Applied Information Technology And Computer Science Vol. 4 No. 3 (2023) 159-174

AITCS

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

Rand-Quiz: Web-Based Quiz System with OTP Using KJY Password Generator

Koh Jia Yee[1], Kamaruddin Malik Mohamad[1]*, Sofia Najwa Ramli[2]

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

DOI: https://doi.org/10.30880/aitcs.2023.04.02.010
Received 02 October 2023; Accepted 06 November 2023; Available online 30 November 2023

Abstract: One-Time Password (OTP) is a security mechanism used to authenticate users when login into web applications or other sensitive operations. It is a unique code that is generated for each use and is only valid for a single login attempt. However, the common 6-digit code OTP is weak and vulnerable to attack. The Quiz system also called as Rand-Quiz, is secured with a unique OTP by using KJY Password Generator. KJY Password Generator generates random values for OTP using combination of user ID, username, and user password. The Quiz system is developed using prototype methodology, PHP, HTML, CSS and JavaScript programming languages. It is featured with mail-based OTP, which will be sent to the user of the system for each time user login. This unique randomized mail-based OTP helps to improve security for the Rand-Quiz, web-based Quiz system.

Keywords: Quiz system, Randomized, One Time Password

1. Introduction

A quiz system is used to evaluate a person's knowledge and performance. This is typically used in the education field to know the understanding of students in each course. Previously, most of the classes and examinations were held physically. However, the COVID-19 pandemic caused a significant impact on our society including the educational system. It leads to the online learning approach becoming the new norm in place of conventional methods [1]. This is due to schools being closed throughout the world. Many students are out of the classroom. Physical class is replaced with the distinctive rise of e-learning, whereby teaching is undertaken remotely and on digital platforms. Research suggests that online learning has been shown to improve retention of material, and take less time, meaning this change might be remained even after the pandemic [2].

In this scenario, mobile devices are primarily used for student-teacher communication and access to school information. Additionally, the quiz and final exams are conducted online. Multi factor authentication features will be implemented into the proposed quiz system as security precautions, and it is a requirement for every user to login each time. Additionally, the proposed quiz system will provide

*Corresponding author: malik@uthm.edu.my

hashing of passwords and OTP when storing into database. As a result, a secure quiz system will offer a secure learning environment for teachers, school administrators and students.

Nowadays, online quizzes can be taken through various web pages, but the system of online quizzes is not perfect enough. Multi factor authentication is not used by the majority of quiz systems. They only required a username and password for login. Multi factor authentication can protect the system from an adversary attack. Besides, the online quiz system usually provides role-based access that is crucial for identifying the user's role within the system. Users will be able to access pages that are only authenticated for them.

If these issues are not resolved, the quiz system will be less secure. The objectives of this project are to design OTP using KJY password generator, to develop a randomized web-based quiz system with OTP using KJY password generator and to test the randomized web-based quiz system. This quiz system with multi factor authentication on the web application will have three main users which are administrators, teachers, and students. Each user can only access certain functions based on the role. The development methodology chosen is the prototype methodology.

Users log in to Rand-Quiz by using a user ID and password, and Email OTP verification. First-time login users are required to change the password because the default password for new users is the same as the user ID. The password will be encrypted using bcrypt algorithm in the database. The login modules would redirect users to their authorized pages based on the roles. In the admin module, admin can register admins, students, lecturers, subjects, and workloads. Admin is allowed to add, update and delete that information. Admin cannot be deleted if there is only one admin in Rand-Quiz. Admin can view all the logs. In the teacher module, teacher can view their workload. Teacher can create quizzes. The teacher is allowed to add, update and delete the quiz. Teacher can view the results of students. In the student module, students can register subjects and view the registered subjects. Students can take the quiz and view the results. At the end of the project, a randomized web-based quiz system with OTP using KJY password generator will be developed. Rand-Quiz will also have a user-friendly interface. Multi-factor authentication can help to make Rand-Quiz more secure as the users need to pass the authentication to make modification within the system.

This report has four sections, including an introduction, related work, methodology, and conclusion. Section 1 determines the project background, problem statement, objective, scope and expected result. The existing quiz system and multi-factor authentication will be analyzed in section 2 for the literature review. Next, section 3 will explain about methodology, and analysis and design. Last but not least, the conclusion will be outlined in section 4.

2. Related Work

2.1 Comparison of Existing Systems with Rand-Quiz

Table 1 compares Quizizz, Google Form, Mercer | Mettl, and proposed system (Rand-Quiz) to determine the features that will or have been implemented in the systems.

Table 1: Comparison between Quizizz, Google Form, Mercer | Mettl, and Rand-Quiz [3][4][5]

	Quizizz	Google Form	Mercer | Mettl	Rand-Quiz
Platform Used	Web-based and mobile-based	Web-based	Web-based	Web-based
User friendly	Yes	Yes	Yes	Yes
Login	Yes	Yes	Yes	Yes
Role-based Access Control	Yes	Yes	Yes	Yes
Authentication Method	Password	Password	MFA	MFA
One-Time Password	No	No	Yes	Yes

A platform that is used for Mercer | Mettl, Google Form, and Rand-Quiz are web-based while Quizizz is available both web-based and mobile-based. Role-based access controls were added for authorization mechanisms in all the systems. Besides, the authentication method used by Quizizz and Google Forms is email and passwords, while Mercer | Mettl and Rand-Quiz use multi-factor authentication (MFA). Mercer | Mettl and Rand-Quiz have implemented the One-Time Password login, while Quizizz and Google Form not implemented.

2.2 Multi-Factor Authentication (MFA)

Authentication can be defined as where a user identifies himself by sending x to the system; the system authenticates his identity by computing F(x) and checking that it equals the stored value y. While in nowadays verification the user from the information technology perspective cannot be done by a simple password as a unique factor. This is because unitary identity authentication is insecure and it will be easily cracked by some means [6]. Therefore, multi-factor authentication will be considered. The conceptual authentication consists of 3 types of factor groups such as ownership factor, knowledge factor and biometric factor.

Mercer Mettl requires multi-factor authentication of user for logging in by using email verification, ID card verification, and mobile authentication through OTP [7]. By having multi-factor authentication technology, it can eliminate the risk of student impersonation.

The multi-factor authentication that will be implemented in Rand-Quiz is OTP verification. The generated OTP must be difficult to guess, collect, or detect by hackers in order to secure the system. Consequently, it is crucial to create a secure OTP generating algorithm. The OTP algorithm uses a number of variables to produce a password that is challenging to guess. Each OTP is distinct and has a short validity period. The timestamp on the device must match the one from the server [8].

2.2.1 One-Time Password (OTP)

One-Time Password is a security mechanism used to authenticate users during logging into web applications or other sensitive operations. OTPs are unique codes that are generated for each use and are only valid for a single login attempt [9]. This makes it difficult for attackers to gain unauthorized access to user accounts even if they have obtained the user's username and password [10].

OTP is a two-factor authentication scheme that is considered a natural enhancement over conventional username and password schemes. OTPs are used to provide an additional layer of security to online accounts. Unlike static passwords, OTPs are dynamic and can only be used once [14]. One-Time Password (OTP) is a breakthrough to two-factor authentication technique. OTP improved the security, protection, and confidence level of the user as it uses a randomized generation of OTP codes sent through secured email account that is free from brute force, dictionary attack, insider attack, and key-logger attacks [15].

2.2.1.1 Justification for OTP Length and Time

The length of the One-Time Passwords (OTP) is 64-bits strings in length, displayed as 8 decimal digits. The possible brute force attacks succeed with probability of close to 10^{-8}. Therefore, this length is long enough to be secure and short enough to be manually entered by users when necessary [9]. However, the length of the OTP is an important factor in determining its security. A longer OTP is generally considered more secure as it increases the number of possible combinations, making it more difficult for an attacker to guess or brute-force the OTP.

The length of OTP is an important factor in determining its security. The longer the OTP, the more secure it is. This is because longer OTPs have a larger key space, making it more difficult for attackers to guess the correct OTP. However, it does suggest that users should be encouraged to create their own formula for composing passwords, which can include OTPs of varying lengths [13].

The length of the OTP is justified by the fact that it is generated based on a combination of several parameters like string of characters, numbers, date, time and weather data. The values generated change over time upon login registration as it extracts location weather data producing approximately 91 possible values or 4,095 possible combinations. This makes it difficult for attackers to guess or brute force the OTP code, thus improving the security of the authentication process [15].

OTP can provide complete protection of the login-time authentication mechanism against replay attacks. OTPs are used to authenticate users for a single session or transaction, and they are valid for a limited period of time [9]. OTPs are valid for a very short period of time, usually between 30 seconds to a few minutes, to ensure that the generated OTP is unique [11]. Based on [12], the validity period of the OTP password is 180 seconds, which means that the password is only valid for 3 minutes. After that, the password becomes invalid, and a new OTP password needs to be generated for the next login session or transaction.

2.2.1.2 Email OTP

Email OTP is a type of One-Time Password that is sent to the user's email address for authentication purposes. It is a security measure used to verify the identity of the user during login or transaction [9]. Email OTP are generally more secure than SMS OTPs because they are less vulnerable to interception and can be encrypted [13].

2.2.1.3 SMS OTP

SMS OTP verification code is sent to the user's mobile device through an SMS gateway. It is a security measure used to verify the identity of the user during login or transaction. The user needs to check the SMS and enter the OTP code into the OTP key insert menu to validate the code. The OTP code is only valid for one login attempt and provides an additional layer of security to the login process [10].

2.2.1.4 Time-based Sequence-oriented One-Time Password (TSOTP)

Time-based Sequence-oriented One-Time Password (TSOTP) is a new effective simple OTP method that generates a unique passcode for each use using time stamps and sequence numbers. The calculation uses both time stamps and sequence numbers. A two-factor authentication prototype for mobile phones using this method has been developed and has been used in practice for a year. TSOTP method generates a 6-digit passcode, which is a common length for OTPs. However, the length of the OTP may vary depending on the specific implementation and security requirements [9].

2.2.2 Type of Key Generation

Email OTP and SMS OTP are typically generated using cryptographic algorithms that produce a unique code that can only be used once. Email OTPs are sent to the user's email address, while SMS OTPs are sent to the user's mobile phone number. The specific method of generating OTPs may vary depending on the system being used [13].

The Email OTP and SMS OTP are generated based on a combination of several parameters like string of characters, numbers, date, time, and weather data. The values generated change over time upon login registration as it extracts location weather data producing approximately 91 possible values or 4,095 possible combinations. The OTP codes are sent through secured email account or SMS that is free from brute force, dictionary attack, insider attack, and key-logger attacks [15].

There are two approaches in generating OTP which are randomization parameters and extraction, and a new concept of mapping out the numbers in 4x4 matrix schedules. The existing OTP bingo grid schemes deals with combination of several seed attributes which include numbers and characters. Therefore, it is managed to handle the procedure with less operations with minimal number of elements [15].

2.2.2.1 Cryptographic Algorithms

Cryptographic algorithms are mathematical functions that are used to generate a unique code that can only be used once. These algorithms are designed to be secure and difficult to predict, making it difficult for attackers to guess or intercept the OTP. The specific method of generating OTPs may vary depending on the system being used, but it typically involves a combination of random number generation and cryptographic functions [13].

2.2.2.2 Randomization Parameters

Randomization parameters refer to the process of generating random values that are used to encrypt and decrypt the OTP image shares. The randomization parameters were used to ensure that the shares are secure and cannot be easily decrypted by unauthorized users [16].

2.2.2.3 Extraction

The extraction approach involves extracting location weather data to generate the initial seed for the OTP codes. The values generated change over time upon login registration as it extracts location weather data, producing approximately 91 possible values or 4,095 possible combinations. This approach improves the security, protection, and confidence level of the user as it uses a randomized generation of OTP codes sent through a secured email account that is free from brute force, dictionary attack, insider attack, and key-logger attacks [15].

2.2.2.4 Mapping Out the Numbers In 4x4 Matrix Schedules

Mapping out the numbers in 4x4 matrix schedules is a concept used to avoid producing duplicate values and to maximize the alphanumeric combination. After the first level of authentication, the Passcode together with alphanumeric combinations, a pair of code will be generated. The XY coordinate will be randomly chosen together with the assigned pair of codes [15].

2.3 Role-based Access Control

Access control is influenced by the responsibilities of particular people within an organization. This covers the description of tasks, accountability, and qualifications. Access control decisions are based on the roles that an individual is permitted to play inside an organization under a role-based access control (RBAC) policy. Access permissions cannot be freely transferred by users to other users [17].

Quizizz system uses role-based access control [3]. When people sign up an account it will ask what the role is and what is the purpose for using the website. For example, as a student, a teacher, or an administrator. Therefore, a student can participate in fun classroom activities. Teacher and administrator can use to instruct, engage, and access their students. In Quizizz, teacher and administrator got extra two function which is they can organize or create my school and classes for their students.

Mercer | Mettl enables users to grant any account sub-user role-based access. Additionally, the user may limit which users have access to particular account features. This feature enables users to create roles and assign each role a constrained set of permissions. Specific users can then be given access to these roles, and each user can only use the features that are associated with his role. For role creation, there are a number of permissions and sub-options available, including Assessment, Questions, Candidate, Results and analytics, Proctoring and authorization, and Other global level permissions [18].

Users of Google forms can also create quizzes and only allow certain users to response to it. The quiz creator can also view automatic summaries for all quiz responses, which include graphs marked with the correct answers, average, median, and range of scores, as well as frequently missed questions. The quiz's creator has control over what participants can see both during and after the test, as well as whether they can view incorrect answers, miss questions, and point totals [19].

3. Methodology

3.1 Prototype Methodology

Prototype methodology is a methodology that is used for developing Rand-Quiz. This methodology will undergo the process of define, sketch and design, create and develop, test and feedback, refine, and implement and maintain. In this methodology, the process test and rework will be repeated until the prototype design is accepted by the user. The methodology has a strong advantage which is receiving feedback can lead to a better solution and design. While this methodology may affect the development and become complex.

3.1.1 Requirements Phase

This stage involves defining, gathering, and understanding the needs of the users, goals, and objectives of Rand-Quiz. It includes gathering information from users, conducting user research, and defining the scope of the prototype. The Gantt chart is made according to the prototype methodology which implies requirements, quick design, build prototype, user evaluation, refining prototype, and implementation and maintenance. This is to make sure all the milestones, deliverables and outcomes can be completed on time.

System development is planned during the requirements phase after defining, gathering, and understanding the requirements of the users. The platform such as Google scholars and IEEE is used to conduct literature reviews. The results of the research are then used to identify the system's functional and non-functional requirements. The existing system and Rand-Quiz have been compared, existing systems have been studied, and multi-factor authentication, one-time password (OTP) and role-based access control are analyzed.

Table 2: Functional Requirements for Rand-Quiz

No.	Functional Requirements
1.	Rand-Quiz should allow all users able to login using user ID, password, and Email OTP.
2.	Rand-Quiz should allow admin to register admin, teacher, student, subject, and workload. The admin also will be able to add, update and delete the registration module.
3.	Rand-Quiz should allow admin to view logs.
4.	Rand-Quiz should allow teachers to view their workload.
5.	Rand-Quiz should allow teachers to create, edit and delete quizzes.
6.	Rand-Quiz should allow teachers to view the result.
7.	Rand-Quiz should allow students to register subjects and view the registered subjects.
8.	Rand-Quiz should allow students to answer the quiz and view the result.
9.	Rand-Quiz should allow all users to logout from the system.

Table 3: Non-Functional Requirements for Rand-Quiz

No.	Non-Functional Requirements
1.	Rand-Quiz should be able to identify the role of the user based on the user ID and password.
2.	Rand-Quiz should only allow users to create a minimum of 8 characters alphanumeric passwords.
3.	Rand-Quiz should be able to hash the password and OTP using bcrypt algorithm before it is stored in the database.
4.	Rand-Quiz should allow only the admin to see users' log activity.
5.	Rand-Quiz should be able to record users' log activity.

3.1.2 Quick Design Phase

The data flow for administrators, teachers, and students as they enter Rand-Quiz is depicted in the context diagram. For Rand-Quiz to allow the three entities access and direct them to the authorized module, they are required enter the correct user ID, password, and Email OTP. Rand-Quiz 's context diagram is shown in Figure 1.

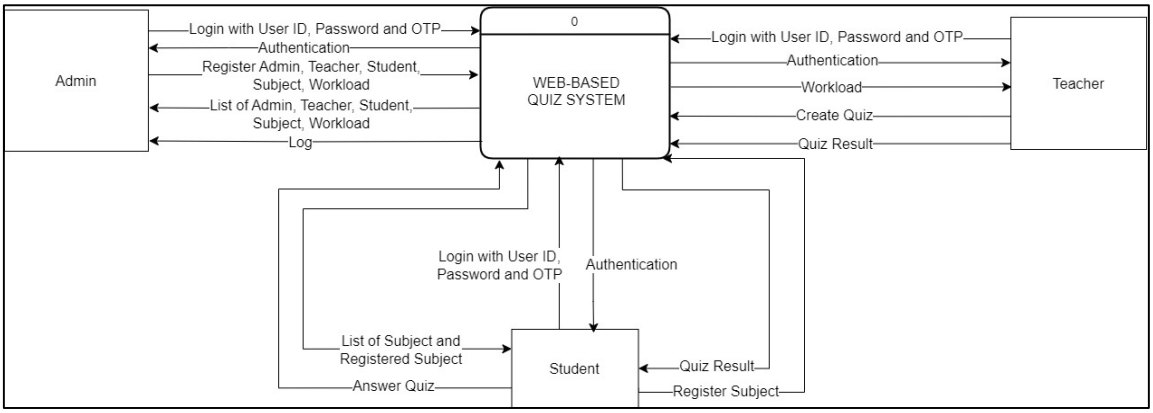

Figure 1: Context Diagram for Rand-Quiz

Before Rand-Quiz is created, data flow diagrams are constructed to depict how data will move through Rand-Quiz as it is being used. The data flow diagram in Figure 2 will involve the three main entities which are admin, teacher, and student. Different entities will have varying access to a particular page. Other than that, the entities can use Rand-Quiz to perform a variety of tasks.

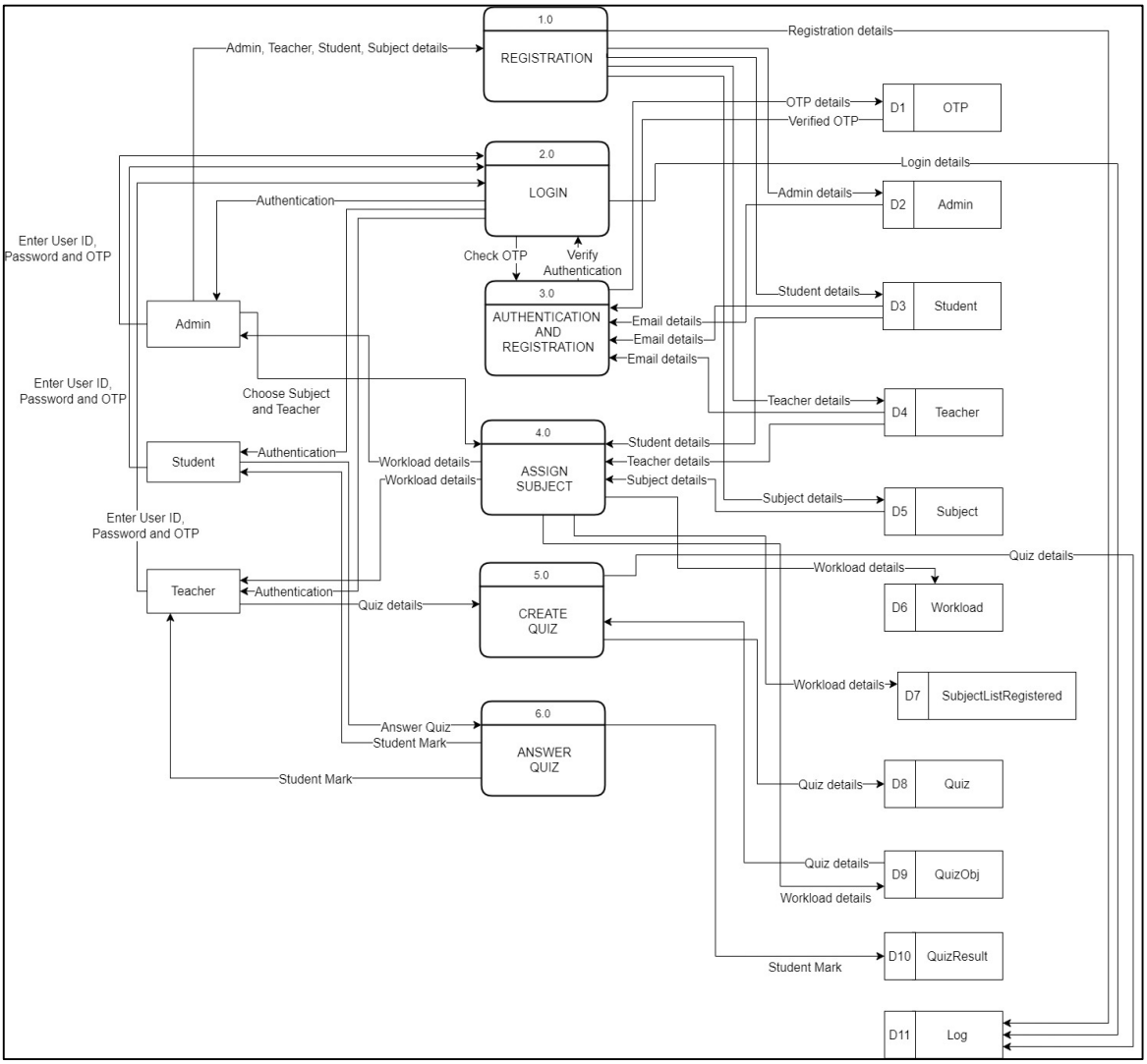

Figure 2: Data Flow Diagram for Rand-Quiz

An entity relationship diagram is designed to identify the relationship between entities in Rand-Quiz. Figure 3 shows the connection between entities in Rand-Quiz.

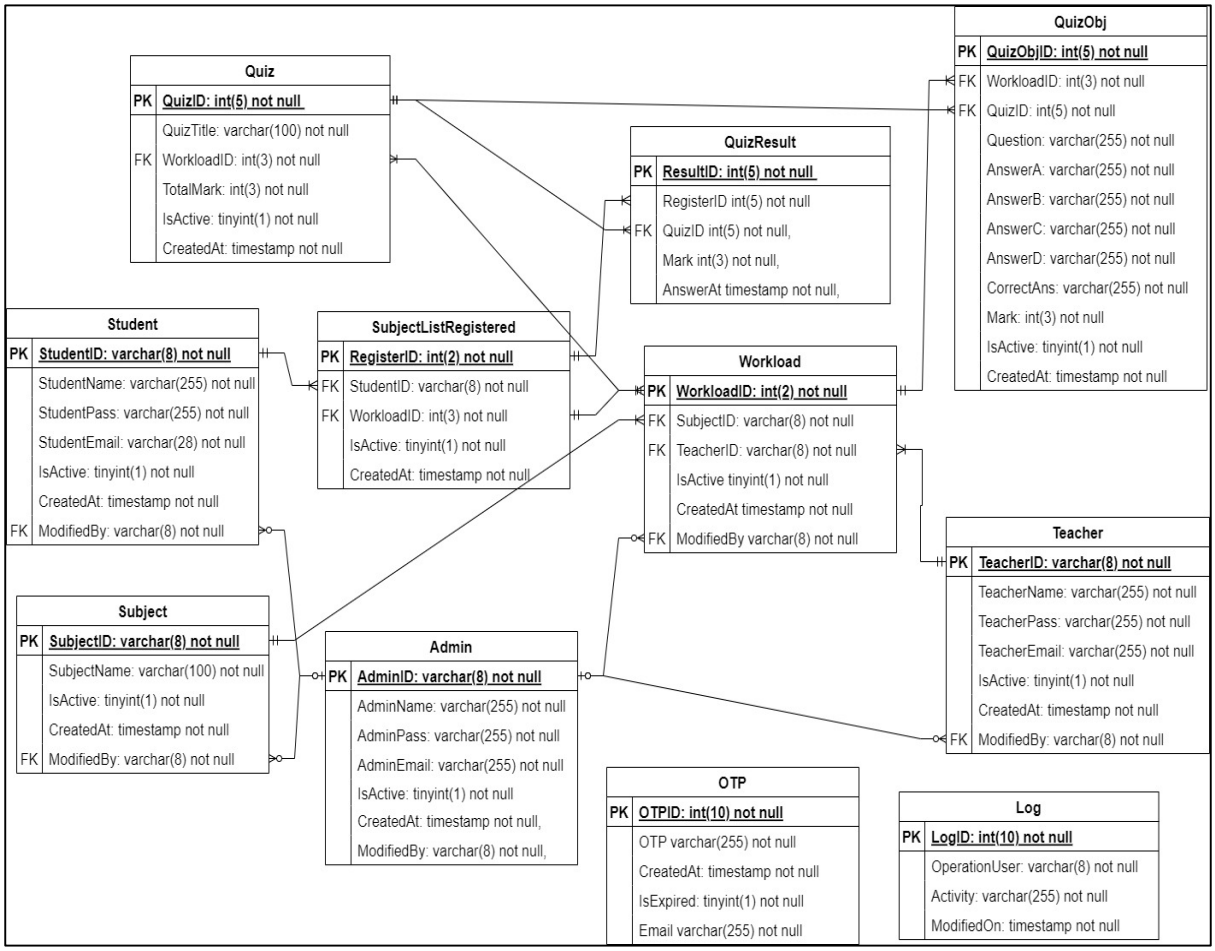

Figure 3: Entity Relationship Diagram for Rand-Quiz

3.1.3 Build Prototype Phase

The development stage focuses on building the Rand-Quiz prototype system based on the design specifications. It involves implementing the functionality, features, and interactions required for the prototype. This stage involves where all the theoretical concepts of Rand-Quiz are created through coding.

In addition, this phase will implement OTP that helps for the confidentiality, integrity and availability of Rand-Quiz. OTP is a mechanism used to enhance security by providing a unique password that is valid for only a single login session or transaction, on a computer system or other digital device. OTPs are commonly used for two-factor authentication to add an extra layer of security.

This phase is divided into three primary components which are hardware, software, and programming language. The hardware that will be used for developing Rand-Quiz is Intel® Core™ i5-1135G7 Processor, 2.40 GHz with 8 GB DDR4 SDRAM. Besides, Visual Studio Code will be used as the software to develop the quiz system. XAMPP will serve as a localhost to store the database. Four programming languages will be used are PHP, HTML, CSS and JavaScript. All the programming languages are used to make Rand-Quiz more systematic and user-friendly.

3.1.4 User Evaluation Phase

This stage involves testing the prototype with users to get feedback and identify any problems. Once the prototype is developed, it needs to be tested to ensure that it meets the desired objectives and requirements. Testing helps to identify any usability issues, functionality problems, or design flaws. Feedback from users is collected during this stage. The evaluation stage involves assessing the prototype's performance, usability, and user satisfaction. It includes usability testing and user feedback surveys. The evaluation helps to determine if the prototype is ready for further development or if additional iterations are required.

Rand-Quiz that was developed in the previous phase is tested in this phase. All of the previously made requirements will also be put into practice during this phase. Depending on the testing strategy, a number of tests must pass. Two different testing methods will be used in this phase to ensure that Rand-Quiz is operating in accordance with the specifications. Rand-Quiz 's functionality will be tested first to make sure the system functions as needed. Then, the user acceptance testing is the second type of testing that will be performed on the developed system to make sure that Rand-Quiz can operate properly, and that the user is satisfied with the system.

In addition, the security feature that has been integrated into Rand-Quiz will be tested by both types of testing. Therefore, any systematic errors will be corrected to ensure that Rand-Quiz can perform as intended.

3.1.5 Refining Prototype Phase

This stage involves making changes to the prototype based on the feedback from users. Based on the feedback received during testing, the prototype may go through multiple iterations. Each iteration involves refining the design, implementing changes, and improving the prototype based on the identified issues. This iterative process continues until the prototype satisfies the desired objectives.

3.1.6 Implementation and Maintenance Phase

This stage involves releasing the product to the market. It involves the actual implementation of the final system and its maintenance. The implementation process involves deploying Rand-Quiz in the production environment and making it available to end-users. The maintenance process involves fixing any issues that arise in Rand-Quiz and making updates or modifications as needed to ensure the system's functionality.

4. Result and Discussion

4.1 Security Module

The security modules implemented are discussed in this section. Confidentiality, integrity, and availability of the quiz system will be included in the security features. The security modules implemented in Rand-Quiz are one-time password (OTP), randomizing questions and answers sequence, hashing for the password and OTP, strong password requirements, and session destruction.

4.1.1 Implementation of One-Time Password (OTP)

One-Time Password login security feature will be this quiz system's most important security feature. This feature covers the Confidentiality component of the CIA Triad. An Email OTP will be sent to the user's email via an email service every time when the user wants to login. Figure 4 will illustrate how the OTP generator works. The user will then be prompted to enter a sent OTP, which will be compared to an OTP stored in the database with a 60 second time limit. Access to authenticated pages will be granted by Rand-Quiz if the OTP is matched.

```php
function generateOTP($userid, $username, $userpass) {
    $otp = "KJY-";

    $user = substr($userid, 2);
    $combinedString = $user . mixCase(str_replace(' ', '$', $username));
    $length = strlen($combinedString);

    for ($i = 0; $i < 8; $i++) {
        $randomIndex = random_int(0, $length - 1);
        $otp .= substr($combinedString, $randomIndex, 1);
    }

    $otp .= "-";

    $length = strlen($userpass);

    for ($i = 0; $i < 8; $i++) {
        $randomIndex = random_int(0, $length - 1);
        $otp .= substr($userpass, $randomIndex, 1);
    }

    return $otp;
}
```

Figure 4: Partial Code for OTP Generation to be Send to User Email

Figure 4 shows the function of generating OTP. It consists of the combination of ownership, user ID, username and hash value of user password. Line 33 initializes the otp variable with the string "KJY-" as a prefix for the OTP. Line 35 extracts a substring from the user ID starting from the third character of the user ID. Line 36 combines the substring of user ID and a mixed-case version of the username. The mixed-case is done by using a function that randomly converts each character of the input string to either lowercase or uppercase and replacing spaces with '$' characters, because the original username is in uppercase and with spaces. Line 37 calculates the length of the combined string. From line 39 to line 42, there is an iteration to randomly choose an index from the combined string and obtain the character. This process is repeated 8 times to obtain a random selection from the combination of user ID and username for the second part of OTP. From line 48 to line 51, it is similar to the previous steps. This process is repeated 8 times to obtain a random selection of characters from the hash value of the user password for the third part. Finally, it returns the resulting OTP with the format "KJY-<random characters from user ID substring and username>-<random characters from userpass>".

←T→	AdminID	AdminName	AdminPass	AdminEmail	IsActive	ModifiedOn	ModifiedBy
Edit Copy Delete	AD230001	KOH JIA YEE	$2y$10$KwxschMQVEKW7ekT245gBuxSDcG1/LLF.h/zXA6oYxa...	kohjiayee24@gmail.com	1	2023-06-18 01:51:08	

Figure 5: User Details in Database That Show the User ID, Username and Hashed Password

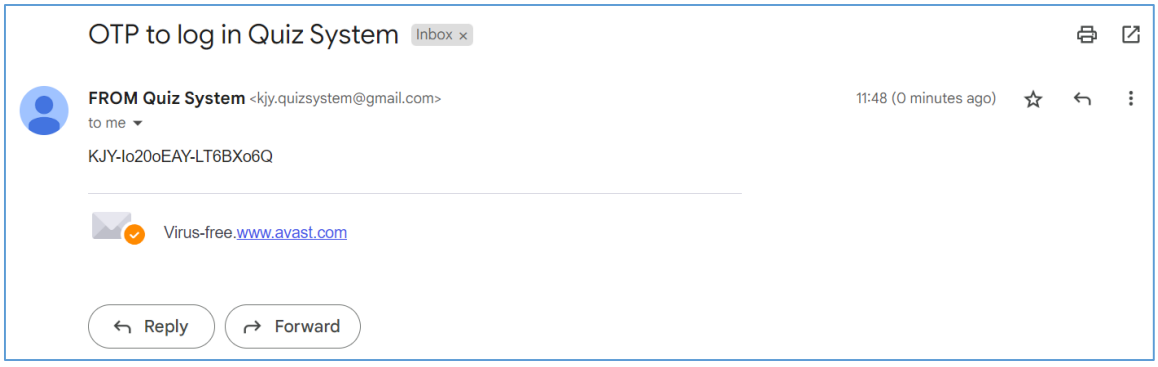

Figure 6: The OTP Generated and Received by User

For example, Figure 5 shows the user details that consists of the user ID, username and hashed password. For the user with user ID ‘AD230001’, username ‘Koh Jia Yee’ and the hashed value of password, the OTP that can be generated is shown in Figure 6.

4.1.2 Implementation of Randomizing Questions and Answers Sequence

Figure 7 shows the partial code to randomize the sequence of questions and answer options. This feature has been put in place to prevent students from discussing the correct response during the quiz session. Line 95 to line 105 retrieve active quiz question details based on the selected quiz ID in random order. Line 107 and line 108 initialize an empty array $quizQuestions and an empty string $selectedQuizTitle. Line 110 to line 113 execute a loop to fetch all the quiz questions and stores the title of the selected quiz based on the data retrieved from lines above. Line 115 shuffles quiz question details to be randomized. Line 117 shows selected quiz title. If students have not submitted the quiz, line 119 to line 137 will be processed. Line 122 to 133 execute a loop to iterate over each question in the $quizQuestions array. Line 124 shuffles answer options for the question. Line 126 to line 131 displayed the shuffled question and answer options. Line 135 shows the submit button for students to submit the quiz after finish answering. Line 137 to line 139 display the quiz result after student submitted the quiz.

```
95  $query = "SELECT q.QuizTitle, obj.QuizObjID, obj.Question, obj.AnswerA, obj.AnswerB, obj.AnswerC, obj.AnswerD, obj.Mark
96            FROM QuizObj AS obj
97            JOIN Quiz AS q ON obj.QuizID = q.QuizID
98            WHERE obj.IsActive = 1
99            AND obj.QuizID = ?
100           ORDER BY RAND()";
101
102 $stmt = $conn->prepare($query);
103 $stmt->bind_param("i", $selectedQuizID);
104 $stmt->execute();
105 $result = $stmt->get_result();
106
107 $quizQuestions = array();
108 $selectedQuizTitle = '';
109
110 while ($data = mysqli_fetch_assoc($result)) {
111   $quizQuestions[] = $data;
112   $selectedQuizTitle = $data['QuizTitle'];
113 }
114
115 shuffle($quizQuestions);
116
117 echo '<h2>' . $selectedQuizTitle . '</h2>';
118
119 if (!$quizSubmitted) {
120   echo '<form method="post" action="">';
121
122   foreach ($quizQuestions as $question) {
123     $answerOptions = array($question['AnswerA'], $question['AnswerB'], $question['AnswerC'], $question['AnswerD']);
124     shuffle($answerOptions);
125     echo '<div>';
126     echo '<p>' . $question['Question'] . '</p>';
127     echo '<input type="hidden" name="quizobj_id[]" value="' . $question['QuizObjID'] . '" />';
128     echo 'A)<input type="radio" name="student_answer[' . $question['QuizObjID'] . ']" value="' . $answerOptions[0] . '" required> ' . $answ
129     echo 'B)<input type="radio" name="student_answer[' . $question['QuizObjID'] . ']" value="' . $answerOptions[1] . '" required> ' . $answ
130     echo 'C)<input type="radio" name="student_answer[' . $question['QuizObjID'] . ']" value="' . $answerOptions[2] . '" required> ' . $answ
131     echo 'D)<input type="radio" name="student_answer[' . $question['QuizObjID'] . ']" value="' . $answerOptions[3] . '" required> ' . $answ
132     echo '</div>';
133   }
134
135   echo '<input type="submit" value="Submit Answers" />';
136   echo '</form>';
137 } else {
138   echo 'Quiz submitted. Your score: ' . $score . ' out of ' . $totalmark;
139 }
```

Figure 7: Partial Code for Randomization of Questions and Answers Sequence

.

Figure 8 and Figure 9 show the sequence of questions and answer options are randomized.

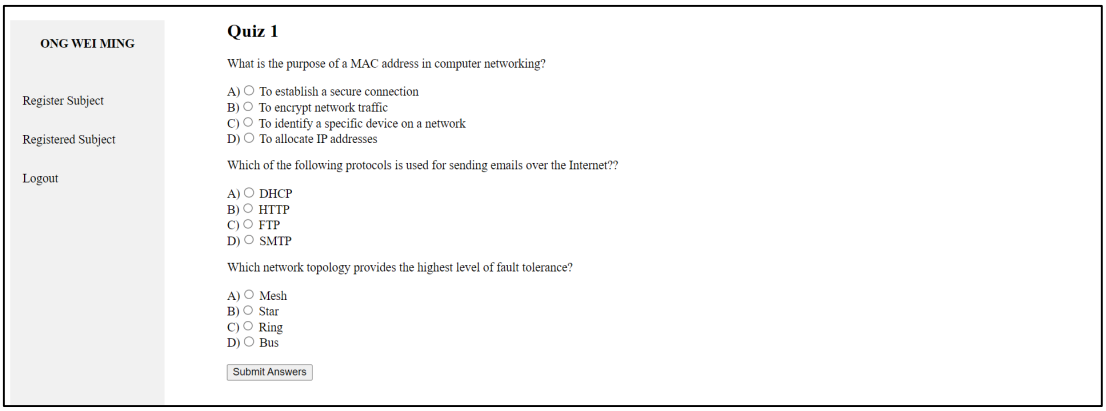

Figure 8: Randomization of Questions and Answers Sequence at Quiz Answering Page

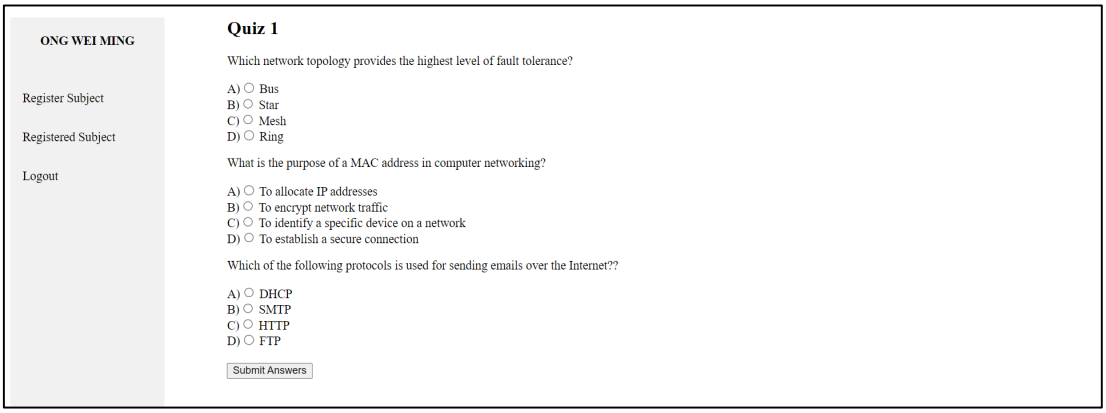

Figure 9: Randomization of Questions and Answers Sequence at Quiz Answering Page

4.1.3 Implementation of Hashing for Password and OTP

Figure 10, Figure 12 and Figure 14 show the partial code for password hashing for admin, teacher and student respectively. The partial code for admin and teacher is similar, while slightly different for the student in the part of email as student's email is default set based on their student ID. For admin and teacher, line 27 shows the format of constructing the ID. Line 29 to line 30 retrieves the name from the user input. Line 31 retrieves the generated ID as the password, this is because the default password is same as the ID. The user will be prompted to change password for the first login. Line 32 to line 33 retrieves the email from the user input. Line 34 hashes the password using bcrypt algorithm and stores in the variable $hashed for later use. Line 36 to line 41 checks if the email input is matched with existing email or sender email because the email is used to receive the OTP for login purpose and should not be duplicated. If the email is not duplicated with others, line 43 to line 45 insert the details into the database.

Figure 11 and Figure 13 show the hashed version of admin and teacher password stored in the database. For students, line 27 shows the format of constructing the ID. Line 29 to line 30 retrieves the name from the user input. Line 31 retrieves the generated ID as the password. Line 32 hashes the password using bcrypt algorithm and stores in the variable $hashed for later use. Line 34 to line 36 insert the details into the database.

```php
$AdminID = 'AD' . $twoDigitYear . str_pad($lastNumber, 4, '0', STR_PAD_LEFT);

$AdminName = stripslashes($_REQUEST['username']);
$AdminName = mysqli_real_escape_string($conn, $AdminName);
$AdminPass = mysqli_real_escape_string($conn, $AdminID);
$AdminEmail = stripslashes($_REQUEST['email']);
$AdminEmail = mysqli_real_escape_string($conn, $AdminEmail);
$hashed = password_hash($AdminPass, PASSWORD_DEFAULT);

$query = "SELECT * FROM `Admin` WHERE AdminEmail = '$AdminEmail'";
$result = mysqli_query($conn, $query);
$rows = mysqli_num_rows($result);

if ($rows == 1 || $AdminEmail == "kjy.quizsystem@gmail.com") {
  echo "<script>alert('Email already existing.');</script>";
} else {
  $query = "INSERT INTO `Admin` (AdminID, AdminName, AdminPass, AdminEmail, ModifiedBy, IsActive)
  VALUES ('$AdminID', '" . strtoupper($AdminName) . "', '$hashed', '" . strtolower($AdminEmail) . "', '$userid', 1)";
  $result = mysqli_query($conn, $query);
```

Figure 10: Partial Code for Hashing of Admin Password

	AdminID	AdminName	AdminPass	AdminEmail	IsActive	ModifiedOn	ModifiedBy
Edit Copy Delete	AD230001	KOH JIA YEE	$2y$10$KwxschMQVEKW7ekT245gBuxSDcG1/LLF.h/zXA6oYxa...	kohjiayee24@gmail.com	1	2023-06-18 01:51:08	
Edit Copy Delete	AD230002	LIANG YOU HOW	$2y$10$oGNscx51avNQ2wNVKAPc/uP3IfwESO637Da45X1roa6...	youhow24@hotmail.com	1	2023-06-18 01:54:22	AD230001

Figure 11: Hashed Admin Password in Database

```php
$TeacherID = 'AT' . $twoDigitYear . str_pad($lastNumber, 4, '0', STR_PAD_LEFT);

$TeacherName = stripslashes($_REQUEST['username']);
$TeacherName = mysqli_real_escape_string($conn, $TeacherName);
$TeacherPass = mysqli_real_escape_string($conn, $TeacherID);
$TeacherEmail = stripslashes($_REQUEST['email']);
$TeacherEmail = mysqli_real_escape_string($conn, $TeacherEmail);
$hashed = password_hash($TeacherPass, PASSWORD_DEFAULT);

$query = "SELECT * FROM `Teacher` WHERE TeacherEmail ='$TeacherEmail'";
$result = mysqli_query($conn, $query);
$rows = mysqli_num_rows($result);

if ($rows == 1 || $TeacherEmail == "kjy.quizsystem@gmail.com") {
  echo "<script>alert('Email already existing.');</script>";
} else {
  $query = "INSERT into `Teacher` (TeacherID, TeacherName, TeacherPass, TeacherEmail, ModifiedBy, IsActive)
          VALUES ('$TeacherID', '" . strtoupper($TeacherName) . "', '$hashed', '" . strtolower($TeacherEmail) . "', '$userid', 1)";
  $result = mysqli_query($conn, $query);
```

Figure 12: Partial Code for Hashing of Teacher Password

	TeacherID	TeacherName	TeacherPass	TeacherEmail	IsActive	ModifiedOn	ModifiedBy
Edit Copy Delete	AT230001	PROF. MADYA TS. DR. HAIRULNIZAM BIN MAHDIN	$2y$10$lu/wMwx6olEwyZkW/sOJnu5omfAKYsJDj7zhq9sR3lU...	hairulnizam@hello.com	1	2023-06-20 22:57:56	AD230001
Edit Copy Delete	AT230002	ENCIK KHAIRUL AMIN BIN MOHAMAD SUKRI	$2y$10$Ss9oLlm0bhdRZaDoHYvAPecBtVCt6ib.owzSiLXPymP...	khairul_amin@gmail.com	1	2023-06-20 23:13:57	AD230002
Edit Copy Delete	AT230003	TS. DR. NURUL HIDAYAH BINTI AB RAHMAN	$2y$10$5A.tegoc00v.ZJzMroRjAeetjXSPdhaBFcJTeDL0A41...	rahman@gmail.com	1	2023-06-18 01:56:04	AD230001
Edit Copy Delete	AT230004	PROF. MADYA DR. KAMARUDDIN MALIK BIN MOHAMAD	$2y$10$Cpj6HeVR/8rSRwcBDsh7wequejPxcdIbygrIpETZt5i...	amarud@gmail.com	1	2023-06-18 05:55:33	AD230002
Edit Copy Delete	AT230005	PUAN IZA NURHIDAYAH BINTI ISMAIL	$2y$10$n13h0qzfn0JaFDOuOgX2ruvgy2i8eqQCdUNtF3dr335...	iza22@gmail.com	1	2023-06-18 01:56:26	AD230002

Figure 13: Hashed Teacher Password in Database

```php
$StudentID = 'AI' . $twoDigitYear . str_pad($lastNumber, 4, '0', STR_PAD_LEFT);

$StudentName = stripslashes($_REQUEST['username']);
$StudentName = mysqli_real_escape_string($conn,$StudentName);
$StudentPass = mysqli_real_escape_string($conn,$StudentID);
$hashed = password_hash($StudentPass, PASSWORD_DEFAULT);

$query = "INSERT into `Student` (StudentID, StudentName, StudentPass, StudentEmail, ModifiedBy, IsActive)
        VALUES ('$StudentID', '" . strtoupper($StudentName) . "', '$hashed', '".$StudentID."@student.uthm.edu.my', '$userid', 1)";
$result = mysqli_query($conn,$query);
```

Figure 14: Partial Code for Hashing of Student Password

	StudentID	StudentName	StudentPass	StudentEmail	IsActive	ModifiedOn	ModifiedBy
Edit Copy Delete	AI230001	ONG WEI MING	$2y$10$o5CJKHY6mLEa02clW.Xne.4inUOg6fzVls0YsqBCOHT...	AI230001@student.uthm.edu.my	1	2023-06-18 05:56:09	AD230001
Edit Copy Delete	AI230002	TERENCE CHEE MING CHUN	$2y$10$FL.HnACG7oAUtt5WTeh/A.tQa2czmMlE1qhZr5nZSDO...	AI230002@student.uthm.edu.my	1	2023-06-14 06:21:16	AD230001
Edit Copy Delete	AI230003	MASITAH ROSLII	$2y$10$UBtFQBxMB0hSaUHjk63tFOrQvGAf9hqRjKU8IUhB6uO...	AI230003@student.uthm.edu.my	1	2023-06-14 06:22:32	AD230002
Edit Copy Delete	AI230004	SHALINI SUBRAMANYY	$2y$10$va6.FI22rlul0T3hl3tNAu/QX0TVVXVtPHMNM1WocLn...	AI230004@student.uthm.edu.my	1	2023-06-14 10:52:31	AD230002
Edit Copy Delete	AI230005	MOHD SHAHMEER HUSNI BIN NASIB	$2y$10$6LNr/7dpt9A2RQufaT4SD.yry4RZBVqFr1v5z/O39DD...	AI230005@student.uthm.edu.my	1	2023-06-14 06:22:39	AD230002

Figure 15: Hashed Student Password in Database

Figure 15 shows the hashed version of the student password stored in the database. Figure 16 shows the partial code for password hashing process for OTP. Line 88 hashed the generated OTP using bcrypt algorithm and stores in the variable $hashedOTP for later use. Line 89 sets the expiration status of OTP as not expired. Line 90 to line 93 insert the OTP details into the database. Figure 17 shows the hashed version of OTP stored in the database.

```
88      $hashedOTP = password_hash($otp, PASSWORD_DEFAULT);
89      $isExpired = 0;
90      $query = "INSERT INTO `OTP` (OTP, IsExpired, Email) VALUES (?, ?, ?)";
91      $stmt = $conn->prepare($query);
92      $stmt->bind_param("sis", $hashedOTP, $isExpired, $email);
93      $stmt->execute();
```

Figure 16: Partial Code for Hashing of OTP

	OTPID	OTP	CreatedAt	IsExpired	Email
Edit Copy Delete	1	$2y$10$AUIIrSFOqICPbtBeIaUyiOc12GS2XMeP233.FCnlgYY...	2023-06-14 04:29:36	1	kohjiayee24@gmail.com
Edit Copy Delete	2	$2y$10$enasyqEQVV.bHIIrDyohUu8.zcuXYZbTPqF9NuGXRjt...	2023-06-14 04:41:17	1	youhow24@hotmail.com
Edit Copy Delete	3	$2y$10$TJjrkD3STfbDWVqmY7ZbR.GdCz1MtlEIzdWG09w2pT9...	2023-06-14 04:50:48	1	hairulnizam@gmail.com
Edit Copy Delete	4	$2y$10$MMeif0sROryJdXl3pUnbHOEv8MM2f4oaKmyq7SMa4S2...	2023-06-14 04:53:37	1	hairulnizam@gmail.com

Figure 17: Hashed OTP in Database

4.1.4 Implementation of Strong Password Policy

Figure 18 shows the partial code for complexity and length of password. The password policy obeys OWASP password policy. The password must contain uppercase, lowercase, number, special character and must be longer than 8 characters.

```
297     $errors = [];
298     if (strlen($newPassword) < 8) {
299         $errors[] = "Password must contain at least 8 characters.";
300     }
301     if (!preg_match("/[A-Z]/", $newPassword)) {
302         $errors[] = "Password must contain at least 1 uppercase letter.";
303     }
304     if (!preg_match("/[a-z]/", $newPassword)) {
305         $errors[] = "Password must contain at least 1 lowercase letter.";
306     }
307     if (!preg_match("/[0-9]/", $newPassword)) {
308         $errors[] = "Password must contain at least 1 number.";
309     }
310     if (!preg_match("/[^a-zA-Z0-9]/", $newPassword)) {
311         $errors[] = "Password must contain at least 1 special character.";
312     }
313     if (empty($confirmPassword)) {
314         $errors[] = "Password confirmation cannot be empty.";
315     }
316     if ($newPassword !== $confirmPassword) {
317         $errors[] = "Password confirmation should be the same as the password.";
318     }
```

Figure 18: Partial Code for password complexity and length

4.1.5 Implementation of Session Destroy

Figure 19 shows the code for destroying sessions. This starts from starting or resuming the current session. Later, unset all session variables. Finally, use the session_destroy() function to destroy the session including all session data, and redirect users to the login page.

```php
<?php
session_start();
session_unset();
session_destroy();
header("Location: login.php");
exit();
?>
```

Figure 19: Code for session destroy

5. Conclusion

Rand-Quiz, a Randomized Web-Based Quiz System, secured with OTP using KJY Password Generator has been successfully developed. An Email OTP will be sent to the user each and every time the user wants to login into the system. The OTP will be in the form of a unique character combination digit consisting of ownership prefix and combines with randomized portions of user ID, username and user password. Rand-Quiz also randoms the sequence of questions to increase security for quiz candidates that are taking the quiz which their computers located next to each other.

Acknowledgment

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

References

[1] T. Gonzalez et al.. (2020). Influence of COVID-19 confinement on students' performance in higher education. PLoS One, vol. 15, no. 10. Retrieved on October 20, 2022, from doi: 10.1371/journal.pone.0239490

[2] Li, Cathy, and Farah Lalani. (2020). The COVID-19 Pandemic Has Changed Education Forever. Retrieved on October 21, 2022, from http://www.weforum.org/agenda/2020/04/coronavirus-education-global-covid19-online-digital-learning/

[3] Quizziz Inc. (2015). Quizizz. Retrieved on November 3, 2022, from https://quizizz.com/?fromBrowserLoad=true

[4] Google. (2008). Google Forms. Retrieved on November 3, 2022, from https://www.google.com/forms/about/

[5] Mercer Mettl. (2010). Mettl online assessment © 2010-2019. Retrieved on November 4, 2022, from https://mettl.com/en/

[6] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy. (2018). Multi-factor authentication: A survey. Cryptography, vol. 2, no. 1, pp. 1–31. Retrieved on November 7, 2022, from doi: 10.3390/cryptography2010001

[7] Mercer LLC. (2022). Proctored Exams: Secure Online Proctoring With AI-Powered Tools. Retrieved on November 7, 2022, from https://mettl.com/online-remote-proctoring/?hsCtaTracking=86e3b5b2-f638-41ca-8928-84c10bf364e7%7Ce5c871a3-4e02-4ad5-94ac-f97acea61bc2#mettl-secure-browser

[8] Amin, Asif, et al. (2017). TWO FACTOR AUTHENTICATION. International Journal of Computer Science and Mobile Computing, vol. 6, no. 7, pp. 5–8.

[9] Huang, Y., Huang, Z., Zhao, H., & Lai, X. (2013). A new One-time Password Method. IERI Procedia, 4, 32–37. Retrieved on May 17, 2023, from doi: 10.1016/j.ieri.2013.11.006

[10] Kurniawan, D. A., Iqbal, M. S., Friadi, J., Hidayat, F., & Permatasari, R. D. (2021). Login Security Using One Time Password (OTP) Application with Encryption Algorithm Performance. Journal of Physics, 1783(1), 012041. Retrieved on May 17, 2023, from doi: 10.1088/1742-6596/1783/1/012041

[11] Acharya, S. (2013). Two Factor Authentication Using Smartphone Generated One Time Password. IOSR Journal of Computer Engineering. Retrieved on May 17, 2023, from doi: 10.9790/0661-1128590

[12] Shally, Aujla, G. S., & Aujla, S. (2014). A REVIEW OF ONE TIME PASSWORD MOBILE VERIFICATION. ResearchGate. Retrieved on May 18, 2023, from https://www.researchgate.net/publication/264155414_A_REVIEW_OF_ONE_TIME_PASSWORD_MOBILE_VERIFICATION

[13] Yildirim, M., & Mackie, I. (2019). Encouraging users to improve password security and memorability. International Journal of Information Security, 18(6), 741–759. Retrieved on May 18, 2023, from doi: 10.1007/s10207-019-00429-y

[14] Erdem, E., & Sandikkaya, M. T. (2019). OTPaaS—One Time Password as a Service. IEEE Transactions on Information Forensics and Security, 14(3), 743–756. Retrieved on May 20, 2023, from doi: 10.1109/tifs.2018.2866025

[15] Balilo, B. B., Gerardo, B. D., Medina, R. P., & Byung, Y. E. (2017). An Improved OTP Grid Authentication Scheme Email-based using Middle-square for Disaster Management System. International Journal of Grid and Distributed Computing, 10(11), 43–56. Retrieved on May 20, 2023, from doi: 10.14257/ijgdc.2017.10.11.05

[16] Christiana, A. O., Oluwatobi, A. N., Victory, G. A., & Ogundokun, R. O. (2019). A Secured One Time Password Authentication Technique using (3, 3) Visual Cryptography Scheme. Journal of Physics, 1299, 012059. Retrieved on May 21, 2023, from doi: 10.1088/1742-6596/1299/1/012059

[17] D. Ferraiolo and D. R. Kuhn. (2009). Role-Based Access Controls. Retrieved on November 10, 2022, from https://www.researchgate.net/publication/24164143

[18] Mercer | Mettl. (2020). Role Based Access Control. Retrieved on November 10, 2022, from https://support.mettl.com/portal/en/kb/articles/role-based-access-control-10-7-2020

[19] Google Help. (n.d.). Create & grade quizzes with Google Forms. Retrieved on November 10, 2022, from https://support.google.com/docs/answer/7032287?hl=en&ref_topic=6063584#zippy=%2Cmake-an-answer-key-assign-points-add-automatic-feedback%2Cchoose-what-people-see-during-and-after-the-quiz%2Csend-your-quiz-to-people-outside-of-your-work-or-school%2Csee-quiz-results%2Cgrade-question-by-question