# PROJECT FINAL REPORT

## Learning Platform for Cyber Security Professionals

Laura Chudzio - C00253150

# Table of Contents

# Abstract

This document outlines the development and features of a cybersecurity learning platform, focusing on its robust admin console. Designed with a secure, role-based access system, the platform ensures that only authorized administrators can manage user profiles and update course content.

The development process revealed challenges and achievements, particularly in implementing a dynamic content management system. This system allows for the easy addition and editing of course chapters and materials, making the platform a comprehensive tool for cybersecurity education.

The paper highlights the platform's functionalities, emphasizing the admin console's pivotal role in managing and enriching the learning experience.

# Introduction

This paper discusses the developments and deployment of the comprehensive cybersecurity learning platform, delving into details of the technological and operational facets needed for improved educational engagement and content management.

It has a robust administrative console with role-based secure access to its various functions, enabling comprehensive management of user profiles, course content updates, and much more. I developed this platform amidst various technical challenges, the most notable being the setup of a dynamic content management system that can support smooth registration and editing of educational materials.

Not only have these been addressed with innovative solutions, but they even turned them into some of the most significant learning opportunities that highly added value to the platform's functionality.

Emphases in this paper are on some of the essential features of the platform—such as the immense value drawn from the role of the admin console in enriching users' learning experiences and further adding value to the broader set of skills emanating from cybersecurity education.

## Homepage (index.php)

The index.php page serves as the entrance to the cybersecurity learning platform, CyberTools, welcoming users with an engaging overview of what the platform offers. This page embodies the initial interaction between the user and the platform, setting the tone for the educational journey ahead.

**Session Management and User Authentication:** Upon landing on **index.php**, the platform initiates or resumes a session, ensuring a seamless user experience. This session management is critical for recognizing returning users, especially those with a persistent login enabled by the **user_logged_in** cookie. If a session username isn't set but a cookie is found, the platform restores the session with the username stored in the cookie, thereby streamlining access for users.

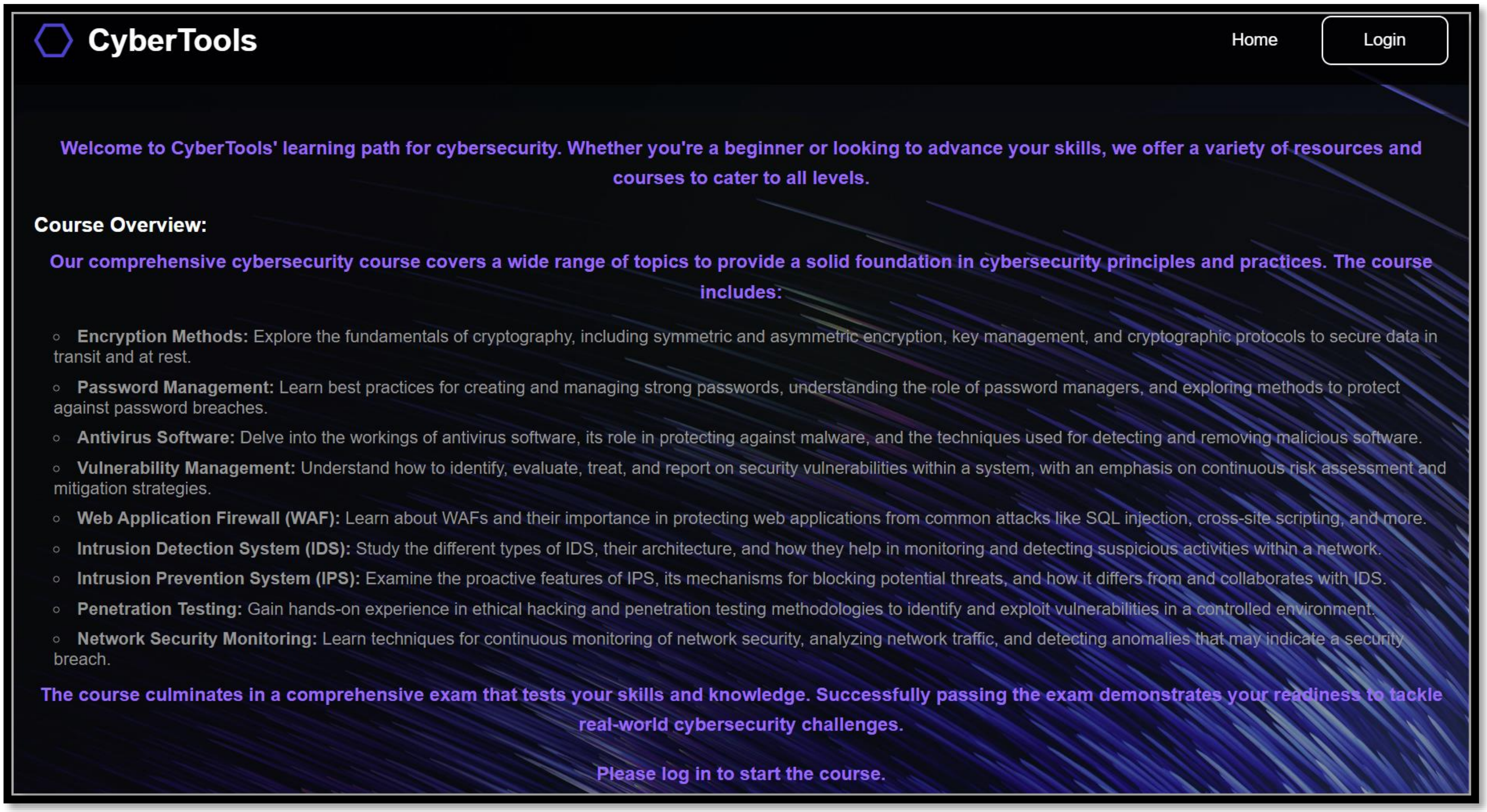
**User Interface and Navigation:** The page features a clean and intuitive interface, with a navigation bar that dynamically adjusts based on the user's authentication state. Logged-in users can navigate directly to course content, their profile, or opt to logout. In contrast, new or unauthenticated visitors are prompted to login, emphasizing the platform's user-centric design that facilitates access to learning materials and user account management.

**Learning Content Overview:** A substantial section of **index.php** is dedicated to outlining the cybersecurity course offerings. It provides prospective learners with a glimpse into the comprehensive curriculum, covering topics from encryption methods to network security monitoring. This overview not only serves to inform but also to entice learners by highlighting the depth and breadth of knowledge that can be gained.

**On authentication**: A prominent call-to-action button with "Start Learning" text lets users jump directly into the course content. This works like a must-have motivator for users to start their learning instantly. For the guest users who are not logged in, they are prompted to make sure that the course content is only visible to authenticated users, hence maintaining the security standards of the platform.

**Registration Success Notification:** Newly registered users are greeted with a confirmation message upon their first visit to **index.php**, reinforcing the platform's responsive and user-friendly design. This immediate feedback mechanism ensures users are aware of their successful registration, fostering a positive user experience from the outset.

## View index.php



Welcome to CyberTools' learning path for cybersecurity. Whether you're a beginner or looking to advance your skills, we offer a variety of resources and courses to cater to all levels.

**Course Overview:**

Our comprehensive cybersecurity course covers a wide range of topics to provide a solid foundation in cybersecurity principles and practices. The course includes:

- **Encryption Methods:** Explore the fundamentals of cryptography, including symmetric and asymmetric encryption, key management, and cryptographic protocols to secure data in transit and at rest.
- **Password Management:** Learn best practices for creating and managing strong passwords, understanding the role of password managers, and exploring methods to protect against password breaches.
- **Antivirus Software:** Delve into the workings of antivirus software, its role in protecting against malware, and the techniques used for detecting and removing malicious software.
- **Vulnerability Management:** Understand how to identify, evaluate, treat, and report on security vulnerabilities within a system, with an emphasis on continuous risk assessment and mitigation strategies.
- **Web Application Firewall (WAF):** Learn about WAFs and their importance in protecting web applications from common attacks like SQL injection, cross-site scripting, and more.
- **Intrusion Detection System (IDS):** Study the different types of IDS, their architecture, and how they help in monitoring and detecting suspicious activities within a network.
- **Intrusion Prevention System (IPS):** Examine the proactive features of IPS, its mechanisms for blocking potential threats, and how it differs from and collaborates with IDS.
- **Penetration Testing:** Gain hands-on experience in ethical hacking and penetration testing methodologies to identify and exploit vulnerabilities in a controlled environment.
- **Network Security Monitoring:** Learn techniques for continuous monitoring of network security, analyzing network traffic, and detecting anomalies that may indicate a security breach.

The course culminates in a comprehensive exam that tests your skills and knowledge. Successfully passing the exam demonstrates your readiness to tackle real-world cybersecurity challenges.

Please log in to start the course.

## Index.php code description

This portion of the code initiates a session or resumes an existing one, ensuring that user state is maintained across page loads. It checks for a specific condition where the user's username is not found in the session, but a 'user_logged_in' cookie exists, suggesting that the user opted for a persistent login.

```
<?php
session_start();

if (!isset($_SESSION['username']) && isset($_COOKIE['user_logged_in'])) {
    $usernameFromCookie = "SampleUser";
    $_SESSION['username'] = $usernameFromCookie;
}
?>
```

The HTML structure showcases a conditional display logic based on the user's authentication state. For logged-in users, it provides navigation links to the course content and other areas. For visitors who are not logged in, it presents a login button, encouraging user authentication to access more features.

```
<header>
    <a class="logo-container">
        <img src="tablogo.png" alt="Logo" class="logo-image">
        <h2 class="logo-text">CyberTools</h2>
    </a>
    <nav class="navigation">
        <a href="index.php">Home</a>
        <?php if (isset($_SESSION['username'])): ?>
            <a href="course.php">Course</a>
            <a href="profile.php"><?= htmlspecialchars($_SESSION['username']) ?></a>
            <a href="logout.php">Logout</a>
        <?php else: ?>
            <button class="btnLogin-popup" id="loginButton">Login</button>
        <?php endif; ?>
    </nav>
</header>
```

This segment outlines the course offerings and presents a ”Start Learning” button for users who are logged in, directly engaging them to proceed to the course materials. For users not logged in, it prompts them to log in to start the course, effectively guiding user actions based on authentication status.

```
<section class=”cybersecurity-info”>
            ---
      <!-- Course Overview and topics -->
            ---
    <?php if (isset($_SESSION[’username’])): ?>
      <button class=”btn” onclick=”window.location.href=’course.php’;”>Start Learning</button>
    <?php else: ?>
      <p>Please log in to start the course.</p>
    <?php endif; ?>
  </section>
```

Here, the script checks for a 'success' query parameter in the URL, a common technique for displaying feedback after an action such as user registration. If present and set to 'true', it displays a success message, enhancing user feedback and engagement.

```
  <?php if (isset($_GET[’success’]) && $_GET[’success’] === ’true’): ?>
    <div id=”registrationSuccess” class=”alert-popup” style=”display:block;”>Registration successful!</div>
  <?php endif; ?>
```

## Registration Process (register.php)

The registration module begins with user session initiation, followed by a connection to the platform's database configured in **config.php**. A crucial step in the registration process is the validation of user-provided data. This includes ensuring that the password meets specific security criteria—such as a minimum length of 8 characters, inclusion of both alphanumeric and special characters—to fortify user accounts against unauthorized access.

The system also verifies that the email address is not already associated with an existing account, a critical measure to prevent duplicate registrations. Successful validation leads to the encryption of the user password using a secure hashing algorithm, followed by the storage of user credentials in the database.

The registration flow diverges based on the user's role; if an admin is creating the account, a notification of successful creation is provided, otherwise, the user is logged in and redirected to the homepage.

## Login Process (login.php)

For the login mechanism, the process commences similarly by initiating a user session and establishing a database connection. The core of the login process involves fetching the user's hashed password from the database based on the provided email and using a password verification function to check the submitted password against the stored hash.

This method ensures that user passwords remain secure and undisclosed, even in the event of direct database access. Upon successful login, session management techniques are employed to maintain the user's signed-in status, with an optional "remember me" feature extending this status through a persistent cookie. In instances of authentication failure—due to incorrect credentials or the absence of a matching user record—an error message is displayed, and the user is redirected back to the login page.

Together, these authentication processes not only protect user accounts but also form the bedrock of the platform's security posture. They exemplify the initial steps towards creating a secure and user-friendly environment for learning cybersecurity concepts, with a particular emphasis on safeguarding access to the admin console, where authorized administrators can manage both users and educational content.

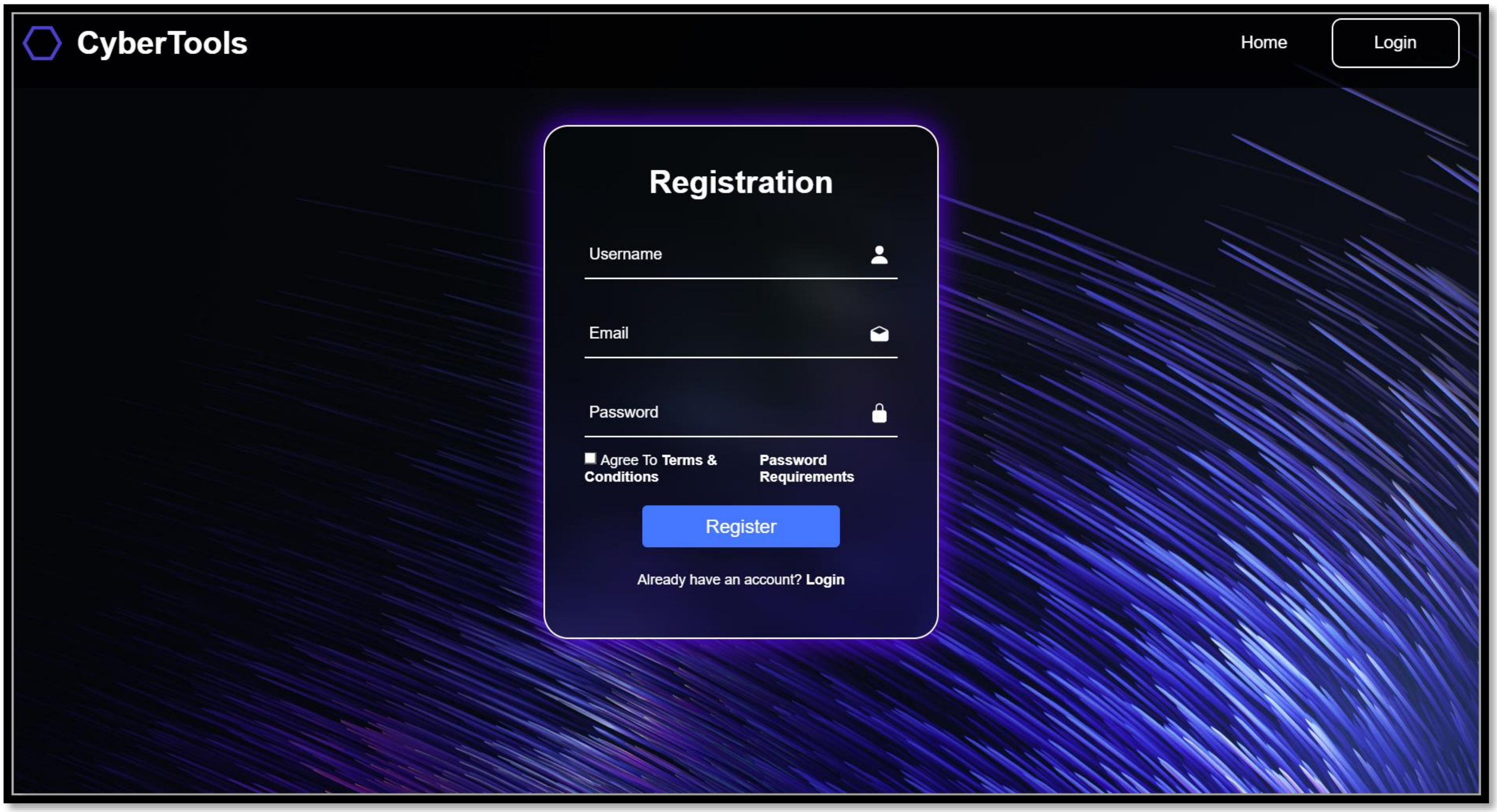
## Logout Process (logout.php)

The logout function is very important since it is required in the logout system of the Cybersecurity Learning platform. This is where users leave the current session. It begins by using the function session_unset() to unset all the session variables to avoid leaving sensitive data there. Then, session_destroy() terminates the session altogether, securing it from possible hijacking by removing traces of the session stored on the server.

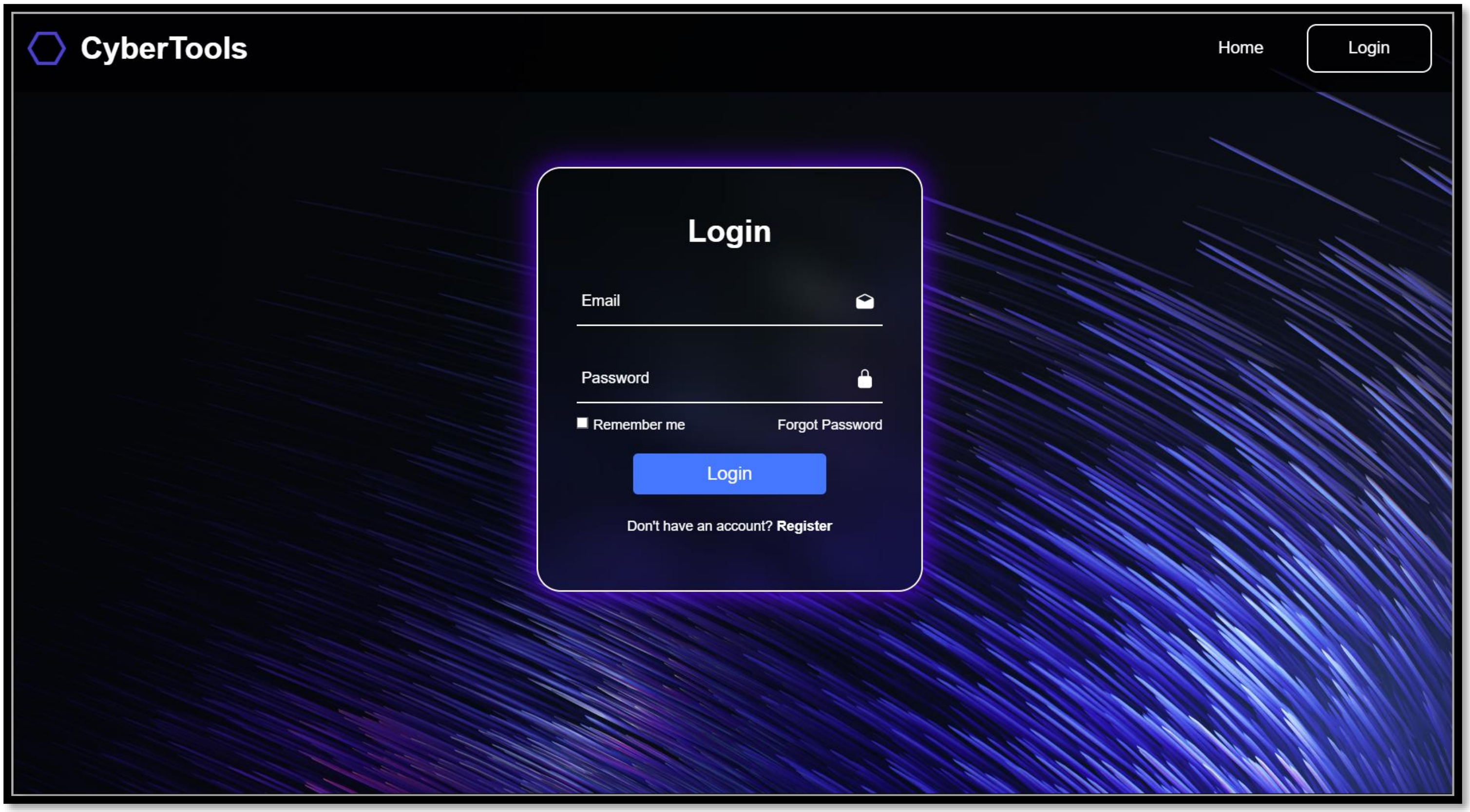
If the user has checked the "remember me" option, the script then invalidates the persistent login cookie, in all probability, by setting the cookie's expiry to a past date, which in turn causes the browser to remove it, disabling automatic re-authentication mechanisms.

Lastly is the homepage to which the script directs the user. This could be conceived as a successful logout confirmation and security because they cannot directly access the authenticated areas post-logout. This brief process outlines how, from starting a session to ending it, the platform commits to solid security, ensuring that learning is done in a secure and private environment.

## View signin.php



Already have an account? **Login**



Don't have an account? **Register**

## Signin.php code Description

This snippet checks the session status and initiates a session if one hasn't already been started. It's essential for tracking user login state and error messages across page requests.

```
if (session_status() == PHP_SESSION_NONE) {
   session_start();
}
```

Error messages from the login or registration process are stored in the session. This code retrieves and clears them, ensuring users are promptly informed of any issues while preventing the persistence of outdated messages.

```
$loginErrorMessage = isset($_SESSION['login_error']) ? $_SESSION['login_error'] : '';
$registerErrorMessage = isset($_SESSION['register_error']) ? $_SESSION['register_error'] : '';
$loginNeededMessage = isset($_SESSION['login_needed']) ? $_SESSION['login_needed'] : '';
```

## Content Functionality (course.php / fetch_chapter_content.php)

The course.php page and its related fetch_chapter_content.php script show how the cybersecurity learning platform uses AJAX and server-side scripting to offer a very interactive and dynamic experience in education. This setup is to ensure effective and secure loading of content about the course so that the learning experience of the user is well facilitated.

Session management and user verification: User access commences from session management. Every time a user tries to access the course without being logged into the system, the user is thus directed to the login page. This ensures the safety of the course content; hence, it is accessible to the people who are logged in. The process checks for both session-based and cookie-based authentication, accommodating the feature opted by the user during login.

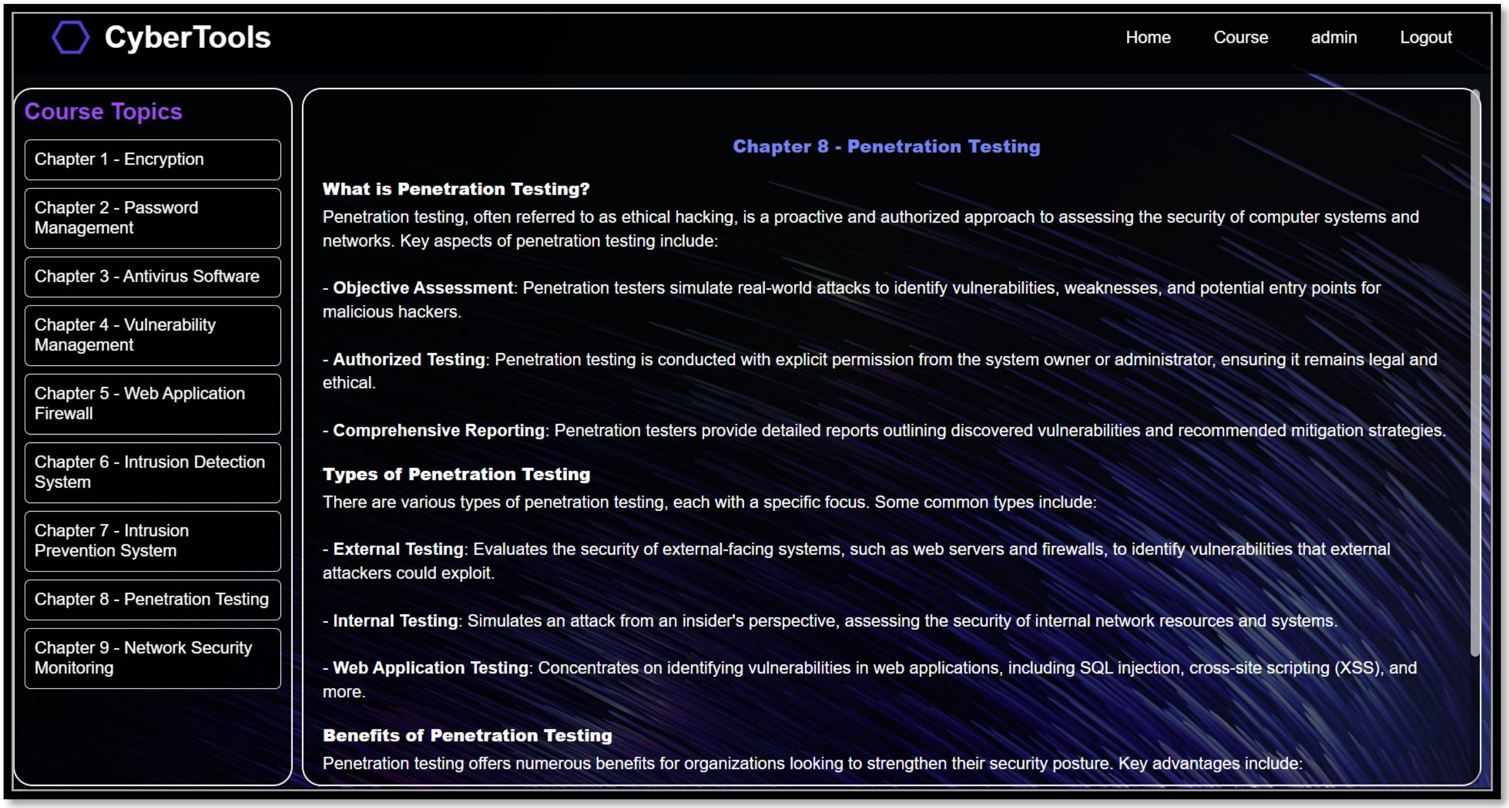
It has a central feature in its page course. php: dynamic AJAX loading of the content of a course. This enables the course chapters and their respective contents to load asynchronously, thus doing away with page reloads. For example, if a user selects a chapter from the sidebar, an AJAX request will be made, related to fetch_chapter_content.php, which queries the database to return the content for the selected chapter.

Reusable Module: Secure Data Handling. Both course.php and fetch_chapter_content.php take precautions to handle data securely. Prepared statements are used for database queries, sessions, and cookies for user authentication. There is also proper sanitation of user inputs using HTMLSPECIALCHARS and outputs.

This will ensure typical web security against the most common vulnerabilities of SQL injection and cross-site scripting (XSS). Server-Side Scripting: Server-side scripting done in PHP is so dynamic to communicate with the database to have the required course content. fetch_chapter_content.php communicates with the database for the details needed concerning the AJAX request from the chapter ID.

It carefully parses the database results to structure them as a JSON format, which is easily consumed and displayed on the client side by the script.

Course.php


CyberTools
Home
Course
admin
Logout
Course Topics
Chapter 1 - Encryption
Chapter 2 - Password Management
Chapter 3 - Antivirus Software
Chapter 4 - Vulnerability Management
Chapter 5 - Web Application Firewall
Chapter 6 - Intrusion Detection System
Chapter 7 - Intrusion Prevention System
Chapter 8 - Penetration Testing
Chapter 9 - Network Security Monitoring

**Chapter 8 - Penetration Testing**

**What is Penetration Testing?**

Penetration testing, often referred to as ethical hacking, is a proactive and authorized approach to assessing the security of computer systems and networks. Key aspects of penetration testing include:

- **Objective Assessment**: Penetration testers simulate real-world attacks to identify vulnerabilities, weaknesses, and potential entry points for malicious hackers.

- **Authorized Testing**: Penetration testing is conducted with explicit permission from the system owner or administrator, ensuring it remains legal and ethical.

- **Comprehensive Reporting**: Penetration testers provide detailed reports outlining discovered vulnerabilities and recommended mitigation strategies.

**Types of Penetration Testing**

There are various types of penetration testing, each with a specific focus. Some common types include:

- **External Testing**: Evaluates the security of external-facing systems, such as web servers and firewalls, to identify vulnerabilities that external attackers could exploit.

- **Internal Testing**: Simulates an attack from an insider's perspective, assessing the security of internal network resources and systems.

- **Web Application Testing**: Concentrates on identifying vulnerabilities in web applications, including SQL injection, cross-site scripting (XSS), and more.

**Benefits of Penetration Testing**

Penetration testing offers numerous benefits for organizations looking to strengthen their security posture. Key advantages include:

## User and Content Management (admin.php)

The admin.php page is the backbone of the cybersecurity learning platform, as it is on this page that the administrator will be able to empower himself with the complete set of tools to manage both users and course content. This encompasses a dedicated administrative panel that acts as a centralized point to control the educational offerings and the platform's user base. This ensures it efficiently allows the Administrator to manage and improve the learning environment effectively.

**Session Verification and Access Control:**

When a user is on admin.php, the session is checked whether it is set or not, and if the session is set, it will further check the authorized role to access the privileged section. Checking of this kind becomes necessary, as it allows access to the admin panel of an authenticated user only having an administrator role. If the session validation fails, the system proceeds to redirect the individual to the login page. This guard ensures that unauthorized personnel are locked out from accessing the sensitive administrative functionalities, thus boosting the overall level of security of the system.

**User Management:** The Registered Users are visible in a much-interrelated window at a glance, showing essential details like User ID, Username, Email ID, etc. The administrator can create a new user account for adding a new student or staff manually. In addition, the current users can easily be managed from the platform. This will involve a detailed user profile being viewed or users who do not need the services offered by the platform being deleted. This level of control is significant for user base integrity and security.

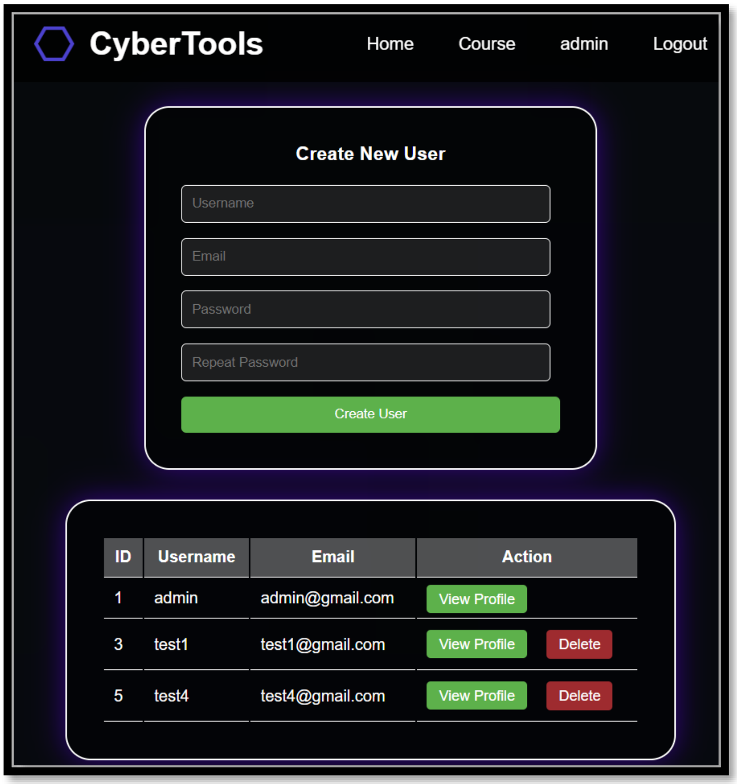
**Course Content Management:** Admin.php is the most essential part of managing the course content. It includes adding all necessary new chapters, headings, and other necessary subtopics related to the course, enriching the curriculum with fresh material. It also allows one to edit or delete an already posted chapter and subheadings, thus allowing the course content to be contemporary and relevant to the user. This flexibility would allow the educational content to be updated continually so it gels with emerging industry standards and, in turn, student needs.

**Content Editing Interactively.** The admin panel of the system offers such an interactive editing mode for the contents of the course, where an administrator can make changes right within the web interface with ease. The need for content updating is thus much easier to manage through this user-friendly approach in ensuring the materials are up-to-date and maintain both accuracy and quality. Further, the platform allows the use of formatting conventions, like Markdown syntax, in allowing for an added flavor of formative content, hence making conducive learning for the student. (w3schools, 2024)

**Conclusion:** Admin.php is a very eminent and potent page within the cybersecurity learning platform. It contains eminent and prominent features that enable the management of users and content. It represents the platform's commitment to ensuring a secure, manageable, and dynamic environment for learning. The administrators, in turn, access the panel through the admin panel, and here they execute their duties of superintendence over the operations of the institution, hence ensuring that the educational content delivered is broad in scope, current, and only accessible to authorized persons.

## View admin.php

### User management:



**Create New User**

Username

Email

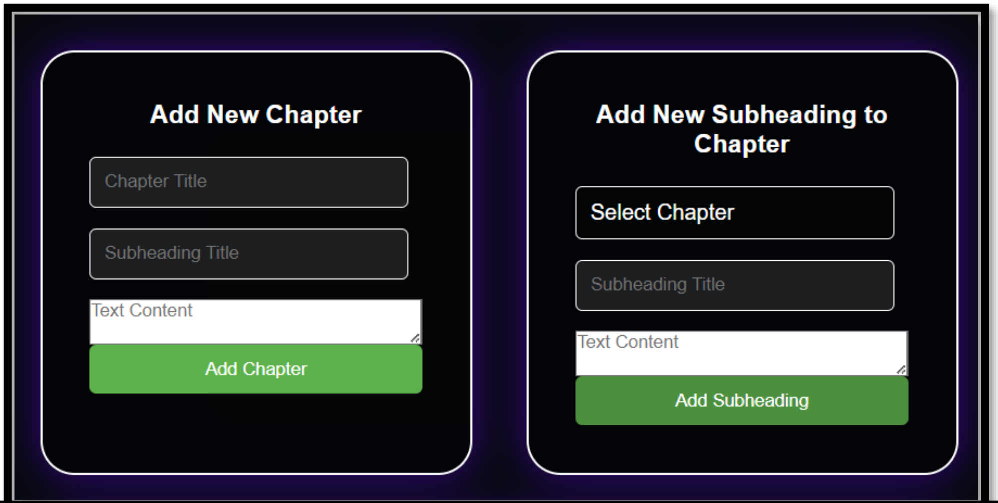
Password

Repeat Password

Create User

| ID | Username | Email | Action |
|---|---|---|---|
| 1 | admin | admin@gmail.com | View Profile |
| 3 | test1 | test1@gmail.com | View Profile Delete |
| 5 | test4 | test4@gmail.com | View Profile Delete |

### Content management:



**Add New Chapter**

Chapter Title

Subheading Title

Text Content

Add Chapter

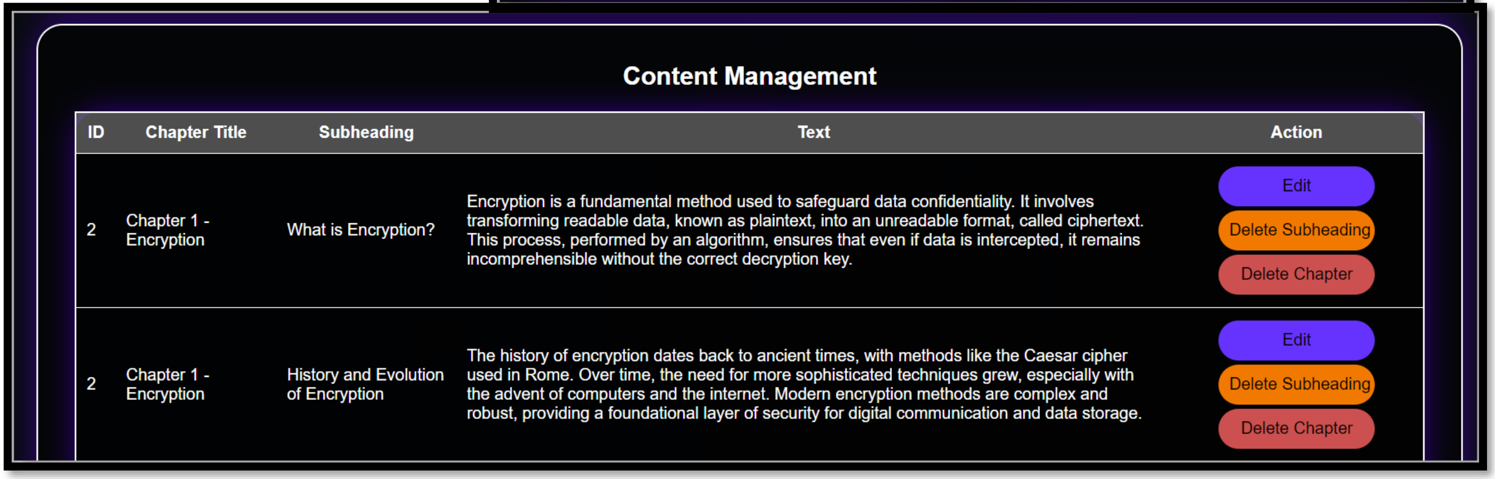
**Add New Subheading to Chapter**

Select Chapter

Subheading Title

Text Content

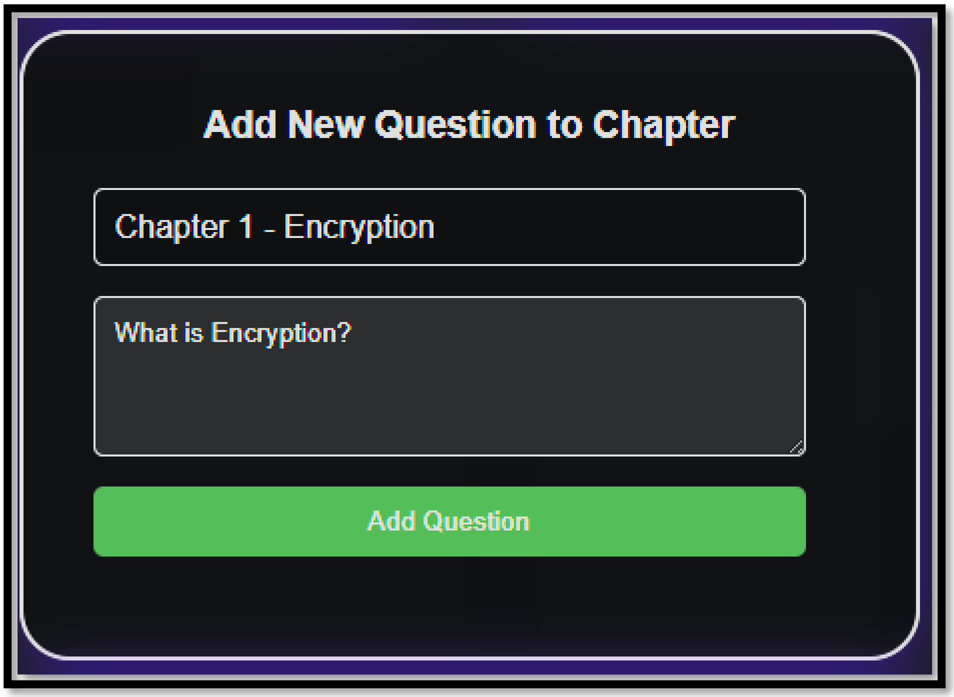
Add Subheading



**Content Management**

| ID | Chapter Title | Subheading | Text | Action |
|---|---|---|---|---|
| 2 | Chapter 1 - Encryption | What is Encryption? | Encryption is a fundamental method used to safeguard data confidentiality. It involves transforming readable data, known as plaintext, into an unreadable format, called ciphertext. This process, performed by an algorithm, ensures that even if data is intercepted, it remains incomprehensible without the correct decryption key. | Edit<br>Delete Subheading<br>Delete Chapter |
| 2 | Chapter 1 - Encryption | History and Evolution of Encryption | The history of encryption dates back to ancient times, with methods like the Caesar cipher used in Rome. Over time, the need for more sophisticated techniques grew, especially with the advent of computers and the internet. Modern encryption methods are complex and robust, providing a foundational layer of security for digital communication and data storage. | Edit<br>Delete Subheading<br>Delete Chapter |

Adding New Questions To chapter:

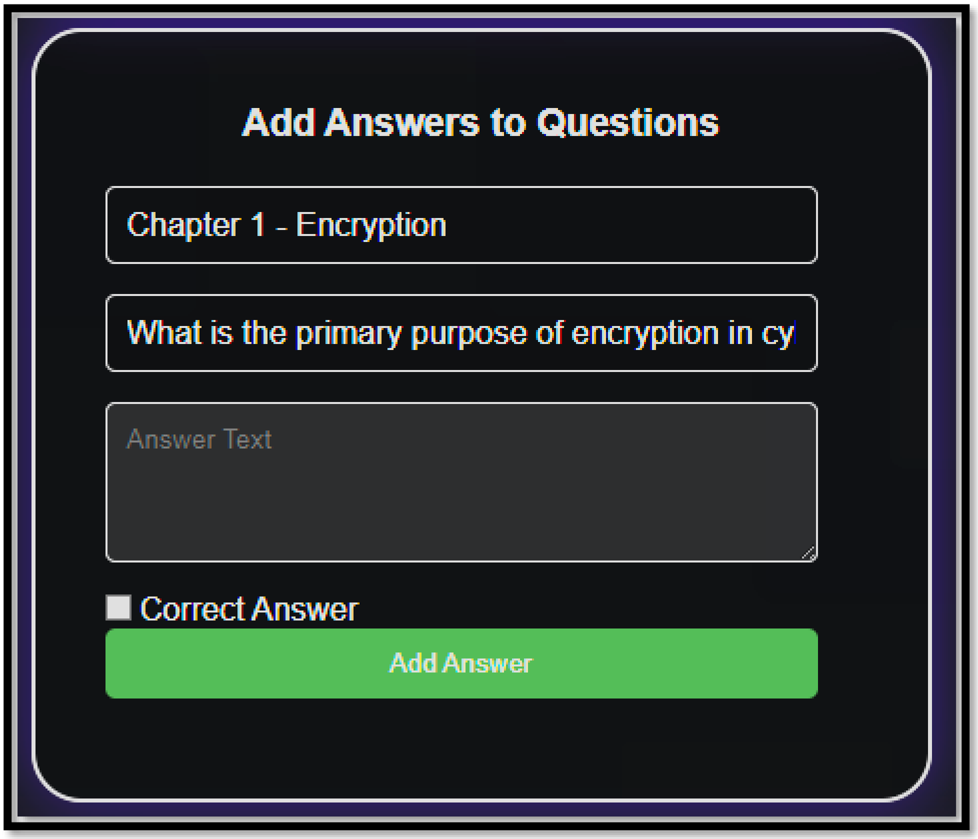


Add New Question to Chapter

Chapter 1 - Encryption

What is Encryption?

Adding Answers to Questions:



Add Answers to Questions

Chapter 1 - Encryption

What is the primary purpose of encryption in cy

Answer Text

□ Correct Answer

## Platform Security

SQL injection (SQLi) is a type of security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. Typically, it involves the insertion of an arbitrary SQL code into the input fields that are then passed to an SQL server for parsing and execution. If not properly sanitized, these inputs can lead to data breaches, unauthorized data manipulations, and even the loss of control over the database server.

```
// Fetch chapters and their subheadings and texts
$sql = "SELECT c.chapter_id, s.subheading_id, c.title AS chapter_title, s.title AS subheading_title, t.content
      FROM Chapters c
      JOIN Subheadings s ON c.chapter_id = s.chapter_id
      JOIN Texts t ON s.subheading_id = t.subheading_id
      ORDER BY c.chapter_id ASC, s.subheading_id ASC"; // Ordered by chapter_id and subheading_id
```

### Mitigation Techniques

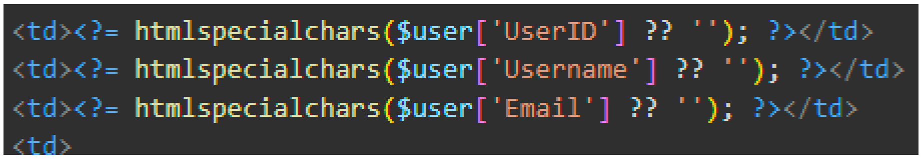
Prepared Statements with Bound Parameters: This is the most effective way to prevent SQL injection. Instead of incorporating user inputs directly into the SQL statement, you use placeholders and then bind the actual input values to these placeholders.

**Example:**

```
$stmt = $conn->prepare("SELECT * FROM users WHERE username = ?");

$stmt->bind_param("s", $username); // 's' specifies the variable type => 'string'

$stmt->execute();

$result = $stmt->get_result();
```

XSS (Cross-Site-Scripting)

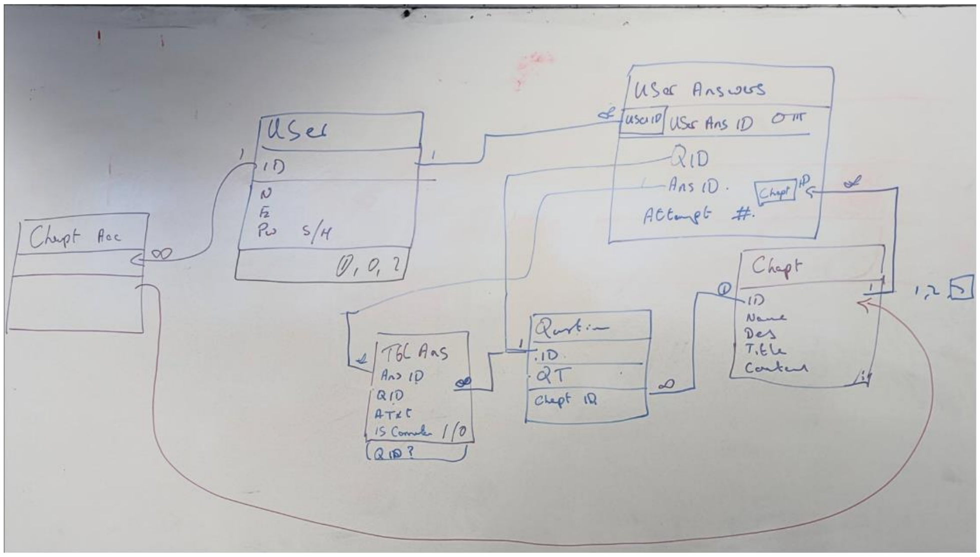
Htmlspecialchars has been implemented across the platform for security against XSS (Cross-Site-Scripting) including Reflected XSS, Persistent XSS.



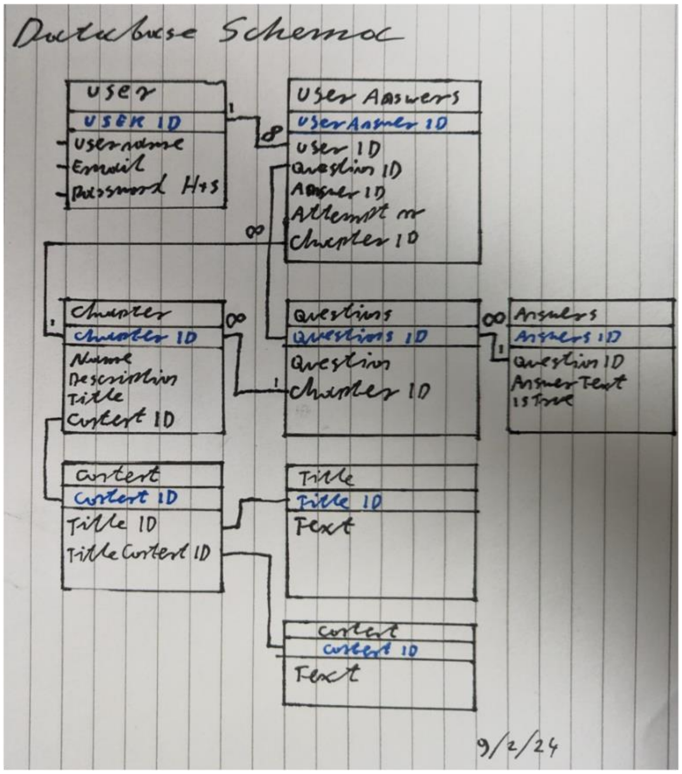
```
<td><?= htmlspecialchars($user['UserID'] ?? ''); ?></td>
<td><?= htmlspecialchars($user['Username'] ?? ''); ?></td>
<td><?= htmlspecialchars($user['Email'] ?? ''); ?></td>
<td>
```

## Database Schema

Initially, I collaborated with my project supervisor, Richard Butler, to accurately design the database schema. Together, we strategized how each table should interconnect to ensure seamless functionality on the website. This involved precise planning to ensure that content could be efficiently fetched, stored, and posted, resulting in a smooth implementation process.



Below, you'll find a comprehensive outline of the database schema:



Following the conceptualization phase, I began crafting the SQL queries necessary for database creation:

```
CREATE DATABASE IF NOT EXISTS CyberTools;

USE CyberTools;

CREATE TABLE IF NOT EXISTS Users (
   UserID INT AUTO_INCREMENT PRIMARY KEY,
   Username VARCHAR(255) NOT NULL UNIQUE,
   Email VARCHAR(255) NOT NULL UNIQUE,
   PasswordHash VARCHAR(255) NOT NULL
);

CREATE TABLE IF NOT EXISTS Chapters (
   chapter_id INT AUTO_INCREMENT PRIMARY KEY,
   title VARCHAR(255) NOT NULL
);

CREATE TABLE IF NOT EXISTS Subheadings (
   subheading_id INT AUTO_INCREMENT PRIMARY KEY,
   chapter_id INT,
   title VARCHAR(255) NOT NULL,
   FOREIGN KEY (chapter_id) REFERENCES Chapters(chapter_id)
);

CREATE TABLE IF NOT EXISTS Texts (
   text_id INT AUTO_INCREMENT PRIMARY KEY,
   subheading_id INT,
   content TEXT NOT NULL,
   FOREIGN KEY (subheading_id) REFERENCES Subheadings(subheading_id)
);

CREATE TABLE IF NOT EXISTS Questions (
   QuestionID INT AUTO_INCREMENT PRIMARY KEY,
   ChapterID INT,
   QuestionText TEXT NOT NULL,
   FOREIGN KEY (ChapterID) REFERENCES Chapters(chapter_id)
);

CREATE TABLE IF NOT EXISTS Answers (
   AnswerID INT AUTO_INCREMENT PRIMARY KEY,
   QuestionID INT,
   AnswerText TEXT NOT NULL,
   IsCorrect BOOLEAN NOT NULL
);

CREATE TABLE IF NOT EXISTS UserAnswers (
   UserAnswerID INT AUTO_INCREMENT PRIMARY KEY,
   UserID INT,
   QuestionID INT,
   AnswerID INT,
   IsCorrect BOOLEAN NOT NULL,
   FOREIGN KEY (UserID) REFERENCES Users(UserID),
   FOREIGN KEY (QuestionID) REFERENCES Questions(QuestionID),
   FOREIGN KEY (AnswerID) REFERENCES Answers(AnswerID)
);
```

## Acknowledgements

This offers a special gratitude to my supervisor, Richard Butler, for his tireless guidance and insightful contributions, without which the development of CyberTools would not have been realized. Richard has indeed been very paramount to the success and realization of this project by maintaining its set timelines and objectives.

Herein, I take this opportunity to extend special thanks to my friends for the cooperation and support extended at every stage of my struggle in the development and preparation of CyberTools. Kind words that inspire and helpful guidance were the sources of constant motivation for me.

First of all, I would like to give big thanks to all my lecturers, who have instilled knowledge and skills in me, which would help me in carrying out this vast project. Their commitment to teaching and being ready to answer my questions, no if they were hard, has been very crucial in shaping both my knowledge and practice of ideas about cybersecurity.

To all these individuals, the collective effort has brought CyberTools to this day; for that, I am deeply grateful.

## Conclusion

This project has successfully developed and launched a robust cybersecurity learning platform tailored specifically for professionals in the field. Throughout the development process, significant advancements were made in terms of security measures, user interface design, and content management. The platform now stands as a comprehensive educational tool that not only enhances the learning experience for users but also significantly contributes to the overall security training landscape.

Key achievements of this project include the implementation of a dynamic content management system, the integration of secure role-based access control, and the development of an intuitive user interface that caters to both beginners and experienced professionals. These features ensure that the platform is not only functional but also secure and easy to navigate.

The real-world application and effectiveness of the platform have been validated through rigorous testing and user feedback. The admin console has been highlighted as a pivotal element, enabling efficient management of user profiles and course content updates. This reinforces the platform's capability to adapt to the ever-evolving demands of cybersecurity education.

The benefits to the cybersecurity industry from this platform are manifold. By providing a structured and secure environment for learning, it facilitates continuous professional development and helps in bridging the skills gap in the cybersecurity workforce. The platform's focus on up-to-date and practical content makes it an invaluable resource for both current practitioners and those aspiring to enter the field, enhancing overall cybersecurity readiness and resilience against threats.

# Plagiarism Declaration

I declare that all material in this submission e.g., thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.

I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams, or other material, including software and other electronic media in which intellectual property rights may reside.

I have provided a complete bibliography of all works and sources used in the preparation of this submission.

I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offence.

| | |
|---|---|
| Student Name: | Laura Chudzio |
| Student Number: | C00253150 |
| Date: | 19/04/2024 |

# Bibliography

w3schools, 2024. *AJAX - The XMLHttpRequest Object.* [Online] Available at: https://www.w3schools.com/xml/ajax_xmlhttprequest_create.asp