COS10026 - Computing Technology in the Inquiry Context

Assignment 2 Report

Tran Thien Thao Vy
104991221

1. Introduction

In the previous assignment, we had the opportunity to build a website from scratch with HTML and CSS, now in order for multiple innovations, we also started implementing PHP scripts and SQL commands to improve our webpage and also to enhance data storage performance.

In this assignment, from a static tech portal webpage, which is used for a tech-job portal for the user to learn about a company, therefore being able to fill in an application form if interested. Their applications now should be stored into a database.

The object of this report is to summarize the process of a member when contributing to making a dynamic website and some notes on the sections of the website.
The structure of my report is distributed into 4 main parts:

1. The flow of the website and its key features
2. Security of the website
3. My contribution and reflection
4. Conclusion

The flow of the website and its key features

There are five pages in our website, including the Homepage (index.php), Jobs (jobs.php), Apply (apply.php), About (about.php), Enhancements (enhancements.php), PHPEnhancements (phpenhancements.php), Login(login.php). The web pages are linked by a navigation bar in every page with the menu items link listed.

1. **Homepage (index.php):**

This page includes the navigation menu and the footer, which will appear in all of the web pages through the website.
Description: The main change regarding headers, nav-bar and footer in which these features are now implemented in the code by the “inc.” files. This file helps to keep the consistency and efficiency in building the pages.

- Screenshots of footer inc. and how it is implemented into the page:

```
<footer class="footer">
  <ul class="quick-nav">
      <li>&copy; 2024 SwinTech Inc. |</li>
      <li><a href="index.php">Homepage</a></li>
      <li><a href="jobs.php">Jobs</a></li>
      <li><a href="apply.php">Apply</a></li>
      <li><a href="enhancements.php">Enhancements</a></li>
      <li><a href="about.php">About us</a></li>
  </ul>
</footer>
```

```php
<?php
    include_once "footer.inc"; //include the footer
?>
</body>
```

2. **Jobs (jobs.php):**

Description: There are the 4 position description boxes of the job openings, job description, salary ranges and other information.The aim of the jobs.php is to provide quick and brief information about the jobs available and keep the flow going by buttons forwarding to the apply.php page.

3. **Apply (apply.php):**

Description: Enable the candidates to apply for the available job openings. The main difference within the assignment 1 page is that the processing page after the applicant submitted their application is the “processEOI.php”, which is used for validating, sanitizing and forwarding data to the database established.

- Screenshot of the code spinnet:

```
rm method="post" action="processEOI.php" novalidate="novalidate">
 <main class="apply-form">
    <section>
    <p class="thank-you">Thank you for considering our company for your career aspirat
    <h1 class="h1-apply">Employment Application Form</h1>
```

4. **ProcessEOI (processEOI.php):**

Purpose: Sanitise/validate data and process it to the database.

Description: Before being inserted into the database, the data of the application form will be sent to the processEOI.php. If the information meets the required standards and is in the correct format, the processEOI.php will navigate them to a page with successful application submission announcements, if not there will be a list of errors to check.

This PHP script also sanitizes and validates input data such that data will be checked and secure the page from security concerns. The ‘processEOI.php’ script also navigates the successful application submission to the database for storage.

Some screenshots of the code where those functions are implemented:

- Connect to the database:

```php
$conn = @mysqli_connect(
    $host,
    $user,
    $pwd,
    $sql_db
);

if (!$conn) {
    die("Can not connect to the database: " . mysqli_connect_error());
}
```

- Validate data:

```php
if (empty($job_ref_num)) {
    throw new Exception("Job Reference Number is missing");
}

if (!preg_match("/^[A-Za-z0-9]{5}$/", $job_ref_num)) {
    throw new Exception(format_invalid_error("Job Reference Number", "Invalid Job Reference Number"));
}

if (!preg_match("/^[A-Za-z]{1,20}$/", $first_name) || !preg_match("/^[A-Za-z]{1,20}$/", $last_name)) {
    throw new Exception(format_invalid_error("First Name and Last Name", "Your First Name and Last Name shoul
    );
}

if (!filter_var($email_address, FILTER_VALIDATE_EMAIL)) {
    throw new Exception(display_invalid_error("Email", "Invalid email address"));
}
```

Screenshots of the successful data application and how it is stored in the database:

5. About (about.php):

Description: this page contains the group information, members information and our timetable.

6. Enhancements (enhancements.php):

Description: Apart from the fundamental sections, there are content boxes to present the enhancements notes like responsiveness, toggle nav-bar and hover buttons, which are the highlight of the previous assignment using HTML and CSS.

7. Login(login.php):

Description: The login page represents a web page for the admin of the website to access to the “manage.php” page for navigation over the application forms .The login section contains a form with input fields for username and password. Upon entering username and password, the form data is sent to “authenticate.php” for processing.

8. Authenticate (authenticate.php):

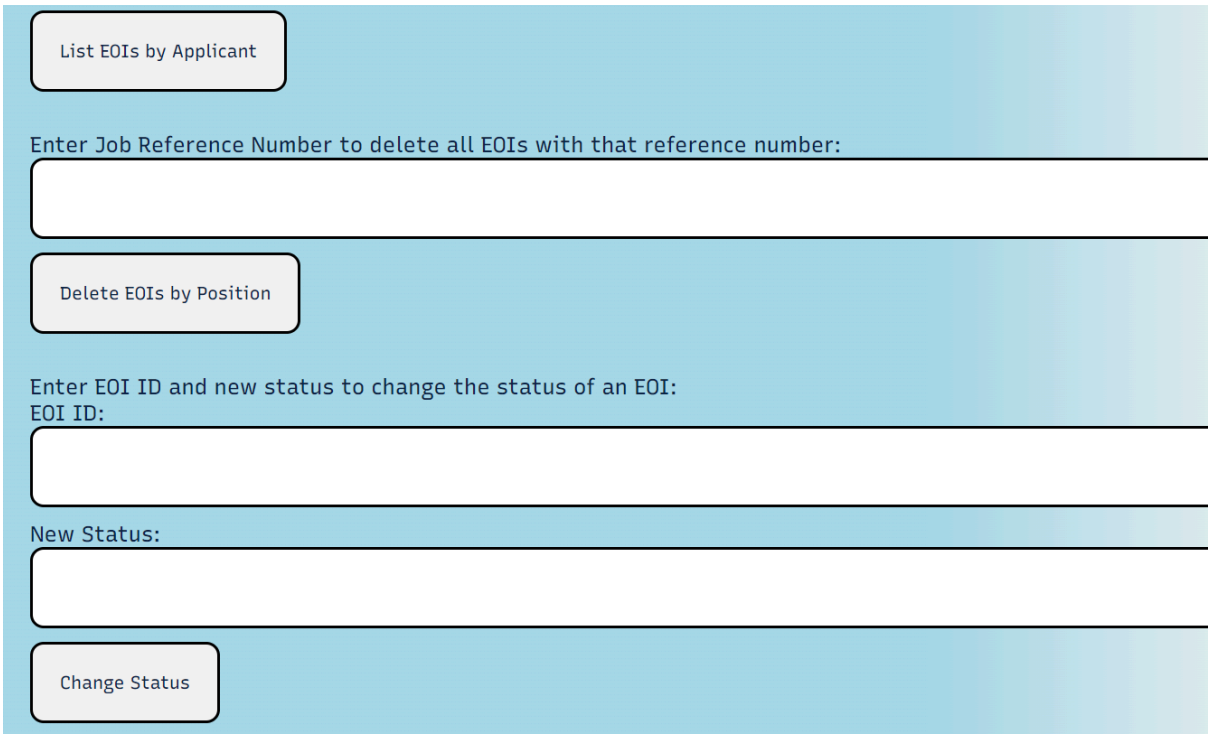
Description: this page is a basic implementation of user authentication using SQL database. When a request is received, the code retrieves the submitted username and password from the request. If the provided credentials match a user in the "users" table. If a match is found, the user is considered authenticated, and their username is stored in a session variable. The code redirects the user to a "manage.php" page. If no match is found, an error message indicating invalid credentials is displayed.

- Screenshot of the database that stored username and password for retrieving:

9. Manage (manage.php):

Purpose: Examine, update and check the data input of the applicants for the job.
Description: When logged in, admin can access the information of the applicants for the job portal. There are functions to navigate, check and examine the information: the list of all EOIs, or the list filter like 'list EOIs by Applicant' by typing their First/Last name; 'delete EOIs by Position' by typing their Job reference number. Admin can also change status.

- Screenshot of the manage.php interface with its features:



Security of the website

- Login/Logout: We implement the login/logout feature to acknowledge it as such an essential component of user authentication systems. It allows users to securely access and exit an application or website. During login, user credentials are validated against stored data, such as usernames and passwords, to authenticate the user's identity. Logout ensures that the user's session is terminated, preventing unauthorized access. These features play a crucial role in protecting user accounts and ensuring that only authenticated individuals can access sensitive information or perform specific actions within the system.

- Validation: Data validation is the process of verifying the accuracy, integrity, and conformity of user input or system data. It ensures that the provided data meets certain criteria or constraints, such as format, length, or range. Validation helps prevent errors, inconsistencies, and security vulnerabilities caused by invalid or malicious input. Failing to properly sanitize user input can lead to severe security risks like SQL injection attacks that allow unauthorized access, as well as manipulation of databases.
- Sanitization: Data sanitization involves the removal or transformation of potentially harmful or unwanted elements from user input or system data. It aims to prevent security vulnerabilities and protect against attacks, such as cross-site scripting (XSS) or SQL injection. Sanitization techniques include removing or escaping special characters, filtering input based on allowed patterns, and applying context-specific transformations. By sanitizing data, applications can enhance security, prevent data corruption or loss, and maintain the integrity of stored information.

My Contribution

I consider myself a valuable contributor to the project, focusing on three main areas: the ProcessEOI functionality, data validation and sanitization, and the design and creation of the database structure using phpMyAdmin:

- For the ProcessEOI feature, I devised a systematic approach to validate and sanitize user input. I implemented data validation techniques to ensure that the submitted data met the required criteria and followed best practices for sanitization, reducing the risk of potential security vulnerabilities.

- Designing the database structure was another significant aspect of my contribution. I worked on creating an efficient database schema that properly normalized the data and established appropriate relationships between tables. This involved carefully considering the data requirements and ensuring data integrity and consistency.

- Throughout the project, my main challenge was finding the right balance between data validation and user experience. I had to implement robust validation processes without compromising the usability and ease of submitting the Expression of Interest (EOI) form
-

Reflection and comments

In reflecting on the project, I am proud of our website and the contributions made by both myself and my teammate. We collaborated effectively, sharing responsibilities and providing feedback to improve the overall outcome. Throughout the project, we maintained effective communication and collaboration. We shared ideas, discussed challenges, and provided constructive feedback on each other's work. I greatly appreciate my teammate's contributions and hope that they recognize and appreciate mine as well. Together, we successfully delivered a high-quality website that meets the project's objectives.

Regarding the website, I focused on the distribution of tasks related to the ProcessEOI functionality, data validation, sanitization, and database design. These elements were crucial for creating a secure and efficient system. Additionally, my teammate made significant contributions to the management system of the website. Their efforts greatly enhanced the overall functionality and usability of the website.