Applied Information Technology And Computer Science Vol. 4 No. 1 (2023) 139-153
© UniversitiTun Hussein Onn Malaysia Publisher's Office

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

A Secure Online Leave Application System Using Dual Authentication and XSS Attack Prevention for M Megalai & Co

Viknesvarma Sathananthan [1], Nor Bakiah Abd Warif[1]*

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, 86400, Parit Raja, Batu Pahat, Johor MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/aitcs.2023.04.01.009
Received 00 Month 2020; Accepted 27 May 2023; Available online 30 June 2023

Abstract: A leave application system is a process by which employees apply for leaves, managers approve them and leave balances are tracked. M. Megalai & Co is a professional body which had used the traditional way to apply and grant leaves. This raises some problems such as difficulty in accessing records, managing records, editing records and even problems like losing applications. In order to counter this problem a secure online leave application system using dual authentication and cross-site scripting attack prevention is proposed. This system will provide a manageable and secure environment for the personnel of M Megalai & Co. Object oriented methodology was handled during the project. UML diagrams were used in order to develop the system. Programming languages such as HTML and CSS were used for the front-end of the project and PHP and MySQL was used for the back-end of the project.

Keywords: Leave Application System, Dual Authentication, Cross-site scripting

1. Introduction

A leave management system is a process by which employees apply for leaves, managers approve them and leave balances are tracked. In this modern era, almost all businesses and agencies practice an employee employer system. A leave management system provides an easy way for human resources or management to administer leave, granting the ability to setup a standard leave scheme or customizes it per employee. This system is done for M Megalai & Co which is a law firm under the Bar Council of Malaysia.

The primary problem is, this legal firm had used the traditional way to apply and grant leaves which is by physical interaction. An additional problem is that the stated firm does not have a system that is secure. Most existing leave application system have one flaw which is lack of security. The main vulnerabilities are from Cross-site Scripting (XSS) attack, brute force and so on. This company has been using manual leave application system all this while and they have encountered multiples issues.

**Corresponding author: norbakiah@uthm.edu.my*

One would be file loss or file damage, as the company being a law firm there are already multiple files and documents involved. Other than that, it is hard to access and manage in this manual system. A futher problem would be when an application is wrongly filed or the information in the application is incorrect it is harder to make changes. There are three main objectives which are to design a secure leave application system for M Megalai & Co using dual authentication and XSS attack prevention, to develop a secure leave management system with visual studio code using HTML, CSS and PHP programming language and to test the security of the developed system with cross site scripting attack and also authentication attack.

The scope of this project is that involves building a Secure Leave Application system for employee to apply leave when they need. The leave application system is developed in accordance with the objectives. The four modules for this project are register, login, employee, and employer.

This project has Cross-site scripting prevention which is very vital at this modern tech era. The key security authentication method used is two factor authentication. The significance of using two factor authentication for this system is 2FA provides an additional layer of security used to ensure only authenticated users gain access to the client management system.

2. Related Work

This section discusses the literature review of the proposed system. Section 2.1 gives an overview on what a leave application system is.

2.1 Leave Application System

An employee leave management system is a platform that allows an organization's or institution's employees and administrators to quickly apply for, appropriately allocate, track, and issue leave. Employees use leave management system to request time off from work, and supervisors use leave management to approve or deny leave depending on company policy. Leave management is one of the most basic yet critical HR tasks that takes up a large percentage of the HR team's time [1].

2.2 Dual authentication

Dual authentication also known as 2 factor authentication is implemented to better protect both a user's credentials and the resources the user can access. Two-factor authentication provides a higher level of security than authentication methods that depend on single-factor authentication. This procedure is simple, but of limited security. MFA (Multi-Factor Authentication) is a type of two-factor authentication. In terms of technology, it's used whenever two authentication factors are necessary to obtain access to a system or service. Using two factors from the same category, on the other hand, does not form 2-factor authentication.

An attacker can usually breach password-based security systems and steal company data given enough time and resources. Because of their low cost, ease of deployment, and familiarity, passwords have remained the most popular form of SFA (Single Factor Authentication) [1]. This means that 2FA might not be the epitome of security, but it makes a system secure.

2.2.1 Email verification

One time password (OTP) confirms that the user has ownership of the device registered with the authentication system, which is commonly a mobile smartphone. OTP is also compromised if an attacker gains access to the device.

2.3 Cross-site scripting attack

Cross-site Scripting refers a group of people who want to intrude add malicious script code to a dynamic web page. The browser will automatically download malicious code embedded in the Web pages, when a user visits certain URL. When the harmful script code is processed, a person who wants to attack can get over the document Object model (DOM) security constraints and obtain the information about the Cookies.[2]. The major element of cross-site scripting vulnerabilities is the web server's lack of information supplied by the user to check the authenticity or verification is insufficient, and the detail given by the user is returned to the client. [2].

2.3.1 Input decoding

The input content must be specific and needs to be judged. If the input content is encoded, the encoded information needs to be decoded to proceed with the following One-step filtering operation. Different encoding will lead to different [3].

2.3.2 Input Filtration

The input filtering module is the main module to defend against XSS attacks. A user should supply a value which is expected to be numeric, validating that the value contains an integer. Filtering input include, validating that input contains only an expected set of characters.

2.3.3 Output Management

Encoding should be applied directly before user-controllable data is written to a page, because the context you're writing into determines what kind of encoding you need to use. For example, values inside a JavaScript string require a different type of escaping to those in an HTML context [4].

2.4 Structured Query Language Injection (SQLi)

SQL Injection Attack is used to get unauthorized access to web application. SQL Injection is the most prevalent web-based attack. There are many distinct varieties of SQLIAs, each with its own strategy to website attacks. The complicated formation is caused by a mix of SQL Injection and XSS attacks, which result in database information being retrieved [5]. Bypass of user validation can be of disabling the client-side validation such as disabling java script in web browser [6].

2.4.1 SQLi Prevention

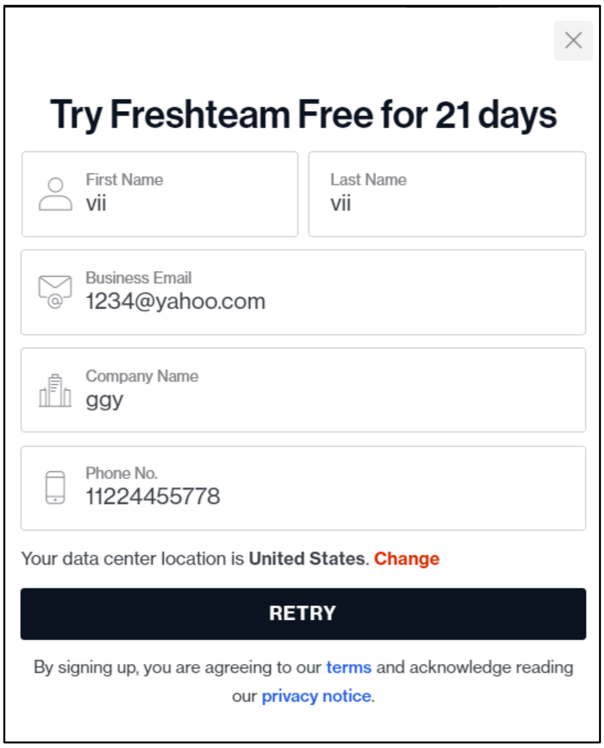
To prevent from modern SQL Injection Attack, it is always advised to use the prepared statements [7]. Users that are willing to attack data may usually find a different way to obtain the same outcome. Instead, code should be used to verify the safe input. This type of validation must be performed on a trustworthy server rather than on a client. Only data must enter the dB or special script in the initial try of validation with this type of validation. [8].

2.5 Comparison of Existing System with Proposed System

There are two existing systems that were compared to the proposed system. The two existing systems are Freshteam Leave Management Software and PlanMyLeave Leave Management System. Freshteam is a web application for leave management. Employees can raise time off requests or report personal time off directly through Freshteam and Managers can foresee upcoming leave to plan their work weeks or spot any absenteeism trends that need attention. PlanMyLeave is a leave management system where employees can apply for leave and managers can approve leave. It is a simple leave management system where employees can upload documents while applying for leave. This system is very convenient for HR Managers.

2.5.1 Freshteam Leave Management Software

Freshteam is a web application for leave management. Employees can raise time off requests or report personal time off directly through Freshteam and Managers can foresee upcoming leave to plan their work weeks or spot any absenteeism trends that need attention. This leave management system is constructed in a way where Employees can apply for their leaves.



Try Freshteam Free for 21 days

First Name: vii

Last Name: vii

Business Email: 1234@yahoo.com

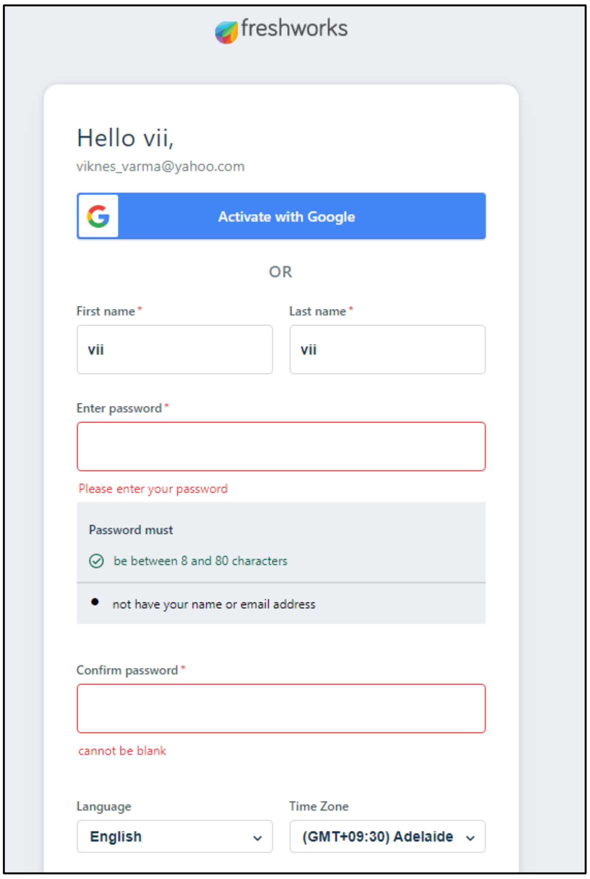
Company Name: ggy

Phone No.: 11224455778

Your data center location is **United States**. Change

By signing up, you are agreeing to our terms and acknowledge reading our privacy notice.

Figure 1: Registration interface for Freshteam Leave Management Software



Hello vii,
viknes_varma@yahoo.com

First name *: vii

Last name *: vii

Enter password *:

Please enter your password

Password must
✓ be between 8 and 80 characters
● not have your name or email address

Confirm password *:

cannot be blank

Language: English

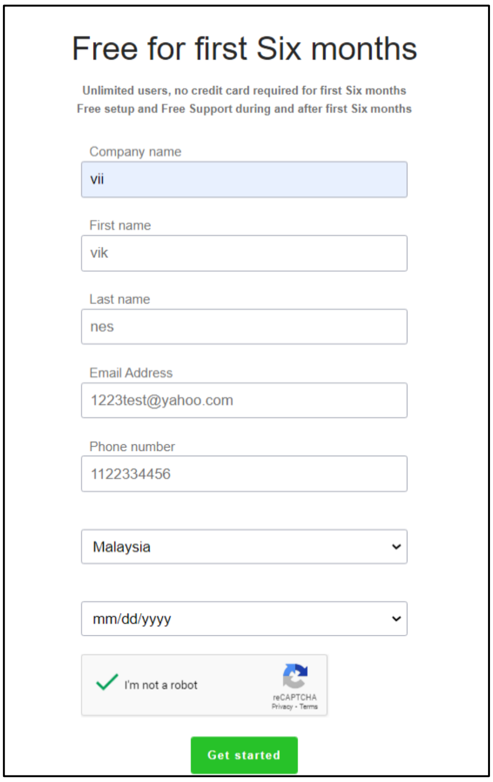
Time Zone: (GMT+09:30) Adelaide

Figure 2: Password interface for Freshteam Leave Management Software

Figure 1 shows the sign-up page for freshteam. This system is secure in terms of email verification, as users cannot input an invalid email address that does not exist to verify themselves. The verification link sends the user the page as shown in Figure 2, where users are required to enter password. This page has a significant flaw where the password validation and verification does not comply with OWASP Guidelines.

2.5.2 PlanMyLeave Leave Management System

PlanMyLeave is a leave management system where employees can apply for leave and managers can approve leave. It is a simple leave management system where employees can upload documents while applying for leave. This system is very convenient for HR Managers. It's known for its easy and flexible interface for both employee and managers.



Free for first Six months

Unlimited users, no credit card required for first Six months
Free setup and Free Support during and after first Six months

Company name: vii

First name: vik

Last name: nes

Email Address: 1223test@yahoo.com

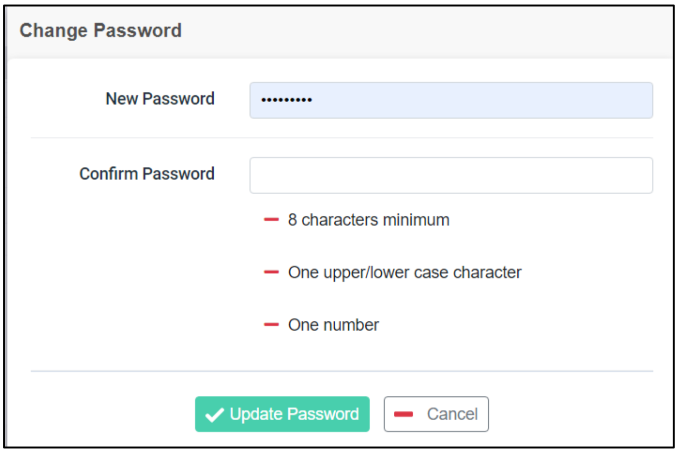
Phone number: 1122334456

Malaysia

mm/dd/yyyy

✓ I'm not a robot

Figure 3: Registration interface for PlanMyLeave Leave Management System



Change Password

New Password: •••••••••

Confirm Password:

— 8 characters minimum

— One upper/lower case character

— One number

Figure 4: Change Password

Figure 3 shows the registration page for PlanMyLeave web application. Users can input an invalid email address that does not exist to verify themselves. Users can enter to the system right after the sign up with no verification and no password. Users can enter the password credentials and soon and they enter the system. As shown in Figure 4, the change password section is following the OWASP guideline.

2.5.3 Comparison of Existing System with Proposed System

The studied existing systems are very important for the analysis by comparing with the proposed system. The output of the comparison is important to determine the differences between the systems. Some concept of the proposed system is like the existing mentioned system. Table 1 shows the difference between the proposed system and the existing systems.

Table 1: Comparison of Existing System with Proposed System

Systems Features	Freshteams Leave Management System	Planmyleave Leave Management System	Proposed System
Password Complexity	x	✓	✓
XSS attack prevention	✓	x	✓
Dual Authentication	✓	x	✓
Verification during password change	✓	x	✓
Email verification	✓	x	✓
Input validation	x	✓	✓

3. Methodology/Framework

3.1 Object-Oriented System Development

There are three main phases in this methodology which are object-oriented analysis, object-oriented design, object-oriented implementation, and testing [9] as depicted in Figure 5.

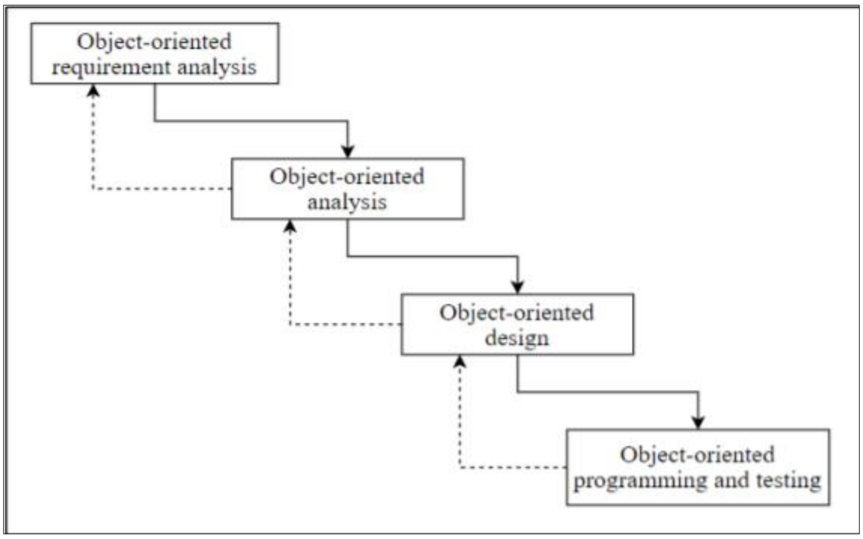

Object-oriented requirement analysis
Object-oriented analysis
Object-oriented design
Object-oriented programming and testing

Figure 5: Object-Oriented System Development Model [1]

3.2 Requirement Analysis Phase

In requirement analysis phase, the vital and necessary requirements are gathered in order to achieve the objective of the proposed system. First up, the objectives and the scope of the systems are identified. The objectives of the system are to design, to develop and to test the proposed secure system. The project's scope includes three main users which are the admin, employee and employer of the company. Interview is also essential in order to obtain vital requirements for the system (Appendix A).

3.3 Analysis Phase

The required software and hardware requirements for developing the proposed system is listed in Table 2.

Table 2: Hardware and Software Requirement

Software Requirement	- Microsoft Windows 10 operating system - Visual Studio Code - Linux
Hardware Requirement	- Laptop
Programming Language	- Php - HTML - CSS
Server and DBMS	- XAMPP - phpMYAdmin

3.4 Design Phase

The entire architecture for the proposed system is designed during the object-oriented design phase. The complete architecture design is necessary in order to develop a complete design model. Unified Modeling Language (UML) diagram is a language to model application structures, it would basically make the relationship between the entities more obvious. There are multiple UML diagrams, but only class diagram, activity diagram and use case diagram are used in this system.

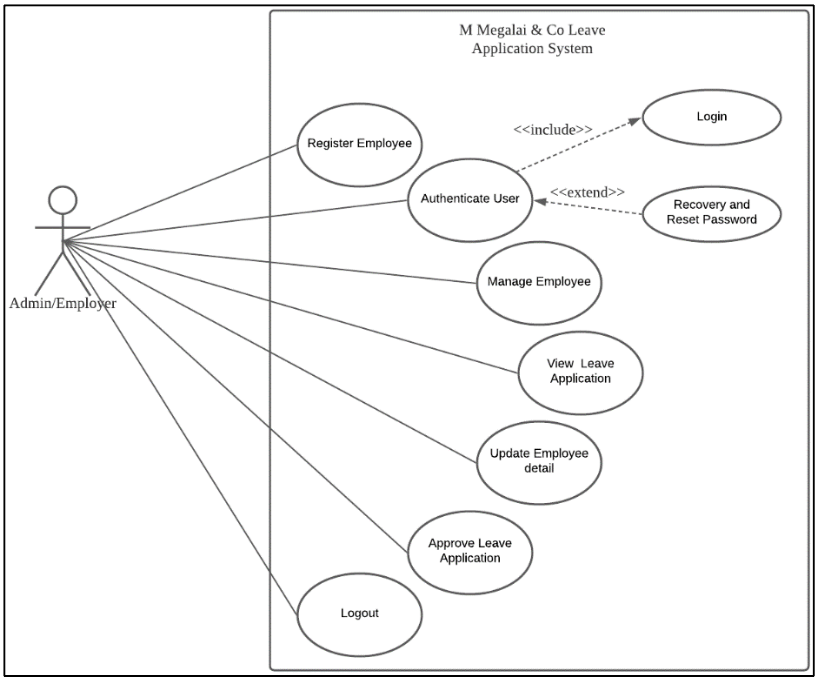


Figure 6: Admin use case diagram

Figure 6 shows the use-case diagram for employees. There are six use cases for employee. The use case is authenticating user, view calendar and set date, attach evidence/leave letter, update personal detail, apply leave application and logout. The authenticate user use case includes login functionality and extends reset password functionality.

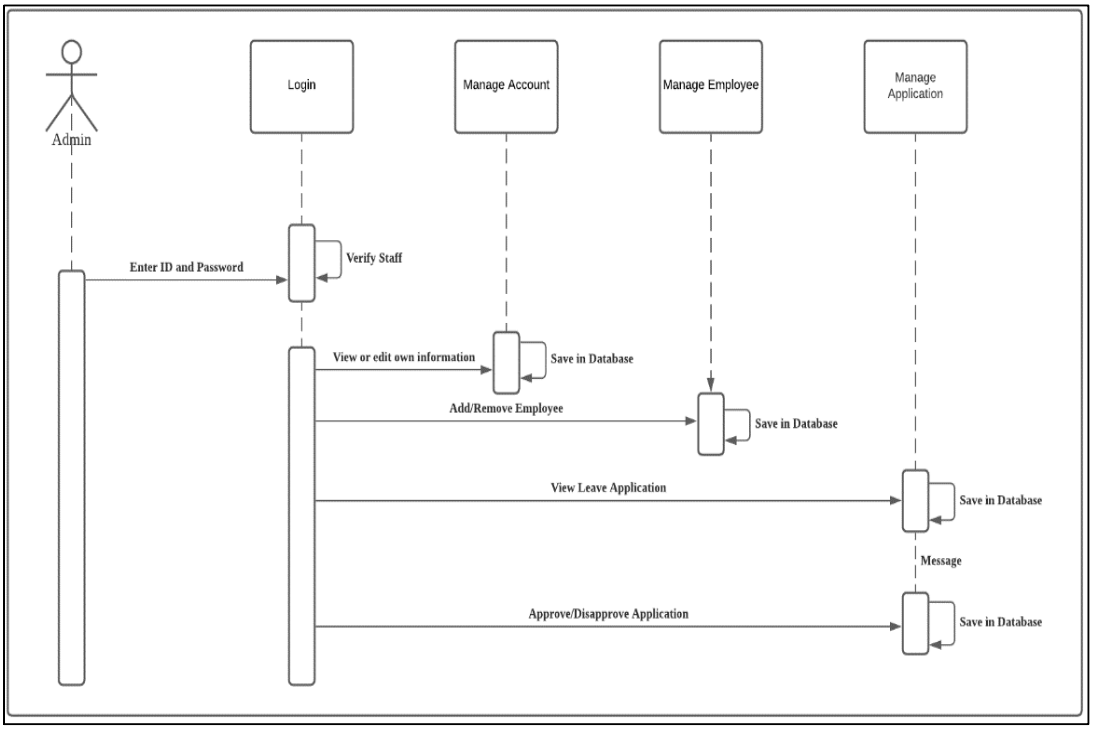


Figure 7: Sequence diagram for admin

Figure 7 shows the complete sequence diagram for admin. First the employee enters ID and password to login into the system. When the user enters proper credentials the system sends a push notification through email to verify the user. When the employee is verified, employee can view or edit own personal information. Employee can set date for leave; they can then attach file as proof of leave. Finally, employee can confirm leave application. All the mentioned progress would be sent to database.

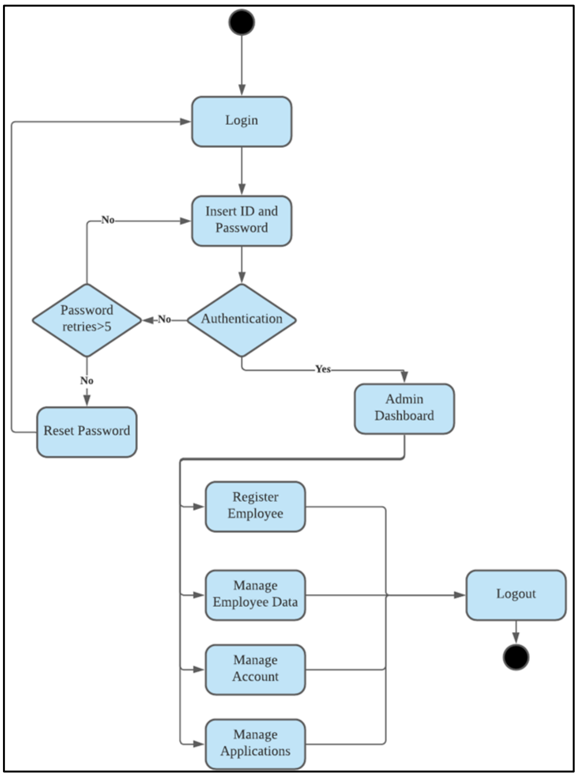


Figure 8: Activity diagram for admin

Figure 8 shows the activity diagram for admin. The activity starts by admin logging into the system. When the admin is authenticated, admin dashboard is presented. From there admin can create, delete, update, and read employee. Admin can also manage employee data and employee account. Admin can also manage employees' leave applications. The activity ends when the admin logs out of the system.

3.5 Testing Phase

Security testing is primary priority as this system is a secure system. It ensures that illegal access to any of the application's resources will not cause the site to crash. It ensures that any sensitive information from leaking. First up, SQL Injection will be done to make sure that it doesn't bypass ambient security checks. Cross-Site Scripting attack is also done to make sure attackers can't inject client-side script into Web pages viewed by other users.

Table 3 shows the shows the system functionality test result for the login part. Table 4 shows the system functionality test result for the registration part. Table 5 shows the system functionality test result for the leave application part.

Table 3: System functionality test result (Login)

Module	Description	Expected Result	Actual Result
Login	Correct Username/Email, password and correct OTP.	Login Successfully	Pass
	Empty input for required field.	Error message displayed. Login Failed	Pass
	Login with correct email and wrongpassword.	Error message displayed. Login Failed	Pass
	Login with wrong email and wrong password	Error message displayed. Login Failed	Pass

Table 4: System functionality test result (Registration)

Module	Description	Expected Result	Actual Result
Registration	Input valid format of username, phone number, password and email.	Register succeeds.	Pass
	Empty input for required field.	Error message displayed. Register failed.	Pass
	Unmatched password combination.	Error message displayed. Register failed.	Pass
	Invalid format of password or phone number.	Error message displayed. Register failed.	Pass

Table 5: System functionality test result (Leave Application)

Module	Description	Expected Result	Actual Result
Leave Application	To date is entered before From Date	Error message displayed. Application Failed.	Pass
	Empty input for required field.	Error message displayed. Application Failed.	Pass
	Leave type not selected	Error message displayed. Application Failed.	Pass
	Valid format leave Application	Leave Application submitted	Pass

Table 6 shows the test case result of multiple scenarios and Table 7 shows the injection attack plan result.

Table 6: Test Case Results

Modules	Test Case	Expected Result	Actual Result
Login & Register Module	User does not fill in allthe required fields	The system prompts the user tofill in all of required fields	Pass
	User enters the information with invalid or incorrect format	The system prompts the user tofill in and provides the sample data with correct format	Pass
	User enters the incorrect username/email or password	The system will check the combinations entered andprompts incorrect id or password	Pass
	User enters the malicious code to the input fields for SQL injection	The system validates and prompts the invalid format ofinputs	Pass
	User enters the correct id and password	The system will log into theuser to its homepage	Pass
	User enters the malicious codeto the input fields for Cross site scripting	The system validates and prompts the invalid format ofinputs	Pass

Table 7: Injection Attack Plan Results

Scenarios	Malicious Input	Vulnerable System Result	Secure System Result
Login	‘ or 1=1 –“ or ‘=”	The first user account had been logged without properinformation	Unable to login to the system.
	' or 1=1; drop table admin; --	Table admin has been dropped fromdatabase	Table admin did not dropfrom database
	"/><script>alert('XSS!');</script>	Localhost displays “alert XSS” message	Unable to login and system shows invalid input

4. Results and Discussion

This section will highlight the main security features of he proposed leave application system. All the source code are developed to ensure the functionality of the system are according to the requirement. The security features will include confidentiality, integrity and availability of the quiz system.

Figure 9 shows the login interface of the leave application system where only users with correct credentials can login. Figure 10 shows the interface where user enter their one-time passcode. Only the correct credentials are allowed in this interface.

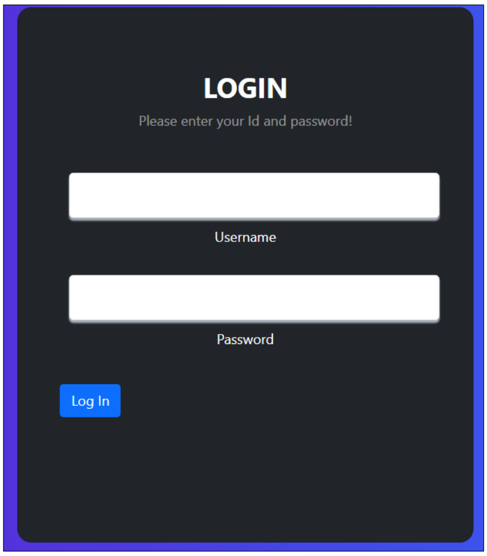


Figure 9: Interface of Login

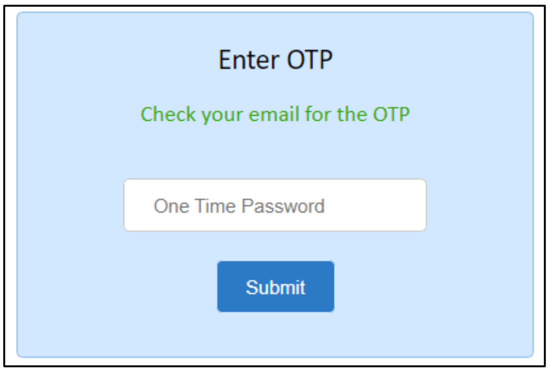
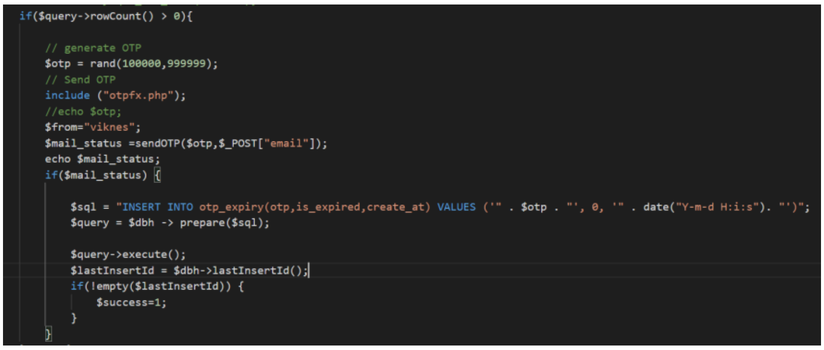


Figure 10: Interface for entering OTP

Figure 11 shows the code for sending OTP to the user requesting it where the code retrieves the email from the database. Figure 12 shows the code for cross-cite scripting. The code structure shows that it is using htmlentities to echo the result requested.



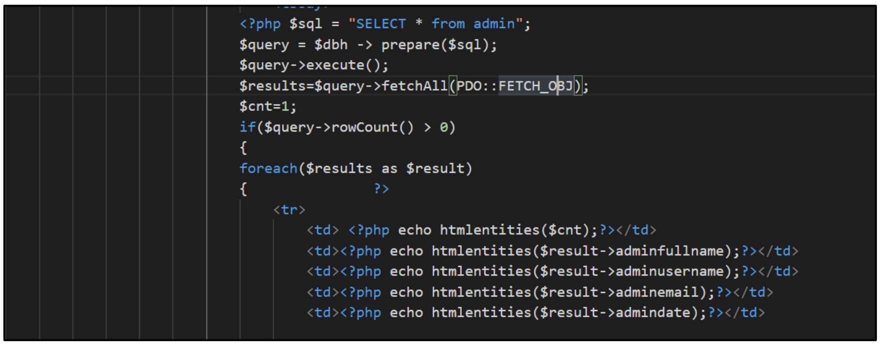
```
if($query->rowCount() > 0){

    // generate OTP
    $otp = rand(100000,999999);
    // Send OTP
    include ("otpfx.php");
    //echo $otp;
    $from="viknes";
    $mail_status =sendOTP($otp,$_POST["email"]);
    echo $mail_status;
    if($mail_status) {

        $sql = "INSERT INTO otp_expiry(otp,is_expired,create_at) VALUES ('" . $otp . "', 0, '" . date("Y-m-d H:i:s"). "')";
        $query = $dbh -> prepare($sql);

        $query->execute();
        $lastInsertId = $dbh->lastInsertId();
        if(!empty($lastInsertId)) {
            $success=1;
        }
    }
```

Figure 11: Code for Sending OTP



```
<?php $sql = "SELECT * from admin";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
{               ?>
    <tr>
        <td> <?php echo htmlentities($cnt);?></td>
        <td><?php echo htmlentities($result->adminfullname);?></td>
        <td><?php echo htmlentities($result->adminusername);?></td>
        <td><?php echo htmlentities($result->adminemail);?></td>
        <td><?php echo htmlentities($result->admindate);?></td>
```

Figure 12: Code for XSS prevention

The result of the user acceptance of the system is collected through an online survey questionnaire using Google Form that was sent to the employees and the employers of the company. The company has a total of four workers and all of them responded to the survey. First are the results about system functionality, second is about the design of the system and last is about the security usability. Figure 13 presents the results of system functionality. All the respondents were able to register and login to the system, manage their personal account and able to apply leave from the system. The result of system usability is shown in Figure 14. The security features implemented are strong password and OTP and they worked for all the respondents.

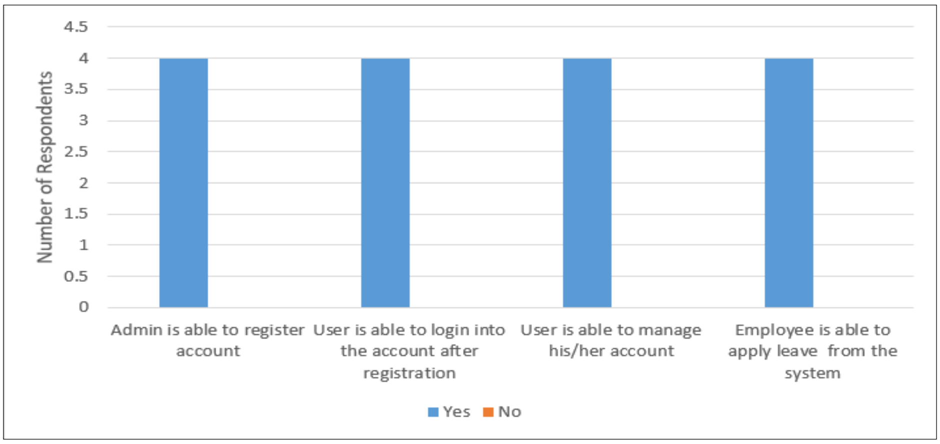


Figure 13: System Functionality Testing Result

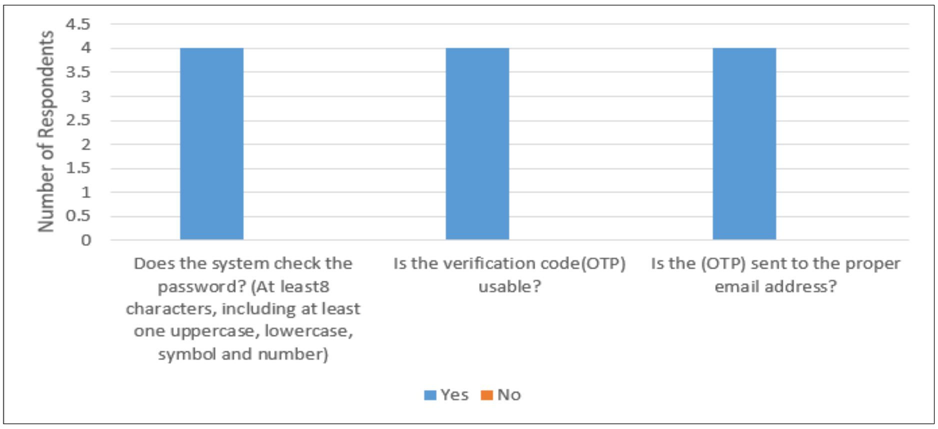


Figure 14: System Security Usability Testing Results

5. Conclusion

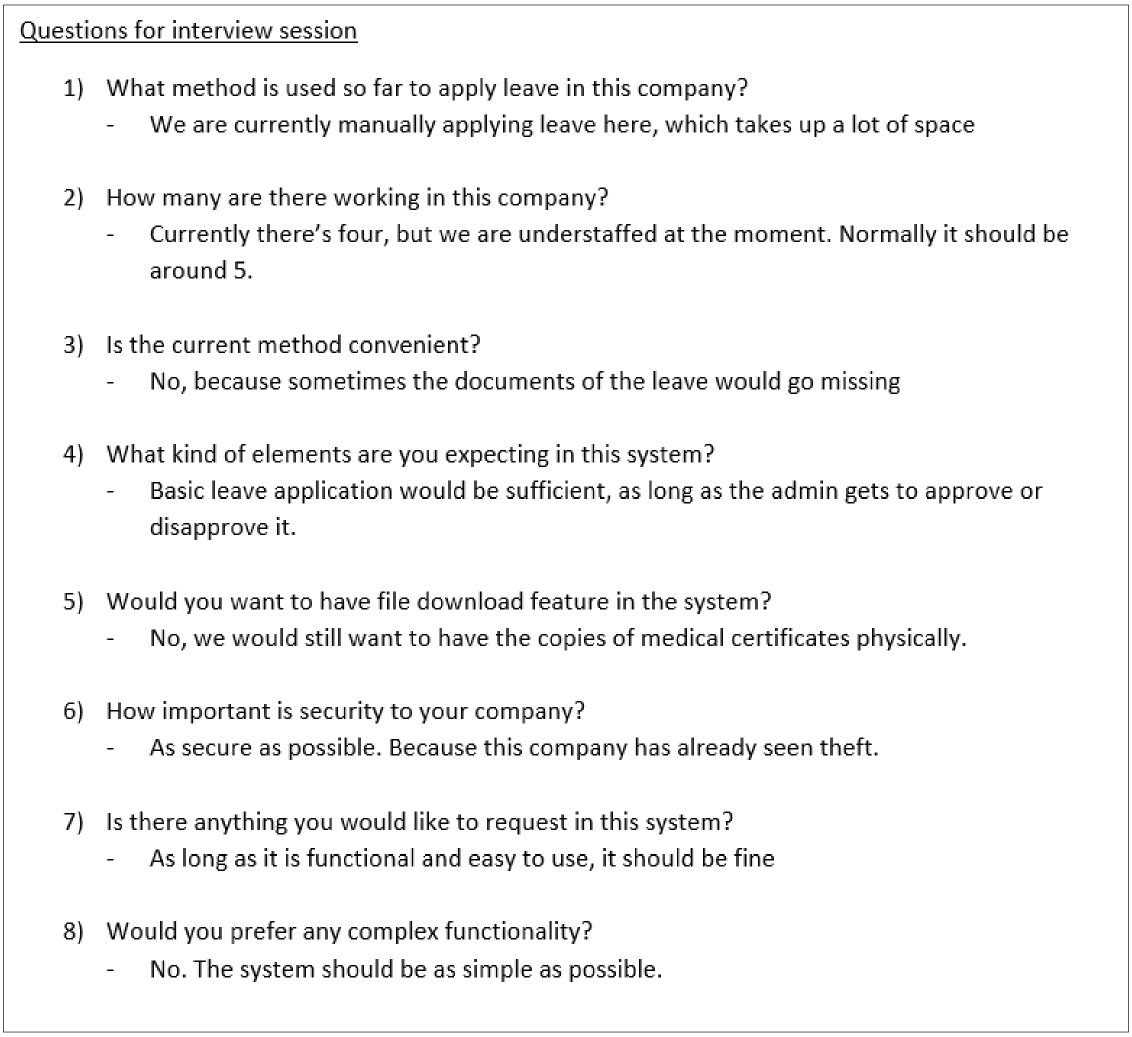
The online leave application system with dual authentication and cross site scripting prevention assists in the lifting of a constraint of manual leave application system. It helps to avoid handling situations like file loss. This system also minimizes the potential vulnerabilities that injection attacks may exploit by using parameterized queries and input validation. At the end of the system development, the overall objectives of the system are successfully achieved.

Since the web-based system cannot be used in mobile, a framework has to be implemented for the mobile application. Employees should be able to view a fully detailed calendar where they can view public holidays and select the days they want to apply leave on. Another functionality that can be added is that, the admin should get a notification 88 whenever an employee applies for a leave.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

APPENDIX A



Questions for interview session

1) What method is used so far to apply leave in this company?
   - We are currently manually applying leave here, which takes up a lot of space

2) How many are there working in this company?
   - Currently there's four, but we are understaffed at the moment. Normally it should be around 5.

3) Is the current method convenient?
   - No, because sometimes the documents of the leave would go missing

4) What kind of elements are you expecting in this system?
   - Basic leave application would be sufficient, as long as the admin gets to approve or disapprove it.

5) Would you want to have file download feature in the system?
   - No, we would still want to have the copies of medical certificates physically.

6) How important is security to your company?
   - As secure as possible. Because this company has already seen theft.

7) Is there anything you would like to request in this system?
   - As long as it is functional and easy to use, it should be fine

8) Would you prefer any complex functionality?
   - No. The system should be as simple as possible.

References

[1] Reese, K., Smith, T., Dutson, J., Armknecht, J., Cameron, J. and Seamons, K., 2021. A Usability Study of Five {Two-Factor} Authentication Methods. [online] Usenix.org. Available at: https://www.usenix.org/conference/soups2019/presentation/reese [Accessed 31 December 2021]

[2] Wang, X. and Zhang, W., 2016. Cross-site scripting attacks procedure and Prevention Strategies. MATEC Web of Conferences, 61, p.03001.

[3] Yadong Wang, 2021. Research on Cross-Site Scripting Attack and Prevention Methods. Computer Science and Application, 11(01), pp.195-206.

[4] Academy, W. and scripting, C., 2022. How to prevent XSS | Web Security Academy. [online] Portswigger.net. Available at: <https://portswigger.net/web-security/cross-site-scripting/preventing#encode-data-on-output> [Accessed 1 January 2022].

[5] Singh, J., 2017. Analysis of SQL Injection Detection Techniques. Theoretical and Applied Informatics, 28(1&2), pp.37-55.

[6] Andodariya, M., 2018. SQL Injection Attack Detection and Prevention Techniques to Secure Web-Site. International Journal of Trend in Scientific Research and Development, Volume-2(Issue-4), pp.624-628.

[7] Arxiv.org. 2022. [online] Available at: <https://arxiv.org/pdf/1303.3047.pdf> [Accessed 1 January 2022].

[8] S. Farde and S. Chaudhari, "SQL Injection (SQLI)," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 5, no. 6, pp. 2278–1323, 2016.

[9] A. R. Hevner, "Object-oriented system development methods," *Advances in Computers*, pp. 135–198, 1992.