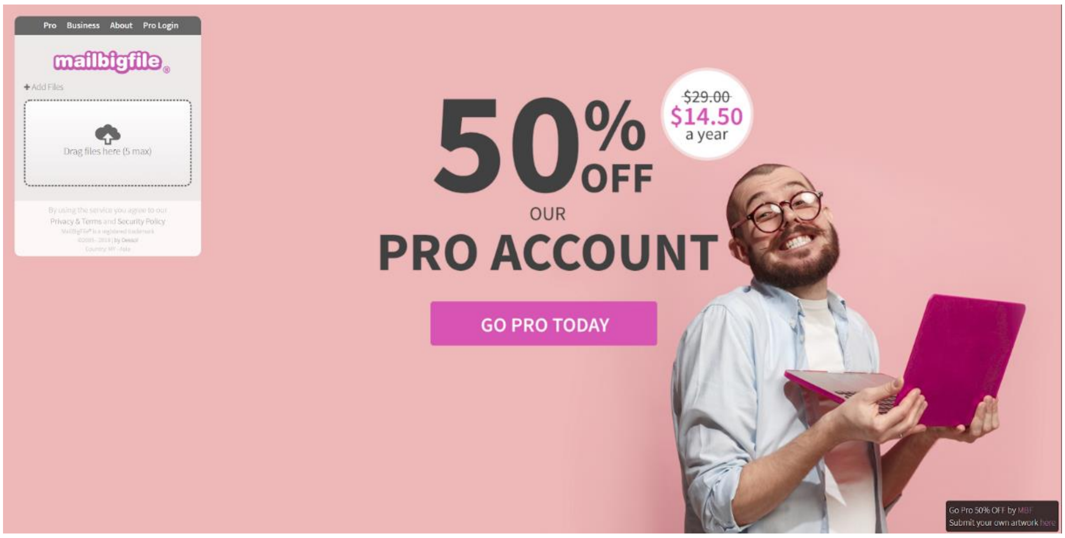
2.1.1 Operation/Structure

A file transfer web service, SendGB provides great security standards as all encryption. All file encryption are all encrypted with 256-bit AES which makes every file stored securely in their server located in France, Paris. The connection is secured with HTTPS with certifications issued by COMODO RSA. Besides that, every files transfer are stored in the server for a fixed time frame of 7 days. All files that are subject to expiration will be automatically deleted from their servers. Once deleted there will be no ways of retrieving the files back. Furthermore, uploaders have a file size limit of 4 GB and the transfer supports well-known file types. Files links can be sent to a maximum of 20 email recipients per transfer.

2.1.2 Features

SendGB provides multiple features to enhance the user experience for their file transferring service. One of the features that stands out from the competitors is that if the user is facing network issue while uploading, the upload will resume as soon as when the connection is back online. Other than that, if the user device runs out of charge, he or she can add back the file and it will resume where it has left off. Besides that, SendGB provides a password for their files to enhance the security level for the downloaders. To enhance the user experience, this service supports ten different languages such as English, Russian, Italian and many more.

2.2 MailBigFile



By using the service you agree to our Privacy & Terms and Security Policy
MailBigFile® is a registered trademark
©2005 - 2018 | by Dessol
Country: MY - Asia

Go Pro 50% OFF by MBF
Submit your own artwork here

Figure 2.2.1: MailBigFile upload page

Dessol, a company based in the United Kingdom has developed a web application called MailBigFile to solve the issue for sending large files in a safe, secure and compliant way. MailBigFile provides a range of different premium packages for their service. Most developed features are only available for their premium users while free users are only able and limited to use their basic functions.

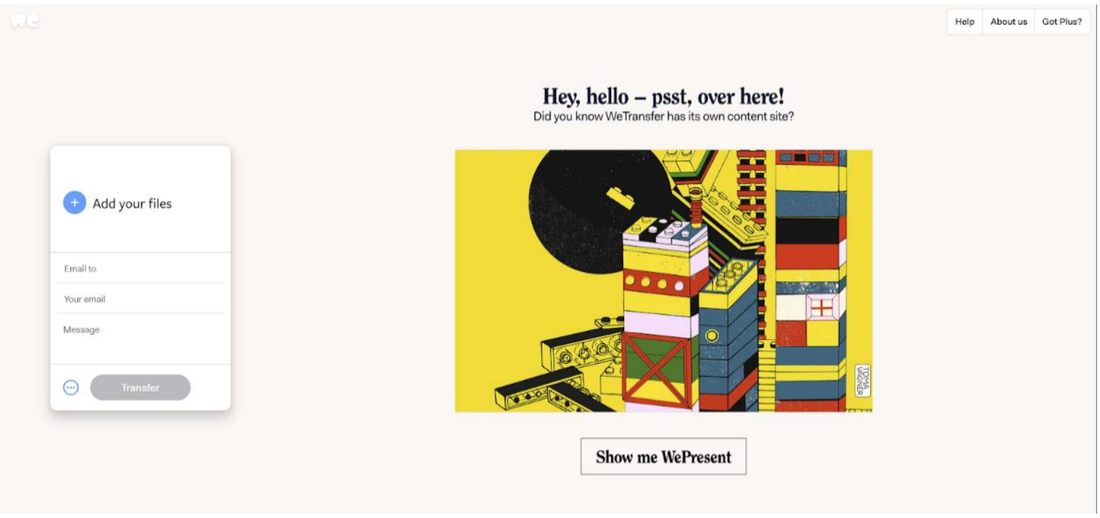
2.2.1 Operation/Structure

MailBigFile equips with secure file transfer for only their premium members. All files uploaded to the cloud are uploaded with 128-bit AES encryption. MailBigFile uses HTTPS to encrypt the connection between the users and the website. The certification is issued by Amazon. All files stored on the server are all encrypted with 256-bit AES. Free and Pro users service runs on AWS in their data centre in the United States and Business package users run on AWS in their datacenter in Iceland. These servers are protected by fully managed AWS firewall and servers software are constantly updated with the latest patch. All the files upload must be lesser than the size limit when uploading the file. 2 GB is the limit file size a free user can upload while premium users have a range of 4 GB to 20 GB based on the service packages. Files uploaded by free users are expired after their 10 days while premium users files will be deleted after 28 days to 60 days. Downloading stored files requires the link generated when the file is uploaded. Premium users can send the links up to 15 recipients via email whereas free users are only available to send 1 link to their recipient.

2.2.2 Features

This service provides multiple features to enhance the service of their main function for their premium users. The system has an FTP integration which offers all file sent to the opposite party is stored directly on the users FTP server. Premium business accounts are available to create a custom form field to receive files to collect all information in one place. Customization is also one of a key feature as users are able to have a custom URL, branding, language and design by uploading their own CSS. Besides that, files uploaded can be deleted from the cloud and files can be tracked. Other than that, a file can be protected by a password besides encryption which increases the security level of the file. Lastly, MailBigFile provides a desktop and iOS experience.

2.3 WeTransfer


Hey, hello – psst, over here!

Did you know WeTransfer has its own content site?

Figure 2.3.1: WeTransfer home page

A team founded in Amsterdam, Netherlands and Los Angeles, United States by Bas Beerens and Naldens in 2009 have developed a cloud-based file transfer service. The service for WeTransfer is free for all with limited features, only the paid users are accessible to all their developed features. This service main mission is to provide the simplest and effortless transfer of files.

2.3.1 Operation

WeTransfer provides excellent security functionality for their users as all files upload are encrypted with TLS and stored in the cloud with 256-bit AES encryption. Users are able to upload with a maximum file size of 20 GB per transfer whereas 2 GB is the maximum file size a free user can upload. The security standards of WeTransfer are to accommodate the Dutch Personal Data Protection Act. Files are only downloaded from the unique generated link created with the uploader send the file to the cloud. All files upload are stored in WeTransfers' server located in Europe and the United States. Files are stored in the server for a temporary time up to 4 weeks and will be destroyed after the file hits the expiry date. The network connection of the website is secured with HTTPS. The certificates are issued by Amazon.

2.3.2 Feature

Apart from transfer files, WeTransfer provides additional features to support their main function for paid premium users. Every transfer, a user can specify up to 100 email recipients to receive to link to download the file. The expiry date of files are flexible as they can be specified by the uploaders or they can be stored without expiry date at their server for a maximum size of 100 GB. Besides that, WeTransfer's premium users can have their own personal or customizable URL ('username.webtransfer.com') profile for their transfer. With this, free users can send a file up to 20 GB instead of 2 GB to the paid user by visiting their profile. Apart from that, a file can have a password defined by the uploader to increase the level of security.

2.4 Conclusion

In conclusion from all the existing system studied, all the level of encryption is all AES 256-bit encryption which makes all the file upload extremely secure. The file size limit of each file is different, MailBigFile and WeTransfer provide a large file limit which is 20 GB limit and SendGB's limit is 4 GB. For this developed system, the file limit is 1 GB as it targets file transfer for organizational documentation and not for commercial files like videos or music. Besides that, the connection is all encrypted with HTTPS. All the data transferred over the network are encrypted which makes the connection a secure connection. For some system or service, they come with a price, MailBigFile and WeTransfer provide multiple packages for their service in accordance with their limitation while SendGB and this system are free without limitations. Furthermore, all these systems studied provides an additional level of security which is file password as the password is required to begin the download.

On the other hand, as all these systems provide the same functionalities which are to transfer files from one party to another, they do have other additional features to enhance the usability of the service. For SendGB, when a network issue has occurred during file upload, it will resume where it has left off as soon as when the connection is back online. Besides that, MailBigFile provides FTP integration, custom web page design with CSS, branding, language, file tracking and file deletion for their premium users. Furthermore, WeTransfer provides custom URL for their consumers.

For this system, it provides multiple features to enhance the functionality of the system which is through file verification to detect any file corruption or damage. Besides that, this service provides administrative control access to monitor the overall usage of the website and user account control. Moreover, for uploaders, this service is able to generate the list of file history of the uploaders completed and pending files. For uploaders, there is a view made that display the progress of who or whom have downloaded or yet to download the files.

Table 2.4.1: Literature Study Comparison

	SendGB	**MailBigFile**	**WeTransfer**	**FYP**
Encryption	256-bit AES	256-bit AES	256-bit AES	256-bit AES
File Size Limit	4 GB	20 GB	20 GB	1 GB
Connection	HTTPS	HTTPS	HTTPS	HTTPS
Fee	Free to use	Free/Paid	Free/Paid	Free
File Password	True	True	True	True

CHAPTER 3

METHODOLOGY

3.1 System Development Life Cycle

In this chapter, it describes the methodology used for this project. This development adapts the system development life cycle. Dora & Dubey (2013) stated the framework that is used to plan, manage, and control the process of developing an information system refers to as a software development methodology. This development will consist of 5 different phases Requirement Analysis, Design, Coding, Testing and Maintenance.

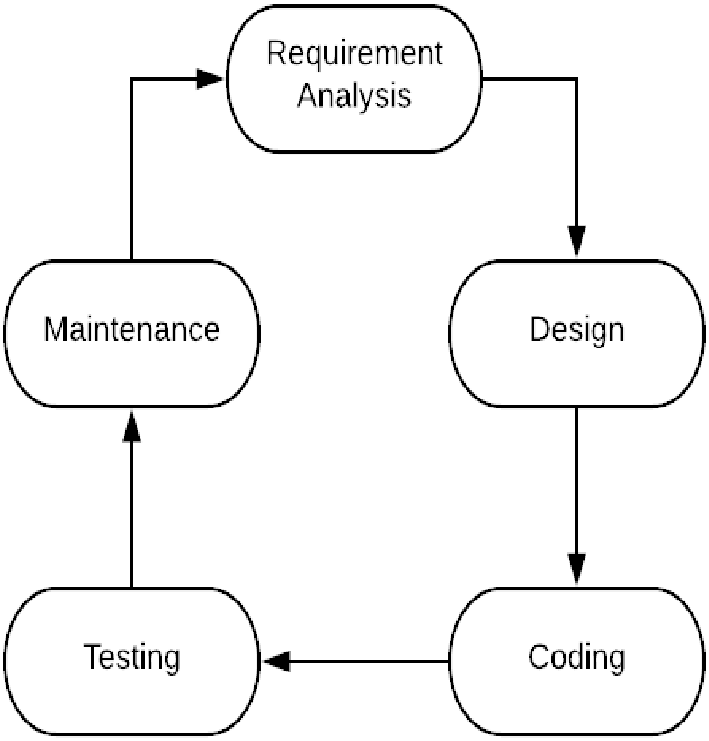


Figure 3.1.1: SDLC Model

In the first phase of the development, the main objectives and goal are to collect the actual project requirement and document properly. Besides that, the key features of the system are also addressed and problem statement is studied. Research on cryptography, cloud computing, access control and web programming is deeply studied and understood. This is a very crucial phase in the life cycle as it marks the starting point of the project of what is required.

In the design phase, stated that the goal of this phase is to transform the requirement specification into a structure. During this phase, the user interface is being designed using a low-fidelity prototyping. The logic and algorithm are developed and transformed into UML diagrams such as flowcharts and use-case diagram. The database logic is reconstructed into ERD diagram for visualisation

After the design phase comes coding. In this phase, the designed planned from the second phase is continued by converting it into code using programming languages. The end result of this phase is the programming code. The programming languages and open source software used are stated on page 13. The software used to write the programming languages is Adobe Dreamweaver.

After the third stage, testing is a very important phase as it is carried out to know if it satisfies the requirement proposed. Errors and bugs are also identified such as logic error, syntax error and semantic error are identified. Kumar, Zadgaonkar, & Shukla (2013) stated that an effective testing will contribute to the delivery of high-quality software products, more satisfied users, lower maintenance costs, and more accurate and reliable results

In the final phase, after the product has been launched, the admin and the system administrator of the system is to maintain the project. The system administrator is to maintain the files stored in the cloud server and make sure the cloud server is running in good condition. The admin main role is to control the access of users to the system as well as to monitor the users' usage.

3.2 Languages and Software

In this part, the types of languages and software used in the development of this project are defined and described. The languages used consists of a Markup Language, Style Sheet Language, Scripting Language and My Structured Query

Language. The software used for the implementations and write the code is Adobe Dreamweaver and to host the cloud resources and data, Amazon Web Server is used.

3.2.1 Markup Language

Markup languages are built for the presentation and display of text. For this development markup language is required and necessary. The project is using Hypertext Markup Language (HTML) to create hypertext documents for the presentation and definition of text. HTML is a language that is mainly used in the production of a website. A web page and web application are required to have a visual appearance for the users to interact with the service. Berners-Lee & Connolly (1995) stated that HTML markup can represent hypertext news, mail, documentation, and hypermedia; menus of options; database query results; simple structured documents with in-lined graphics; and hypertext views of existing bodies of information. HTML provides a range of long list of features that this project is required. HTML can be embedded into scripting languages such as JavaScript and PHP: Hypertext Preprocessor. HTML display hypertext documents in a boring manner, so HTML can be embedded with a style sheet language called Cascading Style Sheets to bring some design layout to the web page and web application.

3.2.2 Style Sheet Language

Style Sheet Language is a language that expresses the design and appearance of the document. Cascading Style Sheets (CSS) is a language used to define the visual expression of the web page and application embedded with the HTML document. This language allows designers to define the presentation of the layout, colours and fonts. Implementing CSS with HTML in a document ensures a more modernized layout and appearance of the web pages. With great defined layout, it brings good navigation around the web page, making users able to understand and explore the page more easily which makes the web pages more user-friendly. Authors can define good and attractive content and visual elements from their choice of fonts and other typographic details. These tiny details can change the perspective of the web page. With Bootstrap, it helps the design the web pages to be more responsive and attractive. Besides that, Bootstrap helps HTML to beautify the typography, forms, buttons, navigation bar, modal and many more.

3.2.3 Scripting Language

PHP is known as PHP: Hypertext Preprocessor or previously known as Personal Home Page is a service-side scripting language that will be used for the development for this project. It is will be used to make dynamic and interactive web pages. The PHP will be acting as the back-end functions of the web service and be working side by side with HTML. For this project, PHP will be used for the connection and communication between the web page and the database, the uploading and downloading of the files, encryption and decryption, the deletion of the files from the cloud server and many more.

JavaScript often abbreviated as JS, is a popular client-side scripting language that is particularly used together with HTML and CSS on the web pages. JS is used to make the web pages more interactive, therefore making it an essential part of web development. In the project, JS will be playing a minor role and it is used for its built-in event-driven functionalities.

Furthermore, to simplify HTML Document Object Model Tree manipulation and event handling, a lightweight JavaScript library that simplifies JavaScript programming called jQuery is used. jQuery has multiple core features such as HTML or Document Object Modal (DOM) manipulation, CSS manipulation, HTML event methods, Ajax, effects and animations. With this library, jQuery wraps the multiple lines of codes written in JavaScript and turns into a single method.

Moreover, to create an asynchronous web application on the client side, a web development technique for accessing web servers from a page called Asynchronous JavaScript and XML or more commonly known as Ajax is used. Using this web technology, this has made possible for a web application to send data and receiving data asynchronously without ever having to refresh or redirect the browser. Because of this, it is able to change the content of the data displayed on the web browser dynamically without refreshing the entire web page. The built-in XMLHttpRequest object is commonly used to execute Ajax to request data from the web server and display or print the data and content to the web page with the help of JavaScript and HTML DOM

3.2.4 Database Management System

My Structured Query Language or also known as MySQL is an open-source relational database management system. Davis & Phillips (2007) stated that MySQL automates the most common tasks related to storing and retrieving specific user information based on your supplied criteria. Besides that, MySQL is also built to manipulate data, update data, delete data and many more. MySQL in the project is used to store user accounts, guest users, access rights, encryption/decryption key, guest information, log and file information. Furthermore, MySQL will be working hand-in-hand with PHP to display data in the website, storing data entered by a user from the website, deleting data and many more.

3.2.5 Cloud

A cloud server is a logical server that is delivered over the Internet through a cloud computing platform. With this type of technology, it is possible to configure resources that can be rapidly provisioned with little or no effort over the Internet. In this project, we have used cloud computing service called Amazon Web Services to aid with one of our main functionalities which are storing data and files. This cloud computing platform is developed by Amazon and launched in 2006. On the other hand, every single file that the users upload through the web server is stored in the cloud server which is called Amazon S3. It is an object storage that offers high availability, performance and security. With these great attributes, S3 makes a great service to store and protect file for this web application. Besides that, to house the data stored, a database is required. In this development, we have used Amazon Relational Database (RDS) Service to store and house the database data. With RDS, it makes it so much easier to run, operate, and scale a relational database. This AWS service provides great performance, security and high availability. Furthermore, they provide multiple types of database engines like Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle and Microsoft SQL Server but for this development, only MySQL database engine is used.

3.2.6 Web Server

A web server is a program that uses HTTP (Hypertext Transfer Protocol) that process incoming network request from the files in the server to form the web pages for the users. For this development, a free and opensource software called XAMPP developed by Apache Friends is used as the web server. XAMPP has a module to deploy an Apache HTTP Server and his server will be mainly used to store the uploaded files, process the files like encryption or decryption or retrieve the files hash value and deliver the web pages to the users. All files uploaded are stored in the web server temporarily and the file is encrypted in the web server. Once encrypted, the file is sent over to the AWS cloud storage and the file in the web server is removed and deleted.

3.2.7 Programming Tools/Software

During the development of this project, in the third phase of the SDLC, a programming development software is required to write the code of the system. In this project, we have chosen a software called Adobe Dreamweaver created by Macromedia and developed by Adobe Systems to develop the web service and web pages. Adobe Dreamweaver is a proprietary web development software that is crafted for novice to expert programmers which makes it very easy to use for any developers. Besides that, Adobe Dreamweaver has a built-in template for designing the appearance of the web pages to assist in the design process. With this feature available, it is able to create very consistent web pages throughout the whole website and the workload is reduced. Adobe Dreamweaver provides some standard code functionalities and features such as syntax highlighting to spot any potential errors, code completion, real-time syntax checking and code examination for generating hints to assist the author. Furthermore, this software supports programming technologies required for this project which are HTML, CSS, PHP, JavaScript and MySQL.

Besides software for programming languages, a program called MySQL Workbench developed by Oracle Corporation is used to connect to the AWS RDS database to integrate SQL development. This workbench enables the developer to visually generate, manage and implement the database required. With this software, it provides multiple great features that aid the developer such as SQL Editor, data modelling, database administration and performance monitoring.

CHAPTER 4

IMPLEMENTATION

4.1 Overview

In this chapter, we will focus on the implementation of the project using multiple graphical diagrams. These graphical diagrams represent multiple independent views of the system and illustrate how the system operates. In this project, we have used a use-case diagram, flowchart diagram and an entity-relationship diagram to give an overview of the whole system.

4.2 Use Case Diagram

When developing this system, the use case diagram is used to identify the roles of the functional requirements.

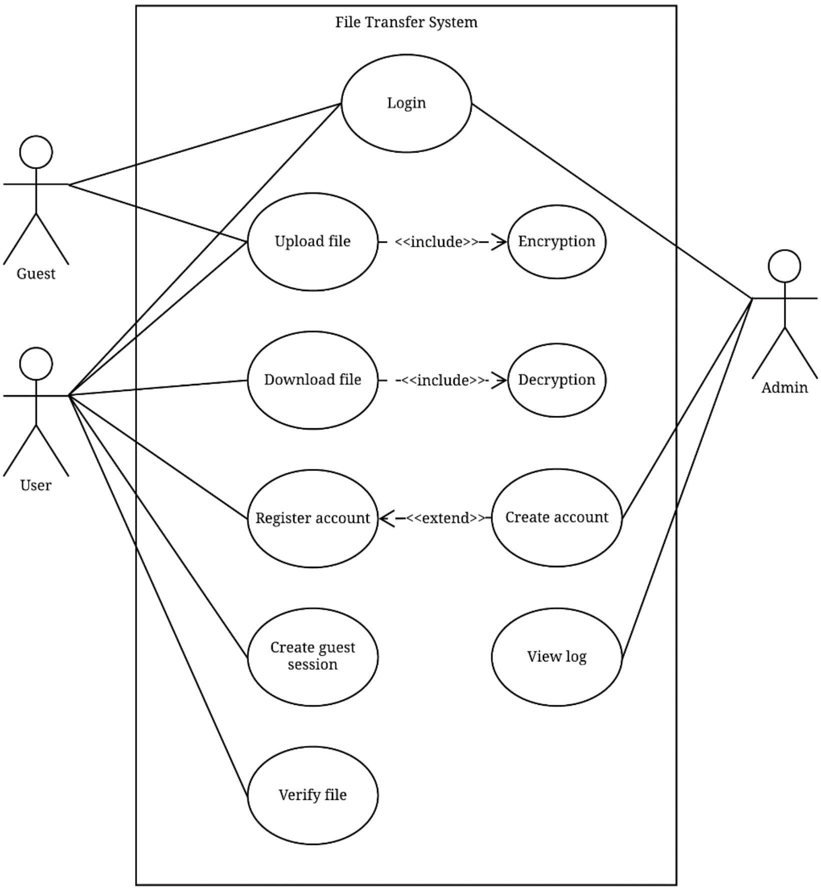


Figure 4.2.1: Use Case Diagram

4.3 Flowchart

The flowchart below shows the login process, register process, uploading and downloading a file sequence.

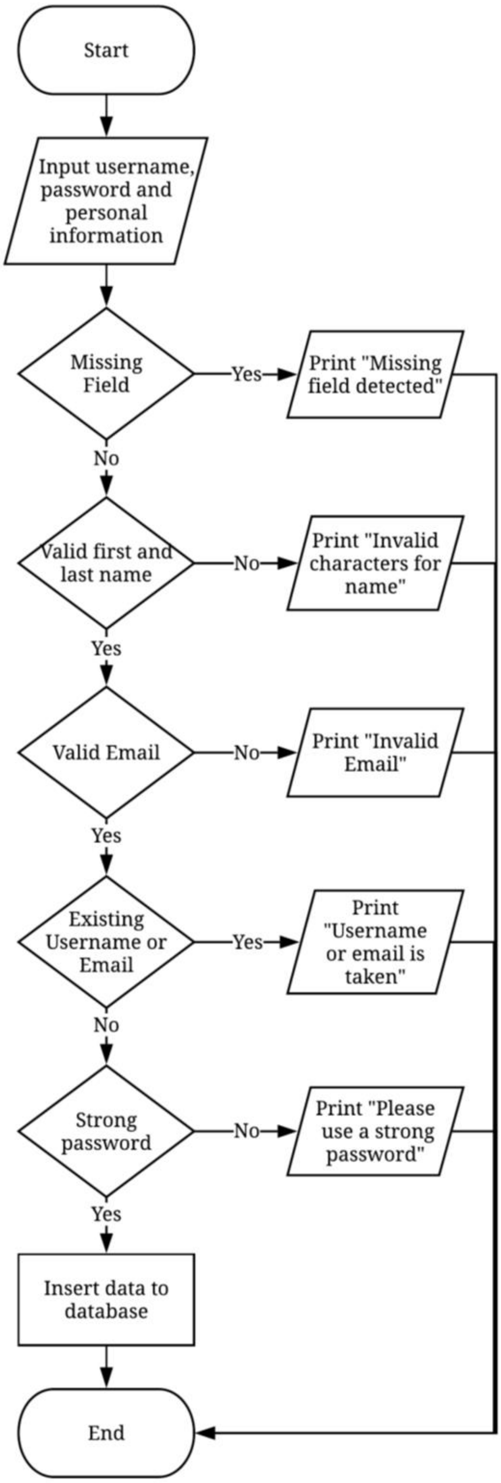

Start
Input username, password and personal information
Missing Field
Yes
Print "Missing field detected"
No
Valid first and last name
No
Print "Invalid characters for name"
Yes
Valid Email
No
Print "Invalid Email"
Yes
Existing Username or Email
Yes
Print "Username or email is taken"
No
Strong password
No
Print "Please use a strong password"
Yes
Insert data to database
End

Figure 4.3.1: Register

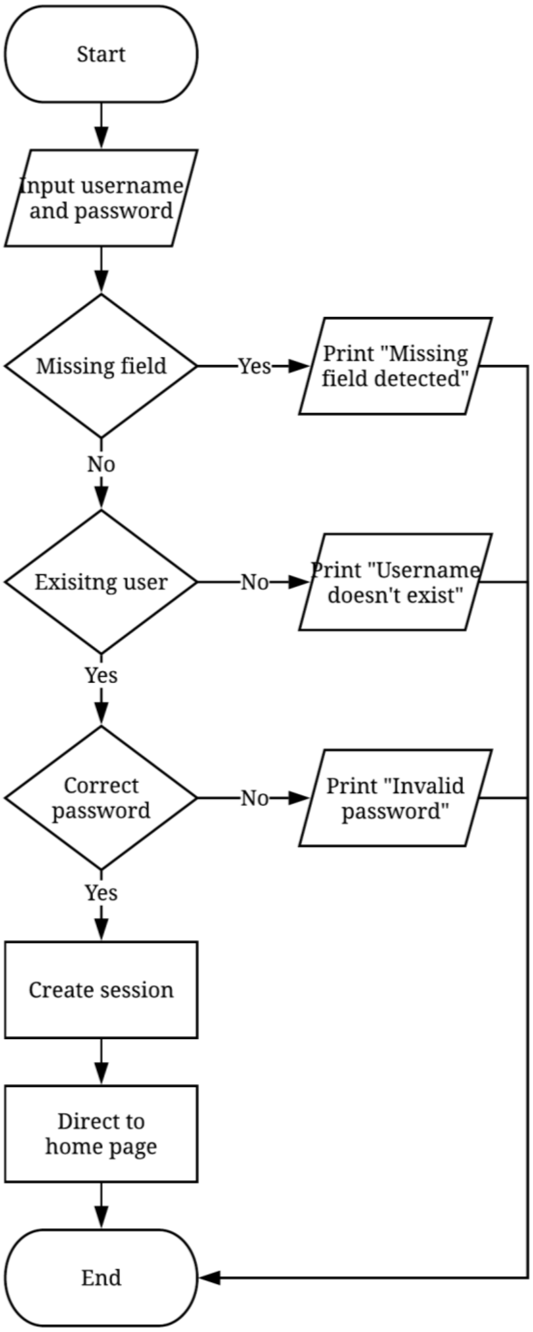


Figure 4.3.2: Login

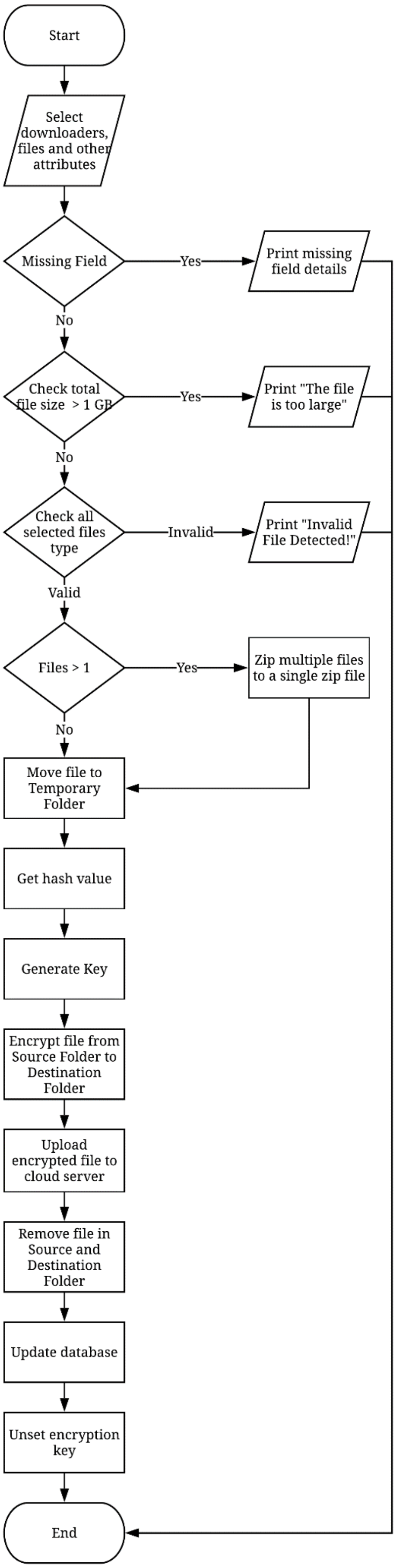

Start
Select downloaders, files and other attributes
Missing Field
Yes
Print missing field details
No
Check total file size > 1 GB
Yes
Print "The file is too large"
No
Check all selected files type
Invalid
Print "Invalid File Detected!"
Valid
Files > 1
Yes
Zip multiple files to a single zip file
No
Move file to Temporary Folder
Get hash value
Generate Key
Encrypt file from Source Folder to Destination Folder
Upload encrypted file to cloud server
Remove file in Source and Destination Folder
Update database
Unset encryption key
End

Figure 4.3.3: Uploading File

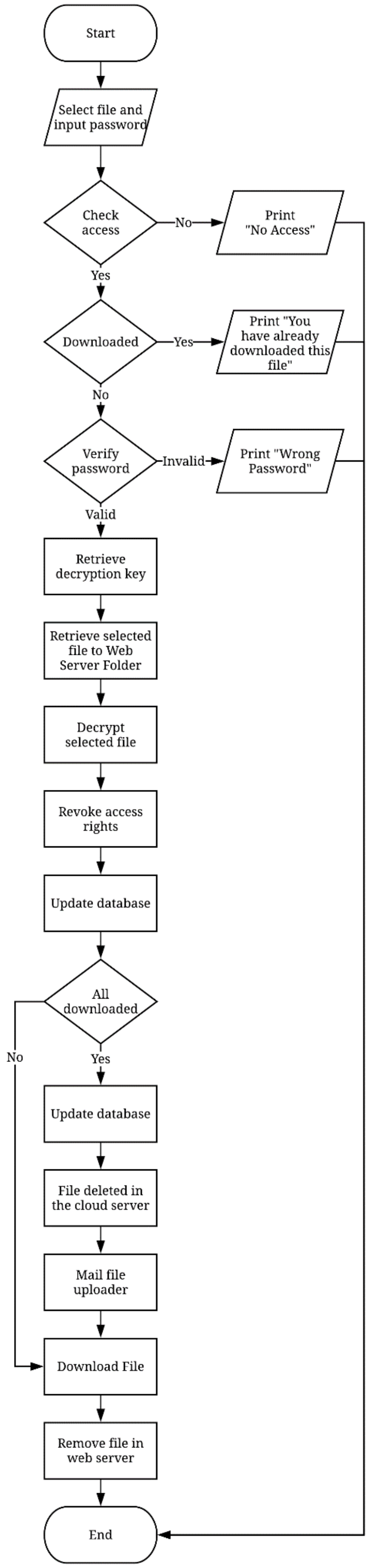


Figure 4.3.4: Downloading File

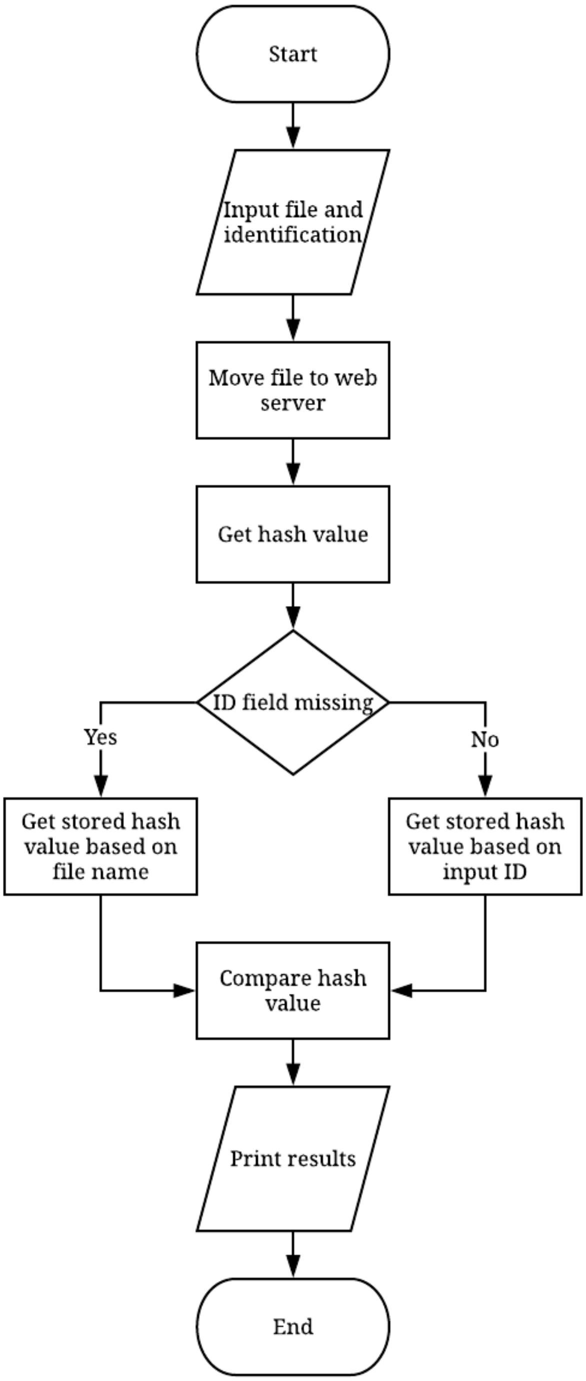

Start
Input file and identification
Move file to web server
Get hash value
ID field missing
Yes
No
Get stored hash value based on file name
Get stored hash value based on input ID
Compare hash value
Print results
End

Figure 4.3.5: Verify file

4.4 Sequence Diagram

The sequence diagram below shows the login process, register process, uploading and downloading a file sequence.

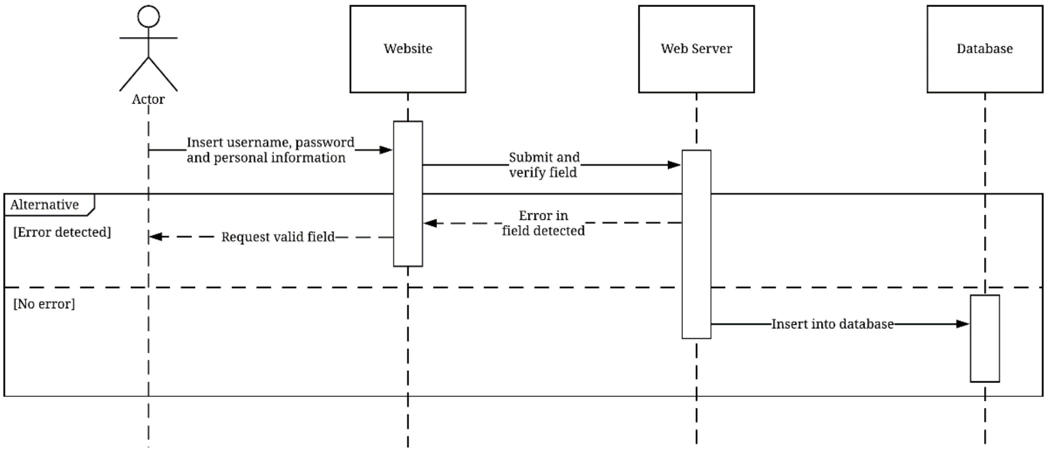


Figure 4.4.1: Sequence Diagram Register

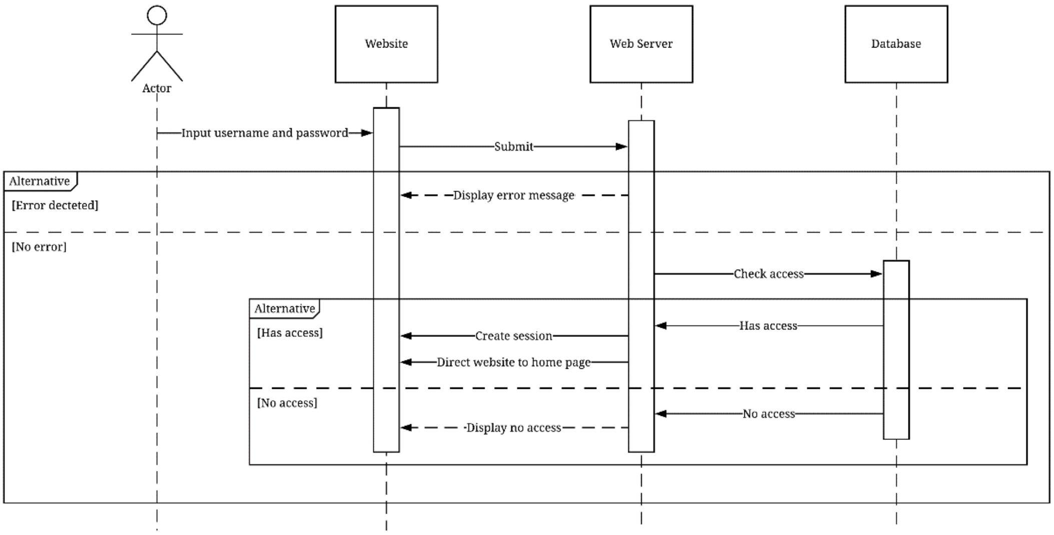


Figure 4.4.2: Sequence Diagram Login

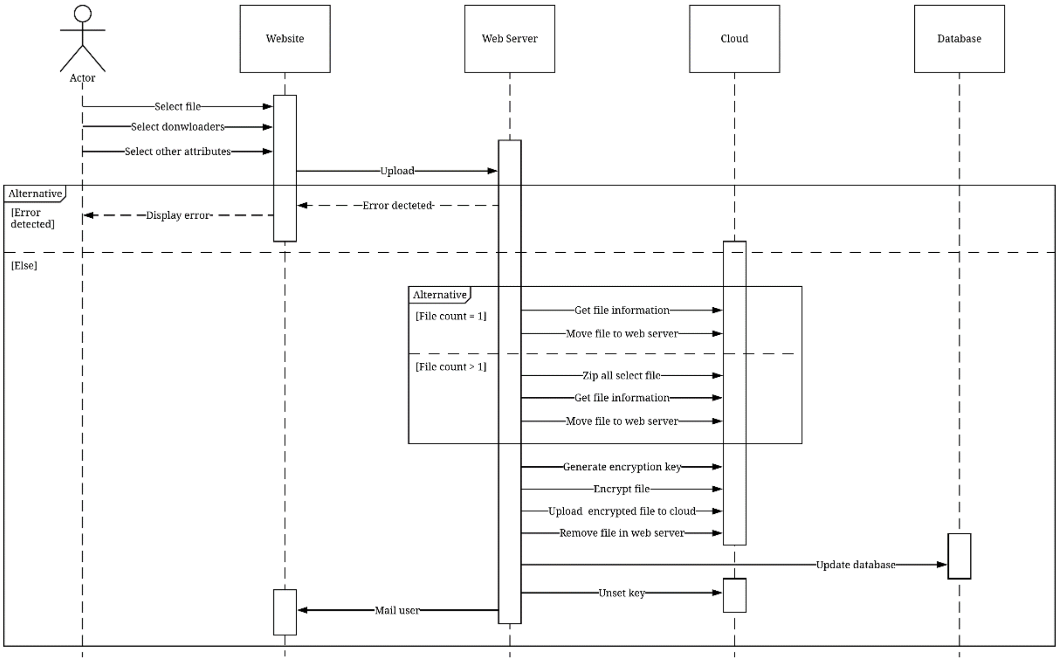


Figure 4.4.3: Sequence Diagram Upload File

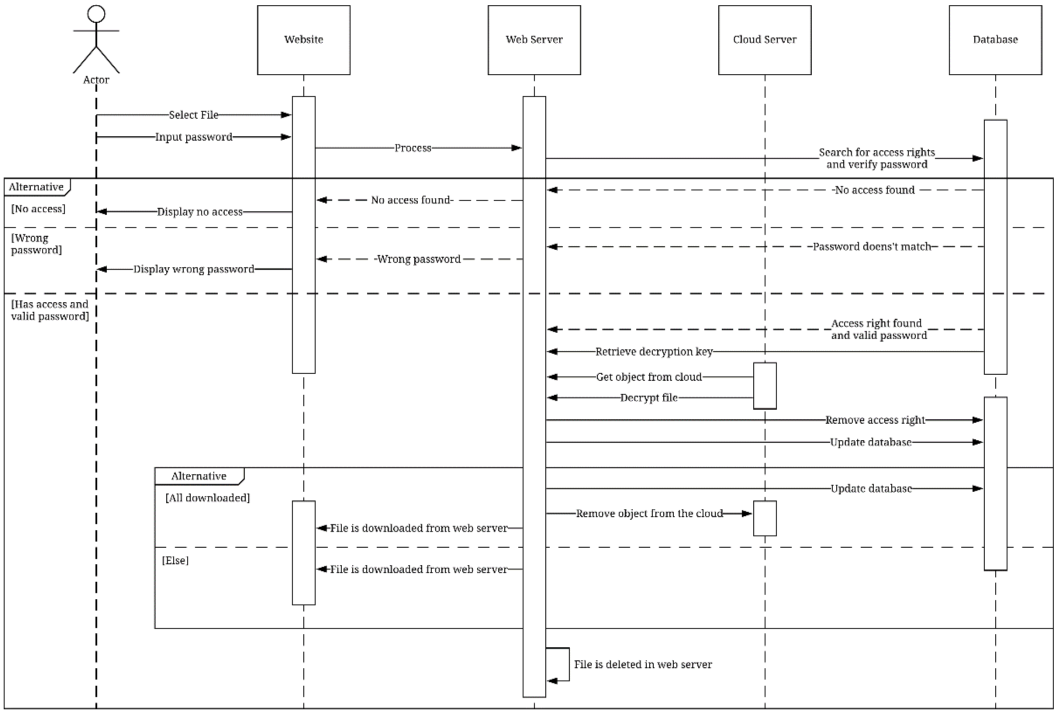


Figure 4.4.4: Sequence Diagram Download File

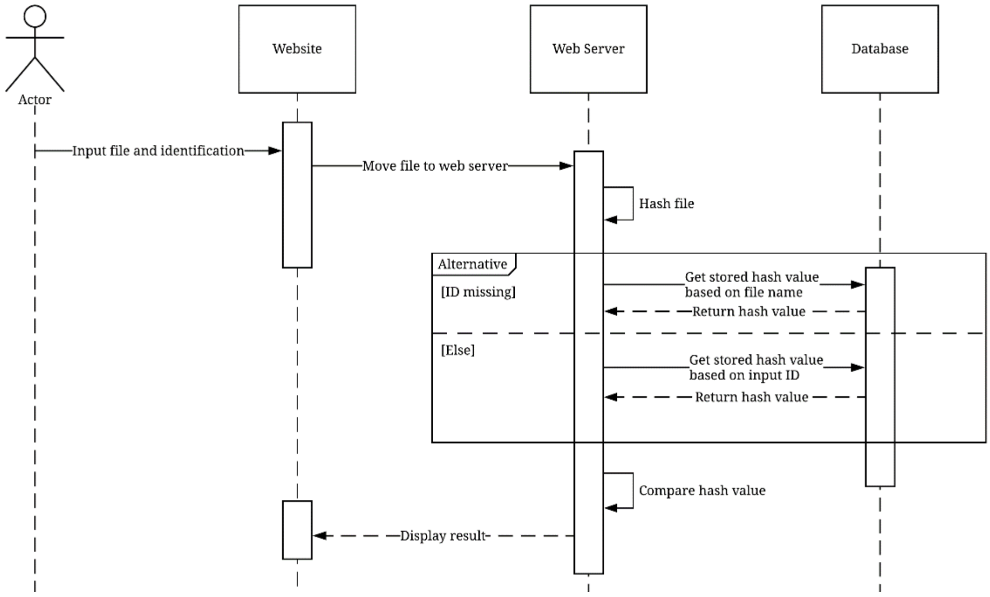
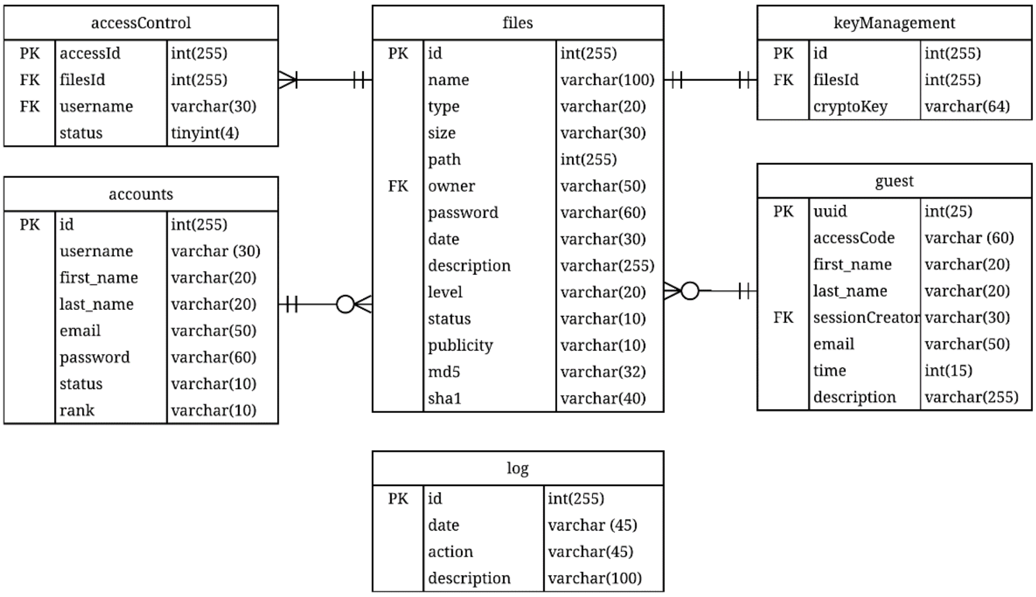


Figure 4.4.5: Verify File

4.5 Entity Relationship Diagram

The diagram shows the primary keys and the foreign keys for each table to know which attribute is unique. The attribute type is also declared in the ERD.



accessControl

PK	accessId	int(255)
FK	filesId	int(255)
FK	username	varchar(30)
	status	tinyint(4)

files

PK	id	int(255)
	name	varchar(100)
	type	varchar(20)
	size	varchar(30)
	path	int(255)
FK	owner	varchar(50)
	password	varchar(60)
	date	varchar(30)
	description	varchar(255)
	level	varchar(20)
	status	varchar(10)
	publicity	varchar(10)
	md5	varchar(32)
	sha1	varchar(40)

keyManagement

PK	id	int(255)
FK	filesId	int(255)
	cryptoKey	varchar(64)

accounts

PK	id	int(255)
	username	varchar (30)
	first_name	varchar(20)
	last_name	varchar(20)
	email	varchar(50)
	password	varchar(60)
	status	varchar(10)
	rank	varchar(10)

guest

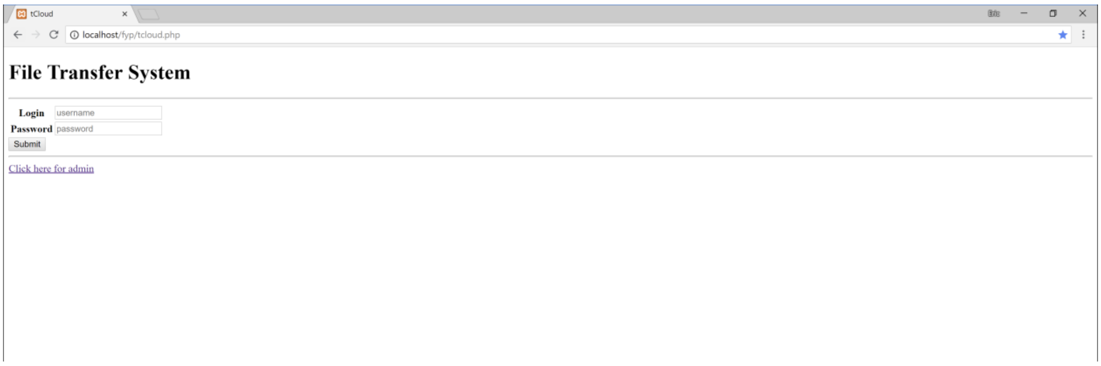
PK	uuid	int(25)
	accessCode	varchar (60)
	first_name	varchar(20)
	last_name	varchar(20)
FK	sessionCreator	varchar(30)
	email	varchar(50)
	time	int(15)
	description	varchar(255)

log

PK	id	int(255)
	date	varchar (45)
	action	varchar(45)
	description	varchar(100)

Figure 4.5.1: ERD Diagram

4.6 Prototype



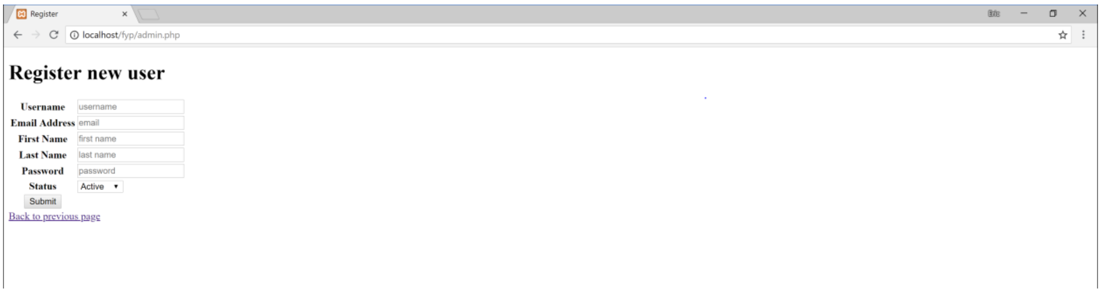
File Transfer System

Login

Password

Click here for admin

Figure 4.6.1: Login page for users and admin prototype



Register new user

Username

Email Address

First Name

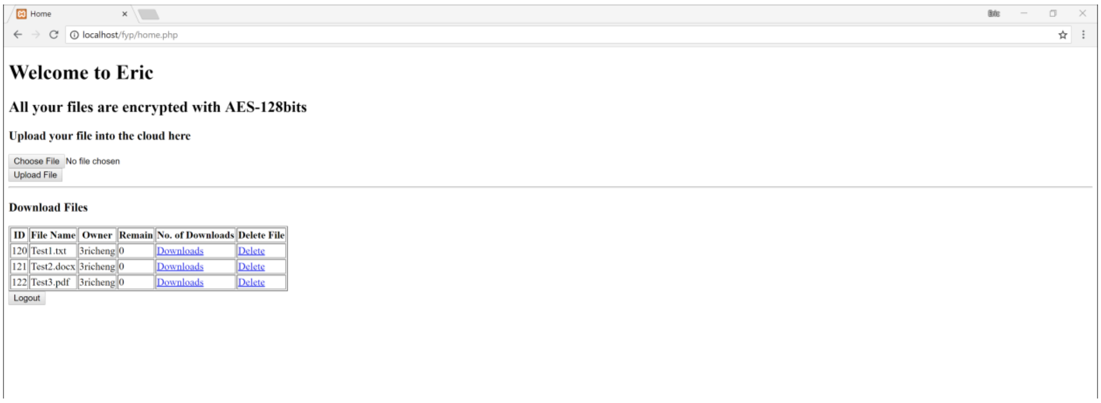
Last Name

Password

Status Active

Back to previous page

Figure 4.6.2: Register page prototype



Welcome to Eric

All your files are encrypted with AES-128bits

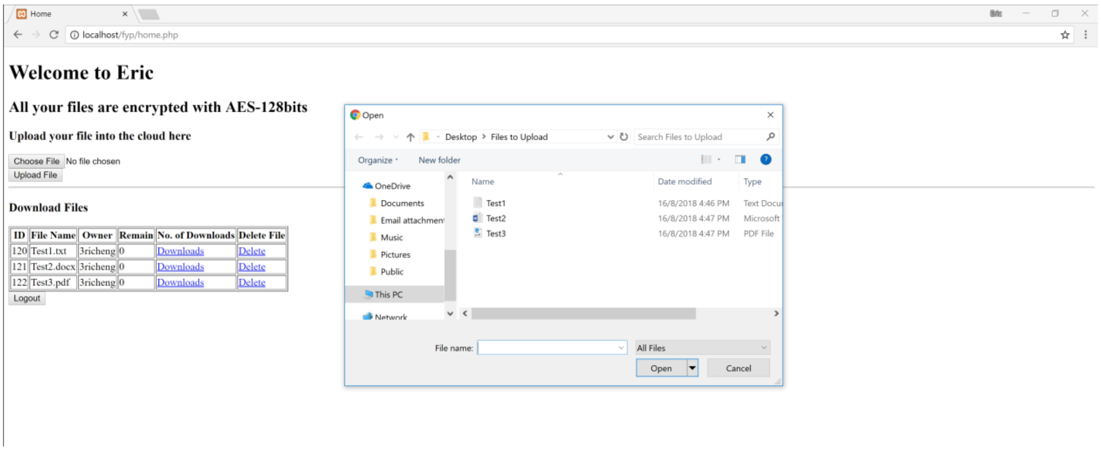
Upload your file into the cloud here

No file chosen

Download Files

ID	File Name	Owner	Remain	No. of Downloads	Delete File
120	Test1.txt	3richeng	0	Downloads	Delete
121	Test2.docx	3richeng	0	Downloads	Delete
122	Test3.pdf	3richeng	0	Downloads	Delete

Figure 4.6.3: Home page prototype



Welcome to Eric

All your files are encrypted with AES-128bits

Upload your file into the cloud here

Download Files

ID	File Name	Owner	Remain	No. of Downloads	Delete File
120	Test1.txt	3richeng	0	Downloads	Delete
121	Test2.docx	3richeng	0	Downloads	Delete
122	Test3.pdf	3richeng	0	Downloads	Delete

Name	Date modified	Type
Test1	16/8/2018 4:46 PM	Text Docu
Test2	16/8/2018 4:47 PM	Microsoft
Test3	16/8/2018 4:47 PM	PDF File

File name:

All Files

Figure 4.6.4: Upload file in home page prototype

CHAPTER 5

THE SOLUTION

5.1 Overview

In this chapter, it covers the explanation of the main functionalities of the system regarding on the web service. As we have discussed, the aim and goals of this web service are to implement a secure way for users to upload their file with a strong encryption algorithm and decrypt their files.

5.2 File

In the project, multiple web languages are integrated to make up to this project. The web languages used are discussed earlier which are HTML, PHP, JavaScript, MySQL and so on. From this, we have used the PHP file type (.php) to write most of the main content of the project. To house the design language of the web page, CSS file type (.css) is used to format the beautification structure of the web contents through alongside with HTML. Besides that, to describe the dependencies of this project used such as PHPMailer and PHPSpreadsheet, JSON file type (.json) is used. Moreover, a JavaScript file type (.js) houses the JavaScript codes. Below are forty-eight files used throughout the development of the project

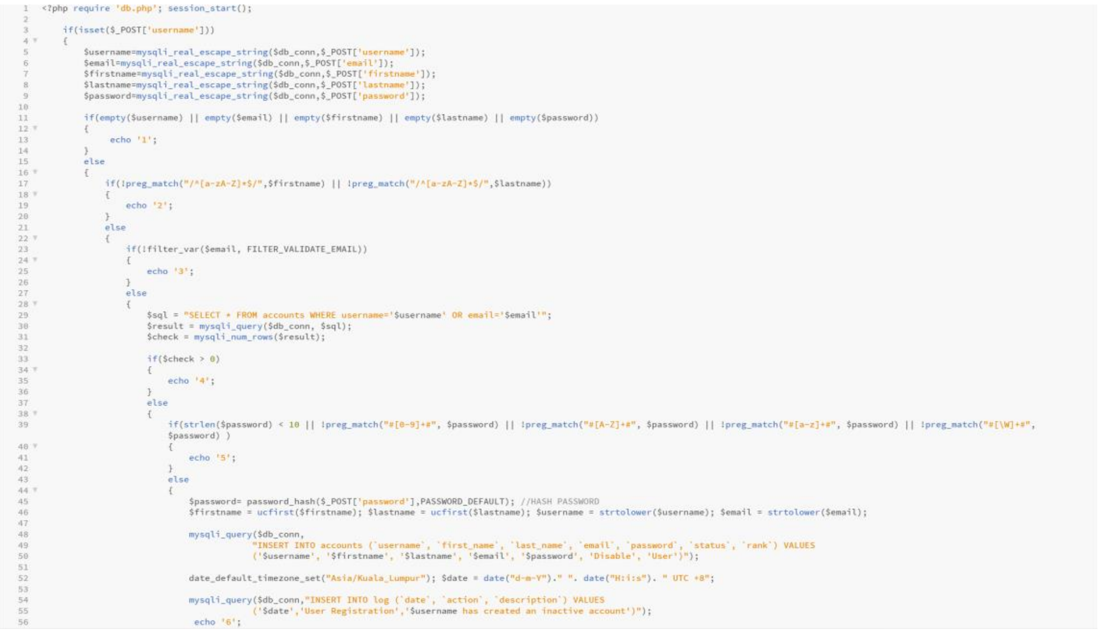
Table 5.2.1: List of Files

File Name	File Type	Description
accessCheck	php	Verify user access and file password
admin	css	Admin page design
admin	php	Admin page dashboard
adminLogin	php	Verify admin login information
adminOverview	php	Get database data for file and account overview
aws	php	AWS Keys
composer	json	Describe the dependencies
composer	lock	Exact version of the package used and installed.
contactAdmin	php	Post subjected information to admin

db	php	Database connection
decryption	php	Decryption algorithm
delete	php	Delete objects from S3 bucket
download	php	Define file header
downloadPage	css	Download page design
downloadPage	php	Download page dashboard
encryption	php	Encryption algorithm
exportData	php	Export uploaded file information to excel file
fetch_access	php	Print file details and file status information
fetch_record	php	Print file details
fileSize	php	Format file size details
generateGuest	php	Create guest session
getObject	php	Get selected file object for downloading
guestIndex	php	Guest login page
guestLogin	php	Guest login back-end process
guestPage	php	Create guest session page
guestUpload	php	Guest upload file page
home	css	Home page design
home	php	Home page
home	js	Interactivity in home
index	css	Login page design
index	php	Login page
login	php	Login back-end process
logout	php	Unset and destroy session
mainConfigure	php	Mail configuration
main	css	Design language for upload, verify and guest page
register	php	Register account back-end process
registerPage	css	Register account page design
registerPage	php	Register account page
tableData	php	Display download page data view
terms	js	Terms and Condition information
upload	php	Upload file back-end process
uploadPage	php	Upload file page

userAttributes	php	Display user’s information and web usage
userConfigure	php	Activate or Disable user account
userData	php	Display activated or disabled account or log data
verifyFile	php	Verification of download file
verifyPage	php	Verification page

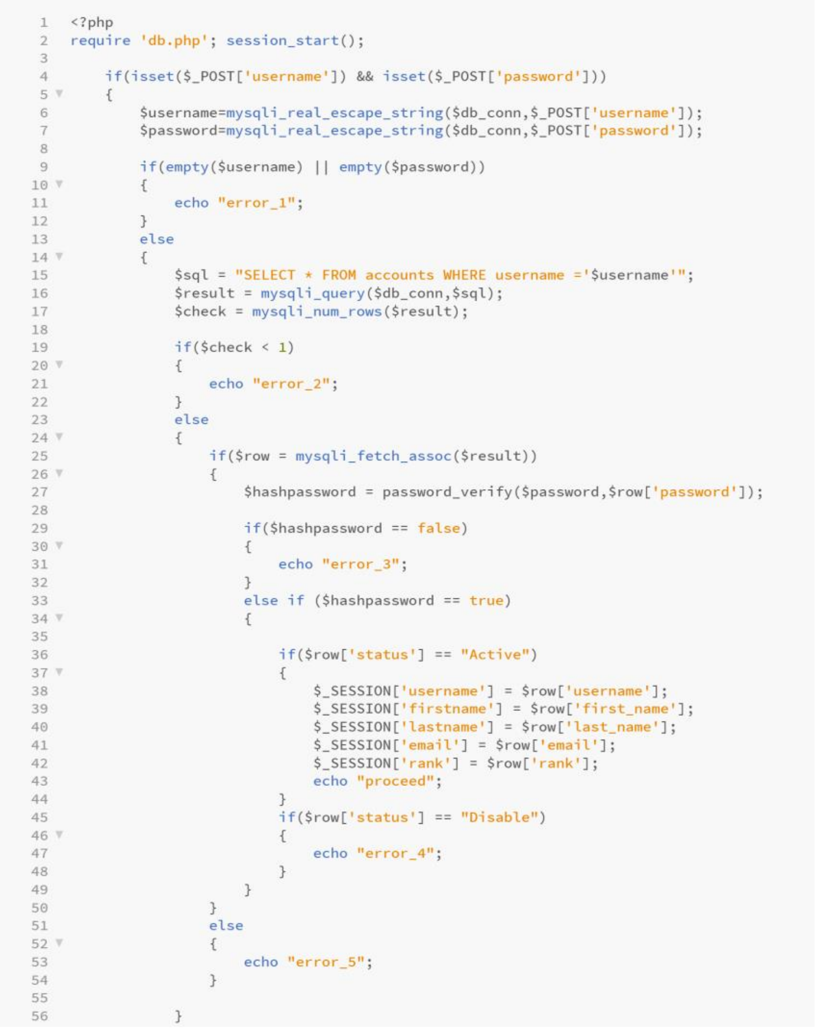
5.3 Login and Register



```php
<?php require 'db.php'; session_start();

    if(isset($_POST['username']))
    {
        $username=mysqli_real_escape_string($db_conn,$_POST['username']);
        $email=mysqli_real_escape_string($db_conn,$_POST['email']);
        $firstname=mysqli_real_escape_string($db_conn,$_POST['firstname']);
        $lastname=mysqli_real_escape_string($db_conn,$_POST['lastname']);
        $password=mysqli_real_escape_string($db_conn,$_POST['password']);

        if(empty($username) || empty($email) || empty($firstname) || empty($lastname) || empty($password))
        {
             echo '1';
        }
        else
        {
            if(!preg_match("/^[a-zA-Z]*$/",$firstname) || !preg_match("/^[a-zA-Z]*$/",$lastname))
            {
                echo '2';
            }
            else
            {
                if(!filter_var($email, FILTER_VALIDATE_EMAIL))
                {
                    echo '3';
                }
                else
                {
                    $sql = "SELECT * FROM accounts WHERE username='$username' OR email='$email'";
                    $result = mysqli_query($db_conn, $sql);
                    $check = mysqli_num_rows($result);

                    if($check > 0)
                    {
                        echo '4';
                    }
                    else
                    {
                        if(strlen($password) < 10 || !preg_match("#[0-9]+#", $password) || !preg_match("#[A-Z]+#", $password) || !preg_match("#[a-z]+#", $password) || !preg_match("#[\W]+#",
                        $password) )
                        {
                            echo '5';
                        }
                        else
                        {
                            $password= password_hash($_POST['password'],PASSWORD_DEFAULT); //HASH PASSWORD
                            $firstname = ucfirst($firstname); $lastname = ucfirst($lastname); $username = strtolower($username); $email = strtolower($email);

                            mysqli_query($db_conn,
                                        "INSERT INTO accounts (`username`, `first_name`, `last_name`, `email`, `password`, `status`, `rank`) VALUES
                                        ('$username', '$firstname', '$lastname', '$email', '$password', 'Disable', 'User')");

                            date_default_timezone_set("Asia/Kuala_Lumpur"); $date = date("d-m-Y")." ". date("H:i:s"). " UTC +8";

                            mysqli_query($db_conn,"INSERT INTO log (`date`, `action`, `description`) VALUES
                                        ('$date','User Registration','$username has created an inactive account')");
                             echo '6';
```

Figure 5.3.1: Registration process

Figure 5.3.1 above shows the code that is written in the “registration.php” file. The main purpose and goal of the following code is to create a user account to access into the website. As shown above, the PHP file gets the user input data from the POST request from the “registerPage.php” file and the data are processed with the built-in PHP function to prevent any potential risk of SQL injections. Once received, the retrieved information must go through a validation process to prevent any misleading, redundant, ambiguous and false data from the user. The first validation is to analyse for any missing field from the input data. Secondly, the validation process examines the first and last name of the user to prevent any special characters or symbols from their given input. Thirdly, the email address of the user is checked to prevent any invalid data entry followed by validating for existing username or existing email address as they are required to be unique. Lastly, a strong password validation with a minimum of ten characters including at least a numeric, lowercase, uppercase and special character is required to as it makes it difficult for humans and computer attacks like brute force attack which makes it effective to prevent any unauthorised access. Once all the data is analysed, the user data is inserted to the RDS database for the admin to authorise the account.



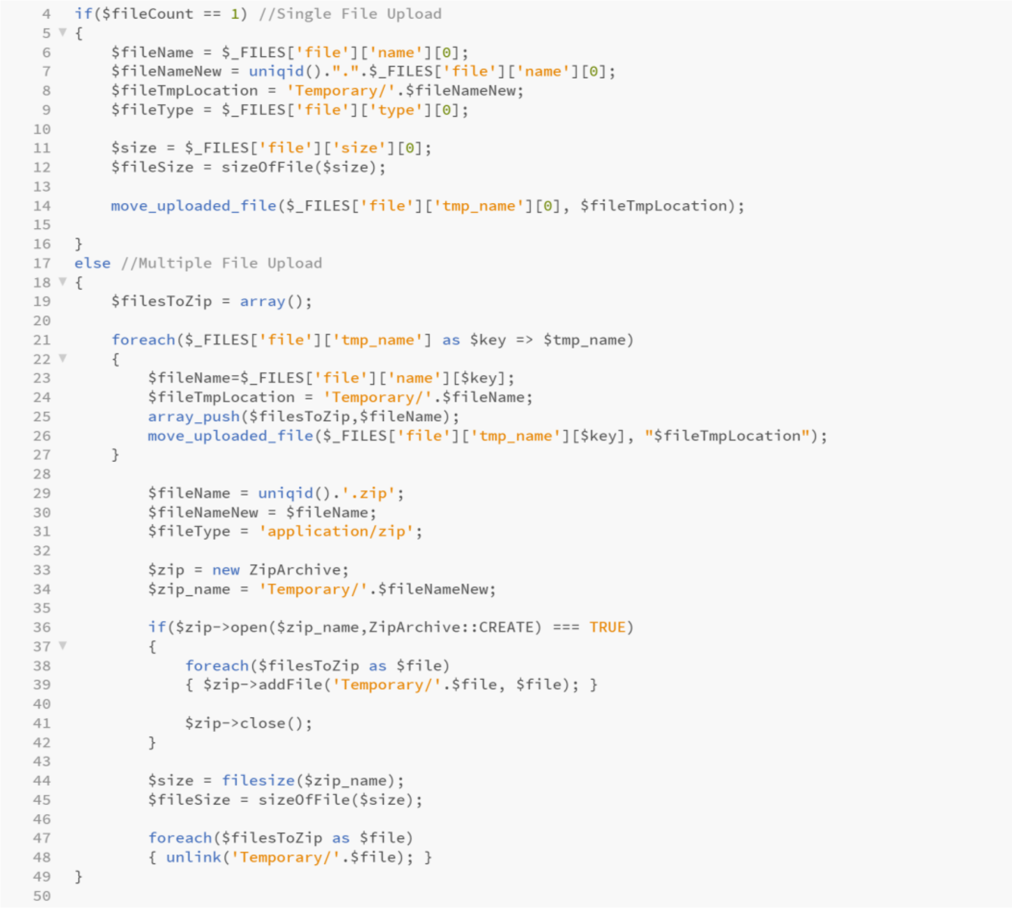
```php
<?php
require 'db.php'; session_start();

    if(isset($_POST['username']) && isset($_POST['password']))
    {
        $username=mysqli_real_escape_string($db_conn,$_POST['username']);
        $password=mysqli_real_escape_string($db_conn,$_POST['password']);

        if(empty($username) || empty($password))
        {
            echo "error_1";
        }
        else
        {
            $sql = "SELECT * FROM accounts WHERE username ='$username'";
            $result = mysqli_query($db_conn,$sql);
            $check = mysqli_num_rows($result);

            if($check < 1)
            {
                echo "error_2";
            }
            else
            {
                if($row = mysqli_fetch_assoc($result))
                {
                    $hashpassword = password_verify($password,$row['password']);

                    if($hashpassword == false)
                    {
                        echo "error_3";
                    }
                    else if ($hashpassword == true)
                    {

                        if($row['status'] == "Active")
                        {
                            $_SESSION['username'] = $row['username'];
                            $_SESSION['firstname'] = $row['first_name'];
                            $_SESSION['lastname'] = $row['last_name'];
                            $_SESSION['email'] = $row['email'];
                            $_SESSION['rank'] = $row['rank'];
                            echo "proceed";
                        }
                        if($row['status'] == "Disable")
                        {
                            echo "error_4";
                        }
                    }
                }
                else
                {
                    echo "error_5";
                }

            }
```

Figure 5.3.2: Login process

The figure above is the written login process in the file “login.php”. This code is built to authenticate the users into the system. The user identifies as whom they claim to be by the username and password. These data are received with the POST request from the file “index.php”. The data goes through a validation process starting with checking for empty input field either username or password. Secondly, the process checks if user whom they claim to exist in the database followed by verifying the user’s password. Next, the process verifies the status of the user. Only users with the status of active are allegeable to be authorized to the systems and PHP generates a session variable for the authorized user.

5.4 Upload

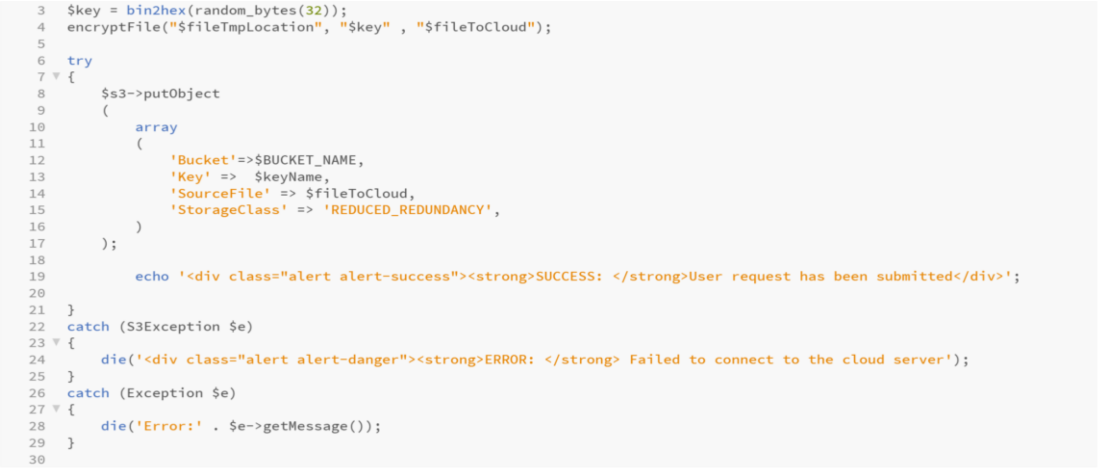
As discussed above, one of the main functionalities of this system is to upload a file. These file requests are handled in the “uploadPage.php” file and with the functionalities from AJAX “FromData” and “XMLHttpRequest”



```php
if($fileCount == 1) //Single File Upload
{
    $fileName = $_FILES['file']['name'][0];
    $fileNameNew = uniqid().".".$_FILES['file']['name'][0];
    $fileTmpLocation = 'Temporary/'.$fileNameNew;
    $fileType = $_FILES['file']['type'][0];

    $size = $_FILES['file']['size'][0];
    $fileSize = sizeOfFile($size);

    move_uploaded_file($_FILES['file']['tmp_name'][0], $fileTmpLocation);

}
else //Multiple File Upload
{
    $filesToZip = array();

    foreach($_FILES['file']['tmp_name'] as $key => $tmp_name)
    {
        $fileName=$_FILES['file']['name'][$key];
        $fileTmpLocation = 'Temporary/'.$fileName;
        array_push($filesToZip,$fileName);
        move_uploaded_file($_FILES['file']['tmp_name'][$key], "$fileTmpLocation");
    }

        $fileName = uniqid().'.zip';
        $fileNameNew = $fileName;
        $fileType = 'application/zip';

        $zip = new ZipArchive;
        $zip_name = 'Temporary/'.$fileNameNew;

        if($zip->open($zip_name,ZipArchive::CREATE) === TRUE)
        {
            foreach($filesToZip as $file)
            { $zip->addFile('Temporary/'.$file, $file); }

            $zip->close();
        }

        $size = filesize($zip_name);
        $fileSize = sizeOfFile($size);

        foreach($filesToZip as $file)
        { unlink('Temporary/'.$file); }
}
```

Figure 5.4.1: Single or multiple file upload

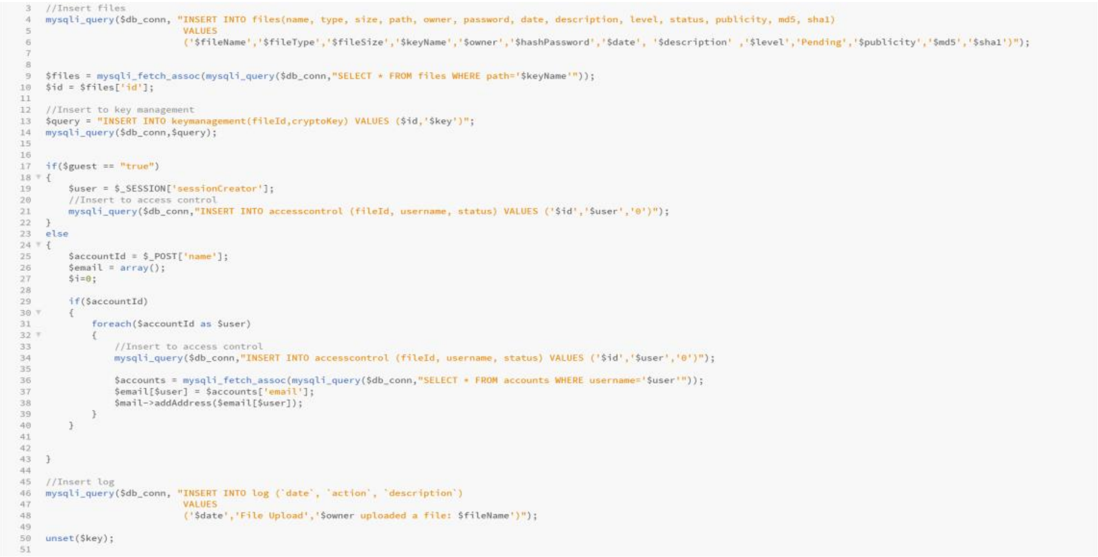
This website provides two features for the upload file functionalities which are single file upload or multiple file upload. The figure above shows how PHP handles with a single file and multiple file upload requests. For single file upload, the process only acquires the file information and moves to a temporary location in the web server for further processing. On the other hand, multiple file upload does relatively same as single file upload but PHP moves all the multiple selected file to the temporary location then zips all the file in a compressed file also known as a zip folder. Next, the zip folder is given with a unique identification name and the temporary folder is emptied.



```php
$key = bin2hex(random_bytes(32));
encryptFile("$fileTmpLocation", "$key" , "$fileToCloud");

try
{
    $s3->putObject
    (
        array
        (
            'Bucket'=>$BUCKET_NAME,
            'Key' =>  $keyName,
            'SourceFile' => $fileToCloud,
            'StorageClass' => 'REDUCED_REDUNDANCY',
        )
    );

        echo '<div class="alert alert-success"><strong>SUCCESS: </strong>User request has been submitted</div>';

}
catch (S3Exception $e)
{
    die('<div class="alert alert-danger"><strong>ERROR: </strong> Failed to connect to the cloud server');
}
catch (Exception $e)
{
    die('Error:' . $e->getMessage());
}

```

Figure 5.4.2: Upload file to the cloud

Once the file is in the web server, it is ready for the next step, encryption. A cryptographic key is generated. The file is passed to an encryption function to encode the intelligible text to an unintelligible form. Next, PHP is going to upload the file to the AWS S3 cloud storage. An array is used to declare the data which are the bucket name, key name, source file and the storage class. If the upload fails, PHP will catch the error message and displays the error code to the user.

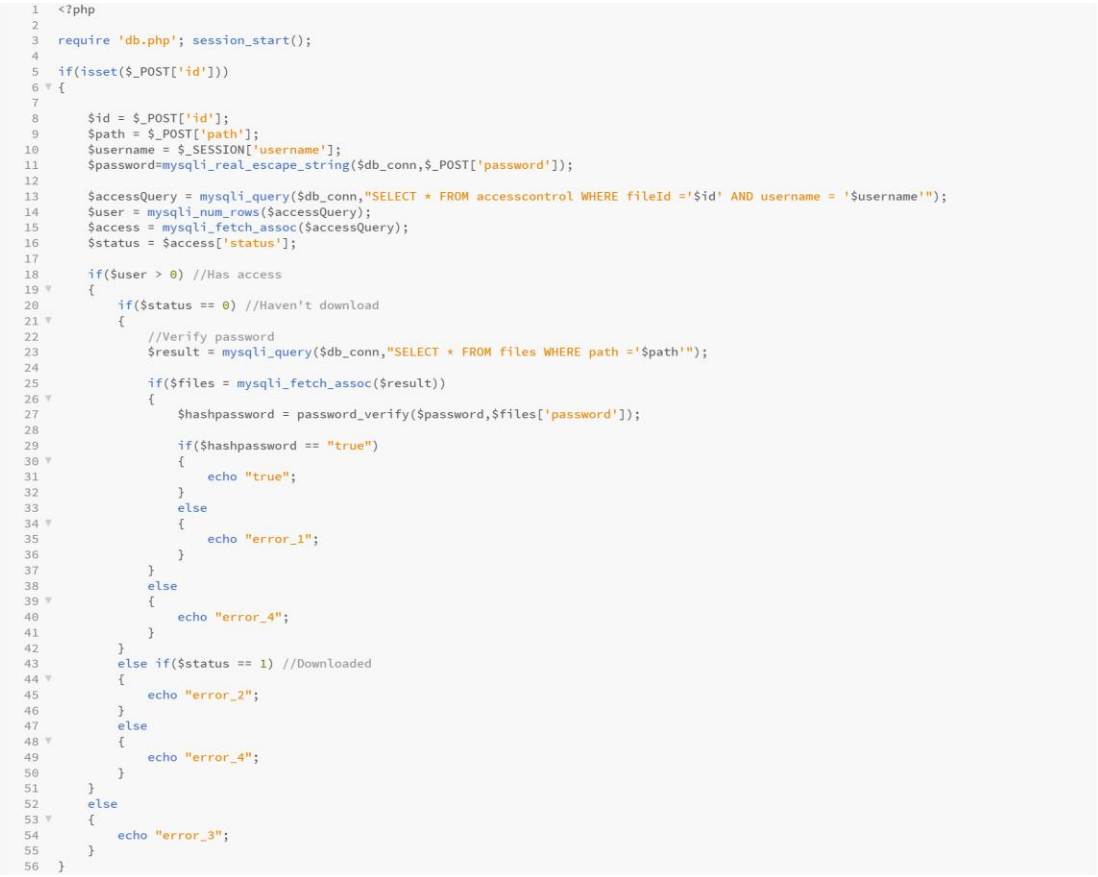


```php
//Insert files
mysqli_query($db_conn, "INSERT INTO files(name, type, size, path, owner, password, date, description, level, status, publicity, md5, sha1)
                        VALUES
                        ('$fileName','$fileType','$fileSize','$keyName','$owner','$hashPassword','$date', '$description' ,'$level','Pending','$publicity','$md5','$sha1')");


$files = mysqli_fetch_assoc(mysqli_query($db_conn,"SELECT * FROM files WHERE path='$keyName'"));
$id = $files['id'];

//Insert to key management
$query = "INSERT INTO keymanagement(fileId,cryptoKey) VALUES ($id,'$key')";
mysqli_query($db_conn,$query);


if($guest == "true")
{
    $user = $_SESSION['sessionCreator'];
    //Insert to access control
    mysqli_query($db_conn,"INSERT INTO accesscontrol (fileId, username, status) VALUES ('$id','$user','0')");
}
else
{
    $accountId = $_POST['name'];
    $email = array();
    $i=0;

    if($accountId)
    {
        foreach($accountId as $user)
        {
            //Insert to access control
            mysqli_query($db_conn,"INSERT INTO accesscontrol (fileId, username, status) VALUES ('$id','$user','0')");

            $accounts = mysqli_fetch_assoc(mysqli_query($db_conn,"SELECT * FROM accounts WHERE username='$user'"));
            $email[$user] = $accounts['email'];
            $mail->addAddress($email[$user]);
        }
    }


}

//Insert log
mysqli_query($db_conn, "INSERT INTO log (`date`, `action`, `description`)
                        VALUES
                        ('$date','File Upload','$owner uploaded a file: $fileName')");

unset($key);

```

Figure 5.4.3: Insert data to the database

Once the file is encrypted and uploaded to the cloud, the data acquired from the files, cryptographic keys and access control are to be stored in the database. All the file status, file password, hash value and attributes are stored in the files table, the cryptographic keys used to encrypt the file is stored in the key management table. Next, the selected users selected by the uploader is inserted into the access control table, but for guest upload, the only the session creator is inserted into the access control table. Following, the email address of the selected users is retrieved then an email is sent to the recipients to notify them about the existence of the file. Lastly, a brief action and description are inserted to the log table and they cryptographic key is deleted.

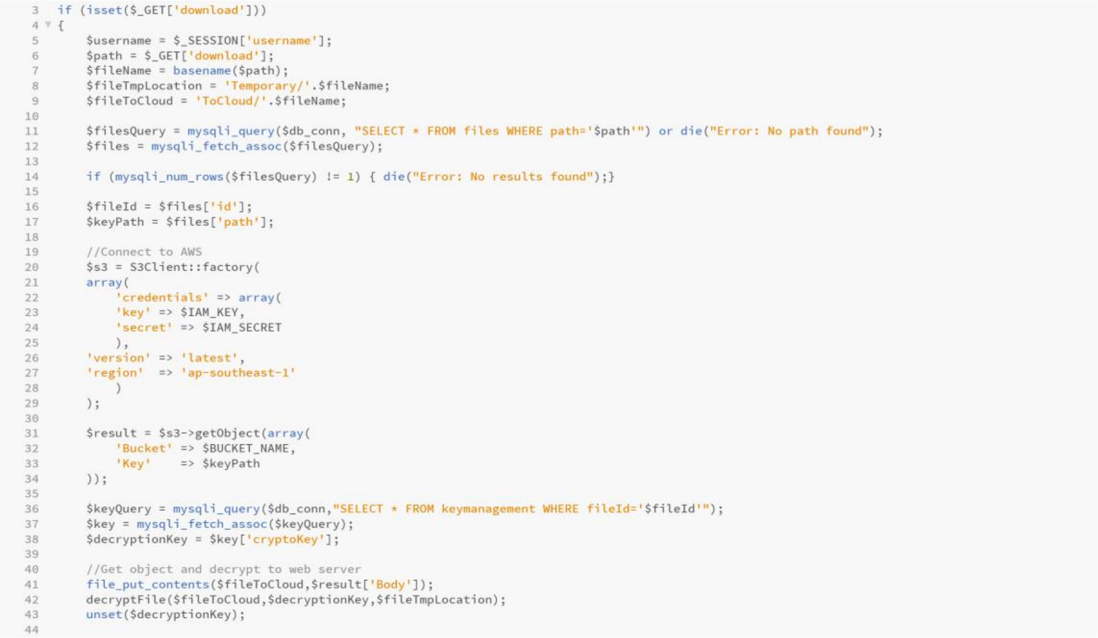
5.5 Download



```
<?php

require 'db.php'; session_start();

if(isset($_POST['id']))
{

    $id = $_POST['id'];
    $path = $_POST['path'];
    $username = $_SESSION['username'];
    $password=mysqli_real_escape_string($db_conn,$_POST['password']);

    $accessQuery = mysqli_query($db_conn,"SELECT * FROM accesscontrol WHERE fileId ='$id' AND username = '$username'");
    $user = mysqli_num_rows($accessQuery);
    $access = mysqli_fetch_assoc($accessQuery);
    $status = $access['status'];

    if($user > 0) //Has access
    {
        if($status == 0) //Haven't download
        {
            //Verify password
            $result = mysqli_query($db_conn,"SELECT * FROM files WHERE path ='$path'");

            if($files = mysqli_fetch_assoc($result))
            {
                $hashpassword = password_verify($password,$files['password']);

                if($hashpassword == "true")
                {
                    echo "true";
                }
                else
                {
                    echo "error_1";
                }
            }
            else
            {
                echo "error_4";
            }
        }
        else if($status == 1) //Downloaded
        {
            echo "error_2";
        }
        else
        {
            echo "error_4";
        }
    }
    else
    {
        echo "error_3";
    }
}
```

Figure 5.5.1: File access control

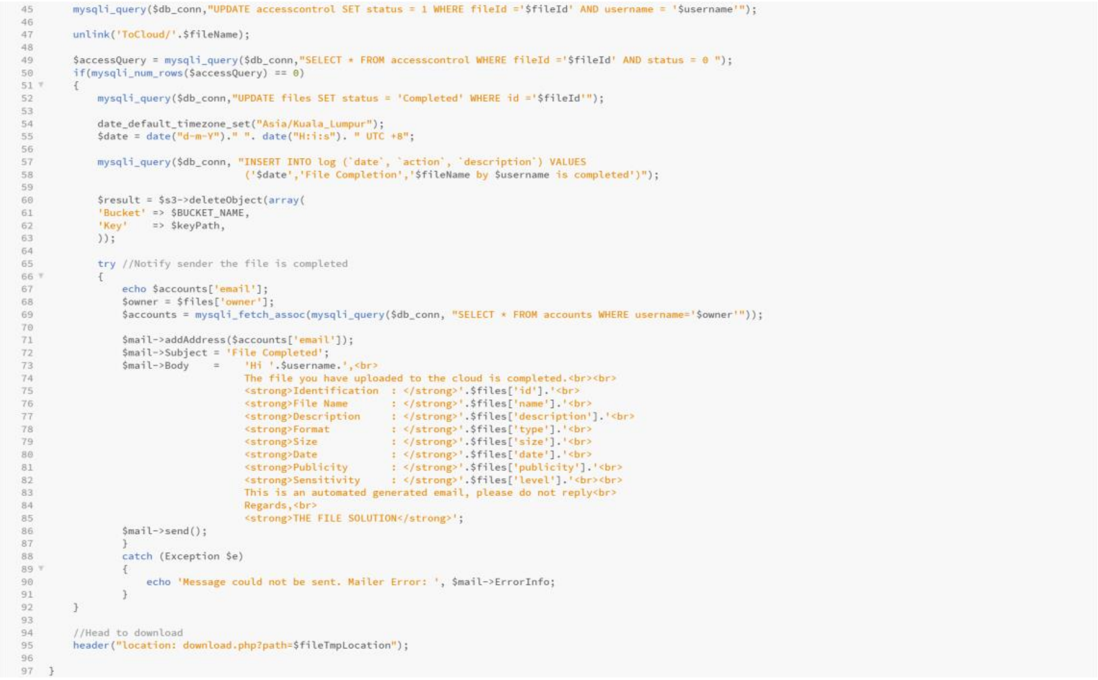
Every file uploaded, only the selected users are authorized to download the file. The file is protected with a password to provide another layer of security. The figure above shows the code for access control and it is written in the "accessCheck.php" file. Once the download button is triggered, PHP checks for whether the user has access to download the file. If the user has access and has already downloaded the file, the request is rejected as the user is only available to download it once. If the user has access and hasn't downloaded, PHP will verify the input password. Only with the correct password, PHP will trigger the download code.



```php
if (isset($_GET['download']))
{
    $username = $_SESSION['username'];
    $path = $_GET['download'];
    $fileName = basename($path);
    $fileTmpLocation = 'Temporary/'.$fileName;
    $fileToCloud = 'ToCloud/'.$fileName;

    $filesQuery = mysqli_query($db_conn, "SELECT * FROM files WHERE path='$path'") or die("Error: No path found");
    $files = mysqli_fetch_assoc($filesQuery);

    if (mysqli_num_rows($filesQuery) != 1) { die("Error: No results found");}

    $fileId = $files['id'];
    $keyPath = $files['path'];

    //Connect to AWS
    $s3 = S3Client::factory(
    array(
        'credentials' => array(
        'key' => $IAM_KEY,
        'secret' => $IAM_SECRET
        ),
    'version' => 'latest',
    'region'  => 'ap-southeast-1'
        )
    );

    $result = $s3->getObject(array(
        'Bucket' => $BUCKET_NAME,
        'Key'    => $keyPath
    ));

    $keyQuery = mysqli_query($db_conn,"SELECT * FROM keymanagement WHERE fileId='$fileId'");
    $key = mysqli_fetch_assoc($keyQuery);
    $decryptionKey = $key['cryptoKey'];

    //Get object and decrypt to web server
    file_put_contents($fileToCloud,$result['Body']);
    decryptFile($fileToCloud,$decryptionKey,$fileTmpLocation);
    unset($decryptionKey);

```

Figure 5.5.2: Fetch object

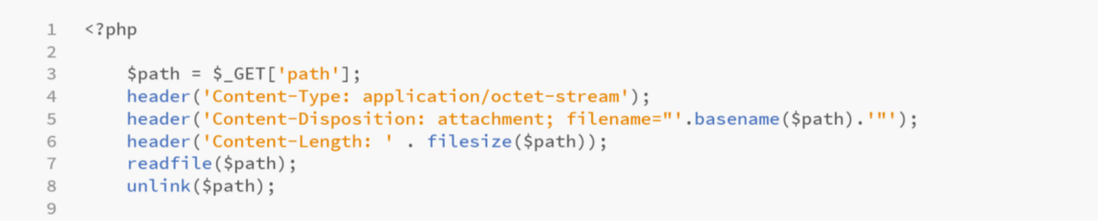
After checking for access, the code is redirected to "getObject.php" file. This file fetches the object from the cloud to the web server. Firstly, the PHP gets the data of the file selected and proceeds with the connection to AWS S3 Server by providing the AWS credentials like the IAM key and secret. Once connected, PHP gets the object from the S3 bucket and stores it in the web server. Then decryption key is retrieved from the database based on the identification of the file and the file is decrypted. Once decrypted, the key is unset.



```
    mysqli_query($db_conn,"UPDATE accesscontrol SET status = 1 WHERE fileId ='$fileId' AND username = '$username'");

    unlink('ToCloud/'.$fileName);

    $accessQuery = mysqli_query($db_conn,"SELECT * FROM accesscontrol WHERE fileId ='$fileId' AND status = 0 ");
    if(mysqli_num_rows($accessQuery) == 0)
    {
        mysqli_query($db_conn,"UPDATE files SET status = 'Completed' WHERE id ='$fileId'");

        date_default_timezone_set("Asia/Kuala_Lumpur");
        $date = date("d-m-Y")." ". date("H:i:s"). " UTC +8";

        mysqli_query($db_conn, "INSERT INTO log (`date`, `action`, `description`) VALUES
                                ('$date','File Completion','$fileName by $username is completed')");

        $result = $s3->deleteObject(array(
        'Bucket' => $BUCKET_NAME,
        'Key'    => $keyPath,
        ));

        try //Notify sender the file is completed
        {
            echo $accounts['email'];
            $owner = $files['owner'];
            $accounts = mysqli_fetch_assoc(mysqli_query($db_conn, "SELECT * FROM accounts WHERE username='$owner'"));

            $mail->addAddress($accounts['email']);
            $mail->Subject = 'File Completed';
            $mail->Body    =    'Hi '.$username.',<br>
                                The file you have uploaded to the cloud is completed.<br><br>
                                <strong>Identification  : </strong>'.$files['id'].'<br>
                                <strong>File Name       : </strong>'.$files['name'].'<br>
                                <strong>Description     : </strong>'.$files['description'].'<br>
                                <strong>Format          : </strong>'.$files['type'].'<br>
                                <strong>Size            : </strong>'.$files['size'].'<br>
                                <strong>Date            : </strong>'.$files['date'].'<br>
                                <strong>Publicity       : </strong>'.$files['publicity'].'<br>
                                <strong>Sensitivity     : </strong>'.$files['level'].'<br><br>
                                This is an automated generated email, please do not reply<br>
                                Regards,<br>
                                <strong>THE FILE SOLUTION</strong>';
            $mail->send();
            }
            catch (Exception $e)
            {
                echo 'Message could not be sent. Mailer Error: ', $mail->ErrorInfo;
            }
    }

    //Head to download
    header("location: download.php?path=$fileTmpLocation");

}
```

Figure 5.5.3: Download

Once decrypted, the database is updated. Next, PHP verifies if everyone has downloaded the file or not. If there are still some users who have yet to download the file, the file remains in the cloud. If all the selected users have downloaded the file, it will update the database stating that the file is completed, and the file is deleted from the cloud then an automated generated email is sent to the uploaders personal email address to notify him or her that their uploaded file is completed. Lastly, the code redirects to "download.php"

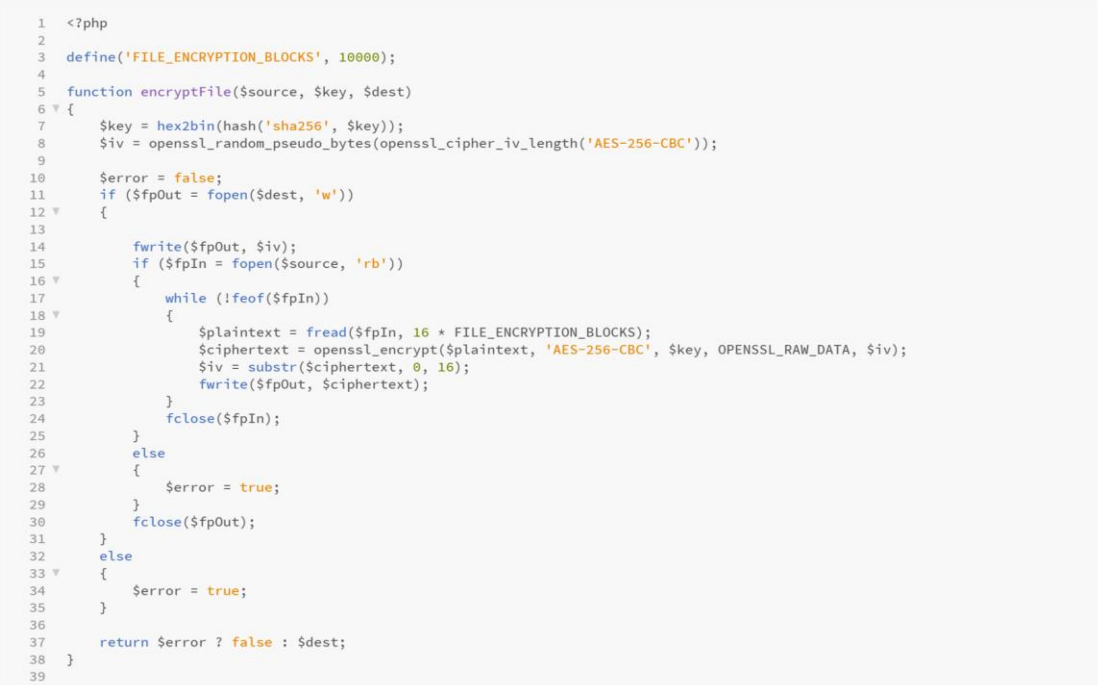


```
<?php

    $path = $_GET['path'];
    header('Content-Type: application/octet-stream');
    header('Content-Disposition: attachment; filename="'.basename($path).'"');
    header('Content-Length: ' . filesize($path));
    readfile($path);
    unlink($path);

```

Figure 5.5.4: Download header

In this last part, in this file, "download.php" it specifies the header content-type, content disposition and content length. This downloads the decrypted file from the web server to the user's personal computer.

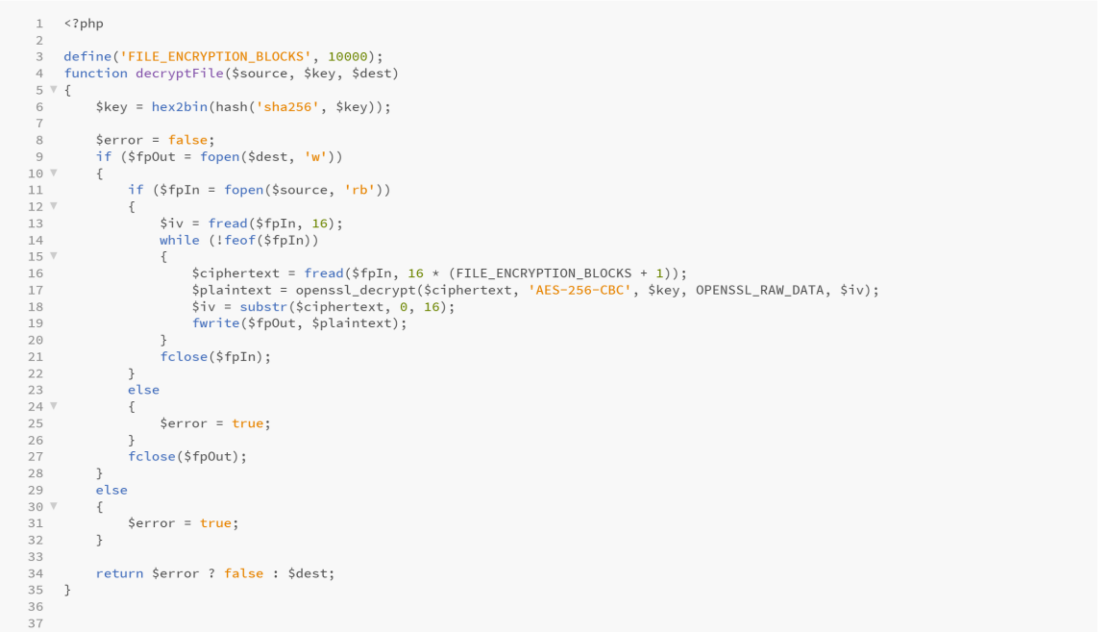
5.6 Cryptography



```php
<?php

define('FILE_ENCRYPTION_BLOCKS', 10000);

function encryptFile($source, $key, $dest)
{
    $key = hex2bin(hash('sha256', $key));
    $iv = openssl_random_pseudo_bytes(openssl_cipher_iv_length('AES-256-CBC'));

    $error = false;
    if ($fpOut = fopen($dest, 'w'))
    {

        fwrite($fpOut, $iv);
        if ($fpIn = fopen($source, 'rb'))
        {
            while (!feof($fpIn))
            {
                $plaintext = fread($fpIn, 16 * FILE_ENCRYPTION_BLOCKS);
                $ciphertext = openssl_encrypt($plaintext, 'AES-256-CBC', $key, OPENSSL_RAW_DATA, $iv);
                $iv = substr($ciphertext, 0, 16);
                fwrite($fpOut, $ciphertext);
            }
            fclose($fpIn);
        }
        else
        {
            $error = true;
        }
        fclose($fpOut);
    }
    else
    {
        $error = true;
    }

    return $error ? false : $dest;
}

```

Figure 5.6.1: Encryption

In this system, encryption is a mandatory as all the uploaded files are automatically encrypted with AES in the background. To begin encryption, the cryptographic key generated is hashed with a cryptographic hash function called SHA-256 which would return a thirty-two-byte character and declare an initialization vector. Next, the destination file is opened with write enabled and the initialization vector is placed at the beginning of the file. After that, the source file is opened with read enabled. The block size of the plaintext is captured from the source file and is passed through the encryption algorithm alongside with the cryptographic key. The ciphertext produced by the encryption algorithm is written to the destination file. This process is an ongoing looping process until the end-of-file is met.

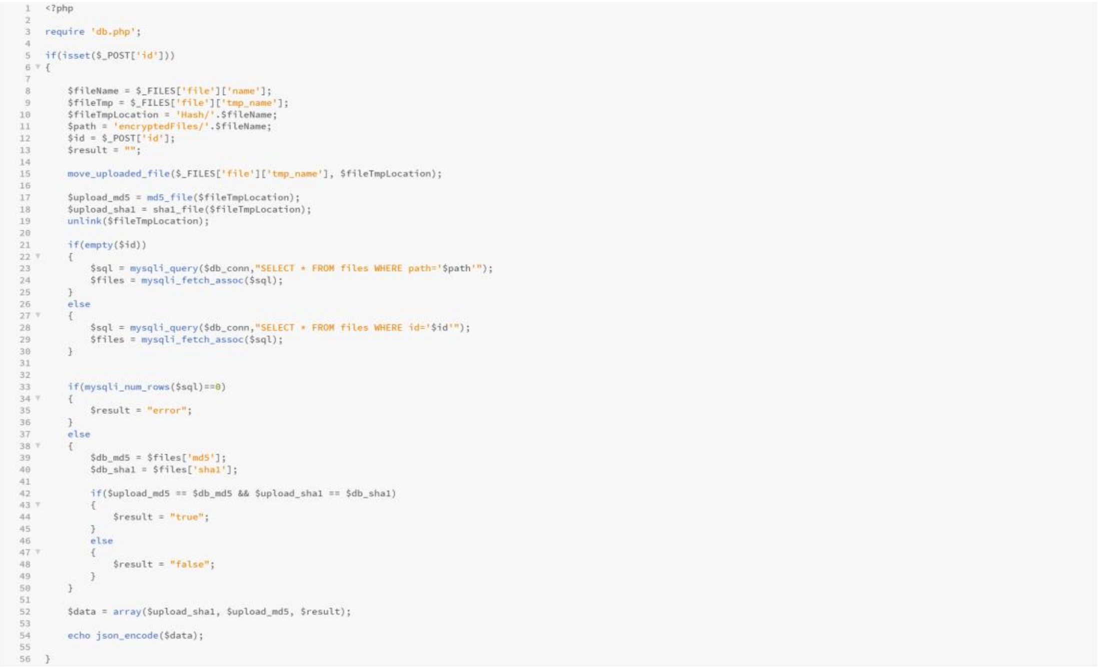


```php
<?php

define('FILE_ENCRYPTION_BLOCKS', 10000);
function decryptFile($source, $key, $dest)
{
    $key = hex2bin(hash('sha256', $key));

    $error = false;
    if ($fpOut = fopen($dest, 'w'))
    {
        if ($fpIn = fopen($source, 'rb'))
        {
            $iv = fread($fpIn, 16);
            while (!feof($fpIn))
            {
                $ciphertext = fread($fpIn, 16 * (FILE_ENCRYPTION_BLOCKS + 1));
                $plaintext = openssl_decrypt($ciphertext, 'AES-256-CBC', $key, OPENSSL_RAW_DATA, $iv);
                $iv = substr($ciphertext, 0, 16);
                fwrite($fpOut, $plaintext);
            }
            fclose($fpIn);
        }
        else
        {
            $error = true;
        }
        fclose($fpOut);
    }
    else
    {
        $error = true;
    }

    return $error ? false : $dest;
}


```

Figure 5.6.2: Decryption

Once an authorized user decides to download the file, this system must decrypt the following file. To start, the cryptographic key used to encrypt the file stored in the database is received and is hashed with SHA-256. The destination file is opened with write enabled and the source file opened with read enabled. The first sixteen bytes is read by the initialization vector. Next, the block size of the ciphertext is captured and goes through a decryption algorithm with the decryption key. The plaintext created from the ciphertext is written to the destination file. This process is an ongoing looping process until the end-of-file is reached.

5.7 Verification

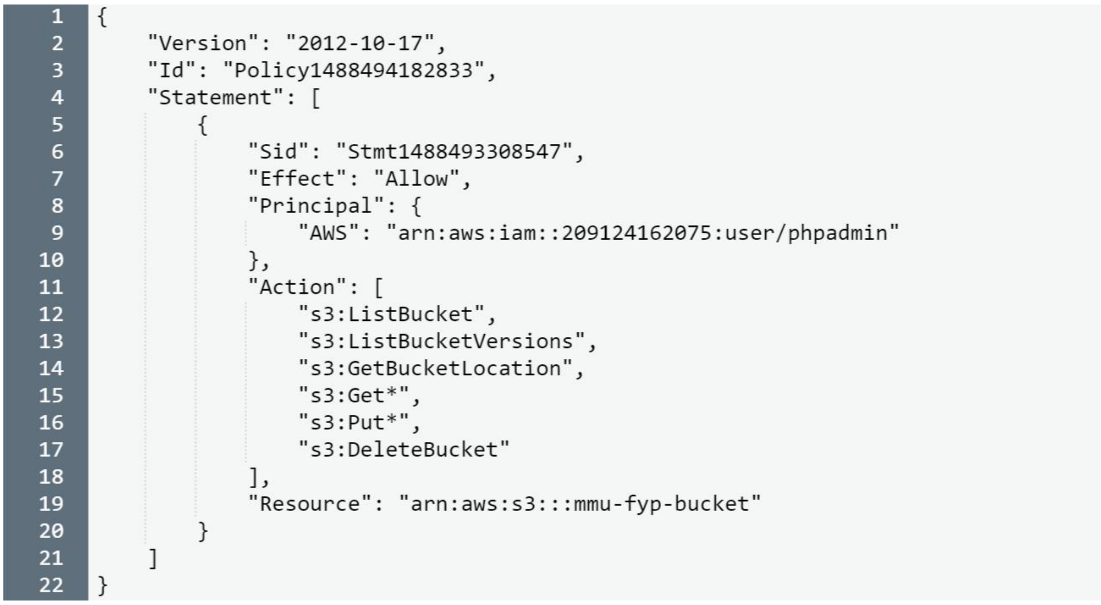


```php
<?php

require 'db.php';

if(isset($_POST['id']))
{

    $fileName = $_FILES['file']['name'];
    $fileTmp = $_FILES['file']['tmp_name'];
    $fileTmpLocation = 'Hash/'.$fileName;
    $path = 'encryptedFiles/'.$fileName;
    $id = $_POST['id'];
    $result = "";

    move_uploaded_file($_FILES['file']['tmp_name'], $fileTmpLocation);

    $upload_md5 = md5_file($fileTmpLocation);
    $upload_sha1 = sha1_file($fileTmpLocation);
    unlink($fileTmpLocation);

    if(empty($id))
    {
        $sql = mysqli_query($db_conn,"SELECT * FROM files WHERE path='$path'");
        $files = mysqli_fetch_assoc($sql);
    }
    else
    {
        $sql = mysqli_query($db_conn,"SELECT * FROM files WHERE id='$id'");
        $files = mysqli_fetch_assoc($sql);
    }


    if(mysqli_num_rows($sql)==0)
    {
        $result = "error";
    }
    else
    {
        $db_md5 = $files['md5'];
        $db_sha1 = $files['sha1'];

        if($upload_md5 == $db_md5 && $upload_sha1 == $db_sha1)
        {
            $result = "true";
        }
        else
        {
            $result = "false";
        }
    }

    $data = array($upload_sha1, $upload_md5, $result);

    echo json_encode($data);

}
```

Figure 5.7.1: File verification

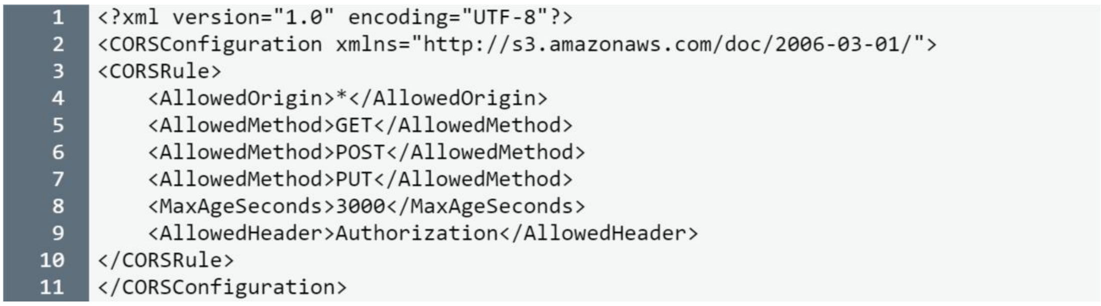
The figure above shows the method for verifying the file. This function is built to determine whether there is any corruption or damages of the file when after it has been downloaded. The user uploads the file that he/she wants to verify, and PHP gets the file details and goes through the hashing algorithm with are MD5 and SHA1. There are two ways to compare the hash value which are compare by file name or by file identification. Comparing the file name, the database gets the stored hash value in the database based on the name of the file or by identification if the user inputs the file identification. Both the generated and stored hash values are compared and if they produce the same value it means that the file has not been altered or corrupted by any means. The results return to display for users with JSON.

5.8 Cloud Policy



```json
{
    "Version": "2012-10-17",
    "Id": "Policy1488494182833",
    "Statement": [
        {
            "Sid": "Stmt1488493308547",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::209124162075:user/phpadmin"
            },
            "Action": [
                "s3:ListBucket",
                "s3:ListBucketVersions",
                "s3:GetBucketLocation",
                "s3:Get*",
                "s3:Put*",
                "s3:DeleteBucket"
            ],
            "Resource": "arn:aws:s3:::mmu-fyp-bucket"
        }
    ]
}
```

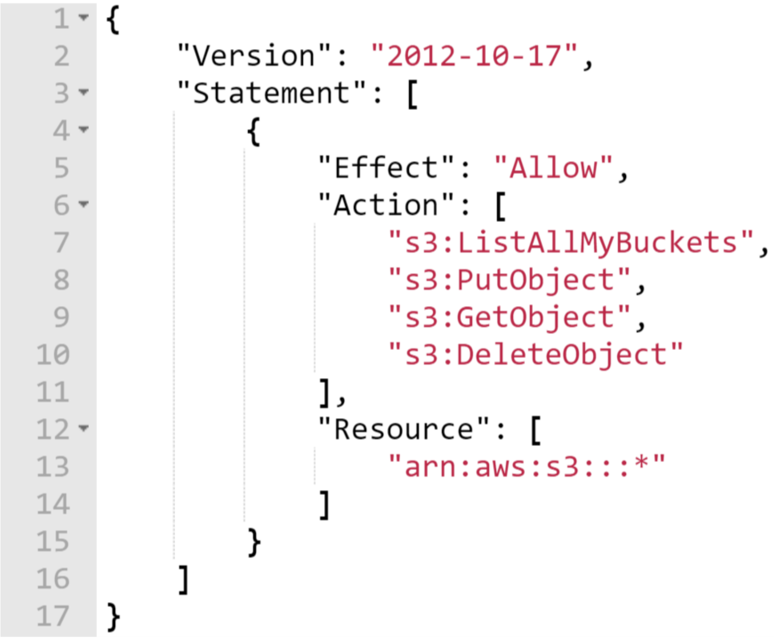
Figure 5.8.1: Bucket policy



```xml
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>PUT</AllowedMethod>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <AllowedHeader>Authorization</AllowedHeader>
</CORSRule>
</CORSConfiguration>
```

Figure 5.8.2: CORS

In figure 5.8.1, it shows the bucket policy of AWS S3 written in JSON. The bucket policy language manages the permission and accessibility of the bucket to the Amazon S3 resources. The policy declares the IAM principal, the list of the available actions that are to be performed by the bucket and the bucket name. Besides that, in the figure 5.8.2, it shows the policy for the cross-origin resource sharing (CORS) configuration and this policy written in Extensible Markup Language (XML) and it defines how the resources from different domain interacts with the clients' web application.



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ]
        }
    ]
}
```

Figure 5.8.3: IAM policy

The figure above shows the policy configuration for the Identify Access Management of the AWS user written in JSON and defines the permission for the user. The following code above, it lists the available action that can be made by the user and the resources of the AWS.

CHAPTER 6

THE RESULT

6.1 Overview

In this chapter, it shows the final display for the development of this project. The design has been changed back and forth during the development during the prototype developed in chapter 4. The user interface has been changed completely and may not have any resemblance compared to the prototype. The design shown in this chapter marks the final design. As more functionalities and web pages are added, the design of the web pages are styled with accordance to the theme of the whole website.

6.2 Web application

As this project is built on mark-up and scripting languages like HTML, CSS and JavaScript stated on chapter 4, a web browser is used like Google Chrome, Mozilla Firefox or Microsoft Edge serves as a platform to run the web application.

6.2.1 Login

For the login, there are two types of login. One is for the user and another is for the guest.

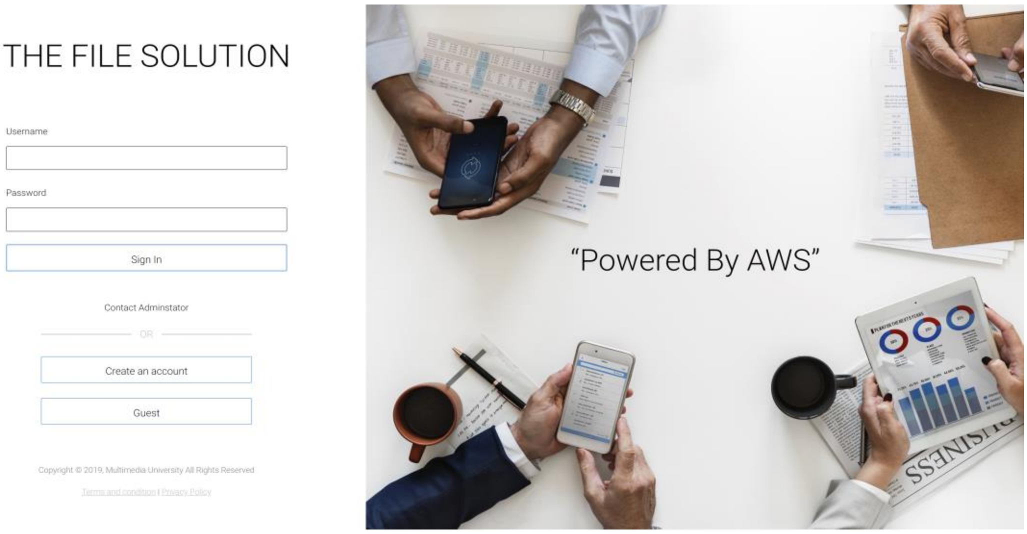


Figure 6.2.1: User Login

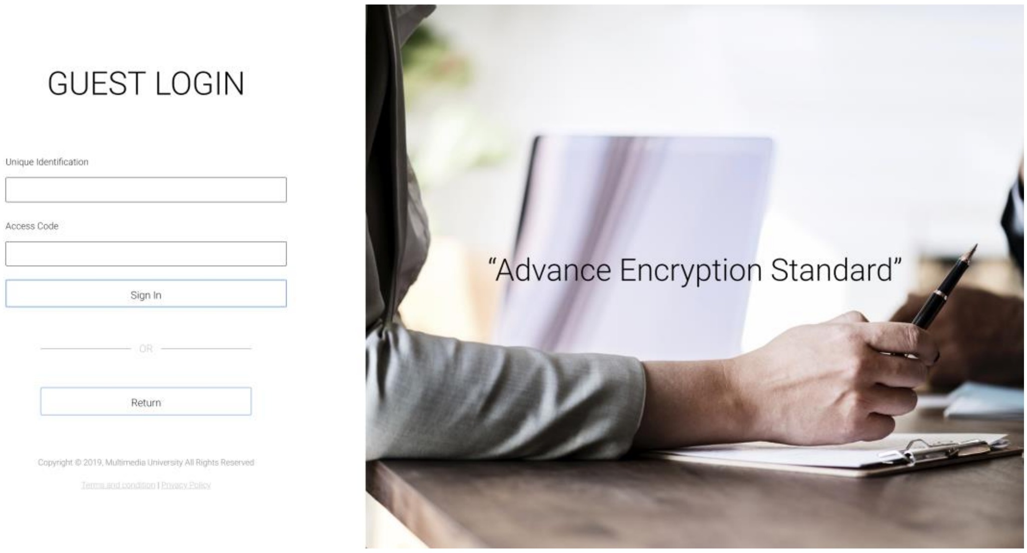

GUEST LOGIN
Unique Identification
Access Code
Sign In
OR
Return
Copyright © 2019, Multimedia University All Rights Reserved
Terms and condition | Privacy Policy
“Advance Encryption Standard”

Figure 6.2.2: Guest login

The login screen, there are divided into two segments. On the left, it is the input box for the identification of the user and guest. For user login, the user is required to insert their login and password while for the guest, they are required to insert their unique identification and access code which are generated when the session is created. Besides that, there is a button on the login page in figure 6.2.1 that redirects the user to the account registration and a form to contact the administrator for any sort of inquiries. On the right segment, it shows a randomized generated quote and displays the type error message for the user

6.2.2 Register

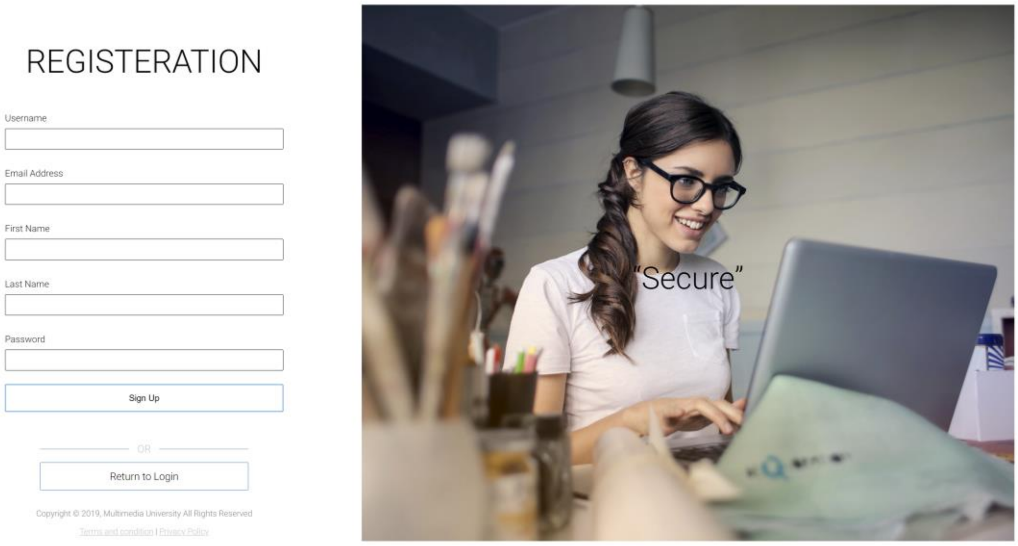

REGISTERATION
Username
Email Address
First Name
Last Name
Password
Sign Up
OR
Return to Login
Copyright © 2019, Multimedia University All Rights Reserved
Terms and condition | Privacy Policy
“Secure”

Figure 6.2.3: Register page

On the figure above, it shows the registration page. As it is the same as the login page, there are two segments. One for user input and another for display of randomized quotes and error messages. To create, the user is required to insert all the required username, email address, first name, last name and password. For the username and email address, there are required to be unique from every other user. For this website, it compiles a policy for a case-sensitive strong password. The password is required to contain with a minimum of ten characters including an uppercase, lowercase, numeric and special character. This is an effective method for preventing unauthorised access. On the other hand, any errors detected like a missing field or a weak password or an existing username and email will prompt the user about the type of error on the right side of the web page.

6.2.3 Home page

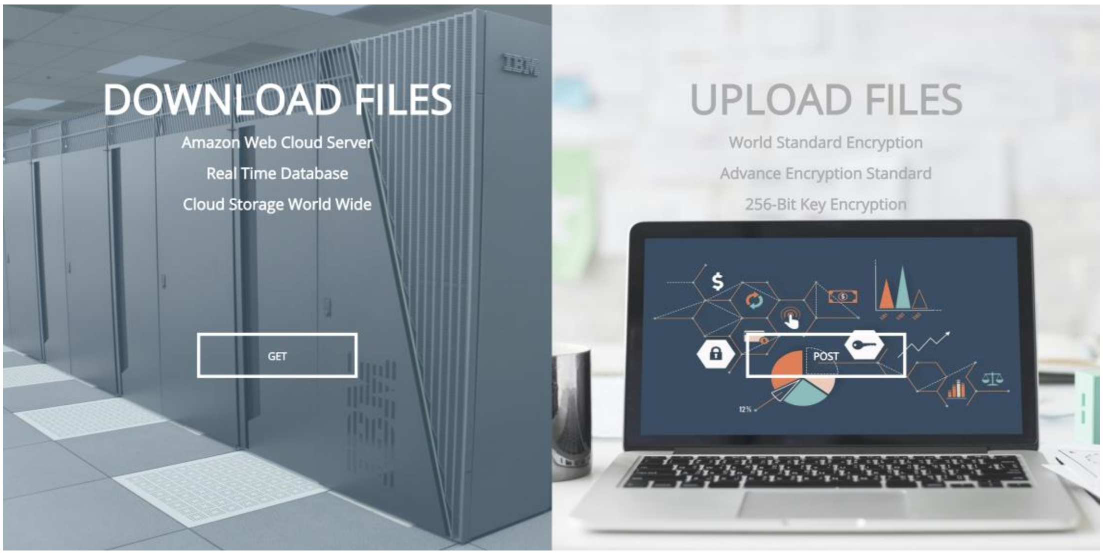
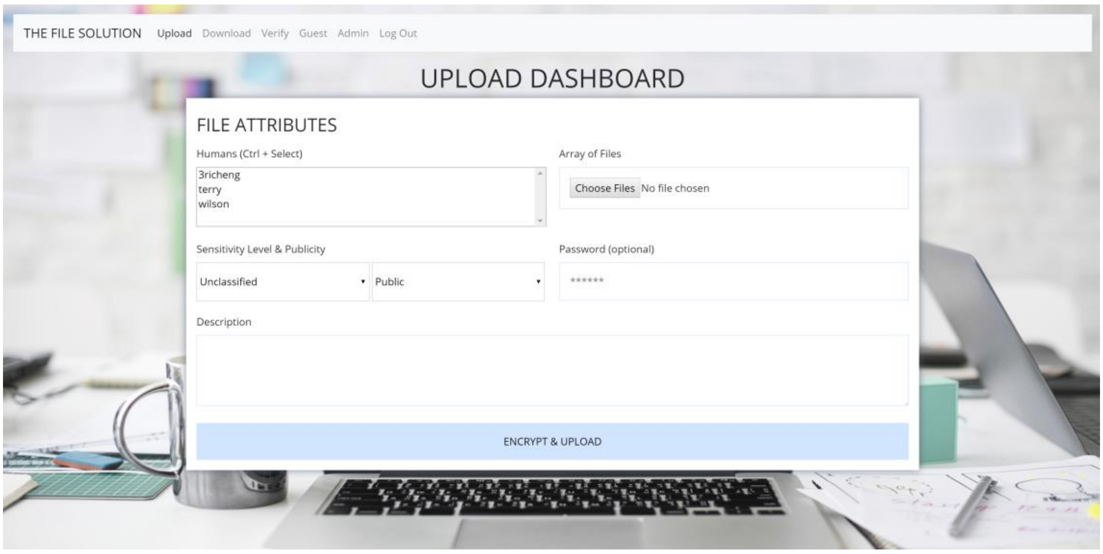


Figure 6.2.4: Home page

The figure above is the entry point to the website. It is an interactive web page that enlargers in size with accordance to the mouse pointer. This page redirects the user to either one of the web pages which are download files or upload file page with a click of a button

6.2.4 Upload page



UPLOAD DASHBOARD

FILE ATTRIBUTES

Humans (Ctrl + Select): 3richeng, terry, wilson

Array of Files: No file chosen

Sensitivity Level & Publicity: Unclassified, Public

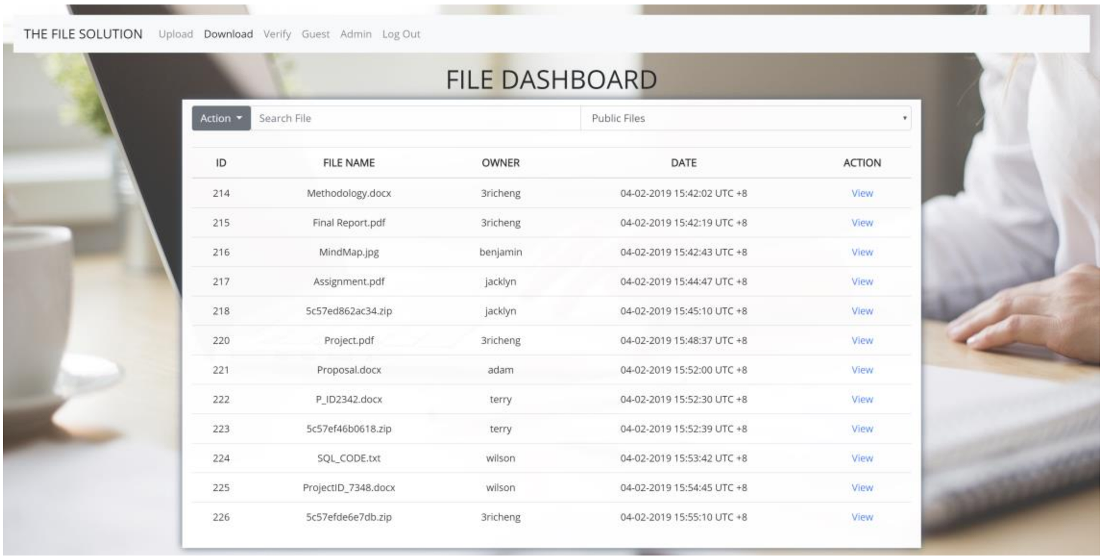
Password (optional): ******

Description:

Figure 6.2.5: Upload dashboard

Uploading files is one of the key features of this website as stated in the previous chapters. The figure above shows the web page for uploading files to the AWS S3 cloud server. In this page, the uppermost header shows a navigation bar that redirects the user to the other web pages. In the centre of the display shows a form for the user to select. First and foremost, the user is required to select who is authorized to download the following file. To select multiple users, just hold the control (CTRL) button and select the users. Secondly, the user may select which file they would like to upload. This service supports multiple file upload. If more than a file is uploaded, it will be automatically be converted into a compressed file (zip). Thirdly, the user may select other attributes like the sensitivity level and the publicity of the file. The sensitivity level is just to let the users know what type of file they are dealing with and for the publicity, it is used to define who can view the file. The file password is an optional feature that gives another layer of protection for the file and normal password of any length will do the job. Lastly, another optional field called description is for the users to describe the file they are dealing with. Once all the attributes are set, the user may submit their file to the AWS S3 cloud storage server.

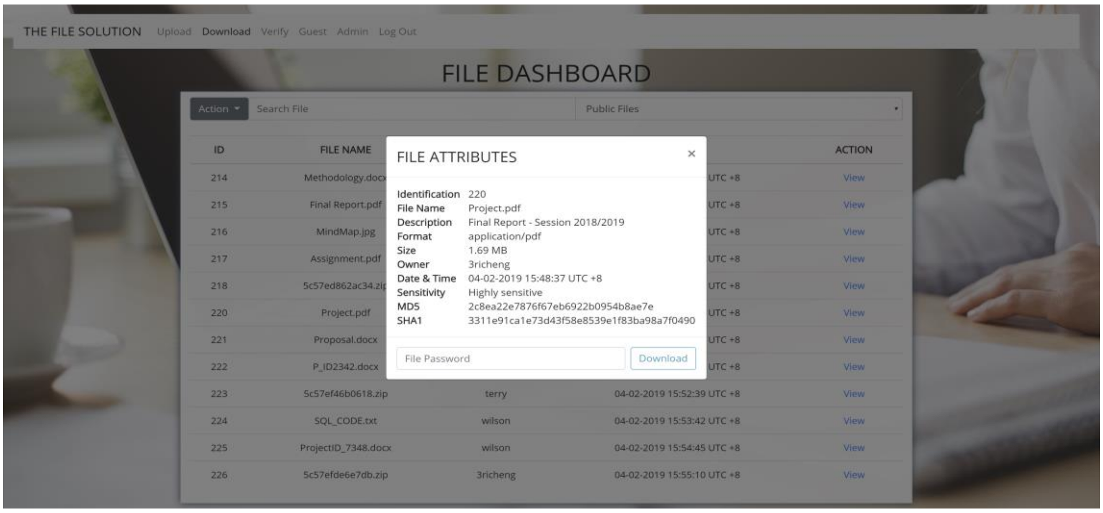
6.2.5 Download Page



ID	FILE NAME	OWNER	DATE	ACTION
214	Methodology.docx	3richeng	04-02-2019 15:42:02 UTC +8	View
215	Final Report.pdf	3richeng	04-02-2019 15:42:19 UTC +8	View
216	MindMap.jpg	benjamin	04-02-2019 15:42:43 UTC +8	View
217	Assignment.pdf	jacklyn	04-02-2019 15:44:47 UTC +8	View
218	5c57ed862ac34.zip	jacklyn	04-02-2019 15:45:10 UTC +8	View
220	Project.pdf	3richeng	04-02-2019 15:48:37 UTC +8	View
221	Proposal.docx	adam	04-02-2019 15:52:00 UTC +8	View
222	P_ID2342.docx	terry	04-02-2019 15:52:30 UTC +8	View
223	5c57ef46b0618.zip	terry	04-02-2019 15:52:39 UTC +8	View
224	SQL_CODE.txt	wilson	04-02-2019 15:53:42 UTC +8	View
225	ProjectID_7348.docx	wilson	04-02-2019 15:54:45 UTC +8	View
226	5c57efde6e7db.zip	3richeng	04-02-2019 15:55:10 UTC +8	View

Figure 6.2.6: Download dashboard

For every file uploaded, a list of selected users is able to download the files. The figure above shows the download web page. The following pages show the list of files uploaded by the users and there are four types of views which are view public files, available files, pending uploads and completed uploads. Each view generates a different result, for public files, it generates a result for all the public files in the database while available files display all the available files to download. For my pending files, the table shows the files that the uploader has uploaded but not everyone has downloaded the file and for completed files, it displays the list of completed files.

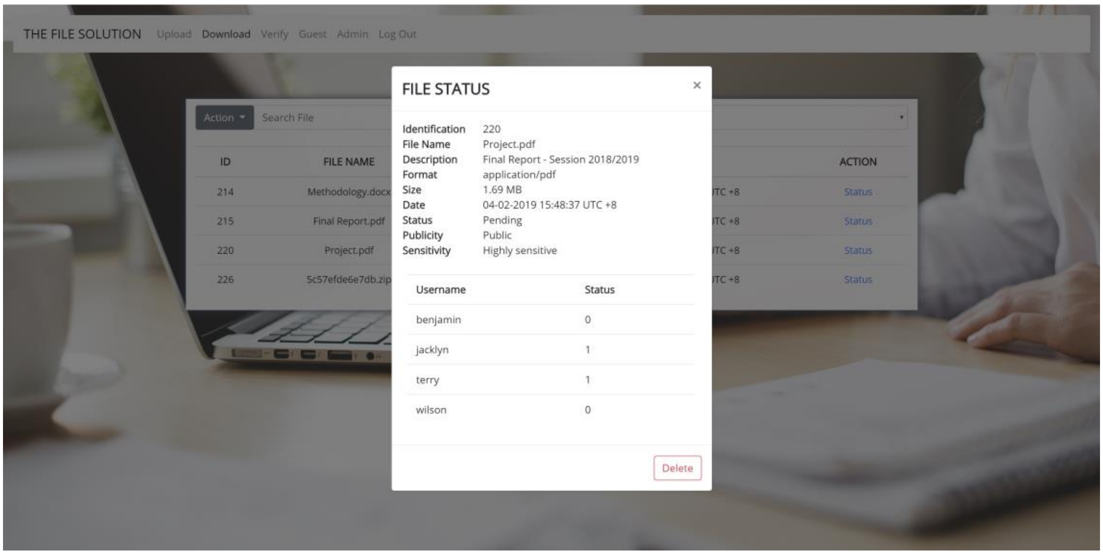


Identification	220
File Name	Project.pdf
Description	Final Report - Session 2018/2019
Format	application/pdf
Size	1.69 MB
Owner	3richeng
Date & Time	04-02-2019 15:48:37 UTC +8
Sensitivity	Highly sensitive
MD5	2c8ea22e7876f67eb6922b0954b8ae7e
SHA1	3311e91ca1e73d43f58e8539e1f83ba98a7f0490

Figure 6.2.7: File attributes

On the last cell of each table lies an action button. This button prompts a bootstrap modal that shows the file attributes as shown as figure 6.2.7. The main content of the modal displays the file identification, name, description, format, size, owner, date and time, sensitivity level, message digest 5 value and the secure hashing algorithm value. The footer of the model shows the input for the file password and the download button.

Besides that, above the file table, there is a function under actions used to export the list of all the users uploaded pending and completed file in Microsoft Excel. Moreover, there is also a search bar that performs quick searches for the content of the table. Furthermore, the table title of the table is clickable, and it sorts the content of the table in ascending or descending order.



ID	FILE NAME	ACTION
214	Methodology.docx	Status
215	Final Report.pdf	Status
220	Project.pdf	Status
226	5c57efde6e7db.zip	Status

FILE STATUS

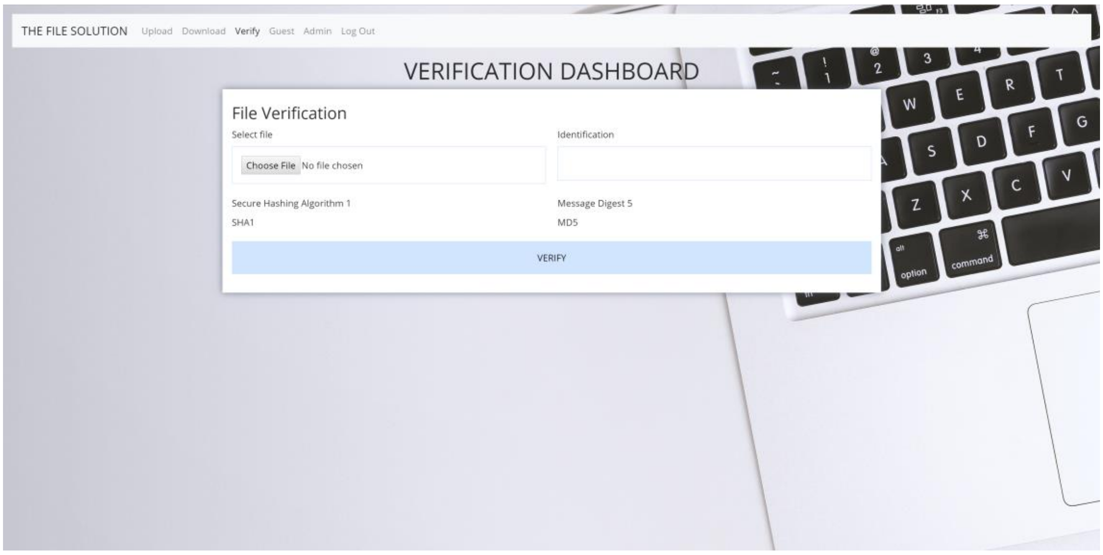
Identification	220
File Name	Project.pdf
Description	Final Report - Session 2018/2019
Format	application/pdf
Size	1.69 MB
Date	04-02-2019 15:48:37 UTC +8
Status	Pending
Publicity	Public
Sensitivity	Highly sensitive

Username	Status
benjamin	0
jacklyn	1
terry	1
wilson	0

Figure 6.2.8: File status

In my pending and completed views, the user can view all their file status of the file. The bootstrap modal shows the file attributes and the progress of the file. The selected users are printed along with their status, for the status of 0 means they have yet to download and 1 for those who have already downloaded the file. At the modal footer, the uploader is able to delete the file from the cloud server at their own wish.

6.2.6 Verification Page



VERIFICATION DASHBOARD

File Verification

Select file: No file chosen

Identification

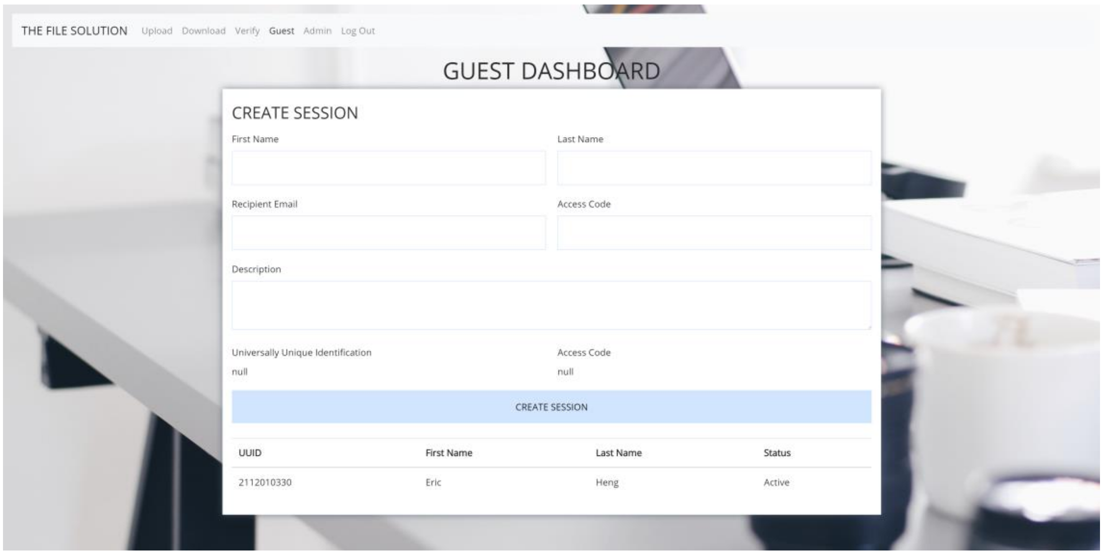
Secure Hashing Algorithm 1: SHA1

Message Digest 5: MD5

Figure 6.2.9: Verification dashboard

The figure above shows the verification dashboard used to detect any file alteration or corruption upon the download files. The user may select the file to verify. There are two methods to verify which are by using the file name extracted from the selected file or file identification. Once verified, the results will be printed out for the user and the MD5 and SHA1 of select file's hash value will also be printed out.

6.2.7 Create Guest Session



GUEST DASHBOARD

CREATE SESSION

First Name

Last Name

Recipient Email

Access Code

Description

Universally Unique Identification: null

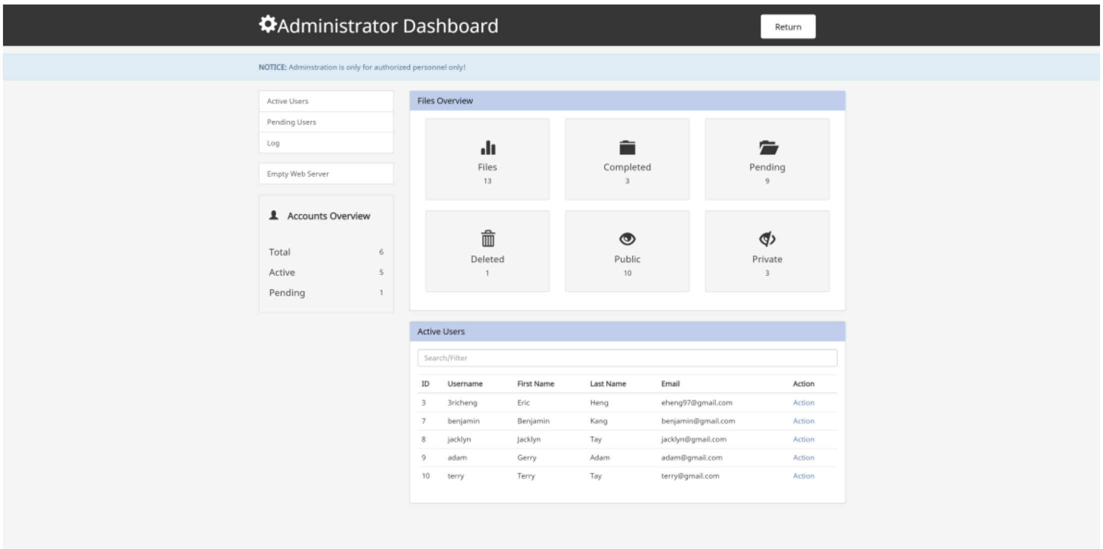
Access Code: null

UUID	First Name	Last Name	Status
2112010330	Eric	Heng	Active

Figure 6.2.10: Guest dashboard

In figure 6.2.10 shows a functionality used to create temporary access for external users to access the system temporarily to upload a couple of files for the internal users. This access temporary guest access that last only for an hour. To create this session, the internal users are required to insert some personal information about the external user which are the first name, last name and the recipient email. Besides that, the internal user is required to provide an access code which acts as a password. The access code is required needs to be a strong password which consist of a minimum of ten characters including an uppercase, lowercase, numeric and special character. An optional description is for the user to clarify or describe the reason for creating this temporary session. Once created, the page will prompt the user that the session has been created and display the unique identification generated and the access code.

6.2.8 Administrator Page



NOTICE: Adminstration is only for authorized personnel only!

Files Overview

Files 13	Completed 3	Pending 9
Deleted 1	Public 10	Private 3

Accounts Overview

Total	6
Active	5
Pending	1

Active Users

ID	Username	First Name	Last Name	Email	Action
3	3richeng	Eric	Heng	eheng97@gmail.com	Action
7	benjamin	Benjamin	Kang	benjamin@gmail.com	Action
8	jacklyn	Jacklyn	Tay	jacklyn@gmail.com	Action
9	adam	Gerry	Adam	adam@gmail.com	Action
10	terry	Terry	Tay	terry@gmail.com	Action

Figure 6.2.11: Administrator Dashboard

The figure above shows the administrator dashboard used to monitor the websites usage and the user's usage. The file usage of the website is displayed on the file overview section and the account status of the user is shown on the left side box. The admin can view the upload usage of every user. This page tabulates the view for the active users, pending users and the log of the processes.

CHAPTER 7

TESTING

7.1 Overview

In this chapter, it covers the four type of test cases used to determine the quality of the web service based on the objectives. Bertolino & Faedo (2007) states that in the simplest terms, it validates whether the software behaves as intended and identify potential malfunctions based on observing the execution of the system.

7.2 Unit Testing

In this testing, it checks for every single individual module of the source code to determine whether they are working properly. Bertolino & Faedo (2007) stated that in unit testing, it can detect even subtle and deeply-hidden faults which would not be able to be detected in system testing because it scrutinizes individual units in isolation. Therefore, it is an essential phase to assure quality assurance.

Table 7.2.1: Login Unit Testing

Login		
Test Case	**Test Results**	**Comment**
Missing username	PASS	
Missing password	PASS	
Valid login and password	PASS	
Invalid login and password	PASS	
Correct error message	PASS	
Contact administrator	PASS	

Table 7.2.2: Register Unit Testing

Create an account		
Test Case	**Test Results**	**Comment**
Missing username	PASS	
Missing email	PASS	

Missing first name	PASS	
Missing last name	PASS	
Missing password name	PASS	
Invalid email	PASS	
Invalid first and last name	PASS	
Strong password	PASS	
Correct error message	PASS	

Table 7.2.3: Guest Unit Testing

Guest Page		
Test Case	**Test Results**	**Comment**
Missing UUID	PASS	
Missing access code	PASS	
Valid UUID and access code	PASS	
Invalid UUID and access code	PASS	
Correct error message	PASS	

Table 7.2.4: Upload Unit Testing

Upload		
Test Case	**Test Results**	**Comment**
Working navigation bar	PASS	
Missing selected users	PASS	
Missing file	PASS	
Select multiple users	PASS	
Select single & multiple files	PASS	

Table 7.2.5: Download Unit Testing

Download		
Test Case	**Test Results**	**Comment**
Working navigation bar	PASS	
Search bar	PASS	
File sorting	PASS	
Table view	PASS	

View file attributes	PASS	
View file status	PASS	

Table 7.2.6: Verify Unit Testing

Verify		
Test Case	**Test Results**	**Comment**
Working navigation bar	PASS	
Select single file only	PASS	
Input numbers only in ID field	PASS	

Table 7.2.7: Guest Unit Testing

Guest		
Test Case	**Test Results**	**Comment**
Missing first name	PASS	
Missing last name	PASS	
Missing email	PASS	
Missing access code	PASS	
Invalid email	PASS	
Strong access code	PASS	
Correct error message	PASS	

Table 7.2.8: Admin Unit Testing

Admin		
Test Case	**Test Results**	**Comment**
Search bar	PASS	
View active and pending users	PASS	
View log	PASS	
Table sorting	PASS	

7.3 Integrated Testing

In this level of testing, individual units of the system are combined and are tested a group. Leung & White (1990) stated that integrated testing isn't at the statement level as in unit testing but rather done at the module level and emphasizes between the modules and their interfaces interaction.

Table 7.3.1: Login Integrated Testing

LOGIN	
TEST CASE	**RESULT**
User able to log in	PASS
Guest able to log in	PASS
Verify guest session	PASS

Table 7.3.2: Register Integrated Testing

REGISTER / SIGNUP	
TEST CASE	**RESULT**
User able to register	PASS
Verify duplicate username and email	PASS
Verify registered data inserted into database	PASS

Table 7.3.3: Upload Integrated Testing

UPLOAD FILE	
TEST CASE	**RESULT**
Users can upload the file to the cloud	PASS
File size limit	PASS
Display selectable approved users only	PASS
Uploaded file is encrypted	PASS
The file is stored in the cloud	PASS
Mail all selected users to download	PASS
Delete file in web server after uploading	FAIL

Table 7.3.4: Download Integrated Testing

DOWNLOAD FILE	
TEST CASE	**RESULT**
Able to tabulate a list of public files	PASS
Able to tabulate a list of available files	PASS
Able to tabulate uploaders pending file	PASS
Able to tabulate uploaders completed file	PASS
Verify file access right	PASS
Database update upon download	PASS
Display correct file attributes in modal	PASS
Display correct file status in modal	PASS
Generate users completed and pending file data in excel	PASS
File deleted in the cloud upon completion	PASS
Database update upon file completion	PASS
Mail uploader once the file is completed	PASS
Delete file in web server after downloading	FAIL

Table 7.3.5: Verify Integrated Testing

VERIFY FILE	
TEST CASE	**RESULT**
Able to verify file using the file name	PASS
Able to verify file using file identification	PASS
Able to hash selected file to verify	PASS
Retrieve stored hash value and compared with the selected file hash	PASS
Display status and display selected file hash value	PASS

Table 7.3.6: Guest Integrated Testing

GUEST FILE	
TEST CASE	**RESULT**
User able to create a guest session	PASS
Verify guest session data inserted into database	PASS

Display UUID and access code upon creation	PASS
Tabulate list of previously created session	PASS
Mail guest UUID and access code	PASS

Table 7.3.7: Admin Integrated Testing

ADMINISTRATOR	
TEST CASE	**RESULT**
Verify users’ administrator access privilege to access admin page	PASS
Display account overview	PASS
Display file overview	PASS
Tabulate active users	PASS
Tabulate pending users	PASS
Tabulate log	PASS
Retrieve and display user attributes and usage in the bootstrap modal	PASS
Approve users accounts	PASS
Deactivate user accounts	PASS

7.4 System Testing

In this level of testing, integrated and unit testing must be completed. This test the whole system and is concerned with the behaviour of the website as a whole unlike in integration testing which focuses among modules. Besides that, in system testing, it checks from end to end scenarios as of how the user would use this system. Other than that, apart from the functional test, a non-functional test is also done.

Table 7.4.1: System Testing

TEST CASE	**RESULT**
Website working in Chrome, Firefox and Microsoft Edge	PASS
Website working in Windows, Mac OS and Linux	PASS
Website working on a mobile phone	PASS
The user interface is straight forward, clean and consistent	PASS
Registration with validation and insert input data into the database	PASS
Upload files are encrypted to the cloud and database is updated	PASS

Objects are uploaded from the cloud in a quick manner	PASS
The upload page user interface is straight forward	PASS
Download files are only to those who or whom they have access to	PASS
Download file is deleted and the database is updated upon completion	PASS
Objects are retrieved from the cloud in a quick manner	PASS
The download page user interface is straight forward	PASS
Verify file is able to compare the hash value	PASS
Verify file is easy to use and straight forward	PASS
Create guest session is easy and straight forward	PASS
Admin can view the overall performance and usage of the website	PASS
Data displayed for admin is easy to navigate	PASS

7.5 Security Testing

McGraw & Potter (2004) stated that in the presence of a malicious attack the software should behave correctly, in the real world, software tends to fail spontaneously without intention mischief. This testing intends to uncover websites vulnerabilities such as SQL injection and cross-site scripting (XSS). To perform this testing, the chosen operating system is Kali Linux and the tools used are Nikto Web Scanner, Uniscan and Web Application Scanner (WAScan)

Table 7.5.1: Nikto Web Scanner

NIKTO WEB SCANNER	
VULNERABILITY	**RESULT**
Anti-Clickjacking X-Frame	PASS
X-XSS Protection	PASS
X-Content-Type-Option	PASS
Cross Site Tracing (XST)	FAIL

Table 7.5.2: Uniscan

UNISCAN	
VULNERABILITY	**RESULT**
Blind SQL Injection	PASS
PHP CGI Argument Injection	PASS
Remote Command Execution	PASS
SQL Injection	PASS
Cross-Site Scripting (XSS)	PASS

Table 7.5.3: WAScan

WASCAN	
VULNERABILITY	**RESULT**
X-XSS Protection	PASS
X-Content-Type	PASS
X-Frame Option	PASS
Script-Transport-Security	FAIL

7.6 Testing Conclusion

Throughout the testing phase, this system has gone through multiple types of software testing. For the testing results, most of its modules and features are have no error. From the test results, the code is unable to delete the encrypted file from the web server because as the delete code is executed, the encrypted file is still uploading to the cloud. To counter this issue, an additional feature is added for the admin to remove or clean the web server's uploaded or download file with a click of a button. Furthermore, this web application has gone through security testing. From the results, this website is able to withstand attacks such as SQL injections and Cross-site Scripting (XSS) attacks. Unfortunately, from the results from WAScan, this web application does not have the web security policy called Script Transport Security. Therefore, it is unable to withstand attack against protocol downgrade attack and cookie hijacking. Furthermore, from the results of Nikto, this web application is vulnerable against HTTP TRACE method, a network security vulnerability as it does not protect against cross-site tracing (XST).

CHAPTER 8

CONCLUSION

Throughout this whole development for developing a website solution to ease the ways and method to transfer file among parties. There are a couple of objectives for this development and all of them are met. Firstly, this system manages to replace traditional methods of transferring files which are using pen drives or DVD. Secondly, the cloud storage infrastructure has been developed as the medium for the transferring files. Thirdly, all the uploaded and downloaded to or from the cloud storage. Next, the file is deleted once it is completed. Lastly, all the files uploaded are to be encrypted and decrypted once downloaded.

In FYP phase one, the problem statement, the objectives and the scope of the project have been stated out. The problem statement condemns the issue addressed to the traditional ways of transferring files and the objectives state the main goals. Moreover, multiple research about the components of this project such as the cloud infrastructure and encryption has been studied tremendously.

Besides that, to understand and study the existing system in our current world, a literature study is conducted. For this report, three existing systems have been studied. The features have been studied and compared with the other existing system. They all perform the main function but each of them has their own unique features to enhance the usability of the system.

Furthermore, the methodology of the system development life cycle for this project has been established and the list of tools and programming languages has been identified. Besides that, the graphical logical diagram of the system is provided to provide a better understanding of the operation of how the system works. The graphical diagram includes flowcharts, use case, sequence diagram and entity relation diagram. On the other hand, a screenshot of the early prototyping of the system is displayed.

In FYP phase two, it consists of more technical information. In chapter 5, the main code of the functions used explained to understand how the function operates and runs. Followed by chapter 6 which explains the user interface of the website alongside with the help of screenshots. It explains how the system works in the user's perspective.

Moreover, to ensure the software behaves as intended and identifying any sorts of potential malfunction or errors, testing is conducted to ensure quality assurance of the web application. Four types of testing are conducted which are unit testing, integrated testing, system testing and security testing. Each testing deals with different purpose and point of view.

Finally, this project is built to hopefully cater to organizations in hope to help solve the traditional method of transferring files. This web application provides a user-friendly user interface and function but there is always room for improvements such as additional usability functionalities or redesign the user interface to be friendlier for users.

REFERENCES

Berners-Lee, T., & Connolly, D. (1995). Hypertext Markup Language - 2.0, 1–77. https://doi.org/10.17487/rfc1866

Bertolino, A., & Faedo, I. A. (2007). Software Testing Research: Achievements, Challenges, Dreams. *Future of Software Engineering(FOSE'07)*, (September), 19. https://doi.org/10.1109/FOSE.2007.25

Davis, M., & Phillips, J. (2007). *Learning PHP & MySQL: Step-by-Step Guide to Creating Database-Driven Web Sites*. Retrieved from http://www.google.com/books?hl=pt-BR&lr=&id=IpuT0GxHvAkC&oi=fnd&pg=PR5&dq=Learning+PHP+and+MySQL&ots=86w_CDabAF&sig=NKs2zGRL2Z_rpWlt9uBXCjPtYFk

Dora, S. K., & Dubey, P. (2013). Software Development Life Cycle (SDLC) Analytical Comparison and Survey on Traditional and Agile Methodology. *National Monthly Refereed Journal of Research in Science & Technology*, *2*(8), 22–30.

Extras for your Business account | MailBigFile. (n.d.). Retrieved from https://www.mailbigfile.com/business/extras/

How secure is your platform? – WeTransfer Support. (n.d.). Retrieved from https://wetransfer.zendesk.com/hc/en-us/articles/210092453-How-secure-is-your-platform-

Kumar, N., Zadgaonkar, A. S., & Shukla, A. (2013). Evolving a New Software Development Life Cycle Model SDLC-2013 with Client Satisfaction. *International Journal of Soft Computing and Engineering(IJSCE)*, *3*(1), 216–221.

Leung, H. K. N., & White, L. (1990). A study of integration testing and software regression at the integration level. *Proceedings. Conference on Software Maintenance 1990*, 290–301. https://doi.org/10.1109/ICSM.1990.131377

MailBigFile | How It Works. (n.d.). Retrieved from https://www.mailbigfile.com/how-mbf-works/

McGraw, G., & Potter, B. (2004). Software Security Testing. *IEEE Security and Privacy*, *2*(5), 81–85. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1341418

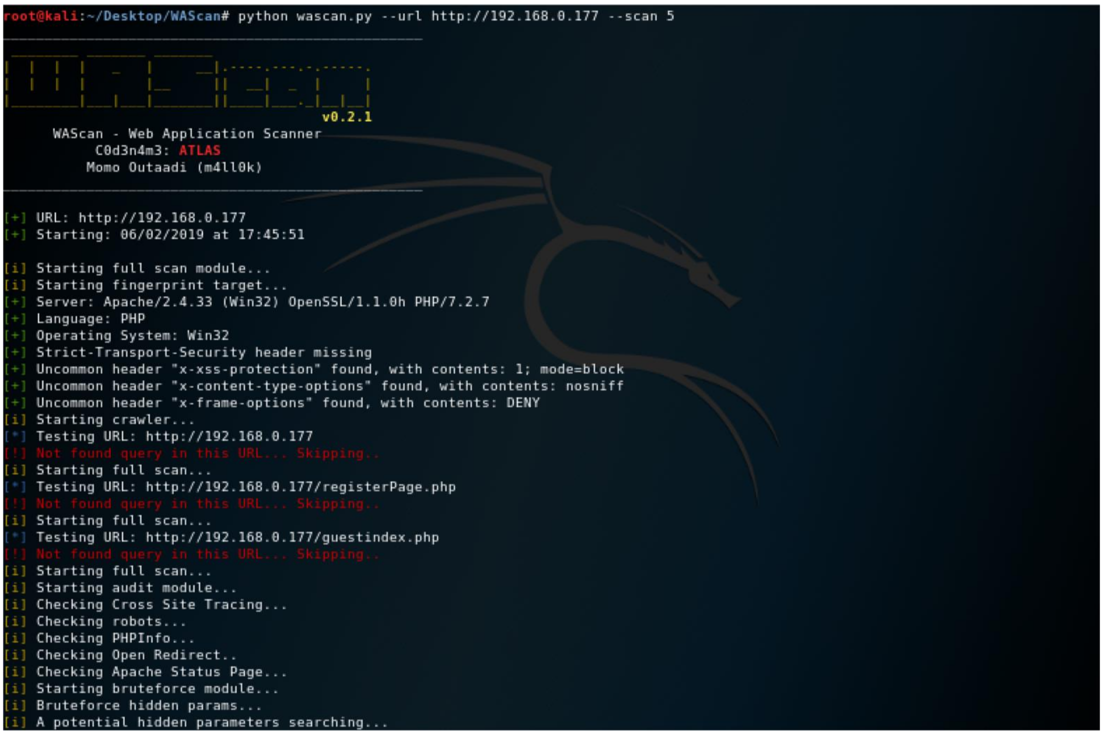
NIST. (2001). FIPS 197 - Announcing the ADVANCED ENCRYPTION

STANDARD (AES). Retrieved from http://csrc.nist.gov/csor/

SendGB | File transfer &amp; File Hosting &amp; File Sharing. (n.d.). Retrieved from https://www.sendgb.com/en/about-us.html

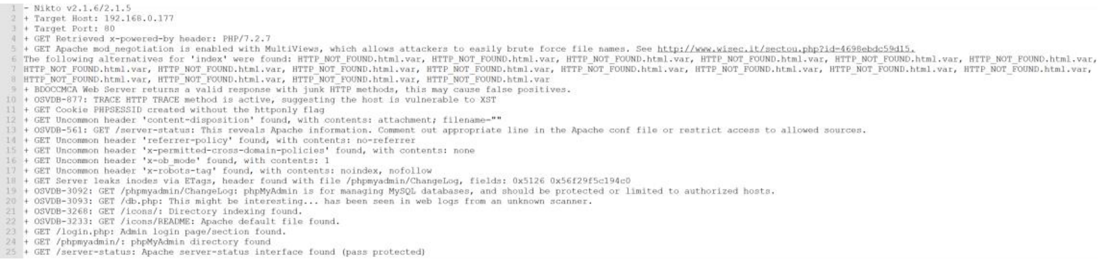
SendGB | Send Large Files, What is SendGB? (n.d.). Retrieved from https://www.sendgb.com/en/faq.html

APPENDIX



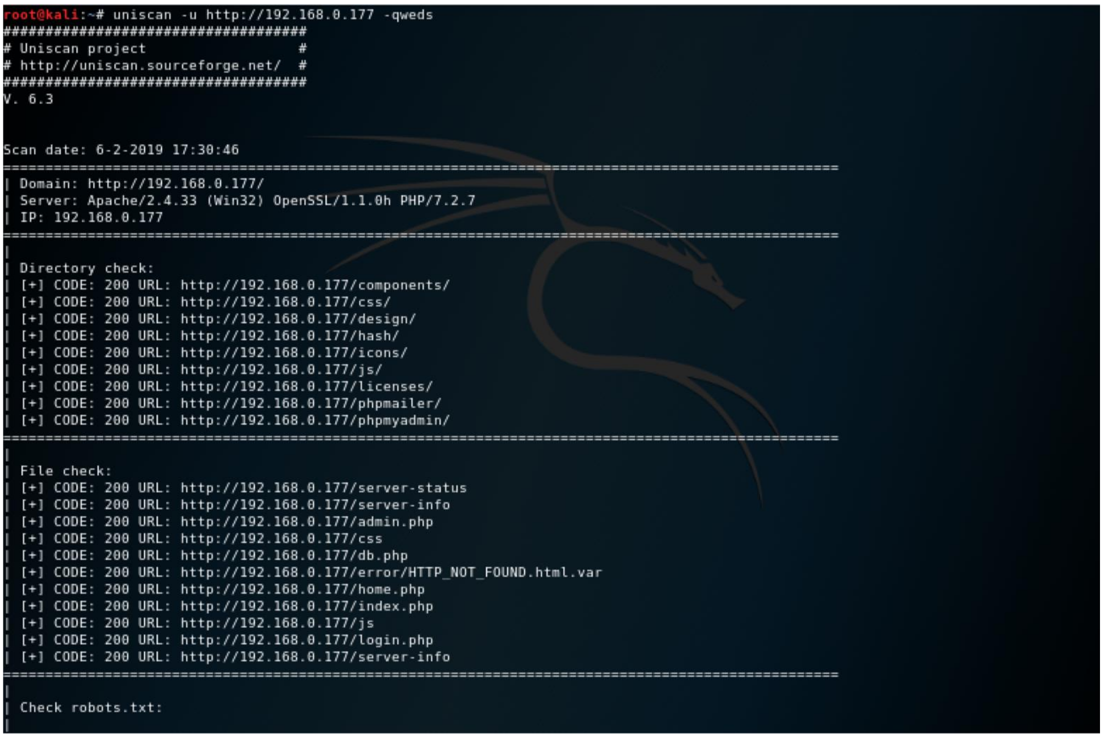
```
root@kali:~/Desktop/WAScan# python wascan.py --url http://192.168.0.177 --scan 5
--------------------------------------------------------
 _______ ______ ______
|   |   |   _  |    __|.----.---.-.-----.
|   |   |      |__    ||  __|  _  |     |
|_______|___|__|______||____|___._|__|__|
                                    v0.2.1
      WAScan - Web Application Scanner
           C0d3n4m3: ATLAS
          Momo Outaadi (m4ll0k)
--------------------------------------------------------

[+] URL: http://192.168.0.177
[+] Starting: 06/02/2019 at 17:45:51

[i] Starting full scan module...
[i] Starting fingerprint target...
[+] Server: Apache/2.4.33 (Win32) OpenSSL/1.1.0h PHP/7.2.7
[+] Language: PHP
[+] Operating System: Win32
[+] Strict-Transport-Security header missing
[+] Uncommon header "x-xss-protection" found, with contents: 1; mode=block
[+] Uncommon header "x-content-type-options" found, with contents: nosniff
[+] Uncommon header "x-frame-options" found, with contents: DENY
[i] Starting crawler...
[*] Testing URL: http://192.168.0.177
[!] Not found query in this URL... Skipping..
[i] Starting full scan...
[*] Testing URL: http://192.168.0.177/registerPage.php
[!] Not found query in this URL... Skipping..
[i] Starting full scan...
[*] Testing URL: http://192.168.0.177/guestindex.php
[!] Not found query in this URL... Skipping..
[i] Starting full scan...
[i] Starting audit module...
[i] Checking Cross Site Tracing...
[i] Checking robots...
[i] Checking PHPInfo...
[i] Checking Open Redirect..
[i] Checking Apache Status Page...
[i] Starting bruteforce module...
[i] Bruteforce hidden params...
[i] A potential hidden parameters searching...
```

Nikto



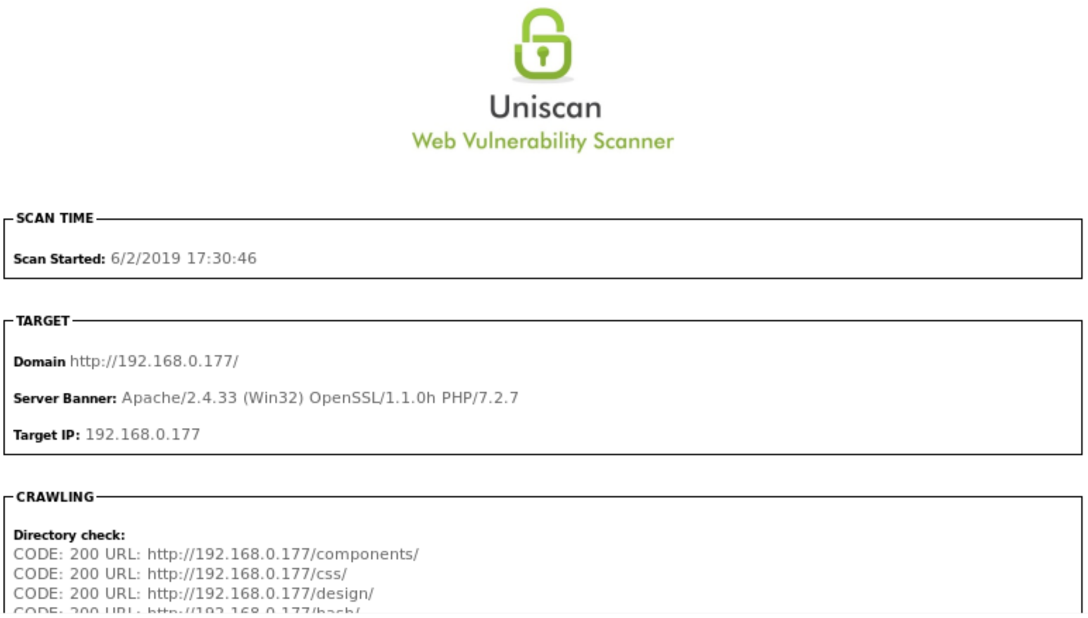
```
- Nikto v2.1.6/2.1.5
+ Target Host: 192.168.0.177
+ Target Port: 80
+ GET Retrieved x-powered-by header: PHP/7.2.7
+ GET Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15.
The following alternatives for 'index' were found: HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var,
HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var
+ BDOCCMCA Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: TRACE HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ GET Cookie PHPSESSID created without the httponly flag
+ GET Uncommon header 'content-disposition' found, with contents: attachment; filename=""
+ OSVDB-561: GET /server-status: This reveals Apache information. Comment out appropriate line in the Apache conf file or restrict access to allowed sources.
+ GET Uncommon header 'referrer-policy' found, with contents: no-referrer
+ GET Uncommon header 'x-permitted-cross-domain-policies' found, with contents: none
+ GET Uncommon header 'x-ob_mode' found, with contents: 1
+ GET Uncommon header 'x-robots-tag' found, with contents: noindex, nofollow
+ GET Server leaks inodes via ETags, header found with file /phpmyadmin/ChangeLog, fields: 0x5126 0x56f29f5c194c0
+ OSVDB-3092: GET /phpmyadmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3093: GET /db.php: This might be interesting... has been seen in web logs from an unknown scanner.
+ OSVDB-3268: GET /icons/: Directory indexing found.
+ OSVDB-3233: GET /icons/README: Apache default file found.
+ GET /login.php: Admin login page/section found.
+ GET /phpmyadmin/: phpMyAdmin directory found
+ GET /server-status: Apache server-status interface found (pass protected)
```

Nikto Generated Report



```
root@kali:~# uniscan -u http://192.168.0.177 -qweds
####################################
# Uniscan project                  #
# http://uniscan.sourceforge.net/  #
####################################
V. 6.3


Scan date: 6-2-2019 17:30:46
===================================================================================================
| Domain: http://192.168.0.177/
| Server: Apache/2.4.33 (Win32) OpenSSL/1.1.0h PHP/7.2.7
| IP: 192.168.0.177
===================================================================================================
|
| Directory check:
| [+] CODE: 200 URL: http://192.168.0.177/components/
| [+] CODE: 200 URL: http://192.168.0.177/css/
| [+] CODE: 200 URL: http://192.168.0.177/design/
| [+] CODE: 200 URL: http://192.168.0.177/hash/
| [+] CODE: 200 URL: http://192.168.0.177/icons/
| [+] CODE: 200 URL: http://192.168.0.177/js/
| [+] CODE: 200 URL: http://192.168.0.177/licenses/
| [+] CODE: 200 URL: http://192.168.0.177/phpmailer/
| [+] CODE: 200 URL: http://192.168.0.177/phpmyadmin/
===================================================================================================
|
| File check:
| [+] CODE: 200 URL: http://192.168.0.177/server-status
| [+] CODE: 200 URL: http://192.168.0.177/server-info
| [+] CODE: 200 URL: http://192.168.0.177/admin.php
| [+] CODE: 200 URL: http://192.168.0.177/css
| [+] CODE: 200 URL: http://192.168.0.177/db.php
| [+] CODE: 200 URL: http://192.168.0.177/error/HTTP_NOT_FOUND.html.var
| [+] CODE: 200 URL: http://192.168.0.177/home.php
| [+] CODE: 200 URL: http://192.168.0.177/index.php
| [+] CODE: 200 URL: http://192.168.0.177/js
| [+] CODE: 200 URL: http://192.168.0.177/login.php
| [+] CODE: 200 URL: http://192.168.0.177/server-info
===================================================================================================
|
| Check robots.txt:
|
```

Uniscan



Uniscan

Web Vulnerability Scanner

SCAN TIME

Scan Started: 6/2/2019 17:30:46

TARGET

Domain http://192.168.0.177/

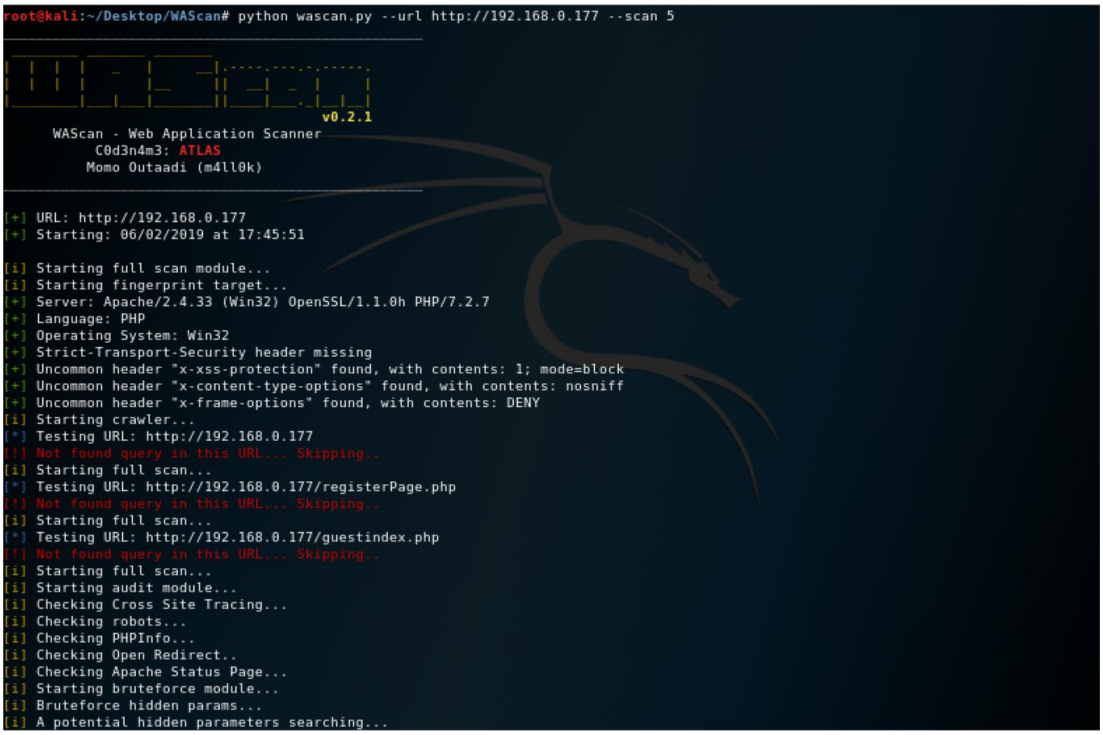
Server Banner: Apache/2.4.33 (Win32) OpenSSL/1.1.0h PHP/7.2.7

Target IP: 192.168.0.177

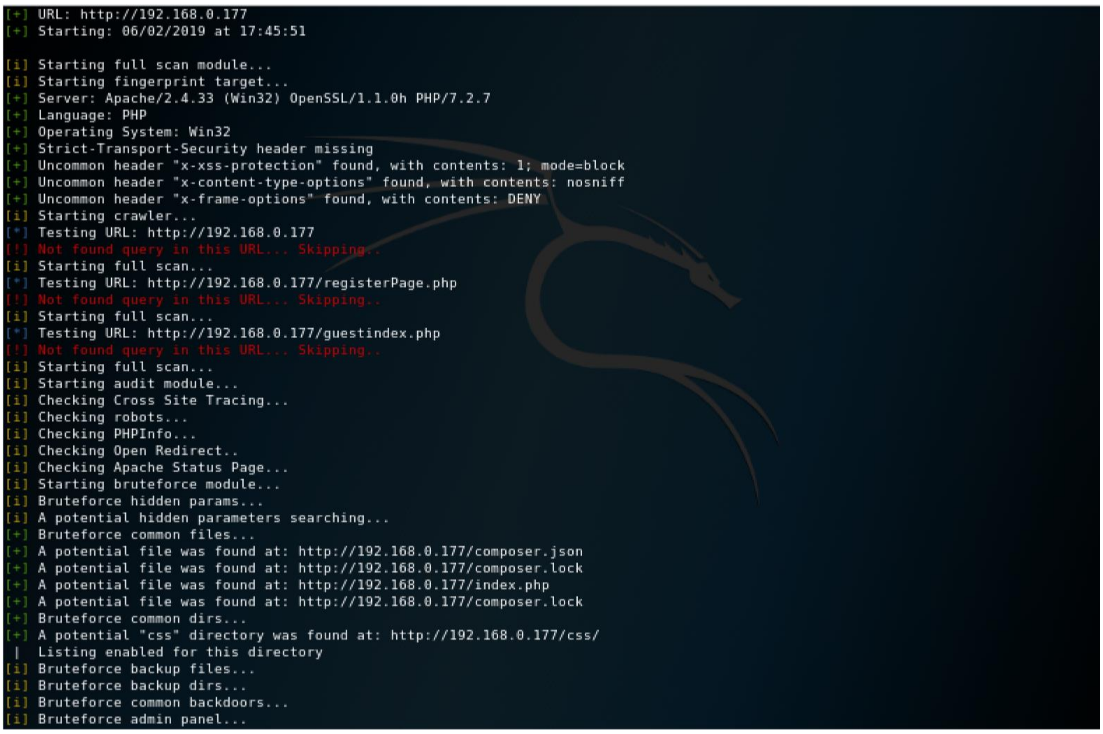
CRAWLING

Directory check:
CODE: 200 URL: http://192.168.0.177/components/
CODE: 200 URL: http://192.168.0.177/css/
CODE: 200 URL: http://192.168.0.177/design/
CODE: 200 URL: http://192.168.0.177/hash/

Uniscan generated report



```
root@kali:~/Desktop/WAScan# python wascan.py --url http://192.168.0.177 --scan 5
----------------------------------------------------------

                                       v0.2.1
      WAScan - Web Application Scanner
           C0d3n4m3: ATLAS
          Momo Outaadi (m4ll0k)

----------------------------------------------------------

[+] URL: http://192.168.0.177
[+] Starting: 06/02/2019 at 17:45:51

[i] Starting full scan module...
[i] Starting fingerprint target...
[+] Server: Apache/2.4.33 (Win32) OpenSSL/1.1.0h PHP/7.2.7
[+] Language: PHP
[+] Operating System: Win32
[+] Strict-Transport-Security header missing
[+] Uncommon header "x-xss-protection" found, with contents: 1; mode=block
[+] Uncommon header "x-content-type-options" found, with contents: nosniff
[+] Uncommon header "x-frame-options" found, with contents: DENY
[i] Starting crawler...
[*] Testing URL: http://192.168.0.177
[!] Not found query in this URL... Skipping..
[i] Starting full scan...
[*] Testing URL: http://192.168.0.177/registerPage.php
[!] Not found query in this URL... Skipping..
[i] Starting full scan...
[*] Testing URL: http://192.168.0.177/guestindex.php
[!] Not found query in this URL... Skipping..
[i] Starting full scan...
[i] Starting audit module...
[i] Checking Cross Site Tracing...
[i] Checking robots...
[i] Checking PHPInfo...
[i] Checking Open Redirect..
[i] Checking Apache Status Page...
[i] Starting bruteforce module...
[i] Bruteforce hidden params...
[i] A potential hidden parameters searching...
```



```
[+] URL: http://192.168.0.177
[+] Starting: 06/02/2019 at 17:45:51

[i] Starting full scan module...
[i] Starting fingerprint target...
[+] Server: Apache/2.4.33 (Win32) OpenSSL/1.1.0h PHP/7.2.7
[+] Language: PHP
[+] Operating System: Win32
[+] Strict-Transport-Security header missing
[+] Uncommon header "x-xss-protection" found, with contents: 1; mode=block
[+] Uncommon header "x-content-type-options" found, with contents: nosniff
[+] Uncommon header "x-frame-options" found, with contents: DENY
[i] Starting crawler...
[*] Testing URL: http://192.168.0.177
[!] Not found query in this URL... Skipping..
[i] Starting full scan...
[*] Testing URL: http://192.168.0.177/registerPage.php
[!] Not found query in this URL... Skipping..
[i] Starting full scan...
[*] Testing URL: http://192.168.0.177/guestindex.php
[!] Not found query in this URL... Skipping..
[i] Starting full scan...
[i] Starting audit module...
[i] Checking Cross Site Tracing...
[i] Checking robots...
[i] Checking PHPInfo...
[i] Checking Open Redirect..
[i] Checking Apache Status Page...
[i] Starting bruteforce module...
[i] Bruteforce hidden params...
[i] A potential hidden parameters searching...
[+] Bruteforce common files...
[+] A potential file was found at: http://192.168.0.177/composer.json
[+] A potential file was found at: http://192.168.0.177/composer.lock
[+] A potential file was found at: http://192.168.0.177/index.php
[+] A potential file was found at: http://192.168.0.177/composer.lock
[+] Bruteforce common dirs...
[+] A potential "css" directory was found at: http://192.168.0.177/css/
 |  Listing enabled for this directory
[i] Bruteforce backup files...
[i] Bruteforce backup dirs...
[i] Bruteforce common backdoors...
[i] Bruteforce admin panel...
```

WAScan