



ASIA PACIFIC UNIVERSITY OF
TECHNOLOGY & INNOVATION

CSS-CT013-3-3

APU/APIDT2505(ISS)/CE/TE

| | |
|--------------------------|--|
| TITLE | SECURITY ENHANCEMENT OF EXSITING WEB APPLICATION |
| NAME / STUDENT ID | SHAIK YASEEN (TP062920) |
| INTAKE | APU4F2505CE |
| LECTURER | DR. KAMALAKANNAN MACHAP |
| HAND-OUT DATE | |
| HAND-IN DATE | 29 th August 2025 |

Contents

| | |
|---|-----------|
| | 1 |
| 1. Introduction | 4 |
| 2. SECURITY ISSUES, POSSIBLE CYBERATTACKS ON THE CURRENT COMPUTER SYSTEM | 5 |
| 2.1 Limitation of Current System: | 5 |
| 2.1.1 No Password Composition Policies: | 5 |
| 2.1.2 No Password Recovery: | 7 |
| 2.1.3 No Authentication Factor in Registration Form: | 8 |
| 2.2 Possible Cyber Attacks: | 8 |
| 2.2.1.1 Brute Force Attacks | 9 |
| 2.2.1.2 Dictionary Attacks: | 9 |
| 2.3 Phishing Attack: | 10 |
| 2.3.1 Account Takeover (ATO) Attack: | 10 |
| 3 REQUIREMENTS FOR DESIGNING A SECURE SYSTEM..... | 11 |
| 3.1 Password Strength Checker..... | 11 |
| 3.2 Password Recovery | 13 |
| 3.3 Encryption:..... | 15 |
| 3.4 Access Control:..... | 15 |
| 4 Implementation and Code Explanation of the selected security method . | 16 |
| 4.1 Software Configuration: | 16 |
| 4.2.1 Source Code Explanation: | 18 |
| 4.2.2 Results: | 22 |
| 4.2.3 Login with Invalid Credentials: | 23 |
| 4.2.4 Successful Login: | 24 |
| 4.2.5 Manage users | 24 |
| 4.2.6 Manage Users – Add User: | 25 |
| 4.2.7 Manage Users – Edit User: | 25 |
| 4.2.8 Manage Users – Delete User: | 26 |
| 4.2.9 Forgot Password Page : | 27 |

| | |
|--|-----------|
| 4.2.9Reset Password Page (via token link):..... | 28 |
| 4.2.10Successful Password Reset: | 29 |
| 4.2.11Change Password Page: | 29 |
| 4.2.12Successful Change Password Message: | 30 |
| 4.2.13Settings Page : | 31 |
| 5 CONCLUSION..... | 31 |
| References :..... | 32 |

1. Introduction

Even in the current digital age, secure online systems are becoming more and more important since sensitive data is handled via web applications in the banking, healthcare, government, and educational sectors. These systems typically handle private user data, such as phone numbers, email addresses, passwords, and names. Weak security measures could make such information public, which could have disastrous repercussions like identity theft, harm to one's reputation, and financial loss.(Cisco, 2022).

Authentication(typically done through usernames and passwords) continues to be used as a preferred means of restricting web apps. It is merely practical when you are combining it with formidable technical protection. The passwords cannot be stored in plain text but must be hashed at least before they touch the database. On the login the system calculates a hash of the input of the user and compares that hash to the saved hash. When the two are equal the user is allowed access (see PHP Manual, n.d.). Weak authentication design is always used by attackers. Consider brute-force or dictionary attacks, phishing, session hijacking or replying to insecure reset links. The 2021 OWASP Top Ten lists broken authentication and weak cryptography as the biggest culprit of breach. Even a simple login page would be exploitarian without defending it, and with lax password policy, poor hashing, token recovery, and weak session controls, is inviting exploits. This class project is a process to beef up a basic login system by addressing those issues. Some of the proposed upgrades are the use of hashed passwords, anti-deresent support to impose strong password policies, establishing a secure / forgot/reset user flow, introducing user management and supplementation of access control on session grounds.

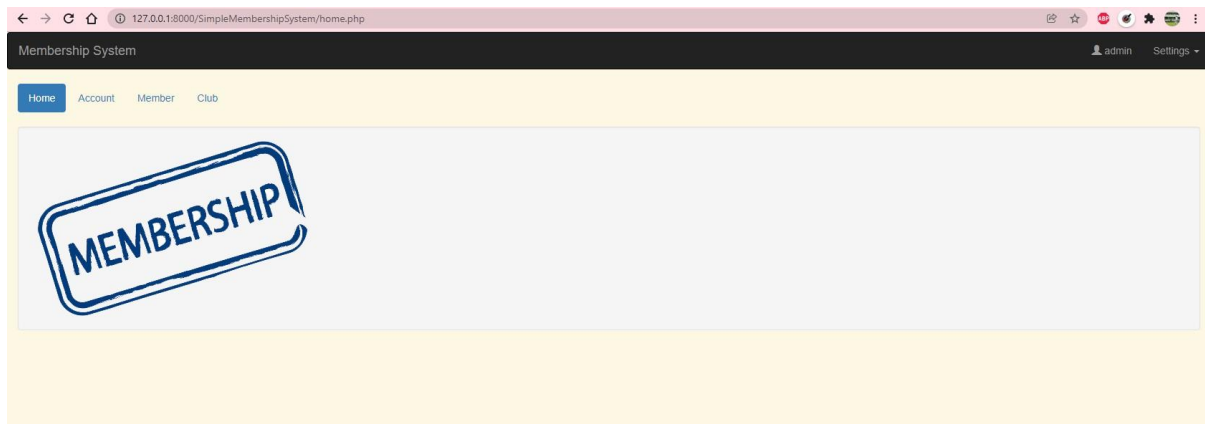


Figure 1 Simple Membership System

We chose a simple admin-log system to explore and talk about its vulnerabilities which will create significant security issues or result in possible cyber attacks. We illuminate and suggest security additions to overcome probable vulnerabilities and threats during the process of log in. Lastly, we take the necessary actions as an impact of protection of the net security of the system. In the course of it, we remember about the three fundamental principles of information security, namely confidentiality, which prevents unauthorized access to any instance of sensitive information; integrity, which ensures that the stored information cannot be ruined through malicious activities; and availability, which ensures that the authorized administrator can access the system whenever he/she wants to.

2. SECURITY ISSUES, POSSIBLE CYBERATTACKS ON THE CURRENT COMPUTER SYSTEM

2.1 Limitation of Current System:

We discovered that the basic login system has some deadly security vulnerabilities which I am going to go through in this section. It is important to identify such weak spots since this allows us to design and implement the correct defenses before the system is subjected to actual attacks by the real world. By addressing the issues at the onset, we can reduce significantly the days of the successful breach of the system, which will enhance the overall system strength.

2.1.1 No Password Composition Policies:

Passwords remain the preferred option of protection over applications at the campus and personal accounts. They are exploited because they are cheap and simple to perform but that unless properly locked down in place they might be the largest vulnerability in a system (OWASP, 2021). Many of us use weak passwords or leave them in a place that the authentication system can notice and this creates a real issue in situations where the authentication system does not provide strong password policies. Unrelated to usability, research carried out in 2015 by National Cyber Security Centre in the UK revealed that users are divided on whether the use of solid passwords was vital. There are those who regard them as being necessary, and those who decry that the rigorous standards are an inconvenience. However, unless standards are established, humans have a tendency of selecting simple-to-crack passwords.

Reinforcement of password policies is the first reason because most of us still use passwords which are simple and can be easily guessed. To affirm this, the NCSC (2021) not only confirms that the use of such things as 123456, password, and qwerty (just to name a few) is one of the most prevalent options that can be broken after a second. Some of the other commonly used passwords include simple names, sporting teams, and usual words in a dictionary. Without any rules whatsoever, it becomes evident that users are just putting their accounts into a jeopardy by choosing insecure passwords. Second, the use of poor passwords contributes to leaving the resources that we require to be accessed. Sensitive data or functionality is made available in any of the systems via the accounts. In case of a weak credential compromise, the even low-privileged accounts can be used against the servers by attackers to inflict negligent harm. E.g., access to an administrative portal may enable a person to give a chance to edit the records, destroy databases, or steal the data about the clients and lead to gigantic operational and reputational consequences (Cisco, 2022).

As we have just observed, cyberattacks can hit even small structures in case they fail to implement a stringent password policy. The fact that users choose weak, easy to remember credentials virtually endangers account confidentiality and increases the chances of attackers succeeding by a large margin. This is why linking and subscribing to your strong password policies is an uncompromising preventive condition of any university log in system.

The screenshot shows a web interface for a 'Membership System'. At the top, there is a dark header with the title 'Membership System'. Below it is a navigation bar with links: 'Home', 'Account' (highlighted in blue), 'Member', and 'Club'. The main content area has a light yellow background. On the left, there is a green 'Back' button. Below it, a yellow box contains the text 'Account / Update'. The main form area is light gray and contains two input fields: 'Username' with the value 'new_user' and 'Password' with a single dot. Below these fields is an orange 'Save Changes' button.

2.1.2 No Password Recovery:

Today, the average user of internet devices has hundreds of online accounts to operate and this figure is only rising. Of course, having so many logins is that people will not remember all of their passwords. Research indicates that there are millions of requests to reset a forgotten password each year, and a good number of users state that they forget their qualifications regularly (NCSC, 2015). This is only evidence that a secure password- reset system is not a luxury anymore; it is a necessity in any modern day login system. A lack of password recovery option may leave the user forever locked out of their accounts, whether it be a personal platform such as social media, key service such as online banking, or daily application such as the website of an e-commerce platform.

The screenshot shows a web interface for 'Administrator Login'. It features a light blue background for the login form. The form has two input fields: 'Username' and 'Password'. Below these fields is a 'Login' button with a right-pointing arrow icon. The entire form is enclosed in a light blue box on a yellow background.

Figure 2 No 'Forgot Password' option – Original System

2.1.3 No Authentication Factor in Registration Form:

In the course of our review, we have learnt that the administrator registration form was excessively simple. It did not need an additional information that can be used later in the authentication or identity check. It basically only required an username and a password. Such minimalism is a problem during efforts to verify the authentic ownership of an account. Use password recovery as an example--there is no way that the system can help verify that the requesting person is who it is supposed to be, or someone out to cause trouble. Figure X illustrates Administrator Registration page in case of the current implementation.

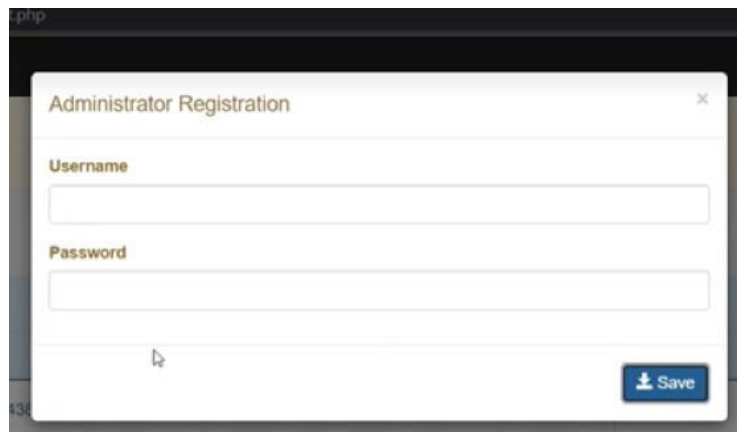
A screenshot of a web browser window displaying a form titled "Administrator Registration". The form has a title bar with a close button (X). It contains two input fields: "Username" and "Password", both with labels above them. Below the password field is a blue button with a white "Save" label and a small icon. The background of the browser window is dark.

Figure 3Admin Registration Screen - original system

2.2 Possible Cyber Attacks:

A cyberattack can simply be defined as an intentional and offensive invasion by a hacker or a cell of interrupting with the computer of another user or organisation. The central aim is to intrude into the system unauthorised, disorient the manner in which things are operated or steal valuable information in order to get a profit or competitive advantage. (Cisco, 2022).

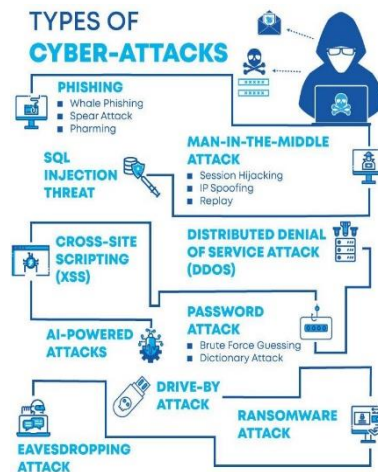


Figure 4Types of Cyber Attacks

2.2.1 Password Attack

To crack passwords, simply stated, means trying to risky to crack the unknown or forgotten password and obtain access to the system or the network and other types of resources with bad intentions (that can be misuse of these resources). What you learn during cracking can be utilized in activities as financial fraud, leaked identity or even leaked private information. Depending on the type of attack, attackers may make use of special software and automate the guessing process either by once trying any possible combination (a brute-force attack) or by searching a known list of common words and patterns (a dictionary attack). These two techniques remain the most widely used techniques of approaching the cracking of passwords. (OWASP, 2021)

2.2.1.1 Brute Force Attacks

In a brutality attack, the login passwords are reached by guessing. By trial and error, hackers induct their fingers to find the correct one. These attacks are executed using brute force which simply implies that such attacks apply undue pressure to infiltrate the private account. Although this method is pretty old-school, it is, however, common and convenient to hackers. Each password may require only several seconds - or it may require years - to crack, depending on the length and complexity of the password.

2.2.1.2 Dictionary Attacks:

Dictionary attacks merely imply that the attacker will simply fling all the words he or she can think of at the system, right? Practically, they are aiming at the words most apt to be used as passwords by the people. The rules of passwords differ greatly and as such, users tend to modify these words-say, by adding numbers or other symbols, simply to make them easy to remember. It is the same maneuver with a dictionary attack: the attacker initially begins with a standard dictionary and then applies a set of mangling rules to each associating an output to a sequence of new guesses in a swarm. As an example, you may add a number two to a word. It is more likely to work quicker than brute-force and is usually the initial move that you use, although there is only so many mangling rules that you can use, particularly when combining them on the top of each other. Suppose you put at the conclusion of every word the sum of two digits: you are multiplying the conjectures a hundredfold. The most difficult in this entire process is the selection of the mangling rules. Each rule you insert expands the attack space by much on the condition that the dictionary is already large.

2.3 Phishing Attack:

Phishing is a type of social engineering attack commonly used to steal sensitive information such as login credentials or financial details. In such attacks, a malicious actor impersonates a legitimate organization and deceives victims into clicking on links or opening fraudulent messages. Once an account is compromised through phishing, immediate action must be taken, most importantly by resetting the affected password to block unauthorized access. However, in the basic admin login system under review, no password recovery mechanism was provided, meaning users would have no secure way to reset their credentials. This limitation makes the system more vulnerable to phishing-related compromise, as attackers could continue exploiting stolen credentials without restriction. Implementing a secure password reset feature is therefore critical to reduce the risks posed by phishing attacks.

2.3.1 Account Takeover (ATO) Attack:



Figure 5 Account Takeover Attack

An account takeover (ATO) attack, as the name implies, occurs when a legitimate account gets under the control of hackers whose intentions are dubious. In effect, the attacker draws up legitimate login credentials, and begins impersonating the legitimate user. As the boom of digital platforms goes on, all people; students and companies are both caught in the gate of such attacks. According to a 2022 report of Cisco, sensitive information may be swiped, a system may be plunged into dysfunction, or other malicious actions may take place using hacked accounts. IT departments, financial groups and even senior executives are particularly vulnerable since their account tends to open key information, funds and network equipment. The initial system of admins login was an easy target as there was no heavy protection in the element of account takeover attacks. The level of threat begins with phishing-based attacks and goes up to fraudulent activities and unauthorized access to confidential documentation. In addition, since the affected account may have greater privileges, the attackers may increase their efforts and start exfiltrating data, carrying out BECs, or otherwise interfering with the internal system. Therefore, the implementation of more powerful authentication and secure password-recovery practices is important to remain safe against ATO.

3 REQUIREMENTS FOR DESIGNING A SECURE SYSTEM

3.1 Password Strength Checker

The unawares way of averting weak passwords danger is ensuring that individuals do not append it even in the beginning. Most secure systems provide detailed stepwise instructions to encourage folks to create a strong password, and auth guidelines encourage complexity by requiring the inclusion of numbers, capital and lower case,

and special characters (OWASP, 2021). Essentially, two broad strategies are available to us, namely educating people on the proper habits, and locking them into the rigid pass-rules. In this project we were using the latter - we used to implement hard policies each time someone signs-up or changes a password. In that manner no one should be able to cheat the system and you will know you will make the security bar with your credentials. Consider a password policy as a rule book whereby all your accounts are required to use a strong albeit cumbersome password. What's the goal? - Make users use passwords that cannot be bruteforced or guessed. - Make passwords difficult to the audience, but not so difficult as to be useless. - Offer clear guidance on storing and rotating passwords safely. Within the upgraded admin entry point, the policy as designed was to use digits, symbols, and lower case and upper-case letters. In short, none of those weak, obvious passwords slip past the check at signup or update. We make the user use stronger creds that can resist an attack of brute-force and dictionary, unlike other systems that allow you to use easier words and phrases (OWASP, 2021). As an example, a username such as admin123 does not qualify, whereas a username such as Ad!mn.2025 does. Efforts such as restrictive measure reduce the probability of compromise since attackers do not have ready guess to play around. This may irritate a few users, but the resulting security payoff far outstrips that irritation most of the times. We experimented with how to quantify password use, and prompt users to use safe but convenient passwords. It has been reported that any easy default has been chosen by people since it has not been forced out of other options (OWASP, 2021). In response to this, our adm system makes it obligatory to use upper and lowercase, figures and symbols when creating or modifying passwords. Ok a snazzy Ad!mn52025 gets through and Admin123 is rejected. Even though there are some grumbleings about the most difficult rules frustrating the users, easy passwords head the list of breach causes (NCSC, 2015). By locking these requirements we decrease brute force/dictionary hits. The following figures demonstrate the password strength feature: weak passwords will

automatically be denied, and users will be suggested to select stronger passwords and secure the principle of login security policy further.

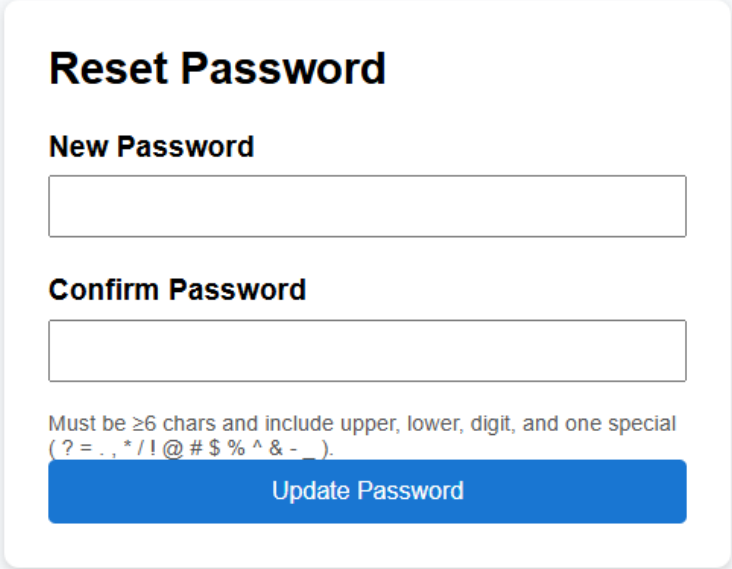
A screenshot of a 'Reset Password' form. The form is white with rounded corners and a subtle shadow, set against a light blue background. It features a title 'Reset Password' in bold black text. Below the title are two input fields: 'New Password' and 'Confirm Password', both with thin grey borders. Under the 'Confirm Password' field, there is a small grey text block stating: 'Must be ≥6 chars and include upper, lower, digit, and one special (? = . , * / ! @ # \$ % ^ & - _)'. At the bottom of the form is a solid blue button with the text 'Update Password' in white.

Figure 6Figure 15Reset Password form accessed via secure token

The sophisticated system has made its users comply with unique password rules during enteries and password resets. The Reset Password page that you will see below requires at least six characters, and you are required to use an uppercase and lowercase letters with numbers and any of the special character. This prevents individuals who highly value a weak password such as admin123 to go to a strong password such as Ad!mn#2025.

3.2 Password Recovery

It is exceedingly crucial to have the tool of the password recovery to prevent the loss of usernames and passwords. Ideally, a good password must contain a combination of uppercase and lowercase letters, numbers and special symbols and must not be less than six characters in need to ensure that accounts cannot be shut by dropping out. But, to tell the truth, one also cannot recall all those fancy combinations. The text-based login woes exist not only with the

technical side but also within the very nature of the way we humans store information as well as the way designs are made, and experience is human. The National Cyber Security Centre of the United Kingdom in a study conducted in 2015 demonstrated that many of us find it challenging to retrieve, including remembering some of the hard words, passwords created, hence resorting to reuses or simplifying them. The 2021 report by OWASP also explains that ineffective password practices remain one of the greatest vulnerabilities in web applications. All these make it clear why we should have a good password recovery system- after all, we people are destined to lose or mismanage our logins at some time.

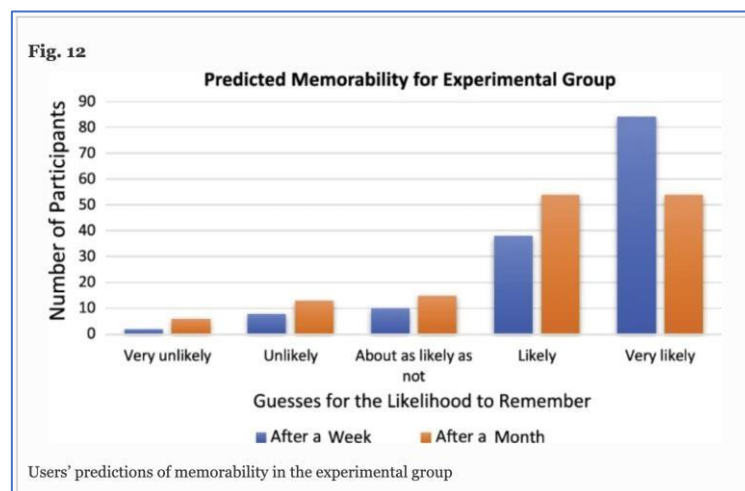
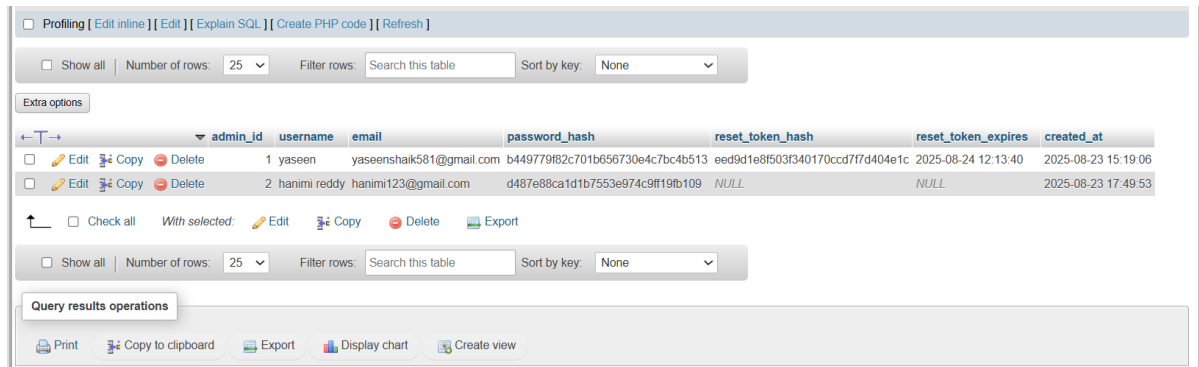


Figure 7 User's prediction of memorability in the experimental group

The initial system of the admin login did not have any remedy in case of password loss. That resulted in a huge security and usability issue. In case an account was compromised or an account holder forgot the password, the password could not be reset safely. It could only change it using brute force on the server or in MySQL database. That is not best practice and risky indeed. It may reveal confidential data and interfere with other functionality. In order to rectify that, we have implemented an email-based password recovery. At the bottom of the login page, there is a they forgot password button, through which reformers can request a reset link. They maintain the registered email and the system uses it to compare it with the system database. In the case that it is found, we dispatch a secure reset link comprising of a one-time token and an expiry period to the verified email. In this manner Ldap only permits authentic admins to change passwords. The application requires SMTP server in order to deliver the link. This is done with the PHPMailer library in PHP. We created a special email account on the server, hence the emails which are sent by the reset arrive safely on the page of the admins. Once the

connection is made and the password is changed, we delete that token on the database in order to prevent reuse and maintaining the security level high.

3.3 Encryption:



The screenshot shows a database management tool interface. At the top, there are tabs for 'Profiling', 'Edit inline', 'Edit', 'Explain SQL', 'Create PHP code', and 'Refresh'. Below this, there are controls for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (a search box), and 'Sort by key' (set to None). An 'Extra options' button is also present. The main table has columns: 'admin_id', 'username', 'email', 'password_hash', 'reset_token_hash', 'reset_token_expires', and 'created_at'. There are two rows of data. The first row has admin_id 1, username 'yaseen', email 'yaseenshaik581@gmail.com', a long hexadecimal password hash, a long hexadecimal reset token hash, a reset token expiration date of '2025-08-24 12:13:40', and a creation date of '2025-08-23 15:19:06'. The second row has admin_id 2, username 'hanimi reddy', email 'hanimi123@gmail.com', a long hexadecimal password hash, a NULL reset token hash, a NULL reset token expiration date, and a creation date of '2025-08-23 17:49:53'. Below the table, there are buttons for 'Check all', 'With selected', 'Edit', 'Copy', 'Delete', and 'Export'. At the bottom, there are buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

| admin_id | username | email | password_hash | reset_token_hash | reset_token_expires | created_at |
|----------|--------------|--------------------------|----------------------------------|----------------------------------|---------------------|---------------------|
| 1 | yaseen | yaseenshaik581@gmail.com | b449779f82c701b656730e4c7bc4b513 | eed9d1e8f503f340170ccd7f7d404e1c | 2025-08-24 12:13:40 | 2025-08-23 15:19:06 |
| 2 | hanimi reddy | hanimi123@gmail.com | d487e88ca1d1b7553e974c9ff19fb109 | NULL | NULL | 2025-08-23 17:49:53 |

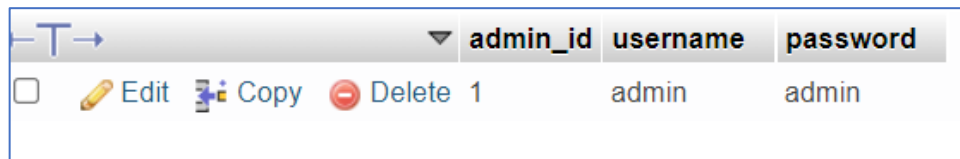
Figure 8 Database records with hashed passwords.

Therefore, the system of passwords of the admin login entails hash passwords calculated using MD5. They are all hashed to 128 bits meaning that the database admins do not necessarily know the actual passwords of other users in the database. I do not mean that MD5 is the most secure option, bcrypt or Argon2 would be much more secure but, as a part of this assignment we are using MD5, which is our advice to consider a more secure algorithm in any actual implementation.

3.4 Access Control:

In practice, access control of the original system was rather poor - anyone was able to pop in a username and a password to spin up a new administrator account. That simplicity also made it difficult to ascertain the people in the loop and it opened up a loophole to mishandled logins. To make it more secure, therefore, the sign-up flow should be beefed up a notch ask them to

provide a valid email on top of the standard creds collected, and then use that address to check identity when others have to recover passwords or to adjust account settings.



| | admin_id | username | password |
|--|----------|----------|----------|
| <input type="checkbox"/> Edit Copy Delete | 1 | admin | admin |

Figure 9Figure 11: Admin Database Table in the Proposed System

The existing database table does not literally contain the information on the amount of user needed that can be uniquely matched to administrators. Due to that, the threat of duplication or confusion between rings up is a genuine possibility in case more than one person register it with the same name and password. In order to rectify it, a unique accounts identifiable to beneficial administrators must be provided to each such account, with such a unique identifier differentiated between two or more accounts. Such a basic modification will reduce the number of collisions and make the system more trustworthy. An email address at sign-up is not merely a unique identifier it provides a flow of email based password recovery. That ensures admins can easily reset their credentials and the overall authentication process gains in security and reliability.

4 Implementation and Code Explanation of the selected security method

4.1 Software Configuration:

The web-based system of managing the membership was set and run on an open-source XAMPP server. It is cross-platform and includes a number of generally Apache-made packages, in particular, the HTTP server and MySQL database.

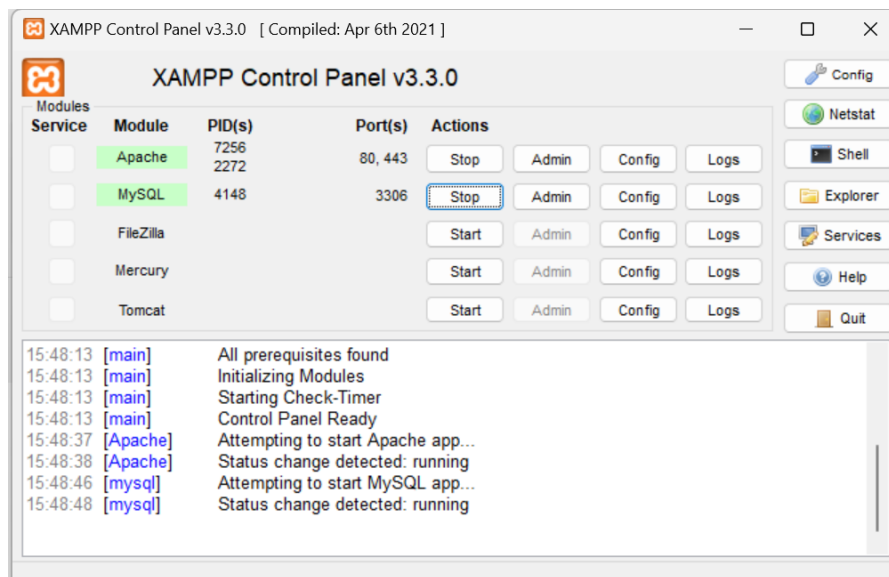


Figure 10XAMPP Control Panel

The interpreter language used for the system was PHP. To run the system an Apache HTTP server, need to be hosted locally along with its configured database.

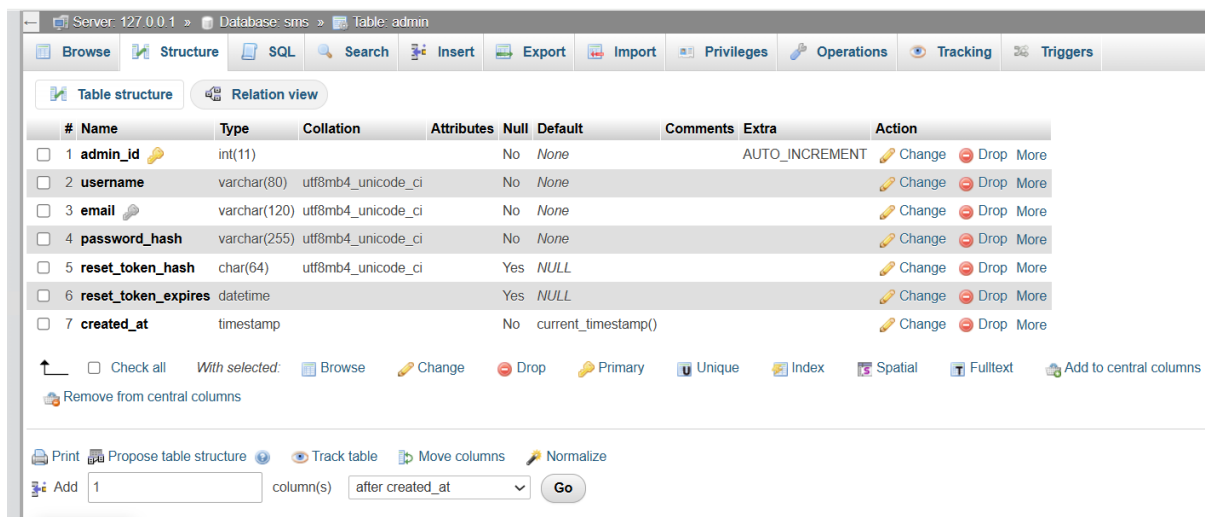


Figure 11phpMyAdmin

Backups: The membership management database was imported into the system repo by phpMyAdmin which is included with XAMPP. The sms.sql file was imported successfully into the local MySQL server and therefore, the system has been given the capability to store and manage the administration account details safely.

This project is started with an HTTP# localhost which refers to the directory of the project. By hitting the said URL, Apache will load the PHPs and everything that depends on them.

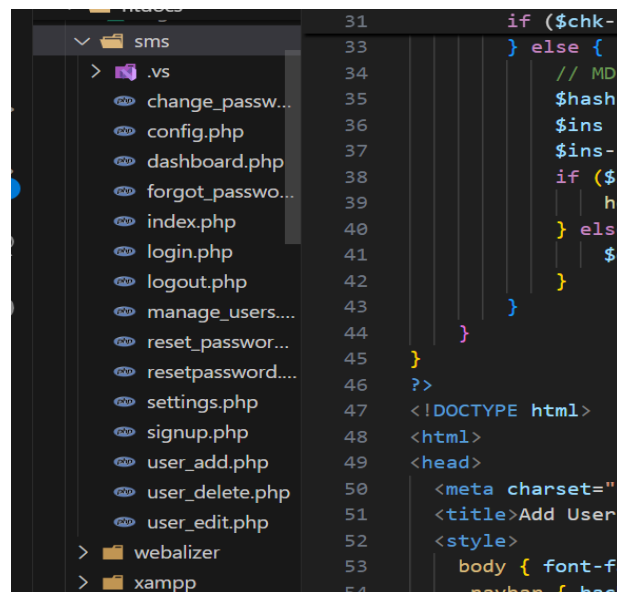


Figure 12Project Directory

We installed PHPMailer because we need to send secure email when people will just forget about their passwords. We do have certain scripts within the project folder such as forgotpassword.php, resetpassword.php, which do the job of creating reset tokens and dispatching recovery emails to the admins. PHPMailer configuration to convey to an SMTP server, with the intent of these password reset links finding their way to the inboxes of the appropriate users, in relative safety..

4.2.1 Source Code Explanation:

Signup.php was developed as a new php file, which has been incorporated in the current system to allow the registration of the administrators. I have done this by connecting the signup.php file with the preexisting account.php file. As it is illustrated in the snapshot, I made an account.php an href command (lines 41-42) to bridge between the two forms. Also, there was the Add New command, meaning that, once an administrator chooses to create a new account the system automatically redirects him to the newly created sign-up page. The last variant worked with a pop up window as a form of registration, the new design opens a specific page, which has a better registration workflow, clearer and more understandable.

```

39      <div class = "col-md-12 well">
40
41          <button type = "button" class = "btn btn-success" data-toggle="modal" data-target="#myModal">
42              <span class = "glyphicon glyphicon-plus"></span><a href="signup.php"> Add new </a></button>
43      <br/>

```

Figure 13Source Code – 1

Then we approached SQL queries to query and verify data in the database. An example is the verification of lines 11 of the code on the registration page, which checks whether the email entered in the record is present in the table of designation (the table of administration). The column of the adventurer is the admin_id which is the primary key making each row unique and easy to retrieve when we make our queries. The database structure is sketched in Figure X below, where tables such as admin and the club among others were created to hold the data you feed to the web application.

| | admin_id | username | email | password_hash | reset_token_hash | reset_token_expires | created_at |
|---|----------|--------------|--------------------------|----------------------------------|----------------------------------|---------------------|---------------------|
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 1 | yaseen | yaseenshaik581@gmail.com | b449779f82c701b656730e4c7bc4b513 | eed9d1e8f503f340170ccd7f7d404e1c | 2025-08-24 12:13:40 | 2025-08-23 15:19:06 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 2 | hanimi reddy | hanimi123@gmail.com | d487e88ca1d1b7553e974c9ff19fb109 | NULL | NULL | 2025-08-23 17:49:53 |

Figure 14 Tables within 'sms' database

Within our classroom organization, the table of administrators is called the table of admins, and contains all registered administrators, with fields (username, email, passwordhash) and other columns to provide password reset features (resettokenhashes, resettokenexpires). We consider the email column as an authentication factor and therefore it contains a unique address and thus it cannot be used to create two accounts with similar address. In the registration, the system constantly infers about whether the one that is attacked is already held in the database, and in this case, we ask the user to provide another, valid email address. After the creation of a successful new account, the database successfully inserts the values of username, email and password hash and rings a createdat timestamp. Not only is this method used to strengthen the security of the account, but also assists a trusted email-driven password recovery system.

```

11 $sql=mysqli_query($con,"select admin_id from admin where email='$email'");
12 $row=mysqli_num_rows($sql);
13 if($row>0)
14 {
15     echo "<script>alert('Email id already exist with another account. Please try with other email id');</script>";
16 } else{
17     $msg=mysqli_query($con,"insert into admin(username,email,password) values('$username','$email','$password')");
18
19     if($msg)
20     {
21         echo "<script>alert('Registered successfully');</script>";
22         echo "<script type='text/javascript'> document.location = 'account.php'; </script>";
23     }

```

Figure 15 Source Code – 2

In the front-end, the system interface had been designed by taking templates provided by Cloudflare that layout the UI components (Figure X). Since the primary aim of this assignment is to peel back the security front end, we are not going to go into much visual design analysis. The single element that is relevant, however, is the `checkpass()` function which begins on line 39 of the code.

However, it verifies that the Password and Confirm Password typed should be equal and administrators cannot continue registering or a password reset until they are exactly equal. Identical

```
26  ?><!DOCTYPE html>
27  <html lang="en">
28  <head>
29    <meta charset="utf-8" />
30    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
31    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
32    <meta name="description" content="" />
33    <meta name="author" content="" />
34    <title>Admin Registration System</title>
35    <link href="css/styles.css" rel="stylesheet" />
36    <script src="https://cdn.jsdelivr.net/npm/font-awesome@5.15.3/js/all.min.js"
37      crossorigin="anonymous"></script>
38    <script type="text/javascript">
39      function checkpass()
40      {
41        if(document.signup.password.value!=document.signup.confirmpassword.value)
42        {
43          alert(' Password and Confirm Password field does not match');
44          document.signup.confirmpassword.focus();
45          return false;
46        }
47        return true;
48      }
49
```

Figure 16Source Code – 3

Additionally, code blocks 69-75 are used to ensure proper styling of the screen. Moreover, since we are also using JavaScript, the webpage becomes scalable. That is even when the window size is decrease, the output adjusts accordingly. Same code chunk is applied to each field.

```

69 <div class="col-md-6">
70 <div class="form-floating mb-3">
71 <input class="form-control" id="username" name="username" type="text"
72 placeholder="Enter your Username" required />
73 <label for="inputUsername">Username</label>
74 </div>
75 </div>
76
77
78
79 <div class="row mb-3">
80 <div class="col-md-6">
81 <div class="form-floating mb-3">
82 <input class="form-control" id="email" name="email" type="email"
83 placeholder="snehadevi@gmail.com" required />
84 <label for="inputEmail">Email address</label>
85 </div>
86 </div>
87

```

Figure 17 Source Code – 4

```

<label>Username</label><br>
<input name="username" required><br><br>

<label>Email</label><br>
<input name="email" type="email" placeholder="you@example.com" required><br><br>

<label>Password</label><br>
<input id="pw" name="password" type="password"
required
pattern="(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\w\s]).{8,}"
title="≥8 chars with upper/lower/digit/special"><br><br>

<label>Confirm Password</label><br>
<input id="cpw" name="confirm" type="password" required><br><br>

<button type="submit">Create Admin</button>

```

Figure 18 Source Code – 4

The placeholder attribute, in this section of this source code, is a means to tell the expected input format. To illustrate this, the form in the email field in Figure X was not filled in with a legitimate email address, but instead, a print display of the user's email address should be added to the email field such as the example; you@example.com. The forces of the usernames, emails, passwords and confirm passwords have also been required, making these inputs obligatory at the time of registration. Also, password and confirm password fields need to be made to match in order to submit it. The design will avoid complete registration and enhance the trustworthiness of the Admin Registration System. In lines 91-92 the password strength checker is brought to reality. Pattern identifier, as illustrated beneath, is used to determine the contents of the string input in the password text box. The following are the requirements:

Password should contain

- at least one special character (? , = , . , * , /)
- at least one integer
- at least one lowercase letter
- at least one uppercase letter
- a combination of at least 6 characters

```
88 <div class="row mb-3">
89 <div class="col-md-6">
90 <div class="form-floating mb-3 mb-md-0">
91 <input class="form-control" id="password" name="password" type="password" placeholder="Create a password"
92 pattern="(?!.*\d)(?!.*[a-z])(?!.*[A-Z]).{6,}"
93 title="at least one number and one uppercase and lowercase letter, and at least 6 characters" required/>
94 <label for="inputPassword">Password</label>
95 </div>
96 </div>
97
```

Figure 19Source Code – 4

Thus, if the contents of the password do not match the requirements stated by ‘pattern’ then it will not be accepted. An alert will then be given, as seen in line 93. This is important to inform the user how to rectify his ‘mistake’.

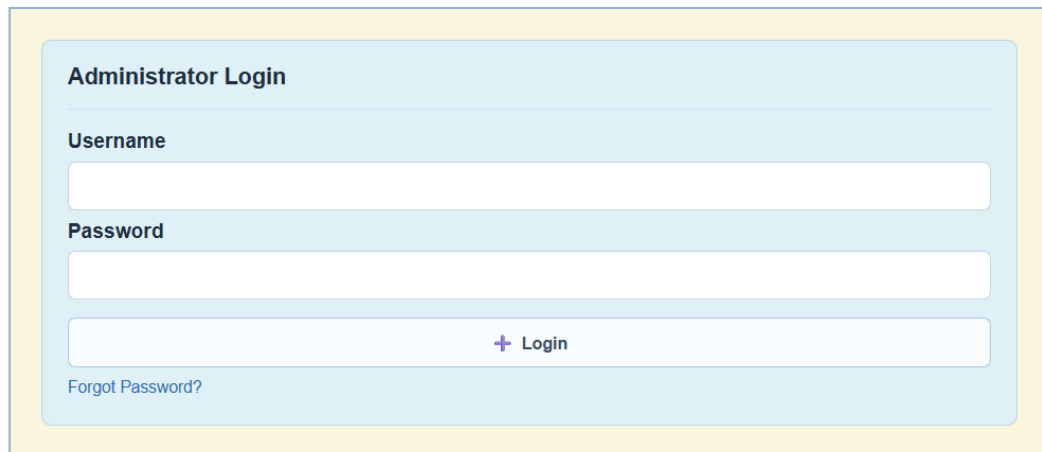
4.2.2 Results:

As it can be seen in the Figure 22, the whole Admin registration screen was updated with the aim of making it more secure and user-friendly. In comparison with the original system, two additional fields were added – email address and confirm password.

Login Page :

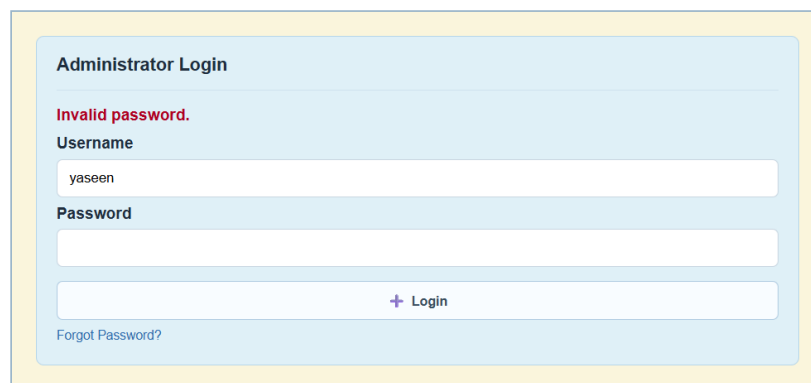
Proves that authentication is enforced:

The login form enforces validation so users must provide both username and password.

Login PageA screenshot of the Administrator Login page. It features a light blue rectangular form with a yellow border. At the top, the text "Administrator Login" is displayed. Below this, there are two input fields: "Username" and "Password". A third input field at the bottom contains a purple plus icon followed by the text "Login". A link labeled "Forgot Password?" is located at the bottom left of the form.*Figure 20 Administrator Login Page – entry point of the system*

The Admin Login Management System begins at the login page the landing point where you run into the first check-point. Before you are allowed into the dashboard, you must input a valid username, password, which secures the entire authentication process and only allows authentic administrators to enter. And by the way there is a Forgot Password link above, so you can readily restate your log-in credentials.

4.2.3 Login with Invalid Credentials:

A screenshot of the Administrator Login page showing an error state. The form is light blue with a yellow border. At the top, it says "Administrator Login". Below this, the text "Invalid password." is displayed in red. The "Username" field contains the text "yaseen". The "Password" field is empty. At the bottom, there is a purple plus icon followed by the text "Login". A link labeled "Forgot Password?" is at the bottom left.*Figure 21 Administrator Login Page – incorrect username or password attempt.*

You enter the wrong user name or password and the system will show you an error that the credentials are invalid. This prevents unauthorised entry and assists you to retry with the correct details. On the internal checks the system verifies your entered credentials with the database securely before admitting you in.

4.2.4 Successful Login:

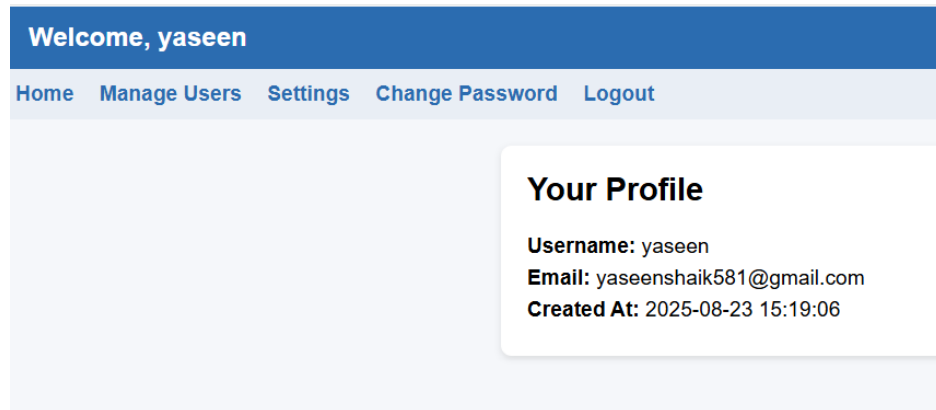


Figure 22 Administrator Login Page – successful login redirects to Dashboard.

Upon typing in legitimate credentials, the user logs in and is redirected to the main dashboard- it is essentially where the user session is launched. This is the initiation of the active work period of the admin. The login is structured in a way that ensures authenticated administrators are only allowed to get access to sensitive system functions such as user management, settings etc.

4.2.5 Manage users

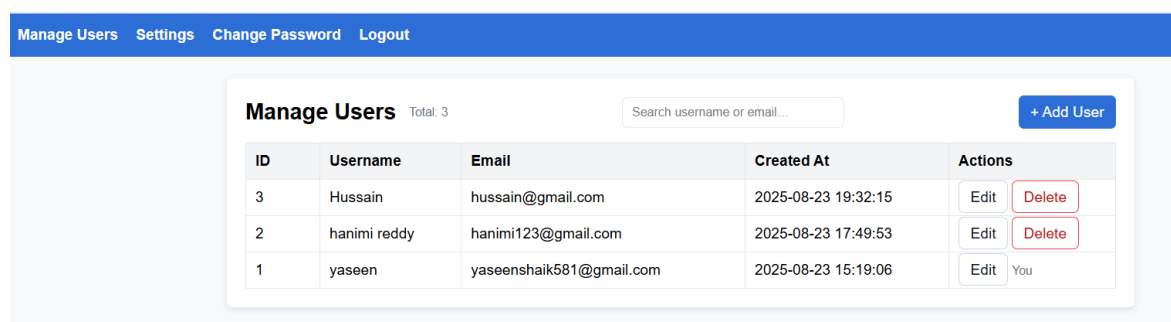
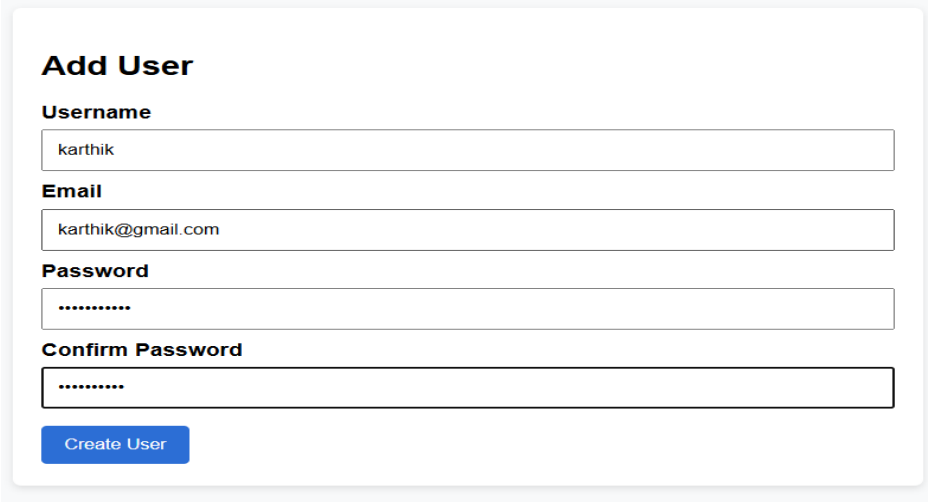


Figure 23 Manage Users page listing all administrators stored in the database.

In the table presented on this page we can see all the accounts of admins including IDs, their

usernames and email addresses along with the date the accounts were created. The data is an in-memory fetched option in the database. Having this opinion in mind, admins could correct or delete the account of other users. It, fundamentally, demonstrates the Read part of CRUD.

4.2.6 Manage Users – Add User:

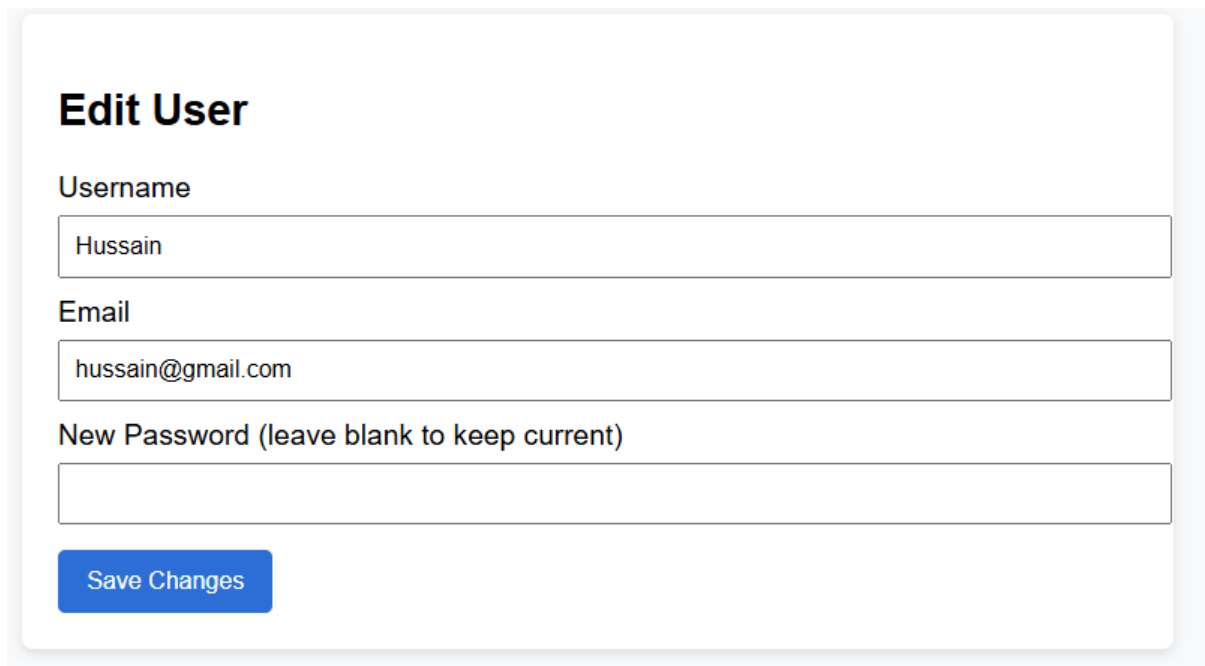


The image shows a web form titled "Add User". It contains four input fields: "Username" with the value "karthik", "Email" with the value "karthik@gmail.com", "Password" with masked characters ".....", and "Confirm Password" with masked characters ".....". Below the fields is a blue button labeled "Create User".

Figure 24 Add User form for creating new administrator accounts

The Create phase of CRUD allows the add of new users who are registered by filling in a form as on registration. The system has checks including, email format, password strength among others before entering the data in the database. This helps to see how input verification is important prior to any write operation.

4.2.7 Manage Users – Edit User:



Edit User

Username

Email

New Password (leave blank to keep current)

[Save Changes](#)

Figure 25 Edit User form to update administrator details.

Note: With our Edit feature, we are able to edit user information including usernames or email addresses. It provides us with a flexibility of managing accounts and maintains the information current. This is more or less the “Update” aspect of CRUD.

4.2.8 Manage Users – Delete User:

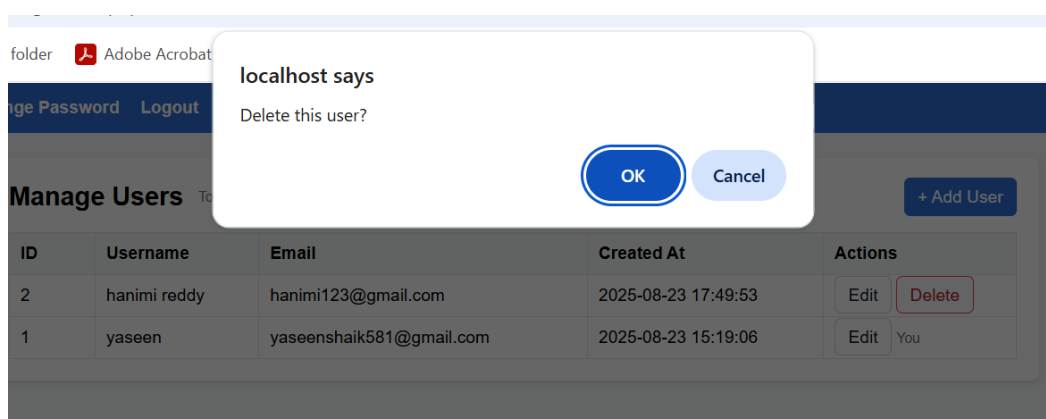


Figure 26 Delete User option with confirmation prompt

The Delete button allows administrators to remove users from the system. A confirmation

message is displayed before deletion to prevent accidental data loss. This action completes the Delete part of CRUD.

4.2.9Forgot Password Page :

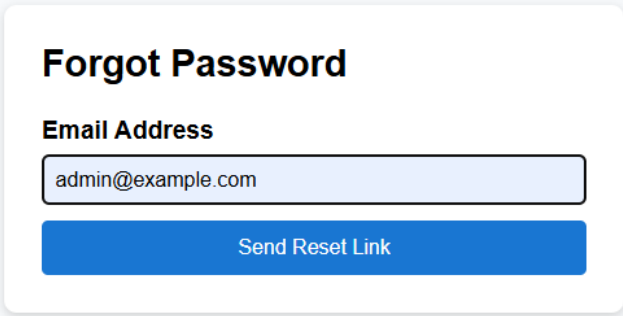
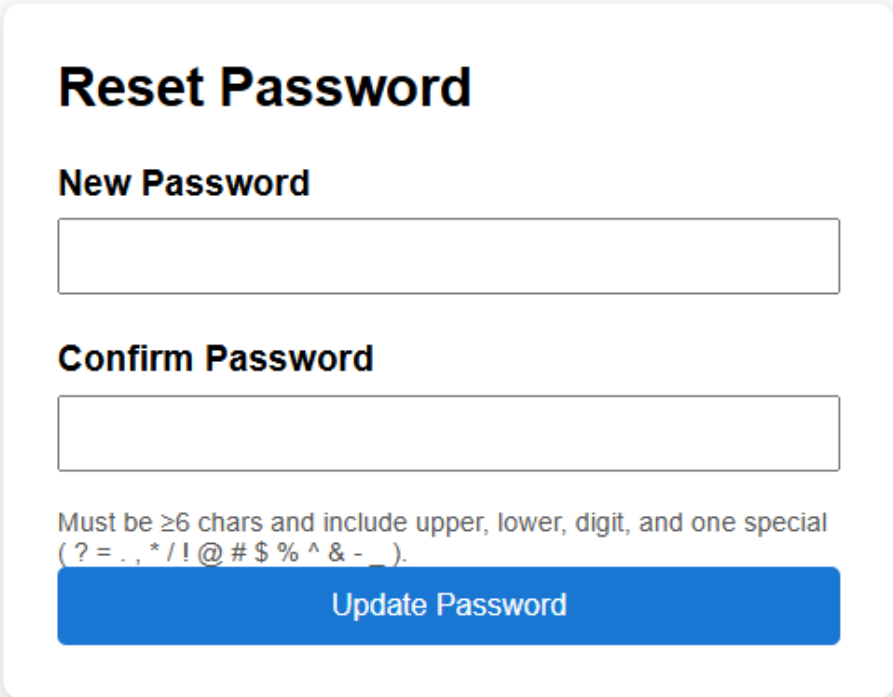
A screenshot of a web form titled "Forgot Password". Below the title is a label "Email Address". Underneath the label is a text input field containing the email address "admin@example.com". Below the input field is a blue button with the text "Send Reset Link". The form is centered on a light gray background.

Figure 27Forgot Password form where administrators enter their registered email address.

When an admin loses their password, all they do is press the reset button and paste their enrolled email in. The system then does two verifications and spits out a secure reset token, and does not reveal the password on the first measurement. In contrast, it transmits a reset connection - preventing un-authorized access by everybody..

4.2.9 Reset Password Page (via token link):



The image shows a 'Reset Password' form. It has a title 'Reset Password' in bold. Below it are two input fields: 'New Password' and 'Confirm Password'. Below the 'Confirm Password' field is a text requirement: 'Must be ≥6 chars and include upper, lower, digit, and one special (? = . , * / ! @ # \$ % ^ & - _)'. At the bottom is a blue button labeled 'Update Password'.

Figure 28 Reset Password form accessed via secure token

A separate token lands the admin in the reset page. The token becomes stored in the database with an expiry time to deter abuse. The new password can safely be set by the admin on this page..

4.2.10 Successful Password Reset:

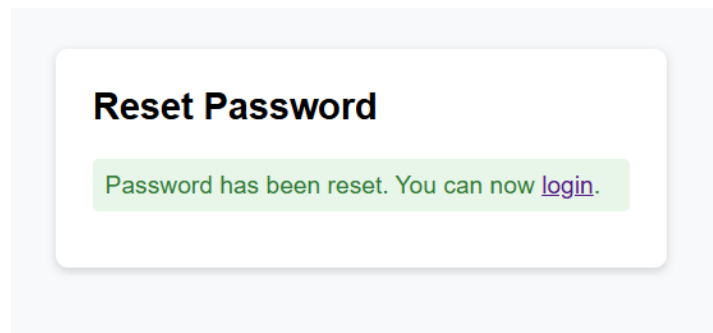


Figure 29 Confirmation of successful password reset

After a good password and confirmation better than your old password, system replaces the old data with a new hash. Thereafter, the user will have his or her changed password to log in. This additional step enhances the general security at the same time making them user friendly.

4.2.11 Change Password Page:

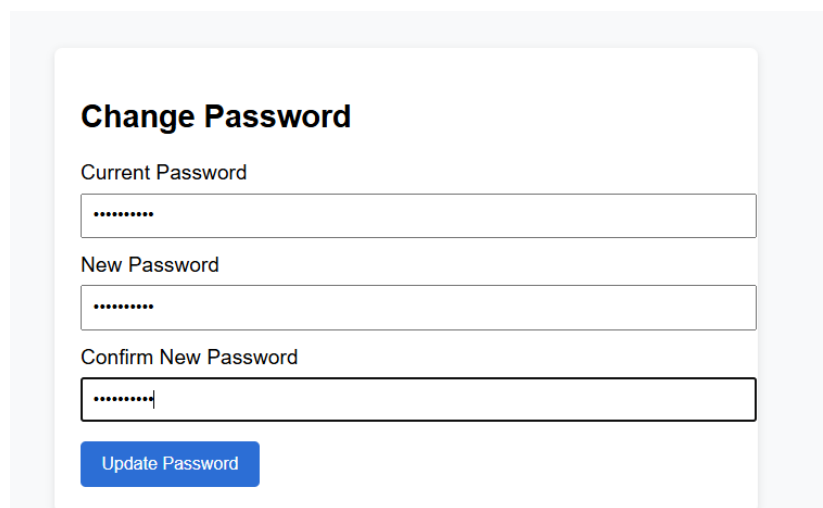
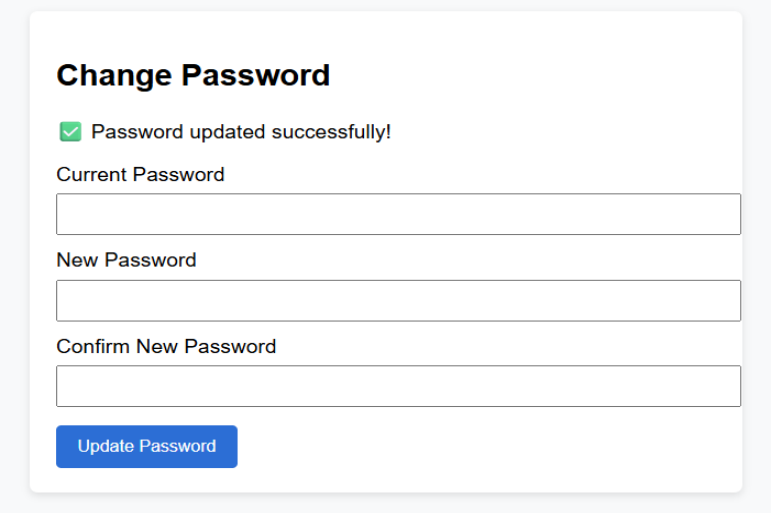
A screenshot of a 'Change Password' form. The title 'Change Password' is in bold black text. Below the title are three input fields: 'Current Password', 'New Password', and 'Confirm New Password'. Each field contains a series of dots representing masked text. At the bottom of the form is a blue button labeled 'Update Password'.

Figure 30 Change Password form where administrator updates their credentials.

Basically, so there is a Reset Password page where the admins as logged-in users can change their old passwords to the new password. The process is relatively straightforward: enter the existing password then enter a new password that you have not used before; you are required to repeat this. This will prevent cases of sneak thieves coming in and compromising your password without due authority. In addition, the system is relatively adamant about the strength of a password. It must be six characters with one upper and one lower character, one digit and one recognized special character. In case your password does not pass such checks or the confirmation does not exactly match, we do not store it and only tough passwords are saved.

4.2.12 Successful Change Password Message:




The image shows a web form titled "Change Password". At the top, there is a green checkmark icon followed by the text "Password updated successfully!". Below this message, there are three input fields: "Current Password", "New Password", and "Confirm New Password". At the bottom of the form, there is a blue button labeled "Update Password".

Figure 31 System confirmation of successful password change

On input of the right current password plus a strong new one the hash being generated by the system is inserted in the database. An prompt Success! banner will jump up in order to provide them with the feeling. They will henceforth have to use the new password to log in. It is simply like implementing security which forces admins to change passwords regularly.

4.2.13 Settings Page :



The screenshot shows a web form titled "Settings". It contains two text input fields. The first field is labeled "Site Name" and has a placeholder text "e.g. Student Management System". The second field is labeled "Admin Email" and has a placeholder text "e.g. admin@example.com". Below these fields is a blue button labeled "Save Settings".

Figure 32 Settings page where administrator can update system preferences

5 CONCLUSION

To sum it up, the assignment objectives were achieved. One up-grading in the web based membership system that I had applied was, primarily, to ensure it was less vulnerable to cyber-attacks. The initial system was flawed (particularly in terms of authentication and data protection) as we explained in Chapter 2. The current project concentrated on two important security aspects, which included implementation of strong password policy and inclusion of efficient password recovery service. Changes have certainly increased the overall security level in the system but this can be still improved. The Multi-Factor Authentication (MFA) may be incorporated into the work of the future to further secure the recovery integrated with MultiFaceted Authentication. In addition, addition of security questions during registration would enhance the process of identifying the identity and reduce cases of unauthorized access.

References :

Cisco. (2022). *What is a cyber attack?* Cisco. <https://www.cisco.com/c/en/us/products/security/what-is-a-cyber-attack.html>

Hooven, B. (2021, November 17). *The most common passwords of 2021 are still shockingly bad.* NordPass. <https://nordpass.com/most-common-passwords-list/>

National Cyber Security Centre (NCSC). (2015). *Password guidance: Simplifying your approach.* NCSC. <https://www.ncsc.gov.uk/collection/passwords>

OWASP Foundation. (2021). *OWASP Top Ten Web Application Security Risks – 2021.* OWASP. <https://owasp.org/Top10/>

Shay, R., Komanduri, S., Kelley, P. G., Leon, P. G., Ur, B., Vidas, T., ... & Cranor, L. F. (2010). Encountering stronger password requirements: User attitudes and behaviors. *Proceedings of the Sixth Symposium on Usable Privacy and Security*, 1-20. <https://doi.org/10.1145/1837110.1837113>

Ur, B., Kelley, P. G., Komanduri, S., Lee, J., Maass, M., Mazurek, M. L., ... & Cranor, L. F. (2015). How does your password measure up? The effect of strength meters on password creation. *USENIX Security Symposium*, 65–80.

Yıldırım, M., & Mackie, I. (2019). Encouraging users to improve password security and memorability. *International Journal of Information Security*, 18(6), 741–759. <https://doi.org/10.1007/s10207-019-00434-3>

PHPMailer. (2022). *PHPMailer – A full-featured email creation and transfer class for PHP.* GitHub. <https://github.com/PHPMailer/PHPMailer>

