

ONE-TIME PASSWORD AUTHENTICATION  
SYSTEM IN WEBSITE

AZHAM HELMI BIN AZMI

Bachelor of Computer Science (Computer System &  
Networking) with Honours

UNIVERSITI MALAYSIA PAHANG

## UNIVERSITI MALAYSIA PAHANG

### DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : AZHAM HELMI BIN AZMI

Date of Birth

Title : ONE-TIME PASSWORD AUTHENTICATION SYSTEM IN WEBSITE

Academic Session : 2021/2022

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)\*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)\*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

\_\_\_\_\_  
(Student's Signature)

\_\_\_\_\_  
(Supervisor's Signature)

\_\_\_\_\_  
New IC/Passport Number  
Date: 8 FEBRUARY 2023

\_\_\_\_\_  
MUHAMMED RAMIZA BIN RAMLI  
Name of Supervisor  
Date:

NOTE : \* If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.



**SUPERVISOR’S DECLARATION**

I/We\* hereby declare that I/We\* have checked this thesis/project\* and in my/our\* opinion, this thesis/project\* is adequate in terms of scope and quality for the award of the degree of \*Doctor of Philosophy/ Master of Engineering/ Master of Science in .....

(Supervisor’s Signature)

Full Name : EN. MUHAMMED RAMIZA BIN RAMLI  
Position : LECTURER  
Date : 8 FEBRUARY 2023

(Co-supervisor’s Signature)

Full Name :  
Position :  
Date :



### **STUDENT'S DECLARATION**

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

A handwritten signature in black ink, appearing to be 'A. Helmi', is positioned above a horizontal line.

(Student's Signature)

Full Name : AZHAM HELMI BIN AZMI  
ID Number : CA19040  
Date : 8 FEBRUARY 2023

ONE-TIME PASSWORD AUTHENTICATION  
SYSTEM IN WEBSITE

AZHAM HELMI BIN AZMI

Thesis submitted in fulfillment of the requirements  
for the award of the degree of  
Bachelor of Computer Science (Computer System & Networking)

Faculty of Science Computer  
UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2023

## **ACKNOWLEDGEMENTS**

First and foremost, I would like to deliver my appreciation and gratitude to my supervisor, En. Muhammed Ramiza bin Ramli for his guidance and support to complete my project. The suggestion and idea from him have given me the knowledge that I have not explore before. It also expands my knowledge to certain topics and expand my experience into other territory.

After that, I would like to sincerely thank my family and friends who gave me encouragement and idea on the project. My friends willingly give contribution even in terms of motivation and support by pushing myself to complete the project.

Last but not least, I want to express my gratitude to the project coordinator Dr Danakorn for being supportive and coordinate my project with clear explanation. The guidance stated in detail with easy-to-understand schedule has helped me in completing the project and with this I am very grateful.

## **ABSTRAK**

Dunia yang kita tahu kebanyakannya menggunakan internet di mana-mana tempat dan masa di dunia. Kami disambungkan ke media sosial, platform permainan, laman blog, kawasan perniagaan, membeli-belah dalam talian dan banyak lagi setiap hari tanpa henti. Internet boleh berguna tetapi ia mempunyai sisi buruknya sendiri. Akhir-akhir ini semasa pandemik orang ramai semakin banyak berhubung dengan internet kerana kawalan pergerakan oleh beberapa negeri untuk membendung penularan Covid-19. Kerana itu ramai penggoda mengambil kesempatan ini dengan mencuri maklumat orang ramai terutamanya dalam laman web. Untuk mengatasi masalah ini, sistem pengesanan kata laluan sekali dicadangkan dalam laman web untuk melindungi pengguna daripada mendapatkan akaun mereka dicuri atau diakses oleh pengguna yang tidak dibenarkan. Sistem ini akan bertindak sebagai kaedah pengesanan kedua di sebelah kaedah log masuk konvensional. Kaedah ini akan menghantar kata laluan sekali sahaja yang hanya boleh diakses oleh pengguna untuk log masuk ke akaun mereka dan tidak boleh digunakan oleh penggoda. Sistem ini seharusnya dapat menyekat mana-mana penggoda yang tidak dibenarkan daripada mencampuri akaun pengguna dan meningkatkan keselamatan tapak web yang melaksanakannya.

## **ABSTRACT**

The world we know how are mostly using the internet in any place and time in the world. We are connected to the social media, gaming platform, blog sites, business area, online shopping and much more every day without stopping. Internet can be useful however it has its own down side. Lately during the pandemic people are more and more connected to the internet because of movement control by several states to curb the spread of Covid-19. Because of that many hackers are taking advantage of this by stealing people information especially in websites. To counter this problem, one-time password authentication system is proposed in websites to protect the user from getting their account stolen or accessed by unauthorized users. This system will act as a second method of authentication next to the conventional login method. The method will send a one-time password that only can be accessed by the user to login into their account and cannot be used by hackers. This system is able to block any unauthorized hacker from meddling with user's account and increase the security of the website that implements it.



## TABLE OF CONTENT

<b>DECLARATION</b>	
<b>TITLE PAGE</b>	
<b>ACKNOWLEDGEMENTS</b>	<b>ii</b>
<b>ABSTRAK</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENT</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>viii</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>LIST OF ABBREVIATIONS</b>	<b>x</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 INTRODUCTION	1
1.2 PROBLEM STATEMENT	2
1.3 OBJECTIVE	4
1.4 SCOPE	4
1.5 SIGNIFICANCE OF PROJECT	5
1.6 REPORT ORGANIZATION	5
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>6</b>
2.1 INTRODUCTION	6
2.2 ONE-TIME PASSWORD	6
2.3 TYPE OF ONE-TIME PASSWORD	7
2.4 TYPE OF ONE-TIME PASSWORD	8
2.4.1 Facebook	9

2.4.2	Twitter	10
2.4.3	Google	12
2.5	SUMMARY OF COMPARISONS OF THREE EXISTING SYSTEM	13
2.6	SUMMARY ON REVIEW EXISTING SYSTEM	14
2.7	PROPOSED SYSTEM	15
<b>CHAPTER 3 METHODOLOGY</b>		<b>16</b>
3.1	INTRODUCTION	16
3.2	PROJECT MANAGEMENT FRAMEWORK	16
3.2.1	AGILE MODEL	17
3.3	PROJECT REQUIREMENT	19
3.3.1	Functional requirement	19
3.3.2	Non-Functional requirement	20
3.3.3	Constraints	20
3.3.4	Limitation	20
3.4	PROPOSED DESIGN	21
3.4.1	Context Diagram	21
3.4.2	Use Case Diagram	22
3.4.3	Activity Diagram	23
3.5	DATA DESIGN	24
3.5.1	Database Dictionary	24
3.6	INITIAL CONCEPT	25
3.6.1	Design Prototype	25
3.7	TESTING PLAN	29
3.8	POTENTIAL USE OF PROPOSED SOLUTION	30
3.9	SUMMARY	30

<b>CHAPTER 4 RESULTS AND DISCUSSION</b>	<b>31</b>
4.1 INTRODUCTION	31
4.2 IMPLEMENTATION PROCESS	31
4.2.1 Development of Website	31
4.2.2 Development of Database	36
4.2.3 Code Use in Visual Studio Code	38
4.3 TESTING	39
4.3.1 User Acceptance Testing	39
4.3.2 Website Testing	40
4.4 RESULT AND DISCUSSION	40
<b>CHAPTER 5 CONCLUSION</b>	<b>42</b>
5.1 INTRODUCTION	42
5.2 RESEARCH CONSTRAINT	42
5.3 FUTURE WORK	43
<b>REFERENCES</b>	<b>44</b>
<b>APPENDIX A</b>	<b>46</b>
<b>APPENDIX B</b>	<b>48</b>
<b>APPENDIX C</b>	<b>61</b>
<b>APPENDIX D</b>	<b>63</b>
<b>APPENDIX E</b>	<b>65</b>

## LIST OF TABLES

Table 1.2.1 Problems arise from cybersecurity	3
Table 2.5.1 System comparison	13
Table 3.4.1 register_user table	24
Table 3.4.2 login_data table	25

## LIST OF FIGURES

Figure 2.1 Example of One-Time Password Authentication using laptop and phone	7
Figure 2.2 HMAC-based One-Time Password	8
Figure 2.3 Time-based One-Time Password	8
Figure 2.4 Facebook website	10
Figure 2.5 Facebook setting page for one-time password authentication	10
Figure 2.6 Twitter website	11
Figure 2.7 Twitter OTP setting web page	12
Figure 2.8 Google website	13
Figure 2.9 Google web page that prompt user to set up OTP	13
Figure 3.1 Agile methodology phases	17
Figure 3.2 Context diagram for designed system	21
Figure 3.3 Use case diagram for designed system	22
Figure 3.4 Activity diagram for designed system	23
Figure 3.5 Website login page	25
Figure 3.6 Website register page	26
Figure 3.7 Website OTP page	26
Figure 3.8 Website home page after login	27
Figure 3.9 Website error page	27
Figure 3.10 Visual Code Studio coding phase	28
Figure 3.11 Database with user account	28
Figure 3.12 Receiving mail with OTP key	29
Figure 4.1 Login page interface	32
Figure 4.2 Register page interface	33
Figure 4.3 OTP page interface	34
Figure 4.4 Error page interface	34
Figure 4.5 Homepage interface of website	35
Figure 4.6 Profile interface of website	35
Figure 4.7 Log out interface	36
Figure 4.8 Database of login and register	36
Figure 4.9 Login data content	37
Figure 4.10 Register data content	38
Figure 4.11 Login page coding in VSC	39
Figure 4.12 Summary of user feedback	40

## **LIST OF ABBREVIATIONS**

OTP	One-Time Password
HOTP	HMAC-based One-Time Password
TOTP	Time-based One-Time Password
HMAC	Hash-based Message Authentication Code

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 INTRODUCTION**

Currently the majority of people in this world are using devices in their daily life. It can be mobile devices, laptops, PCs, tablets and other. We are connected to the internet almost every day as we surf through web, chatting, going to class, watching videos, playing games and so on. Most of the activity we do in the internet or websites requires account where our personal information is stored and used under our own terms. To protect our info, we use certain security to block other users from reaching our information. Even a simple security can make our life better but we are still prone to be attacked as people will always find a way to steal information. With this in mind, security is one of the things to be concerned about in today's society. Security measures such as passwords, firewalls, antivirus, data protection, OS and mobile encryption, education and network monitoring are aspects to consider protecting most important assets (Teodor Topalov, 2015).

Security is the freedom from danger or risk. It is also extended by defining risk as the potential loss from threats, vulnerabilities, value, and countermeasures. To understand a security design process involves determining the organization's security needs. There are four pieces of security framework which is attack prevention, detection, isolation, and recovery. Some threats can be human, external, internal, malignant, malicious and environmental. Some threats are unable to control and some threats are often overlooked in the security framework (Tariq Bin Azad, 2008). These threats can be mitigated by having a proactive and strict security awareness. Security is a quality aspect that constrains the behaviour of applications by imposing any access and use restrictions on the data and other assets, this means

that the requirements stage is the appropriate stage to start addressing security (Eduardo B. Fernandez, 2022).

One-time password or passcode (OTP) is a string of numbers or characters that authenticates user for a single login attempt. An algorithm generates a unique value for each password by factoring in contextual information, like time-based data or like previous login events (Teju Shyamsundar, 2020). With the increase of cyber security threats, it has become more necessary to upgrade security standards of web applications to make sure users' accounts are safe. Web applications nowadays ask users to add an extra layer of security to their account by enabling 2-factor authentication such as Time-based One-Time Password (TOTP) authentication (Prakash Sharma, 2018). In this project, a security method which is one-time password authentication is developed. This security approach has benefits and potential to increase the security and safety of website applications from being attacked and exploited by unauthorized people. It will block any attacker from accessing users' account even after passing the first layer of security which is the widely used text-based password.

## **1.2 PROBLEM STATEMENT**

Security breach is one of the critical security issues where it can be a threat to confidential or private data of a person or even an organization. The usual and common type of password used for security is text-based. Single-factor authentication is considered as the simplest form of authentication methods where a person matches one credential to verify themselves online (Steven Feltner, 2016). Unfortunately, this type of passwords can be easily hacked and can result to someone losing their private data to unauthorized person. Security issues related to login have become a major concern with the increasing amount of cyber-crime. A simple security such as single level authentication might not be safe anymore and will not protect us from cyber threats. Malicious user may guess peoples' password by knowing them personally or even knowing peoples' personal information such as birthdate, favourite food, pet's name, and so on. They also use bot to crack passwords by generate right combination of letters or numbers to match simple identification method of a user.



Passwords that is text based are easier to crack or guessed as people tends to make easy to remember passwords which are short and simple. Because of that the password is restricted to numbers, letters, or symbols due to policy of certain organization to make the password stronger but in exchange of it being harder to remember. Having an account without a strong password will most likely be attacked and stolen from third party and thus decreasing the security level of a system. Users must come up with a strong password by combining numbers, letters, and symbol to make it harder to guess. Other methods beside making long and hard to remember password is using two-factor authentication (2FA). Two-factor authentication is a security system that needs two distinct forms of identification in order to gain access to something and can help to strengthen the security of an online account (Will Kenton, 2020).

Today's widely used websites are frequently targeted by attackers, posing a significant risk to users. Recent data breaches include Cognyte, which experienced a database breach of 5 billion records in May 2021, LinkedIn, which exposed 700 million records from June to August, Facebook, which was breached affecting more than 533 million accounts in March, Bykea, a Pakistani ride-hailing app, that suffered a breach of 400 million records in November, and the Brazilian Ministry of Health, which lost 223 million records in January (Ashley Watters, 2022). According to research from Verizon, web application attacks are involved in 26% of all breaches, making it the second most common type of attack. In 2020, global search traffic has increased dramatically, often spiking during periods of lockdown to curb the COVID-19 pandemic as the world continues to embrace virtual platforms for classes, study, entertainment, conferencing and more (Caitlin Jones, 2022).

Table 1.2.1 Problems arise from cybersecurity

No	Problem	Description	Effect
1	Single factor authentication password is not solid enough for security	Malicious user that has the resources can crack and pass the authority check on a single factor authentication easily.	Accounts will be compromised easily by unwanted users.

2	Users often and prefer to made easy to remember password	Most people tend to make password that is short and easy to remember so that they can access their account easily without trouble to remember.	Attackers can guess or brute force people's password easily.
3	Web based application are attacked frequently	Many websites that contain users' account are targeted by attacker who wants to harvest information.	Web application's security decreasing and have pose threat to users.

### 1.3 OBJECTIVE

The objective of this one-time password authentication is as follow:

- i. To strengthen the existing single factor password authentication in a website.
- ii. To design and develop a one-time password authentication in a website.
- iii. To evaluate the functionality of the developed one-time password authentication.

### 1.4 SCOPE

User Scope:

- i. Students who are studying in area pekan.
- ii. Online users who are using the web.

System Scope:

- i. Two existing security authentication which are text based, and numeric password.
- ii. Covered topic in security and authentication only.

Development Scope:

- i. Contain user input for register and login purposes such as username and password.
- ii. Using HTML, PHP, Apache and MYSQL.

## **1.5 SIGNIFICANCE OF PROJECT**

Students

- i. Student will have a better security in accessing website and worry less about security breach.
- ii. Student's personal information is kept more tightly.

Organization/Community

- i. Organization or the community around it will have better security in accessing websites.

## **1.6 REPORT ORGANIZATION**

This thesis consists of three chapters. Chapter one discusses on the introduction to the project where the goal is to develop a system where it needs a one-time password authentication in order to access certain web application or data. Users need to be aware of their security as it has a big impact to their daily life as well as organization keeping their secret confidential data in good state. Chapter two discusses about the existing application and comparison between the applications. In this project, Facebook, Twitter and Google is compared in terms of security implementation. Chapter 3 covers about methodology; this project will be using Agile model as the SDLC.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 INTRODUCTION**

During this pandemic of Covid-19 many people are staying at home to curb the spread of the virus. In that period, many companies, schools and institutions has embraced the online learning and conference in everyday work. Because of that, the usage of internet is increasing bit by bit and it opens up for malicious users to launch an attack to the users. The security of website must not take lightly as it heavily affects the user who lost their information to unauthorised people. With that matter in mind, many company has come up with a solid security to mitigate the attacks and one of the ways to do is by implementing one-time password (OTP) to the system or websites. There are many websites that use this security method as it is proven to reduce the number of attacks caused by hackers.

#### **2.2 ONE-TIME PASSWORD**

One-time password is a security method that has been used by many big and small companies around the world to strengthen their security and keep user's account safe from hijackers. It is more secure than a common static password which can be weak or reused across multiple accounts. OTP may replace verification login information or may be utilized in addition to it to include another layer of security. OTPs can be generated in many ways and each one has trade-offs in terms of convenience, cost, security, and accuracy. One-time password is a sequence of symbols which is generated for only one use which makes it unnecessary and useless to eavesdrop, render it worthless to an attacker (Sergey Babkin & Anna Epishkina, 2019).



Figure 2.1 Example of One-Time Password Authentication using laptop and phone

### 2.3 TYPE OF ONE-TIME PASSWORD

Securing access to websites, apps, and cloud-based software is a persistent challenge for businesses in various industries. Ensuring that users have simple and reliable security measures in place is vital for protecting user information and sensitive company data. One common method for achieving this is through the use of One-Time Passwords (OTP), which come in three main types: normal OTP, HMAC-based OTP, and Time-based OTP. OTPs can be accessed through smartphones, text messages, or proprietary tokens. One example of an OTP generator is Authy, which can be used on a phone. HMAC-based OTPs, such as Yubiko's Yubikey, generate codes based on a counter, which is incremented each time the code is requested and validated. Time-based OTPs, on the other hand, use a time factor instead of a counter and have a set time period, usually 30 or 60 seconds, during which the password is valid. If the password is not used within this time frame, a new one must be requested to gain access to the application or website.

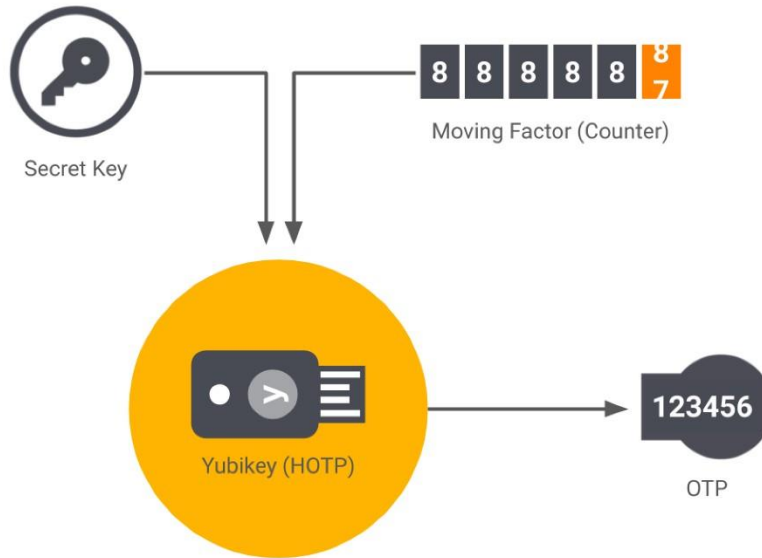


Figure 2.2 HMAC-based One-Time Password

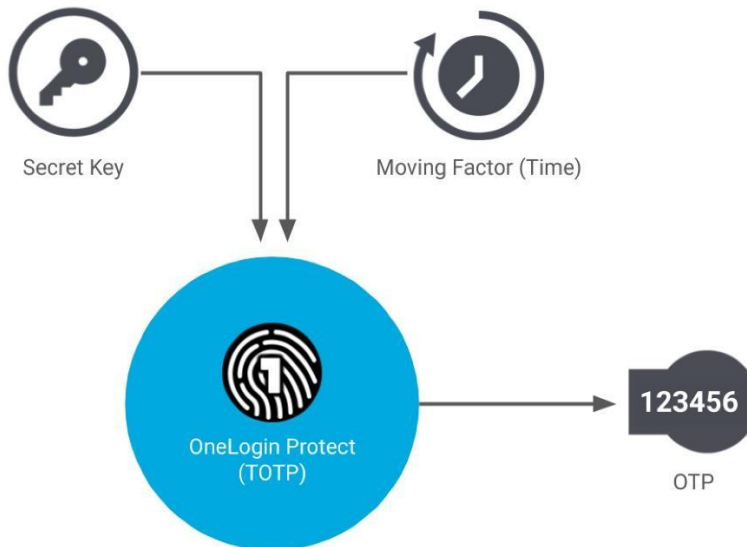


Figure 2.3 Time-based One-Time Password

## 2.4 TYPE OF ONE-TIME PASSWORD

This section will explain about the review of three existing application of One-Time Password in websites. The applications are Facebook, Twitter, and

Google. These websites are using One-Time Password authentication as an option to set up a two-step verification on a user account.

### **2.4.1 Facebook**

Facebook is an online social networking website where people from all around the world can create profiles, share information such as status and photos, and respond or link to the information posted by other people. There are over 2.912 billion users who are currently using Facebook by this year (January 2022). This social network website is popular and being used every day among peoples like students, entrepreneurs, workers, sportsman, and more. They use this website to communicate or even doing business online as they can get their potential target. Being one of the most popular websites has its risks, malicious users are often attracted to Facebook and intent to steal and harm users by hacking people's account. Because of that, Facebook has implemented a two-factor authentication which involve OTP. Figure 2.4 shows the official logo used by Facebook website.

Facebook has a security option inside their settings on user account. It is under Security and Login menu where it shows three methods of setting up a security. The first method is by applying authentication app in which the user will use an app like Google Authenticator or Duo Mobile to generate a verification code (OTP) for more protection. The second option is by security key, user will use a physical security key to access their account and it does not need a code to enter. The third option is by using text message (SMS) where the system will send a text message (SMS) that contain verification codes to the user's registered phone. Figure 2.5 is a web page of Facebook's security setting for users to set up their OTP authentication.



Figure 2.4 Facebook website

Security and Login > Two-factor authentication

The screenshot shows the 'Two-factor authentication' setup page on Facebook. At the top, there is a yellow padlock icon with a keyhole. Below it, the heading 'Help protect your account' is centered, followed by a paragraph: 'If we notice an attempted login from a device or browser we don't recognize, we'll ask for your password and a verification code.' The main section is titled 'Select a security method' and contains three options: 1. 'Authentication app' (marked as 'Recommended'), which suggests using apps like Google Authenticator or Duo Mobile. 2. 'Text message (SMS)', which explains that phone numbers used for two-factor authentication cannot be used to reset passwords. 3. 'Security key', which describes using a physical security key for protection. Each option has a corresponding icon and a button to proceed.

Figure 2.5 Facebook setting page for one-time password authentication

## 2.4.2 Twitter

Twitter is a social networking, microblogging service that permits registered members to broadcast posts called tweets. The users can broadcast tweets and follow other user's tweets by using multiple devices and platforms. The replies and tweets can be sent by using mobile phones, desktop client or by posting in the Twitter website. Based on a statistic, Twitter annual users in 2020 are approximately 186 million users (Mansoor Iqbal, 2022). This makes twitter a popular website to share



and create post to share experience and thoughts. Unfortunately, like other popular website, Twitter has faced serious security breach by attackers and in need of strengthening method for security matter. As of now Twitter already embraced the usage of two-factor authentication or OTP to create a second layer of security. Figure 2.6 shows the logo of the Twitter website.

Twitter security settings can be found inside security and account access and it provides three option to set up OTP authentication. The first option is by text message means that whenever the user wants to login to Twitter, they will receive text message with an authentication code (OTP) to key in first for verification. The second option to setup OTP is by using authentication app, user need to use authentication app to get verification code (OTP) to enter every time a login is being done. Last option is by security key, this option needs the user to get a security key that inserts into the computer or syncs to mobile device when login.



Figure 2.6 Twitter website

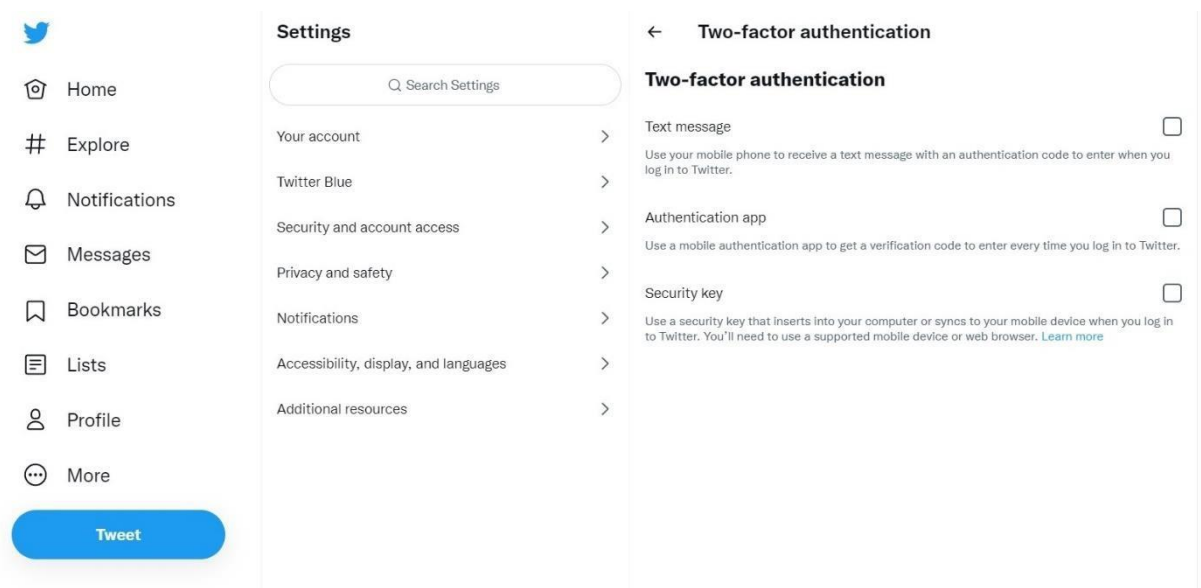


Figure 2.7 Twitter OTP setting web page

### 2.4.3 Google

Google is an internet search engine used frequently in this era. It uses a proprietary algorithm designed to retrieve and order search results to provide relevant dependable sources of data possible. Google has a mission which is to organize the world's information and make it available, accessible, and useful. The flow of online information is influenced by its position as the top search engine in the world as google can give almost any answer that user needs. Google also has made its way to become a platform where people can use to communicate, learn, and become a creator. Students are familiar with google as they are using it for study purposes such as going to online class, research, and collaborate. Users need to make an account to store data and use the tools that Google provide. For the safety of the accounts, Google also implement the usage of OTP into their company for security. Figure 2.7 shows the official logo Google use in their website.

Google security can be set up inside the account settings. There are currently two options available to choose to set up OTP which is by text message and a phone call. The user needs to key in their mobile phone number first so that the system can contact them. The text message option will make the system send a message to the user's phone containing the OTP. The phone call in the other hand will call the user's phone number to tell them the code via voice message.



Figure 2.8 Google website

Figure 2.9 Google web page that prompt user to set up OTP

## 2.5 SUMMARY OF COMPARISONS OF THREE EXISTING SYSTEM

Based on the review in Section 2.4, The following Table 2.1 shows the summary of the comparison of three existing system. There are 6 elements highlighted for comparison which is number of security options, one-time password (OTP) option, HMAC-based One-Time Password (HOTP) option, Time-Based One-Time Password (TOTP) option, advantages and disadvantages.

Table 2.5.1 System comparison

Application Name	Facebook	Twitter	Google
------------------	----------	---------	--------

<b>Number of security options</b>	3	3	2
<b>One-Time Password (OTP) option</b>	✓	✓	✓
<b>HMAC-based One-Time Password (HOTP) option</b>	✓	✓	✗
<b>Time-Based One-Time Password (TOTP) option</b>	✓	✓	✗
<b>Advantages</b>	- User have various choice to pick on setting up security. - Options is clearly described and easy to understand	- User have various choice to pick on setting up security.	- Simple and easy security setup with clean guide
<b>Disadvantages</b>	- Phone numbers used for two-factor authentication cannot be used to reset password when two-factor is on.	- Security key needs a supported mobile device or web server.	- Does not support or use security key. - Vulnerable to be attacked

## 2.6 SUMMARY ON REVIEW EXISTING SYSTEM

There are three website application that have been compared in terms of security implementation (OTP) which is Facebook, Twitter, and Google. Each have its own way of providing security.

In terms of number of options, Facebook and Twitter have three options to pick while Google only provide two. Facebook has given the users three option of OTP to implement which is the Authentication app (TOTP), security key (HOTP), and text message (SMS) (OTP) along with Twitter as well. Both websites provide the same options and both are social networking website. Meanwhile Google only focus on one type which is OTP with two options of implementing it.

Facebook has its own advantages when deploying their security setting which is having to provide users with various choice to pick OTP. User can activate their security based on their preference and convenient method. Some people prefer the normal OTP and some others are more cautious and want to use HOTP or TOTP. The only disadvantages is that the phone number used for OTP authentication cannot be used on to reset password when two-factor is on. This means that when user forgot their password, they will have a hard time resetting their password because of OTP.

As for Twitter, the website also provides the same way of providing OTP like Facebook which is giving three options to pick. This is also convenient for the users to set up their two way based on what method they comfortable with. The disadvantages is that their security key need a supported mobile device or web browser to use. If their device did not support then they cannot use or verify the OTP.

Google has provided two options of security based off OTP. Using mobile phones to get SMS or call the user directly to give the OTP via voice. One good characteristic that Google have is that they manage to design a clean and simple setting page for user to navigate and edit. However, with the lack of other type of OTP, it opens up a way for attacker to do a brute force the normal OTP.

## **2.7 PROPOSED SYSTEM**

In this project, the proposed system is One-Time Password (OTP) authentication implemented inside a website. The target of this system is student in UMP Pekan and online users that accesses websites. The security topic only covered One-Time Passwords, HMAC-based One-Time Password, and Time-Based One-Time Password. This system needs to have a clear online connection to work and need an account to be registered first and login before logging in the websites.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 INTRODUCTION**

This chapter clarifies different strategies that were utilized in gathering data and analysis which are pertinent to the one-time password authentication development. The methodology will include the project management framework, project requirement, proposed design, data design, proof of initial concept, testing plan, and potential use of proposed solution. This will give a detailed idea on how the one-time password is developed and how it is produced to meet the requirement or desired goal. The project will undergo various planning in which needs to be recorded to constantly match the criteria of proposed solution. The SDLC models are analysed and one will be chosen as the right methodology to develop the system.

#### **3.2 PROJECT MANAGEMENT FRAMEWORK**

In this project, the SDLC model that is used to develop the system is Agile model. Agile is a combination of an incremental and iterative approach where the product is released on a progressing cycle then tested and improved at each iteration. The reason why Agile model is used for this particular project is that the time estimated for this project to complete is still uncertain. Agile model is fast and keep progressing which is what the project needs since the time to develop the system is short. The model also helps in making good progress to the development through fixing minor issues. As the time goes, the one-time password authentication system will be constantly updated and fixed based on issues arise at each iteration making it easier to develop. The user or client will also continuously be involved in the project which can give respond rapidly for quickly adapt to changes. The agile model has 6 phases which is plan,

design, build, test, review, and launch and each phase will have its own role and function in the project.

### 3.2.1 AGILE MODEL



Figure 3.1 Agile methodology phases

#### PLANNING PHASE

In the planning phase, the whole development of the system will be drafted and confirmed for further reassess. A discussion with client is arranged to get the key necessities and then plan a documentation to generate an outline of the project. The required component to develop a functional web and functional one-time password is gathered through research. Several potential components such as JavaScript, HTML, CSS, PHP, Apache, MySQL are analysed and used as a base component of the project. Various types of existing system will be analysed too to create a general idea of the system's structure.

#### DESIGNING PHASE

In this phase, a meeting with client is conducted and several requirements are gathered. After the client agreed on the plan, the project is then started by designing the rough design of the user interface for the web. The web will have a login page, OTP page, and the main page of the website after login is successful. The earliest design to be done is the login page as it is the first page greeted by the user or client. Several web designs are made and one design is chosen as the final one to be implemented. The initial system will be in a basic stage and will be incremented as the iteration goes on.

## **BUILDING PHASE**

After finish designing the website and functionality, the system is now ready to be build. The coding for the website is made with PHP to implement the preferred website design. The medium that is used to launch and run the website is Apache while MySQL is used to store information inside a database gathered from the user interface. The current state of the system is light and simple as it is being built with initial requirement. Further improvement is planned to accept and be ready with new changes in the future.

## **TESTING PHASE**

The system is ready and become available for the client to use or test. During the test, user feedback on the login page and functionality of the OTP authentication is recorded. Any difficulties or bugs such as not working OTP, not receive email, not getting the right code during the test is highlighted and assisted. Customer feedback on the current system is documented and improvement plan is produced based on the client experience. The improvement on the system is planned to be implemented on the next cycle of iteration and goes on from time to time.

## **REVIEWING PHASE**



In the last phase which is the review phase, the cycle is on its conclusion and the OTP system is evaluated. The evaluation will be scaled based on how successful the system delivers the requirements to the user and how it impact the user. The OTP system is ready to undergo a new iteration for improvement and testing.

### **3.3 PROJECT REQUIREMENT**

Project requirements are the features, function, and assignments that have to be completed for a project to be deemed as successful. In this project there are functional, non-functional, constrain, and limitation to be covered and highlighted to emphasize the functionality of the OTP system. These requirements will give a clear set of parameters to work toward to and set various goals. This also set the expectation of the outcome or product of the project after it is completed.

#### **3.3.1 Functional requirement**

Functional requirement are the features or functions that must be implemented into the system to enable users or clients to accomplish their task. In this project, the functional requirement is to implement OTP authentication in a website after the user login into their account registered inside the website. The functional requirement for this project is al follow:

- i. The system should register and make account for the user by getting input such as email and password.
- ii. The system will send mail that contain digits to the user for verification after the user successfully key in the right information.

The website will prompt the user to key in username and password on the login page first, then it will direct the user to the OTP page where the user has to get the secret code from their email and key in the code into the website.

### **3.3.2 Non-Functional requirement**

Non-functional requirement defines how the system should perform which are not related to the system functionality. In this project, the OTP system should be running on a web with features that comforts the user such as the following:

- i. The website login page must have a clear GUI and a button option for the user to register a new account.
- ii. The website should have a main page with content for the user to access to.

The user that accesses the website should have no problem interacting with the pages and they can access the content inside the web after they successfully pass the second factor authentication which is the one-time password.

### **3.3.3 Constraints**

The constraints are the limitations of the project including time, risk, and costs. These constraints are important to understand as they affect the project performance. In this project, the constraint that emerge from the system is that it did not have an option for the user to retrieve their account if they forgot their password.

### **3.3.4 Limitation**

The limitation of project is any restriction faced by the product from the project. In this OTP project, the limitation is that it has a risk where the secret code sent by the system might not match with the one inside the website resulting fail to authenticate and verify the user.

### 3.4 PROPOSED DESIGN

In this section, proposed design is explained with the use of context diagram, use case diagram, and activity diagram to further explain the design of the system. The design of this system is referenced from the development of a web application that uses html coding and relevant system related to the design.

#### 3.4.1 Context Diagram

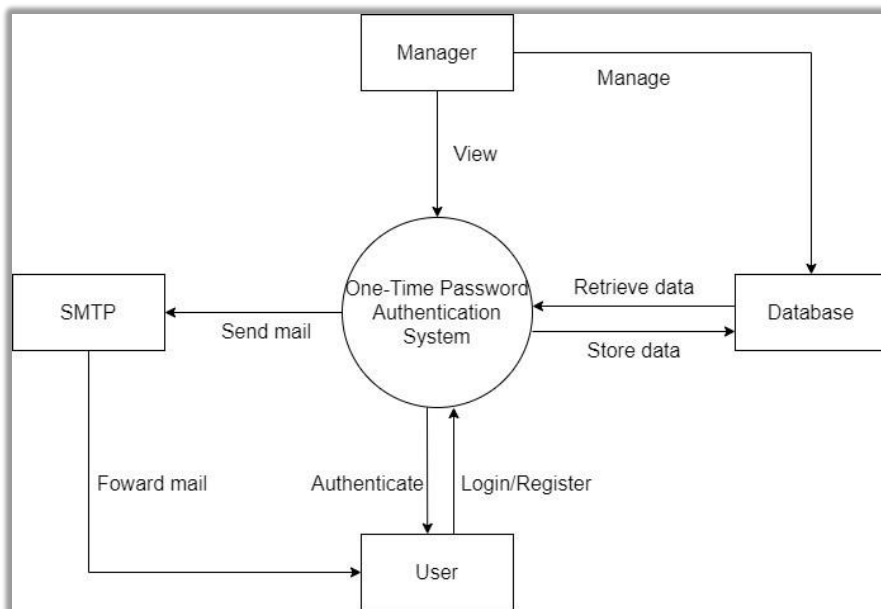


Figure 3.2 Context diagram for designed system

In this context diagram, the connection and relation of the system with other factors are showed in detail. The user will interact with the system by logging into the website, or register a new account if they do not have yet. The system will check if the user has an account or not by accessing the data inside the database. If the user registers new account, the system will store the data inside the database. If an account is found for the user, the system will generate an OTP key and send it to the SMTP server via mail. The SMTP server will act as a sender that forwards the mail to the user who wanted to login to the website. The user will receive the mail with the OTP key and the

user need to key in the detail into the website in order for the system to authenticate and verify them. After the user key in the right OTP key, they will be able to enter the website. The manager role is to keep the system running normally and also manage the database.

### 3.4.2 Use Case Diagram

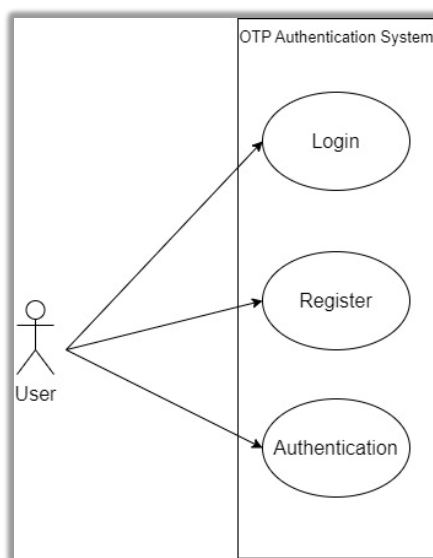


Figure 3.3 Use case diagram for designed system

This use case diagram shows how the system operates and what interaction can be done by the actors. There are 1 actor which is the user. The user is the visitor that wants to login or register to the website and need to go through the OTP authentication system first in order to access the website. The data can be manipulated in case of system error to keep the database working as intended. The SMTP server here acts as a sender to send the generated OTP key from the system to the user via mail.

### 3.4.3 Activity Diagram

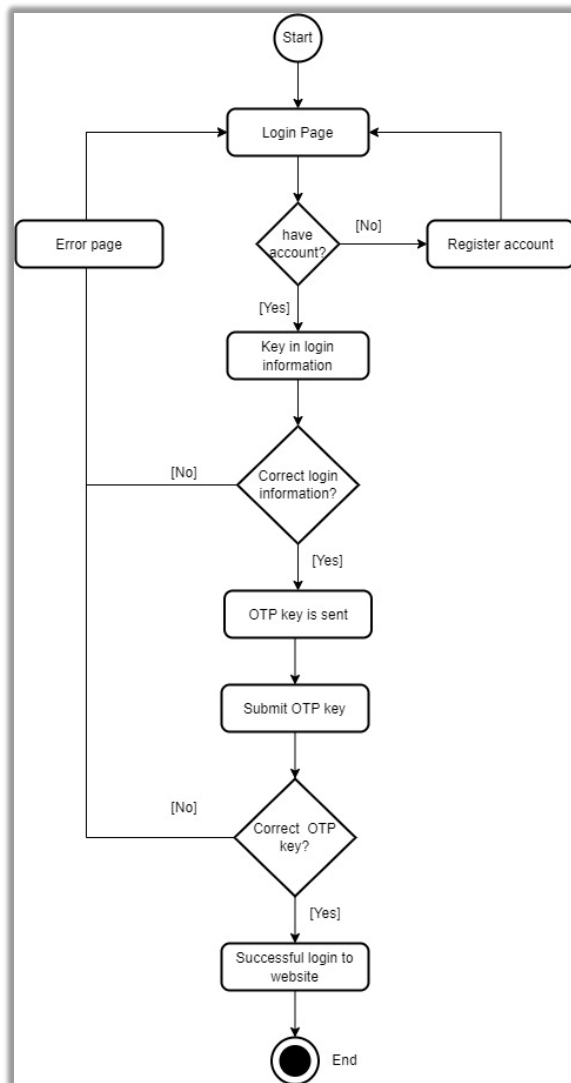


Figure 3.4 Activity diagram for designed system

The activity diagram shows the message flow of the system where the user interacts with the system. At the start of the flow, the user is at the login page. If the user does not have an account, then they need to register. If they have an account then they need to key in the login information. If the login information is incorrect, then it will direct the user to the error page and then to the login page. If the user key in the correct information, then the OTP key will be sent to them. The user will get the OTP key via email and they need to input it to the system. If the OTP key input is incorrect, then the system will direct the user to the error page and then back to the login page. If

the OTP key input is correct then the user successfully login into the website. Then after that the system flow is finished.

### 3.5 DATA DESIGN

The data design is where the data is organized according to the database model, in this project the data that is organized are the user who register to the system and how their OTP key is generated. It is also organized to store information for authentication purposes. When the user login their data will be matched with the data inside the database.

#### 3.5.1 Database Dictionary

Table 3.5.1 register\_user table

Attribute Name	Type	Size	Description
register_user_id	INT	11	Primary Key
user_name	VARCHAR	250	Required
user_email	VARCHAR	250	Required
user_password	VARCHAR	250	Required
user_activation_code	VARCHAR	250	For activation
user_email_status	ENUM	'not verified', 'verified'	Status of email
user_otp	INT	11	For authentication
user_datetime	TIMESTAMP	-	To capture time and date

Table 3.5.2 login\_data table

Attribute Name	Type	Size	Description
user_id	INT	11	Tract user id
login_otp	INT	11	For OTP authentication during login
last_activity	DATE	-	To get the date of activity

### 3.6 INITIAL CONCEPT

Proof of concept is the evidence that shows the proposed system can be made and demonstrated as intended. In this project the proposed system is produced with the functionality to verify user with the use of One-Time Password authentication when login into a website. The prototype is made with certain applications and can be demonstrated using appropriate tools.

#### 3.6.1 Design Prototype

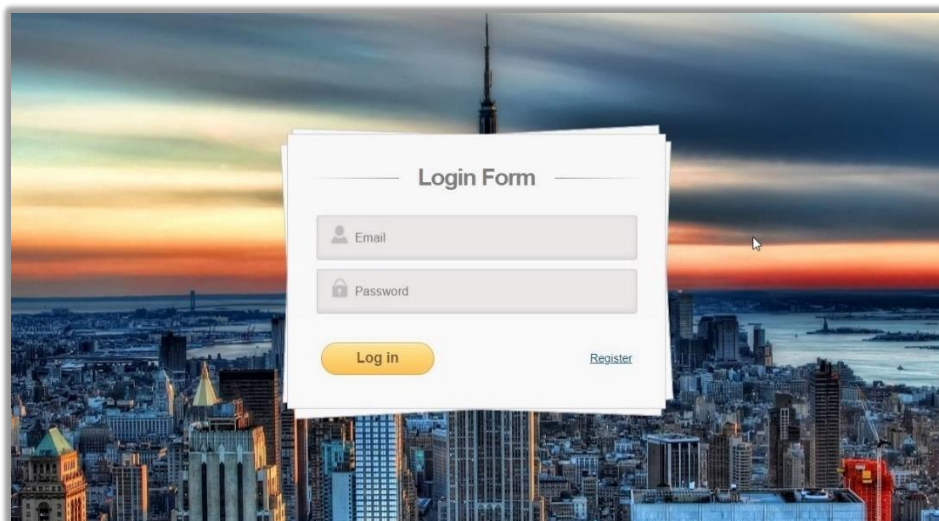


Figure 3.5 Website login page

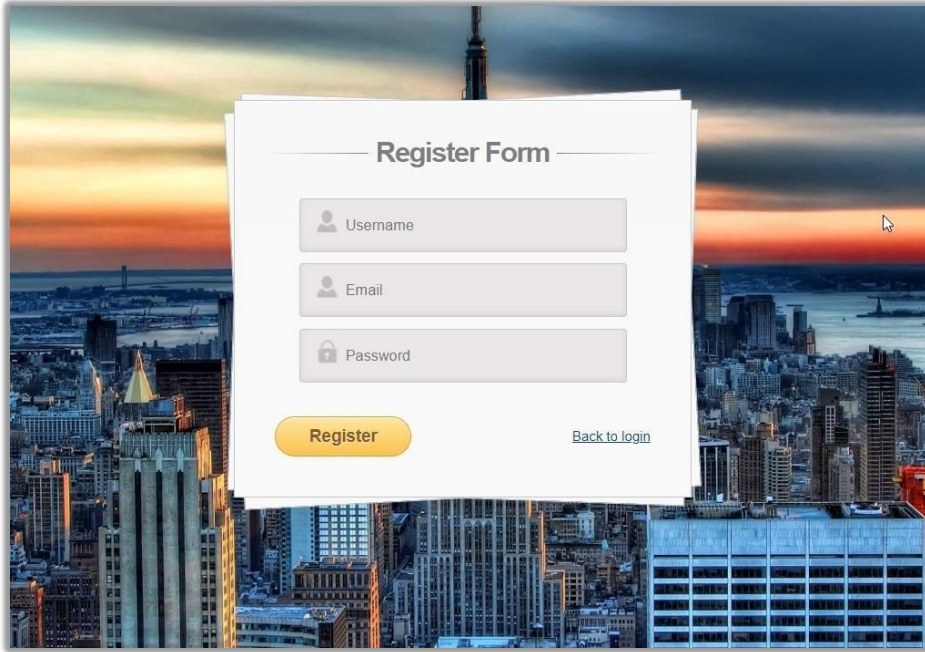


Figure 3.6 Website register page

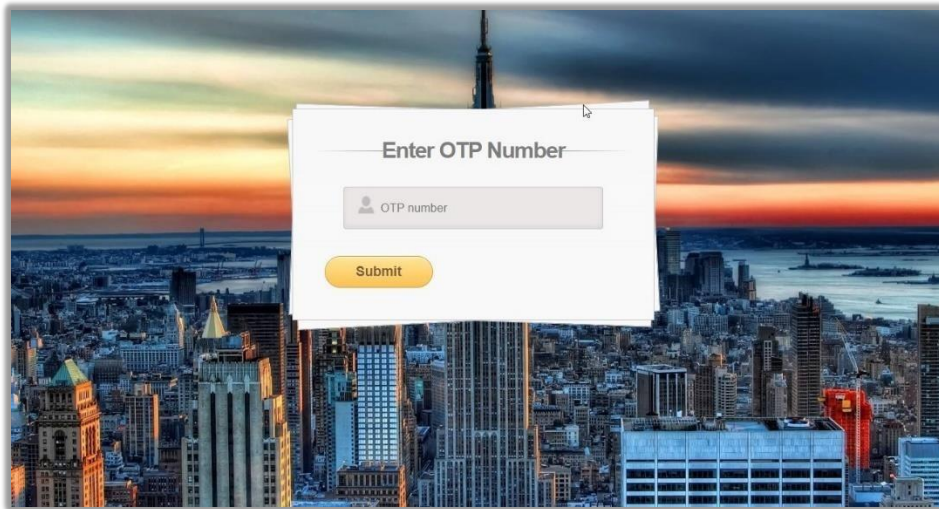


Figure 3.7 Website OTP page



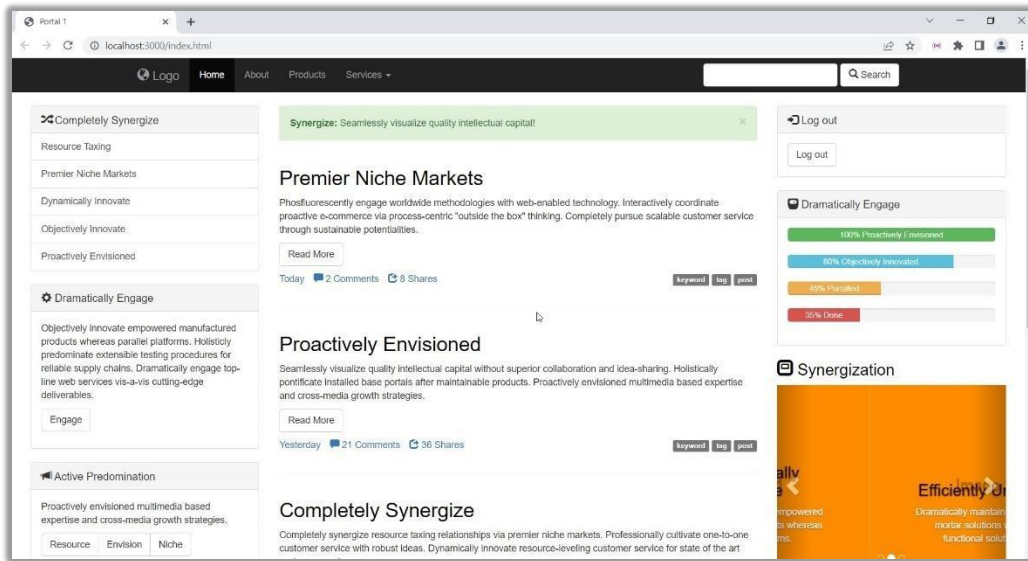


Figure 3.8 Website home page after login

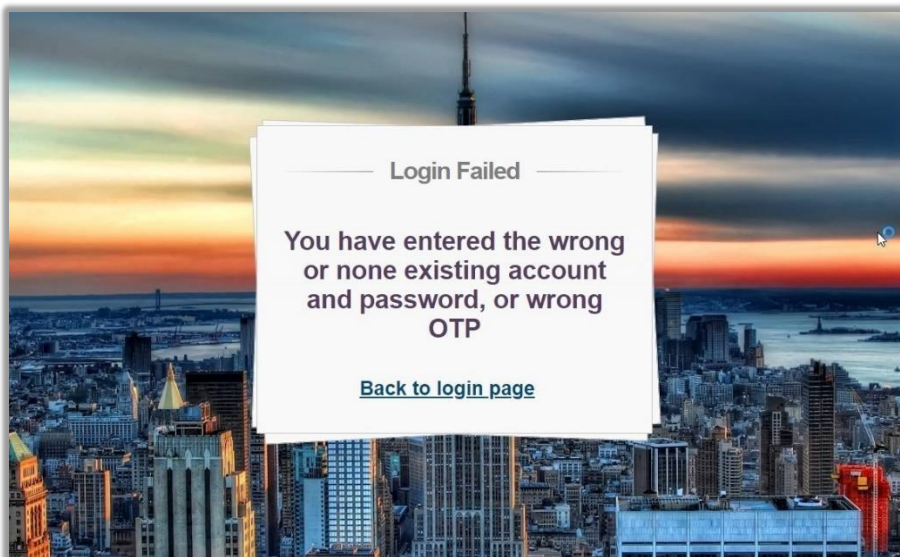


Figure 3.9 Website error page

The figures above show the output of the produced prototype. This prototype is made with Visual Studio Code along with several tools such as MySQL, Xampp, and php script. Firstly, the user will see the login page where they need to input their login information or else register a new account on the register page. If the login info match with the database, then the user will be directed to the OTP page where the user needs to check their email to get the OTP key. After receiving and key in the right OTP key

then the user is authenticated and brought into the home page of the website. However, if the user key in info that does not exist or does not match with the database, the user will be directed to the error page. This error page will be displayed to the user who entered the wrong OTP number as well.

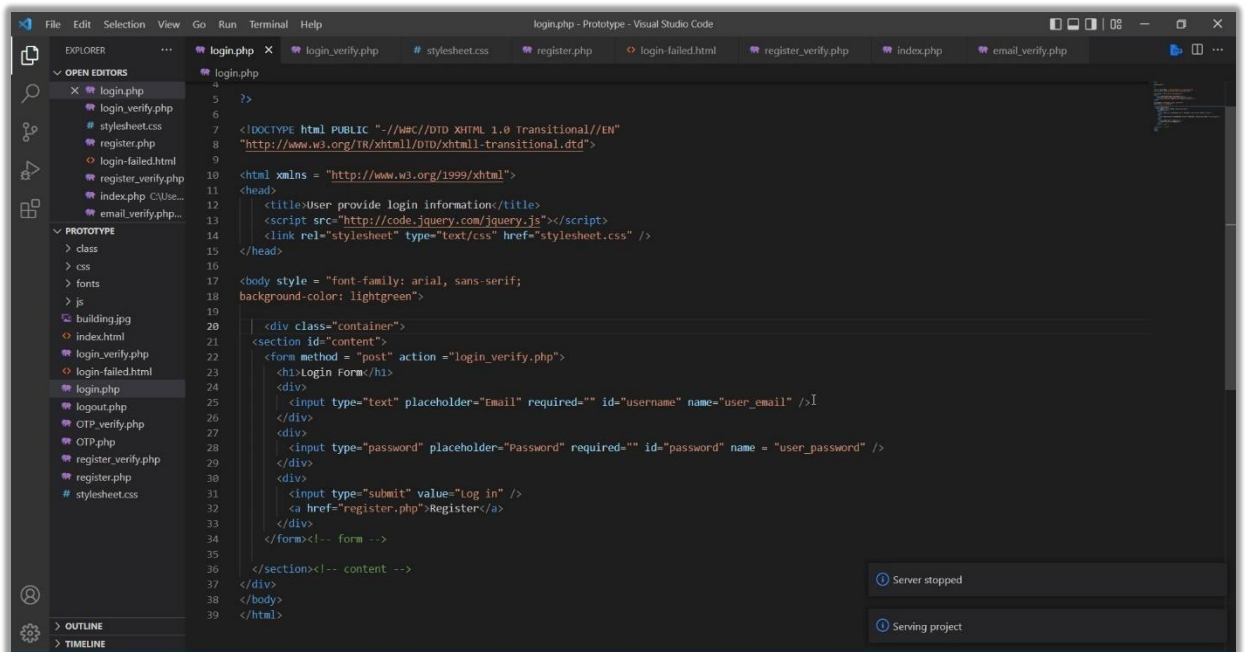


Figure 3.10 Visual Code Studio coding phase

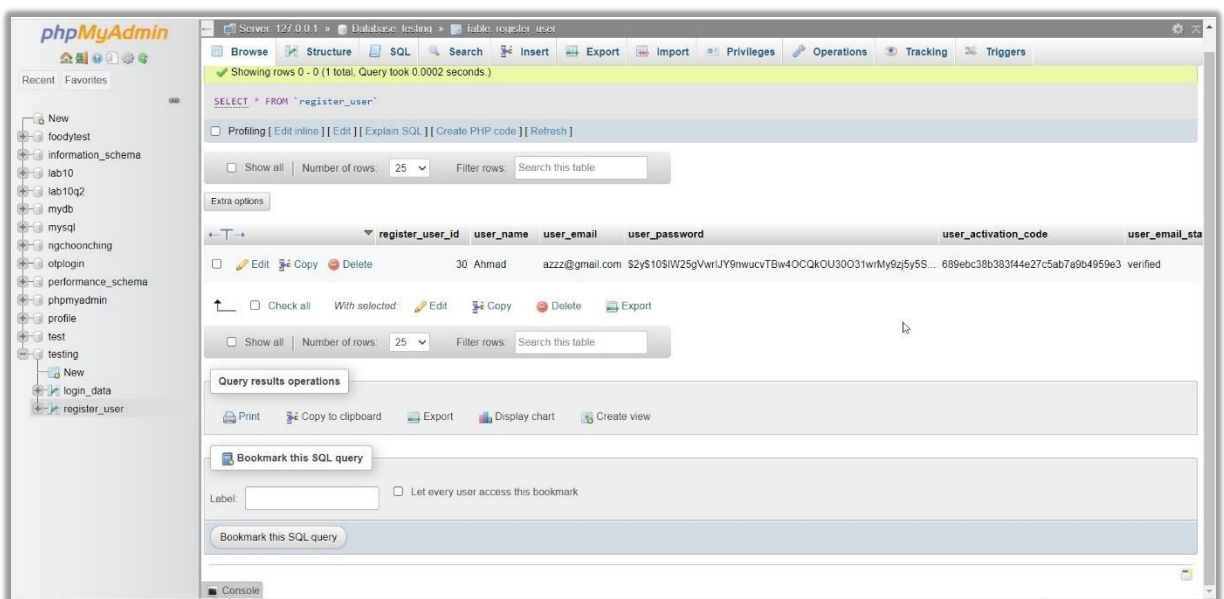


Figure 3.11 Database with user account

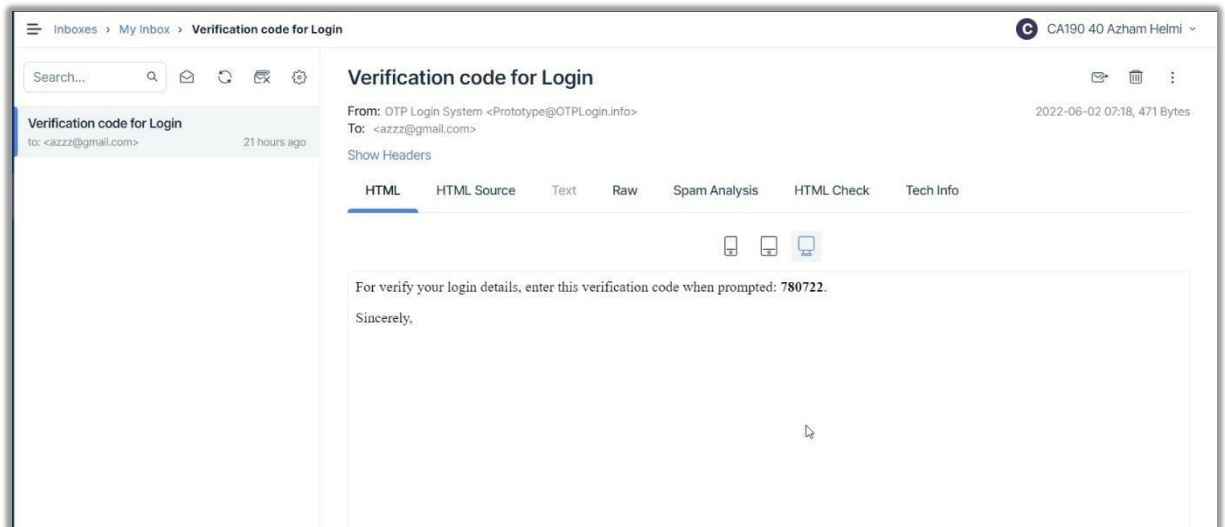


Figure 3.12 Receiving mail with OTP key

The figures show the development proses for the prototype. The website html and php coding is coded with the Visual Studio Code applications and then connected to the database which is MySQL to store the input data and retrieve account information for authentication. If the user manages to pass the login page, they will get a mail from the SMTP server. The mail will contain the OTP key as shown above. The OTP key is generated randomly by the php as coded inside the login page. The key was stored inside the database and used to authenticate the user. If the OTP key generated match with the one that the user gets then it will let the user in the website.

### 3.7 TESTING PLAN

Testing plan is essential in ensuring the designed system running smoothly with little to no error. It will keep the quality of the system in check after the development is finished. The system will deliver the function to the user with best possible quality and reducing maintenance cost in the future. As for this project, User Acceptance Test (UAT) Form are used to determine if the OTP authentication system is working as intended or not. The error detected inside the code will be analysed and fixed to mitigate error after the code is finished. Several volunteer will be picked to test the system if it can run and the results will be documented for further improvement. The test will be

conducted several times until the system meets its initial goal which is successful OTP authentication in the website.

### **3.8 POTENTIAL USE OF PROPOSED SOLUTION**

The proposed solution of this project which is the OTP authentication system have a potential use in real time situation such as:

- i. User can strengthen their account security in a website with extra method or second layer password.
- ii. The system can block unwanted login from fraud or robots that tried to brute force their way into users' account.
- iii. Website integrity will be maintained and is trustworthy to user who access them.

### **3.9 SUMMARY**

In this chapter, the framework used for the system development has been chosen which is the Agile Methodology along with the elaboration of the functional and non-functional requirements of the system. This chapter also has the proposed design which is the OTP authentication system prototype with working modules and outputs. The flow of the system is shown with relation to the database and how it works with the website sending OTP to the user via mail. The system managed to execute its function with several user test and able to operate on normal circumstance.

## **CHAPTER 4**

### **RESULTS AND DISCUSSION**

#### **4.1 INTRODUCTION**

In this chapter, the development and implementation of the One-Time Password Authentication is discussed. This project requires certain important component to run as intended such as web server, MySQL database, GUI and PHP. This one-time password authentication security system is implemented to the university students. Testing of the system was performed to find possible risk and errors in the coding and issued to fix immediately.

#### **4.2 IMPLEMENTATION PROCESS**

The implementation process records all the steps taken in developing the one-time password authentication system. The system need to run on a personal computer (PC) as a device with any web browser installed for user to access the website where the one-time password is implemented. The user should be able to see content of the website such as the login page along with the options to register an account.

##### **4.2.1 Development of Website**

Html coding is used to make the website of the system. HTML or Hypertext Markup Language is a markup language for the web that defines structure of webpages (Kolade Chris, 2021). The login page of the website is the first page that is produced to indicates the user they have entered the website. The login page has a set of inputs and interactions for the user to see and fill. In this project, the html code is designed by using Visual Studio Code to display appropriate GUI of the system.

#### 4.2.1.1 Designing login interface page

Login interface is where user will enter when they load up the one-time password website. The page has a box on the middle of the webpage where it contains several interaction and inputs. The first input is a text box with the label “Email” where user put their email address that is registered in the database. The second text input with the “Password” label is where user type their password that match with the registered email. Next one is a button with the label “Log in”, this button act as a submit button and when it is pressed it will take the input from the two textboxes into the system for query. The “Register” text on the right down side of the box is an option that redirect user to another page which is the register page. User that wanted to register into the system should go to the register page first before logging in.

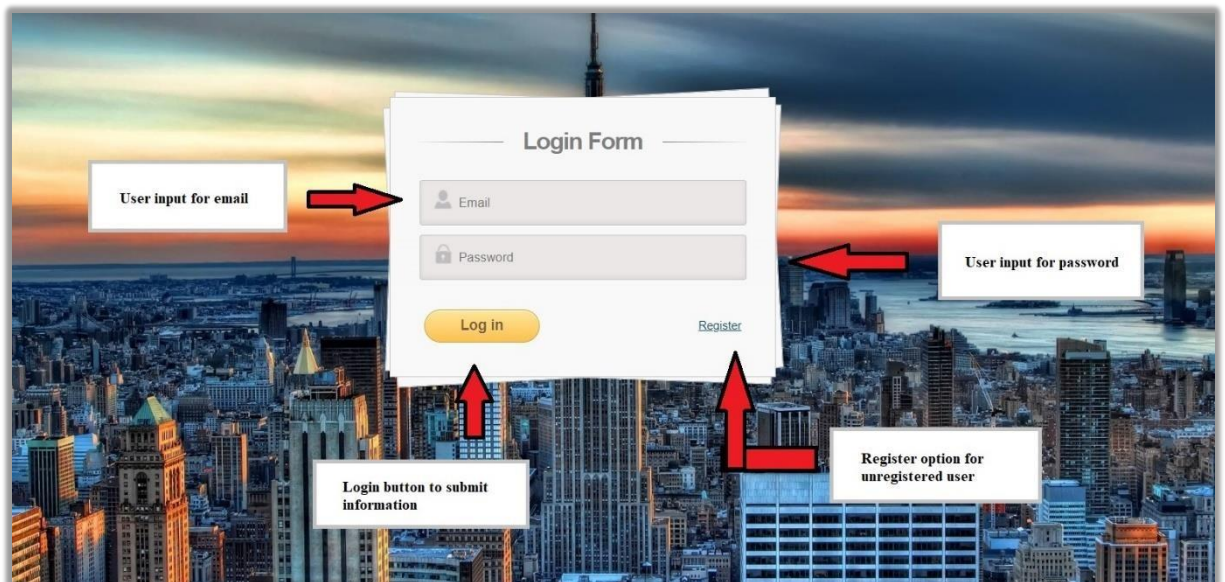


Figure 4.1 Login page interface

#### 4.2.1.2 Designing register interface page

For the register page, the layout of the GUI is the same as the login page but with several differences. In order for the user to register their account in the website, they need to key in their information in the provided input text box first and click the register button. The first input text box with the label “Username” tells user that they have to key in their desired username in the box. The second one is for email, and the third one is for password in which each have their own label so that user will not



confused. On the bottom of the box is a “Register” button and a link that redirect user to the first page which is the login page. The “Register” button act as submit button similar to the login button from the login page.

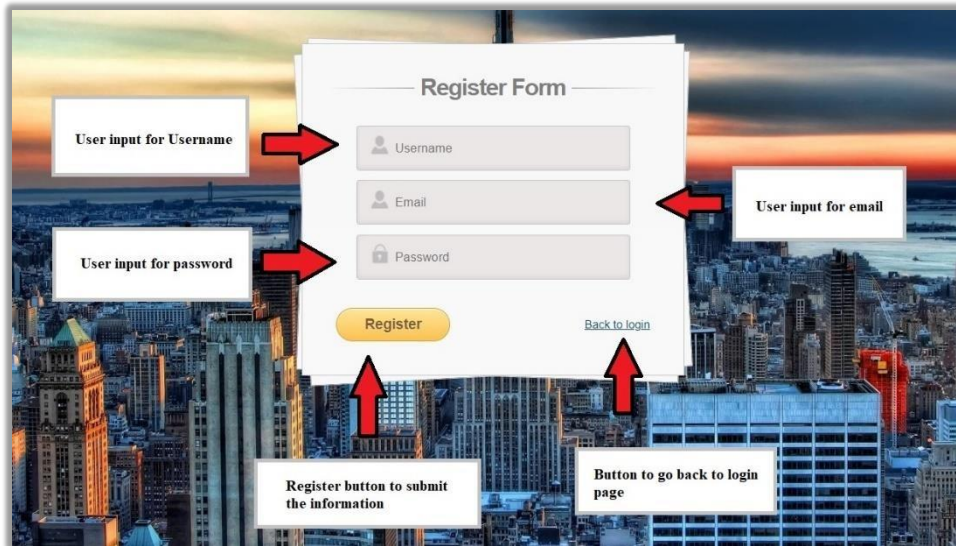


Figure 4.2 Register page interface

#### 4.2.1.3 Designing OTP and error interface page

After a successful matching for account login, the user will be directed to the OTP page. The OTP page have only one input text and one button. The input text space is where the user key in the OTP number that they receive. The “Submit” button will process the input of the user to verify the OTP number. If the OTP number is wrong, user will be directed to an error page. There will be a text that inform the user that they failed to login based on several reason along with a link that sends back the user to the login page.

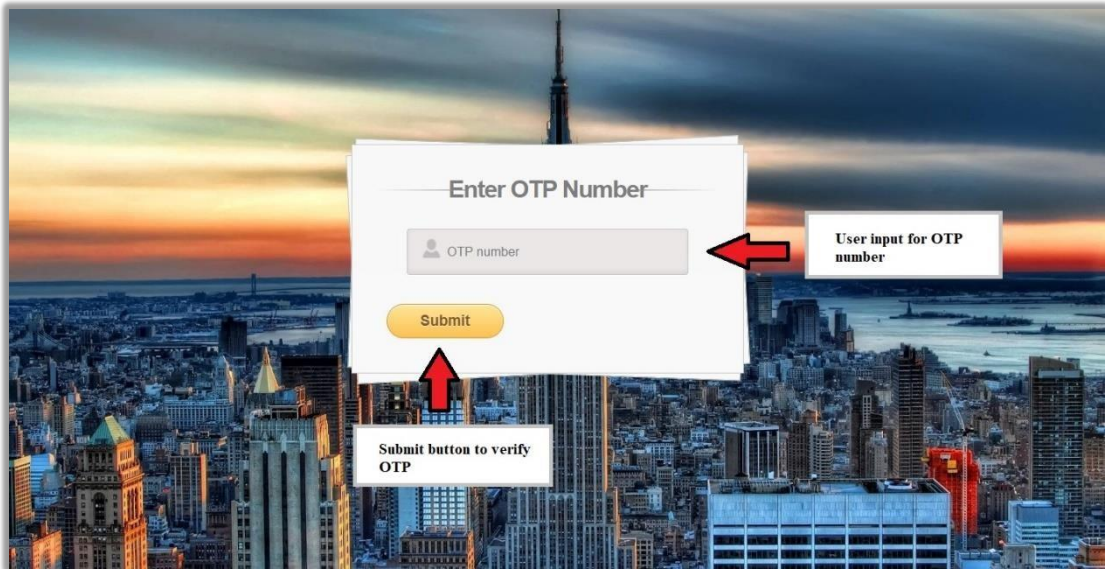


Figure 4.3 OTP page interface

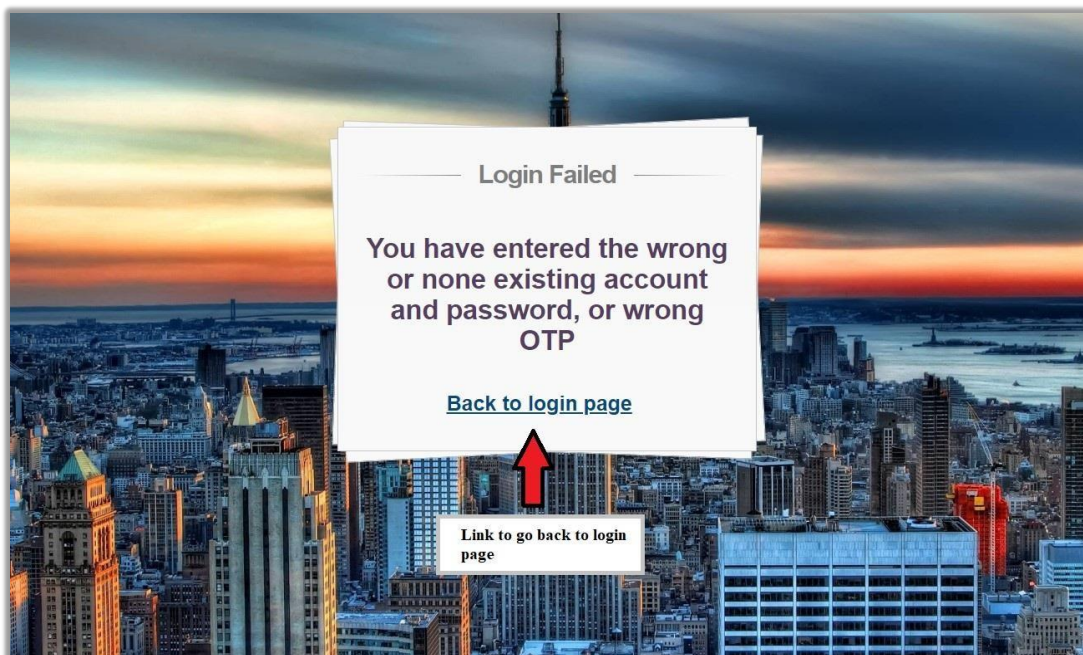


Figure 4.4 Error page interface

#### 4.2.1.4 Designing homepage, profile and logout interface page

A successful login with correct OTP number will direct the user to the homepage of the website. In this website, the content inside it can be anything that is important and require strict access. The user then will be able to access their desired



information in the website protected with OTP authentication. The website also has a log out button for the user to sign out from the website. When they log out, the user will be directed to the log out page which consist of a text that indicates that the user logged out and a link to go back to the login page.

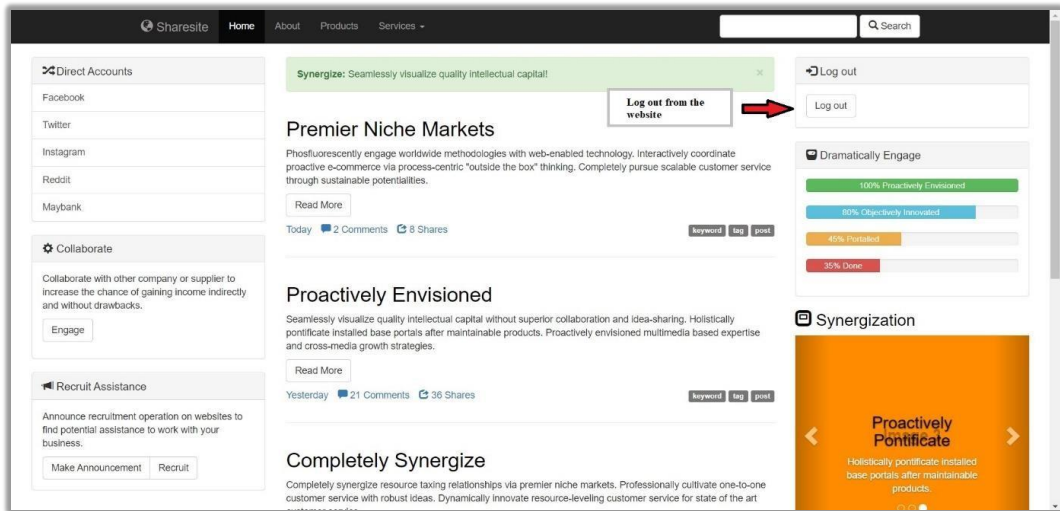


Figure 4.5 Homepage interface of website

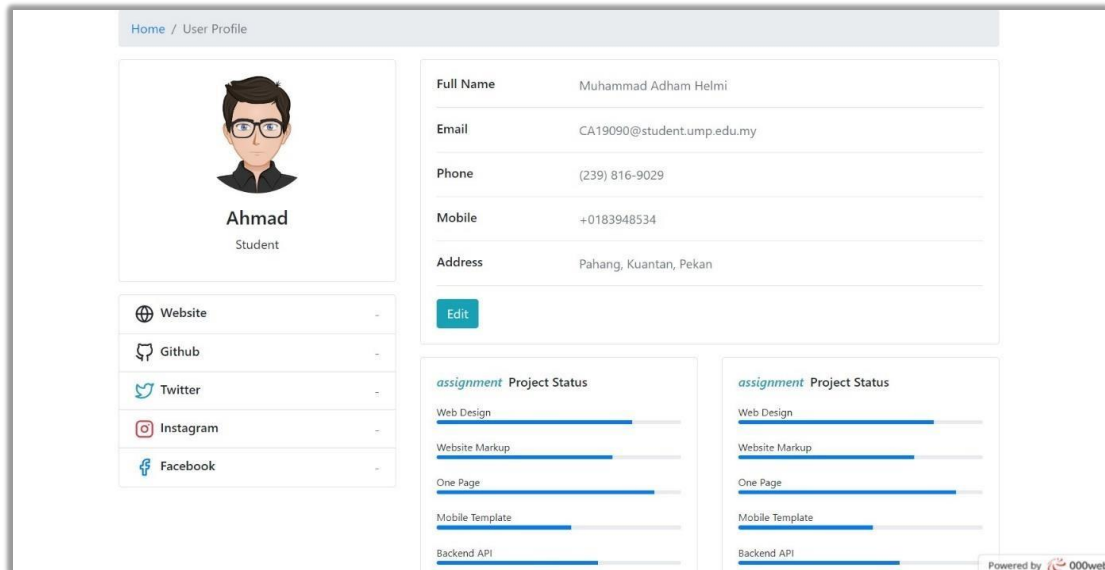


Figure 4.6 Profile interface of website

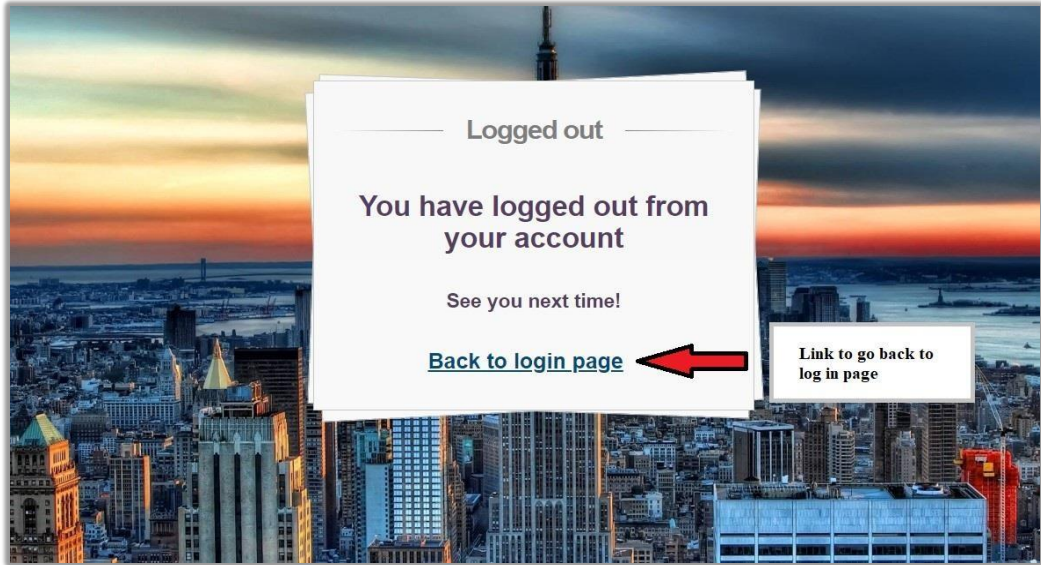


Figure 4.7 Log out interface

#### 4.2.2 Development of Database

After finished designing the interfaces of the website pages, the development of the database of the OTP system is started using MySQL database.

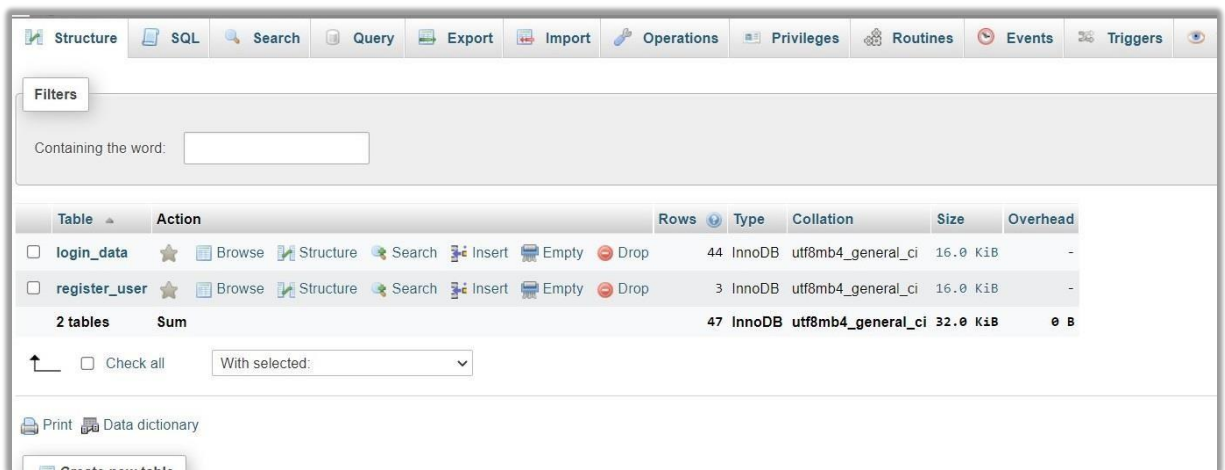
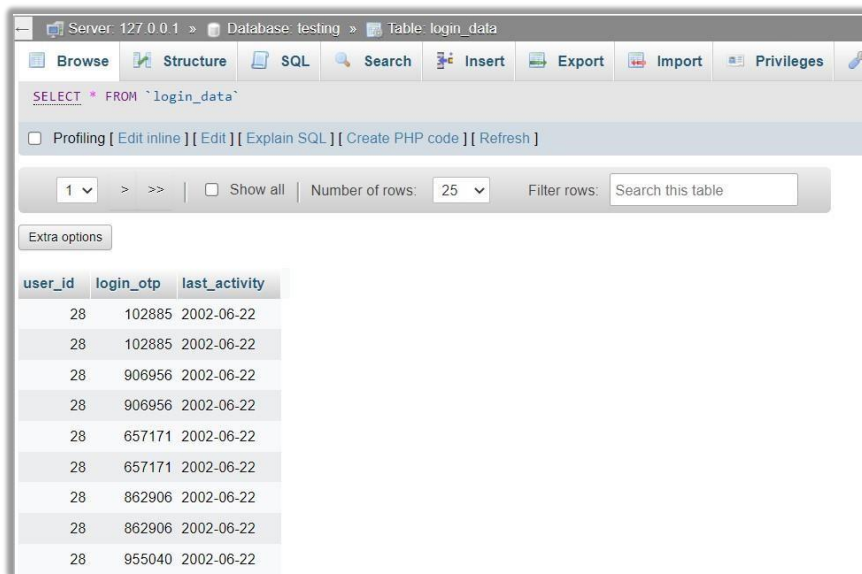


Figure 4.8 Database of login and register

For the database, there are 2 table that is needed for the system to store and recall the user that are registered. The first table is the “login\_data” and second one is “register\_user”. The login data table have 3 column which is “user\_id”, “login\_otp”, and last\_activity”. The purpose of this table is to capture the login information of the

user that tried to login into the website. The key part of this table is the OTP number generated, it must be matched with the user input to verify that they are authorised to login into the website. The register table have 8 column which is “register\_user\_id”, “user\_name”, “user\_email”, “user\_password”, “user\_activation\_code”, “user\_email\_status”, “user\_otp”, “user\_datetime”. New user who registered inside the system will be kept in this table. The password and user activation code is encrypted for safety purposes.



Server: 127.0.0.1 » Database: testing » Table: login\_data

SELECT \* FROM `login\_data`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

1 > >> |  Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

user_id	login_otp	last_activity
28	102885	2002-06-22
28	102885	2002-06-22
28	906956	2002-06-22
28	906956	2002-06-22
28	657171	2002-06-22
28	657171	2002-06-22
28	862906	2002-06-22
28	862906	2002-06-22
28	955040	2002-06-22

Figure 4.9 Login data content

Showing rows 0 - 3 (4 total. Query took 0.0002 seconds)

```
SELECT * FROM `register_user`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	register_user_id	user_name	user_email	user_password	user_activation_code	user_email_status	user_otp	user_datetime
<input type="checkbox"/>	30	Ahmad	azzz@gmail.com	\$2y\$10\$IW25gVwrlJY9nwucvTBw4OQKOU30031wrMy9z5y5S...	689ebc38b38344e27c5ab7a9b4959e3	verified	250377	2022-05-02 13:50:18
<input type="checkbox"/>	35	Userstest1	Userstest1@gmail.com	\$2y\$10\$S6bpZD.t8PQU7XiuOehSeUOMxri7eA1XZyCvJPZupN...	a513a814c5b4b79406b3ab434354f7ad	not verified	771036	2022-05-15 11:43:00
<input type="checkbox"/>	37	Abu	azzz@gmail.com	\$2y\$10\$akvY2xeJZwxvAknCumb2d.sU9Wvsl5eS11JzTsOjmfFn...	f7898f2c677647a540eef3dda9088af	not verified	208656	2022-10-26 08:29:59
<input type="checkbox"/>	39	Ahmad	Ahmad@gmail.com	\$2y\$10\$1GfTqYnIXj1zsGaeADruIW5GintvovsHy5dYzUjKL...	e55a2fb972ac89d5ab7b1d93ee4d2d93	not verified	181849	2022-11-09 16:14:32

Query results operations: Print, Copy to clipboard, Export, Display chart, Create view

Figure 4.10 Register data content

### 4.2.3 Code Use in Visual Studio Code

The data and interfaces that has been coded using html and php are done in Visual Studio Code. This application is suitable for producing detailed websites and easy to implement it. There are several php and html coding produced for the system to work and the coding will be displayed inside the Appendix B due to long coding. Visual Studio Code also have many extensions that can be installed in order to make the production process easier along with troubleshooting error.

```
login.php
1  <?php
2
3  session_start();
4
5  ?>
6
7  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
8  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
9
10 <html xmlns = "http://www.w3.org/1999/xhtml">
11 <head>
12     <title>User provide login information</title>
13     <script src="http://code.jquery.com/jquery.js"></script>
14     <link rel="stylesheet" type="text/css" href="stylesheet.css" />
15 </head>
16
17 <body style = "font-family: arial, sans-serif;
18 background-color: lightgreen">
19
20     <div class="container">
21     <section id="content">
22     <form method = "post" action = "login_verify.php">
23     <h1>Login Form</h1>
24     <div>
25     <input type="text" placeholder="Email" required="" id="username" name="user_email" />
26     </div>
27     <div>
28     <input type="password" placeholder="Password" required="" id="password" name = "user_password" />
29     </div>
30     <div>
31     <input type="submit" value="Log in" />
32     <a href="register.php">Register</a>
33     </div>
34 </form><!-- form -->
```

Figure 4.11 Login page coding in VSC

### 4.3 TESTING

After the interface and coding development of One-Time Password Authentication system is completed, the testing phase is carried out to test the functionality along with the usability of the system. The device used to test the system are phone and tablet. Several users are taken as testers which consist of 10 students of UMP. User Acceptance Test (UAT) is used as a method to carry out the system's functionality and checks if the features are available in the system.

#### 4.3.1 User Acceptance Testing

User Acceptance test is used to test the functions of the one-time password authentication system from the start to the end. Any problem occurred will be noted in

the form. After the UAT form is distributed, the findings indicates that all the tested function are pass. The UAT form can be referred in Appendix C.

### 4.3.2 Website Testing

The one-time password authentication system is conducted on 10 undergraduate UMP students. Ater the testing is conducted; a feedback form will be given to the testers (Refer Appendix D). The form is made to collect feedback from testers after they used the website. The responses gathered shows that the user rated the system between 4 (Agree) and 5 (Strongly Agree).

## 4.4 RESULT AND DISCUSSION

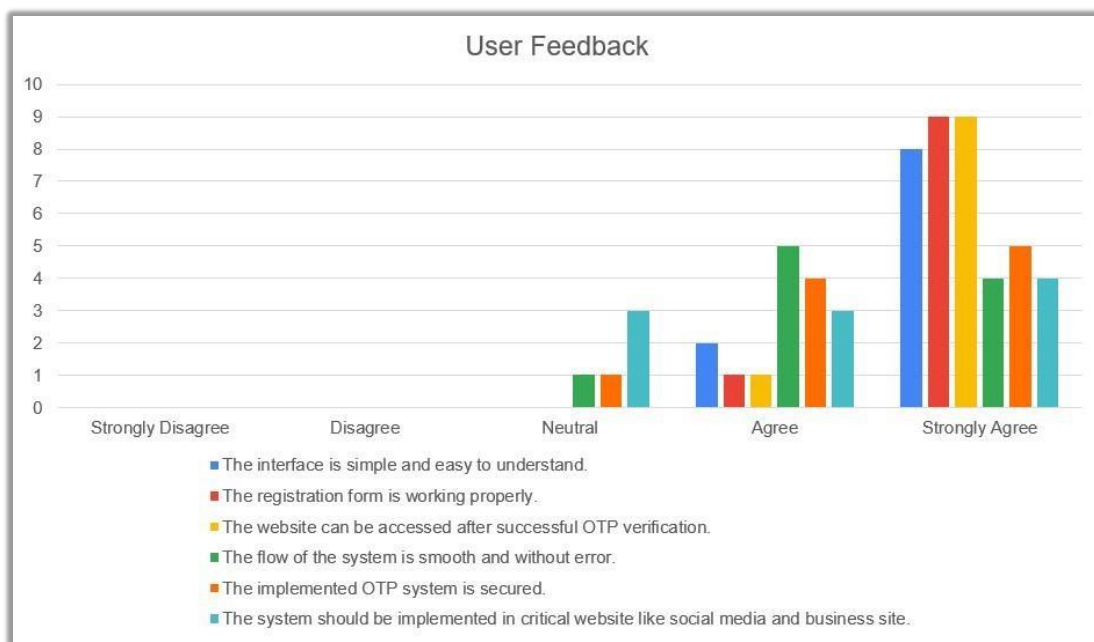


Figure 4.12 Summary of user feedback

Figure 33 shows the result of findings on user feedback after testing the One-Time Password Authentication System in website. 10 UMP students answered the feedback form. They have to evaluate the statement based on the scale strongly disagree, disagree, neutral, agree, and strongly agree. Based on the graph, 80% of the

students' rate strongly agree as they find the interface simple and easy to understand and 20% students' rate for agree. There are 90% of students rate strongly agree that the registration form is working properly along with website can be accessed after successful OTP verification; only 10% student rate agree. For the flow of the system, 40% of students strongly agree that the system is smooth and without error; 50% agree and only 10% is neutral. There are 50% of students who strongly agree that the implemented OTP system is secured, 40% of them agree and 10% rate neutral. Lastly, 40% of student strongly agree that the system should be implemented in critical website like social media and business site, 30% agree to it and the rest 30% student feel neutral. The mean of the result is 3.33. Based on the result, we can conclude that the students give moderately positive feedback toward the system.

Based on the feedback given by the testers, most of them are confident that the system's interface is easy to understand as most of the interface designed based on most popular websites which is simple and straight forward. Next, users very confident that the registration form is working properly as they successfully register their account in the system without any problem or error. It can be confirmed that users also very confident that the website can be accessed after successful OTP verification as they directed to the website when they enter the right OTP number. However, users are less confident that the flow of the system is smooth and without error. This is because they think that the waiting time for the system to register and login are quite long compared to other popular websites even without error. Users also feel less confident that the implemented OTP system is secured; this is because the system implemented are quite ordinary compared to other implementation from popular websites. The OTP system still work as intended just that user tends to feel less confident to a new produced system. Last but not least, users' opinion on should the system be implemented in critical website like social media and business site is the least confident. It can be said that users do not really need the OTP even in critical site as it will slow down their time in logging into their account.

## **CHAPTER 5**

### **CONCLUSION**

#### **5.1 INTRODUCTION**

Chapter 5 discuss the summarization of findings in developing one-time password authentication system in website for students in order to reach the objectives and solve the problems that have been stated in problem statement in Chapter 1. Nowadays, people around the world especially students are prone to cyber-attacks such as unauthorized access to account in various websites. This authentication system acts as a functional tool to mitigate attacks by applying another layer of authentication to verify user before accessing the account. Students can protect their account in websites by applying this authentication system. The application and tools used to develop this system is Visual Studio Code, MySQL, Apache, PHP, and HTML code. This system is developed by using Agile methodology which effectively reduces time and development cost. This system is executed and evaluated by UMP students to test the practicality and functionality of the system. The evaluation assessment shows that the students give moderately positive feedback and the system reach the objectives of the system.

#### **5.2 RESEARCH CONSTRAINT**

The constraints in this project are:

##### **Sending mail with Google SMTP**

The authentication system requires to send mail to the user who login into the website. Sending a mail to an email with proper SMTP is needed so that the system can



work in desired method. Several desired SMTP such as Google have constrain and block the ports which render the SMTP unusable.

### **Dummy account**

The current system can be registered with any account. The account registered inside the system are not based on real account and only applicable on the system.

### **Improving algorithm**

The current system algorithm needs time to improve. The system used an existing algorithm which is prone to attacks. A new algorithm should be able to generate a unique OTP key compared to the existing algorithm.

## **5.3 FUTURE WORK**

There are several improvements that can be done for future enhancement of One-Time Password Authentication System in Website for general people and students.

- i. Developer has to implement the system on a workable website that contains crucial component involving real accounts of the user.
- ii. Developer has to send mail to the real accounts of user who are registering into the website. The accounts should be able to receive the mail sent by the system.
- iii. Developer has to add an option to resend the OTP to the user in the interface so that user can receive the OTP successfully.

## REFERENCES

1. Teju Shyamsundar (2020, June 24). What is a One-Time Password (OTP)? Retrieved April 11, 2022, from <https://www.okta.com/blog/2020/06/what-is-a-one-timepassword-otp/>
2. Will Kenton (2020, September 28). Two-Factor Authentication (2FA). Retrieved April 12, 2022, from <https://www.investopedia.com/terms/t/twofactor-authentication-2fa.asp>
3. Caitlin Jones (2022, January 24). 50 Web Security Stats You Should Know in 2022. Retrieved April 12, 2022, from <https://expertinsights.com/insights/50-web-securitystats-you-should-know/>
4. Steven Feltner (2016, December 28). Single-factor Authentication (SFA) vs. Multifactor Authentication (MFA). Retrieved April 13, 2022, from <https://www.centrify.com/blog/sfa-mfa-difference/>
5. Ashley Watters (2022, January 11). Top 50 Cybersecurity Statistics, Figures and Facts. Retrieved April 13, 2022, from <https://connect.comptia.org/blog/cyber-security-statsfacts#:~:text=6.95%20million%20new%20phishing%20and,bot%20traffice%2C%20ac%20ording%20to%20Imperva.>
6. Prakash Sharma (2018, June 18). How Time-based One-Time Passwords work and why you should use them in your app. Retrieved April 13, 2022, from <https://www.freecodecamp.org/news/how-time-based-one-time-passwords-work-andwhy-you-should-use-them-in-your-app-fdd2b9ed43c3/>
7. Richards, K., & Wigmore, I. (2021, September 29). One-time password (OTP). SearchSecurity. Retrieved April 13, 2022, from [https://www.techtarget.com/searchsecurity/definition/one-time-password-OTP#:~:text=A%20one%2Dtime%20password%20\(OTP\)%20is%20an%20automatically%20generated,or%20reused%20across%20multiple%20accounts](https://www.techtarget.com/searchsecurity/definition/one-time-password-OTP#:~:text=A%20one%2Dtime%20password%20(OTP)%20is%20an%20automatically%20generated,or%20reused%20across%20multiple%20accounts)

8. Jacob, J. (2021, October 15). What is One Time Password and How It Can Help Your Business. TapTalk.Io Blog. Retrieved April 13, 2022, from <https://taptalk.io/blog/whatis-one-time-password-and-how-it-can-help-your-business/>
9. Reed, J. (2021, December 15). One-Time Password Security Might Fail 80% of the Time. IAM is Better. Security Intelligence. Retrieved April 13, 2022, from <https://securityintelligence.com/articles/one-time-password-security-fails-80-percentiam-better/>
10. R. (2022, May 20). One-time Passwords (OTP): A Beginner's Guide 2022. 10Duke. Retrieved April 13, 2022, from <https://www.10duke.com/blog/one-time-passwords-abeginners-guide/>

## APPENDIX A

Gantt Chart of project

Activity	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
First Meeting with Supervisor	█	█												
Chapter 1 – Introduction		█	█											
Chapter 1 – Submission			█											
Second Meeting with Supervisor			█											
Third meeting with Supervisor				█										
Chapter 2 – Literature review				█	█									
Chapter 2 – Submission					█									
Chapter 3 – Methodology					█	█	█	█	█	█	█	█	█	
Forth meeting with Supervisor												█		
Report submission													█	
Presentation submission														█

Activity	Week																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
First Meeting with Supervisor	█	█															
Chapter 4 – Draft		█	█	█	█	█											
Chapter 4 – Submission							█										
Second Meeting with Supervisor							█										

Third meeting with Supervisor																			
Chapter 5 – Draft																			
Chapter 5 – Submission																			
Thesis - Draft																			
Thesis, Turnitin, Logbook, Poster - Submission																			
Presentation																			
Final report Submission																			

## APPENDIX B

```
<?php
session_start();

?>

<!DOCTYPE html PUBLIC "-//W#C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
    <title>User provide login information</title>
    <script src="http://code.jquery.com/jquery.js"></script>
    <link rel="stylesheet" type="text/css" href="stylesheet.css" />
</head>

<body style = "font-family: arial, sans-serif;
background-color: lightgreen">

    <div class="container">
<section id="content">
    <form method = "post" action = "login_verify.php">
        <h1>Login Form</h1>
        <div>
            <input type="text" placeholder="Email" required="" id="username"
name="user_email" />
        </div>
        <div>
            <input type="password" placeholder="Password" required=""
id="password" name = "user_password" />
        </div>
        <div>
            <input type="submit" value="Log in" />
            <a href="register.php">Register</a>
        </div>
    </form><!--form ->

    </section><!--content ->
</div>
</body>
</html>
```

Login page coding

```

<?php

//login_verify.php

$connect = new PDO("mysql:host=localhost;dbname=testing", "root", "");

session_start();

if($_POST["user_email"] != '')
{
    $data = array(
        ':user_email' => $_POST["user_email"]
    );

    $query = "
SELECT * FROM register_user
WHERE user_email = :user_email
";

    $statement = $connect->prepare($query);

    $statement->execute($data);

    $total_row = $statement->rowCount();

    if($total_row == 0)
    {
        header("Location: login-failed.html");
    }
    else
    {
        $result = $statement->fetchAll();

        foreach($result as $row)
        {
            $_SESSION["register_user_id"] =
$row["register_user_id"];

            $_SESSION["user_name"] = $row["user_name"];

            $_SESSION['user_email'] = $row["user_email"];

            $_SESSION["user_password"] = $row["user_password"];
        }
    }
}
else

```

```

{
    header("Location: login-failed.html");
}

if($_POST["user_password"] != '')
{
    if(password_verify($_POST["user_password"],
$_SESSION["user_password"]))
    {
        $login_otp = rand(100000,999999);

        $data = array(
            ':user_id'      => $_SESSION["register_user_id"],
            ':login_otp'    => $login_otp,
            ':last_activity'=> date('d-m-y h:i:s')
        );

        $query = "
INSERT INTO login_data
(user_id, login_otp, last_activity)
VALUES (:user_id, :login_otp, :last_activity)
";

        $statement = $connect->prepare($query);

        $statement->execute($data);

        if($statement->execute($data))
        {
            $_SESSION['login_id'] = $connect->lastInsertId();
            $_SESSION['login_otp'] = $login_otp;

            require 'class/vendor/autoload.php';
            require 'class/PHPMailer.php';

            $mail = new PHPMailer\PHPMailer\PHPMailer();
            $mail->IsSMTP();
            $mail->Host = 'smtp.mailtrap.io';
            $mail->Port = 2525;
            $mail->SMTPAuth = true;
            $mail->Username = '5ebd1b7b2855a8';
            $mail->Password = '99324fe26d6968';
            $mail->SMTPSecure = 'tls';
            $mail->From = 'Prototype@OTPLogin.info';
            $mail->FromName = 'OTP Login System';
            $mail->AddAddress($_SESSION["user_email"]);
            $mail->WordWrap = 50;
            $mail->IsHTML(true);

```



```

        $mail->Subject = 'Verification code for Login';

        $message_body = '
        <p>For verify your login details, enter this
verification code when prompted: <b>'.$login_otp.'</b>.</p>
        <p>Sincerely,</p>
        '

        $mail->Body = $message_body;

        if($mail->Send())
        {
            header("Location: OTP.php");
        }
    }
    else
    {
        header("Location: login-failed.html");
    }
}
else
{
    header("Location: login-failed.html");
}

?>

```

Login verify coding

```

<?php

session_start();

if(isset($_SESSION["user_id"]))
{
    header("location:home.php");
}

$connect = new PDO("mysql:host=localhost; dbname=testing", "root", "");

$message = '';
$error_user_name = '';
$error_user_email = '';
$error_user_password = '';

```

```

$user_name = '';
$user_email = '';
$user_password = '';

if(isset($_POST["register"]))
{
    if(empty($_POST["user_name"]))
    {
        $error_user_name = "<label class='text-danger'>Enter Name</label>";
    }
    else
    {
        $user_name = trim($_POST["user_name"]);
        $user_name = htmlentities($user_name);
    }

    if(empty($_POST["user_email"]))
    {
        $error_user_email = '<label class="text-danger">Enter Email
Address</label>';
    }
    else
    {
        $user_email = trim($_POST["user_email"]);
        if(!filter_var($user_email, FILTER_VALIDATE_EMAIL))
        {
            $error_user_email = '<label class="text-danger">Enter Valid
Email Address</label>';
        }
    }

    if(empty($_POST["user_password"]))
    {
        $error_user_password = '<label class="text-danger">Enter
Password</label>';
    }
    else
    {
        $user_password = trim($_POST["user_password"]);
        $user_password = password_hash($user_password, PASSWORD_DEFAULT);
    }

    if($error_user_name == '' && $error_user_email == '' &&
$error_user_password == '')
    {
        $user_activation_code = md5(rand());

        $user_otp = rand(100000, 999999);
    }
}

```

```

$data = array(
    ':user_name'      => $user_name,
    ':user_email'    => $user_email,
    ':user_password' => $user_password,
    ':user_activation_code' => $user_activation_code,
    ':user_email_status'=> 'not verified',
    ':user_otp'      => $user_otp
);

$query = "
INSERT INTO register_user
(user_name, user_email, user_password, user_activation_code,
user_email_status, user_otp)
SELECT * FROM (SELECT :user_name, :user_email, :user_password,
:user_activation_code, :user_email_status, :user_otp) AS tmp
WHERE NOT EXISTS (
    SELECT user_email FROM register_user WHERE user_email =
:user_email
) LIMIT 1
";

(statement = $connect->prepare($query);

(statement->execute($data);

if($connect->lastInsertId() == 0)
{
    $message = '<label class="text-danger">Email Already
Register</label>';
}
else
{

    require 'class/vendor/autoload.php';
    require 'class/PHPMailer.php';
    $mail = new PHPMailer\PHPMailer\PHPMailer();
    $mail->IsSMTP();
    $mail->Host = 'smtp.mailtrap.io';
    $mail->Port = 2525;
    $mail->SMTPAuth = true;
    $mail->Username = '5ebd1b7b2855a8';
    $mail->Password = '99324fe26d6968';
    $mail->SMTPSecure = 'tls';
    $mail->From = 'Prototype@OTPLLogin.info';
    $mail->FromName = 'OTP Login System';
    $mail->AddAddress($user_email);

```

```

        $mail->WordWrap = 50;
        $mail->IsHTML(true);
        $mail->Subject = 'Verification code for Verify Your Email
Address';

        $message_body = '
        <p>For verify your email address, enter this verification code
when prompted: <b>'. $user_otp. '</b>.</p>
        <p>Sincerely,</p>
        ';
        $mail->Body = $message_body;

        if($mail->Send())
        {
            echo '<script>alert("Please Check Your Email for
Verification Code")</script>';

            header('location:login.php');
        }
        else
        {
            $message = $mail->ErrorInfo;
        }
    }
}

?>
?>
<!DOCTYPE html PUBLIC "-//W#C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
    <title>User Register Information</title>
    <link rel="stylesheet" type="text/css" href="stylesheet.css" />
</head>

<body style = "font-family: arial, sans-serif;
background-color: lightgreen">

    <div class="container">
<section id="content">
    <form method = "post">
        <h1>Register Form</h1>
    </div>

```

```

        <input type="text" placeholder="Username" required="" id="username"
name="user_name" />
    </div>
    <div>
        <input type="text" placeholder="Email" required="" id="username"
name="user_email" />
    </div>
    <div>
        <input type="password" placeholder="Password" required=""
id="password" name = "user_password" />
    </div>
    <div>
        <input type="submit" name="register" value="Register" />
        <a href="login.php">Back to login</a>
    </div>
</form><!--form ->

</section><!--content ->
</div>
</body>
</html>

```

Register page coding

```

<?php

//email_verify.php

$connect = new PDO("mysql:host=localhost;dbname=testing", "root", "");

$error_user_otp = '';
$user_activation_code = '';
$message = '';

if(isset($_GET["code"]))
{
    $user_activation_code = $_GET["code"];

    if(isset($_POST["submit"]))
    {
        if(empty($_POST["user_otp"]))
        {
            $error_user_otp = 'Enter OTP Number';
        }
    }
}

```

```

else
{
    $query = "
SELECT * FROM register_user
WHERE user_activation_code = ".$user_activation_code."
AND user_otp = ".trim($_POST["user_otp"])."
";

    $statement = $connect->prepare($query);

    $statement->execute();

    $total_row = $statement->rowCount();

    if($total_row > 0)
    {
        $query = "
UPDATE register_user
SET user_email_status = 'verified'
WHERE user_activation_code = ".$user_activation_code."
";

        $statement = $connect->prepare($query);

        if($statement->execute())
        {
            header('location:login.php');
        }
    }
    else
    {
        $message = '<label class="text-danger">Invalid OTP
Number</label>';
    }
}

}
else
{
    $message = '<label class="text-danger">Invalid Url</label>';
}

?>
?>

<!DOCTYPE html PUBLIC "-//W#C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns = "http://www.w3.org/1999/xhtml">

```

```

<head>
  <title>User provide login information</title>
  <link rel="stylesheet" type="text/css" href="stylesheet.css" />
</head>

<body style = "font-family: arial, sans-serif;
background-color: lightgreen">

  <div class="container">
<section id="content">
  <form method = "post" >
    <h1>Enter OTP Number</h1>
    <div>
      <input type="text" placeholder="OTP number" required=""
id="username" name="user_otp" />
    </div>
    <div>
      <input type="hidden" name="<?php echo $login_otp ?>"
value="login_otp">
      <input type="submit" name="submit" value="Submit" />
    </div>
  </form><!--form ->

  </section><!--content ->
</div>
</body>
</html>

```

Register verify coding

```

<?php
  $connect = new PDO("mysql:host=localhost;dbname=testing", "root", "");

  session_start();
  $login_otp = $_SESSION['login_otp'];
  $_SESSION['login_otp'] = $login_otp;
  ?>
<!DOCTYPE html PUBLIC "-//W#C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
  <title>User provide login information</title>
  <link rel="stylesheet" type="text/css" href="stylesheet.css" />
</head>

<body style = "font-family: arial, sans-serif;

```

```

background-color: lightgreen">

    <div class="container">
    <section id="content">
        <form method = "post" action="OTP_verify.php">
            <h1>Enter OTP Number</h1>
            <div>
                <input type="text" placeholder="OTP number" required=""
id="username" name="user_otp" />
            </div>
            <div>
                <input type="hidden" name="<?php echo $login_otp ?>"
value="login_otp">
                <input type="submit" value="Submit" />
            </div>
        </form><!--form →

    </section><!--content →
</div>
</body>
</html>

```

OTP page coding

```

<?php

//otp_verify.php

$connect = new PDO("mysql:host=localhost;dbname=testing", "root", "");

session_start();

$login_otp = $_SESSION['login_otp'];

    if($_POST["user_otp"] != '')
    {
        if($login_otp == $_POST["user_otp"])
        {
            $_SESSION['user_id'] = $_SESSION['register_user_id'];
            unset($_SESSION["register_user_id"]);
            unset($_SESSION["user_email"]);
            unset($_SESSION["user_password"]);
            unset($_SESSION["login_otp"]);
            header("Location: index.html");
            exit();
        }
    }

```



```

    }
    else
    {
        header("Location: login-failed.html");
        exit();
    }
}
else
{
    echo'OTP Number is required';
    exit();
}

?>

```

OTP verify coding

```

<!DOCTYPE html>
<html>
  <head>
    <title>Login failed</title>
    <link rel="stylesheet" type="text/css" href="stylesheet.css" />
  </head>
  <body>

    <div class="container">
      <section id="content">
        <form action="">
          <h1>Login Failed</h1>
          <div>
            <br>
          </div>
          <div>
            <h2>You have entered the wrong or none existing account and
password, or wrong OTP</h2>
          </div>
          <br><br>
          <div>
            <h3><a href="login.php" style="float:none;font-size:
20px;">Back to login page</a></h3>
          </div>
          <div>
            <br><br>
          </div>
        </form>
      </section>
    </div>

```

```
        </section><!--content →  
    </div>  
</body>  
</html>
```

Login failed coding

## APPENDIX C

### USER ACCEPTANCE TEST (UAT)

No	Event	Pass / Fail
Login Page		
1	Able to click “Log in” button	Pass
2	Able to click “Register” button	Pass
3	Able to fill text in “email” textbox	Pass
4	Able to fill text in “password” textbox	Pass
Register Page		
1	Able to click “Register” button	Pass
2	Able to click “Back to login” button	Pass
3	Able to fill text in “Username” textbox	Pass
4	Able to fill text in “Email” textbox	Pass
5	Able to fill text in “Password” textbox	Pass
OTP page		
1	Able to click “Submit” button	Pass
2	Able to fill number in “OTP number” textbox	Pass
Error Page		
1	Able to click “Back to login page” button	Pass
Homepage		
1	Able to click “Profile” button	Pass
2	Able to click “Facebook” button	Pass
3	Able to click “Twitter” button	Pass
4	Able to click “Instagram” button	Pass
5	Able to click “Reddit” button	Pass
6	Able to click “Maybank” button	Pass
7	Able to click “Log out” button	Pass
Profile Page		
1	Able to click “Home” button	Pass
Log out Page		

1	Able to click “Back to login page” button	Pass
---	---	------

## APPENDIX D

Feedback form that is given to the user who already test the system:

### One-Time Password Authentication System in Website

Thank you for participating on this OTP system testing. Please rate the system to indicate your opinion towards this project. I want to hear your feedback so I can keep improving the system. Please fill this quick survey and let me know your thoughts.

[Sign in to Google](#) to save your progress. [Learn more](#)

**\* Required**

1. The interface is simple and easy to understand. \*

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

2. The registration form is working properly. \*

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

3. The website can be accessed after successful OTP verification. \*

Strongly Disagree

Disagree

Neutral

Agree

Strongly Agree

4. The flow of the system is smooth and without error. \*

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

5. The implemented OTP system is secured. \*

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

6. The system should be implemented in critical website like social media and business site. \*

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

**Submit**

[Clear form](#)

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#).

Google Forms

## **APPENDIX E**

### **USER MANUAL FOR ONE-TIME PASSWORD AUTHENTICATION SYSTEM**

#### **TABLES OF CONTENT**

#### **1.0 GENERAL INFORMATION**

1.1 System Overview

#### **2.0 SYSTEM SUMMARY**

2.1 System Configuration

#### **3.0 GETTING STARTED**

3.1 System Page

3.1.1 Login Page

3.1.2 Register Page

3.1.3 OTP Page

## **1.0 GENERAL INFORMATION**

### **1.1 System Overview**

One-Time Password Authentication system in website is a security system for students and general user to secure accounts registered in a website. This system is designed to protect accounts of users like students from being attacked or steal from unauthorised users. At the same time, the website integrity and security is improved.

## **2.0 SYSTEM SUMMARY**

### **2.1 System Configuration**

One-time password authentication system operates on devices such as personal computers (PC), tablets, and mobile phones through web application. It requires internet connection to execute and Mailtrap tool to retrieve the password sent by the system.

## **3.0 GETTING STARTED**

### **3.1 System Page**

#### **3.1.1 Login Page**

Login page will appear when the user enters the website. Figure 1 shows the login page in One-Time Password Authentication website. Table 1 shows the description of the login page.



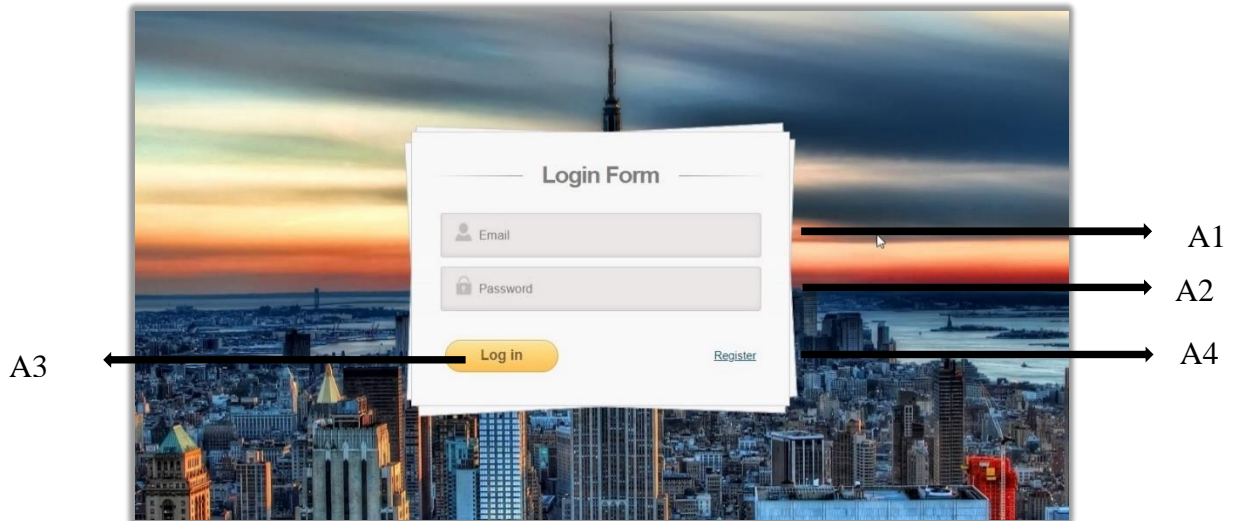


Figure 1 Login Page interface

Table 1 Login page interface description

Function	Description
A1	Input for email
A2	Input for password
A3	Submit information
A4	Navigate user to register page

### 3.1.2 Register Page

Register page is where user will register a new account in the website. Figure 2 shows the register page in One-Time Password Authentication website. Table 2 shows the description of the register page.

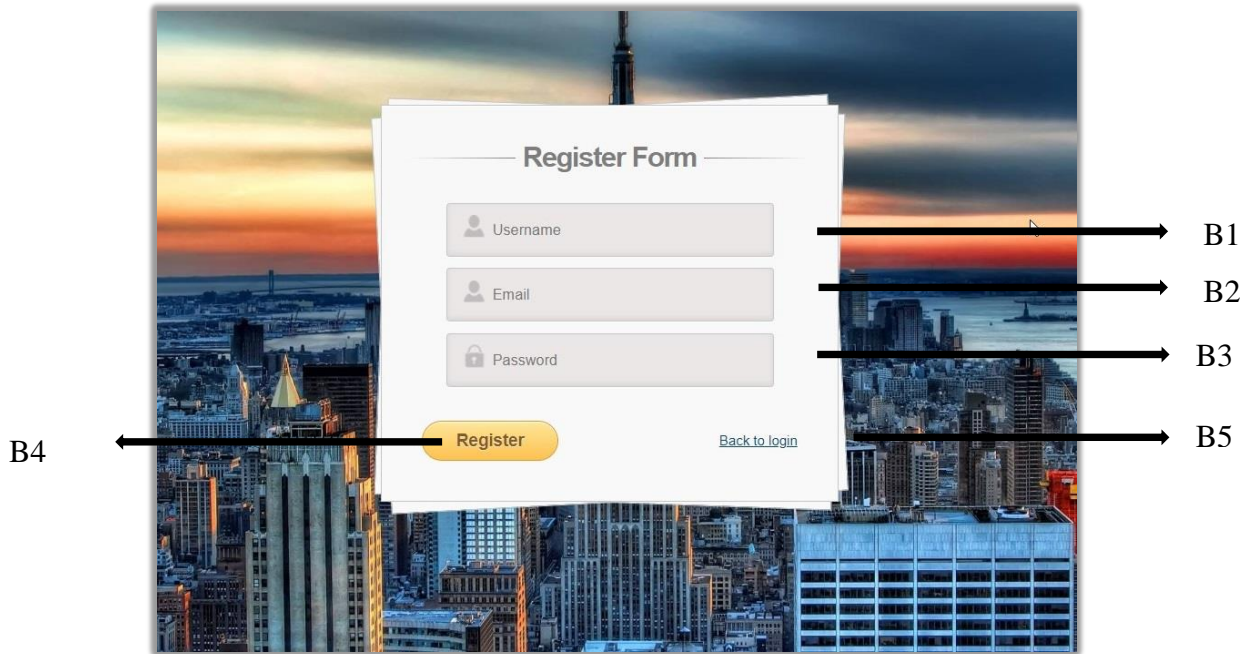


Figure 2 Register Page interface

Table 2 Register page interface description

Function	Description
B1	Input for username
B2	Input for email
B3	Input for password
B4	Submit information
B5	Navigate user to login page

### 3.1.3 OTP Page

OTP page is where user will submit the password given to the user via email. Figure 3 shows the OTP page in One-Time Password Authentication website. Table 3 shows the description of the OTP page.

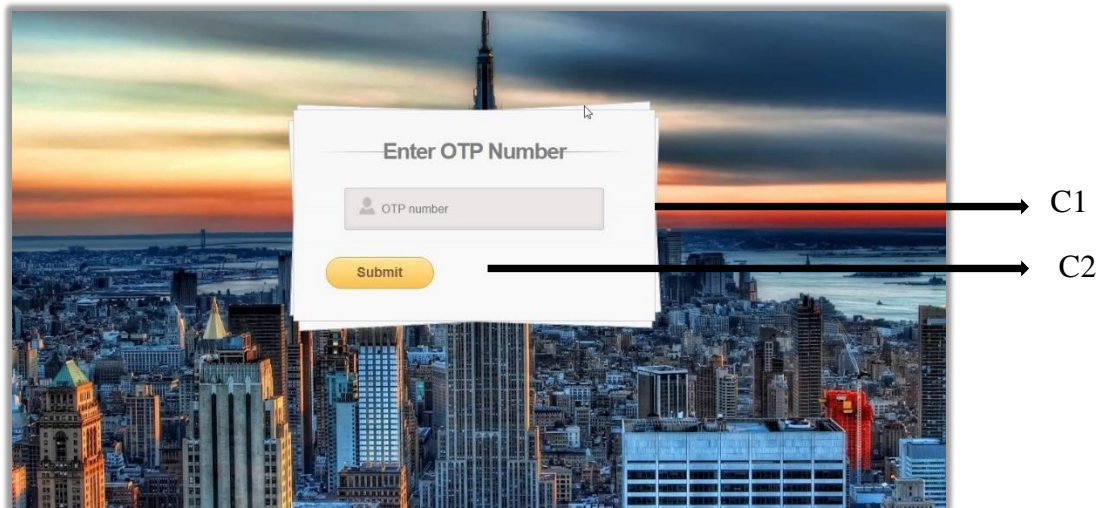


Figure 3 OTP Page interface

Table 3 OTP page interface description

Function	Description
C1	Input for OTP number
C2	Submit information