HOLY ANGEL UNIVERSITY

Finals Requirement

School Year 2023 - 2024

Software and Hardware Security

Submitted to:
Doc. Tayag, Marlon

Submitted by:
Garcia, Reinel
Razon, Iñigo Marie
Singson, John Florence

Section:
CYB - 301

Submission date:
April 17, 2024

Abstract

This study examines security vulnerabilities in a website for a coffee shop that was created by an undergraduate student for a Web Tools course taught by Sir Almocera, Chris. After a security breach that allows information and data to be taken from the compromised database. We turn our attention to identifying the exploits that were used by the attacker and strengthening the website’s defenses to stop similar attacks in the future. Critical vulnerabilities are found in the initial analysis, which includes badly structured HTML and PHP code. The upload functionality for open-access databases and loose limitations on the kinds of files that can be uploaded.

Our mitigation strategy prioritizes the implementation of secure file upload protocols and revision of admin privileges to restrict unauthorized access and restructuring of the website to fix and secure sensitive areas. By fixing these vulnerabilities, we aim to restore confidence in the coffee shop's website presence and secure its digital assets against online threats.

Table of Contents

Introduction

The coffee shop website is a website that can be accessed by customers to buy the products from the coffee shop, it ranges from various products such as coffee, tea, bread, donuts, etc. The website was created by a part-timer and published in a rush because of the coffee shop's opening in a few hours.

The website's owner discovered that the database contained a shell.php file. Already unknowingly compromised, the owner hired cybersecurity specialists (us in this case) out of concern that they could take more of the information they had secured.

The Cybersecurity Specialists will now identify the vulnerabilities within the website and will try to fix them, considering that the creator of the website was a part-timer who was in a rush, they had to consider a lot of possibilities and listed them, they also did not ask for the credentials of the website so that they can learn how the hacker did it without knowing the security credentials—stating that they have to put themselves on the hacker's shoes to learn how they executed the exploitation.

The Cybersecurity Specialists will now start with scanning and checking the vulnerabilities of the website.

Description of the Vulnerability

These vulnerabilities are commonly found on websites created by freshly graduated students, people who do not know how to apply security on their websites, sloppy AI-created websites, or modified public-use websites.

Commented out Credentials: There were credentials on the Coffee Shop website commented out on the bottom of products.php.

- The credentials can easily be used to log into the Admin account as the credentials are easily found in the source code, It is extremely vulnerable to veterans who know how to use the inspect element on browsers and people who are attentive to the HTML structures of websites.

Public File Upload Access: An upload file button is open for public use on the products.php section of the website.

- The upload button allows all types of file extensions and that includes executable bat files or shells. The attacker can program a bat file that could execute automatically once downloaded or opened, they could also make various types of attacks such as the famous shell attacks which direct the attackers towards the database and then they could just execute a privilege escalation to gain full access.

PHP Reverse Shell: The website is vulnerable to a PHP reverse shell because of the file upload option and the login option.

- As a result of the public file upload, it is a common vulnerability on websites that have file upload buttons. It allows the attacker to connect their PC to the attackers' Database via the use of the PHP or any type of Shell as per the required mode of attack as their medium.

Sign-up allows a Similar name to Admin: The website allows you to use the username Admin for signing up, which allows you to access the website and the database.

- The attackers can create multiple accounts with Admin as their username with the password of their choice, which allows them to freely use Admin Privileges.

Directory Traversal: Users can access the admin.php directly.

- When websitename.com/admin.php is typed on the URL of the website, the attackers can access the Admin.php page which allows them to make use of various tools such as the file upload button.

Hacker knows admin credentials: The previous hacker knows the previous admin credentials.

- The previous attackers may have saved the previous credentials on the website, which allows them to re-enter the Admin page again once inputted.

Cross-site scripting (XSS): The website is also vulnerable through cross-site scripting because an unsanitized input from an HTTP parameter flows into the **echo** statement, where it is used to render an HTML page returned to the user. In this case we used an alert XSS script for our website and mitigated it by removing the error part of the php so that the script will not show any alerts.

- Alerts can be used to display passwords and usernames from the database by inputting scripts on the vulnerable url

SQL Injection: Another vulnerability is SQL Injection because another unsanitized HTTP parameter flows into the **mysqli_query**, where it is used in an SQL query. The website has a login form and can be used for SQLI.

- SQLI can be used on the login form of the website as it allows the users to use the username Admin and put a payload for the password.

Please proceed to the Mitigation and Strategies for more information on how to mitigate the said vulnerabilities.

Review of Related Work

An article by Luisa Cabato (2023) discussed the cyber attack targeting the Philippine House of Representatives website. A group calling themselves "3MUSKETEERZ" defaced the site with a “troll face” meme and a message indicating the breach: “You’ve been hacked.”.The House then reached out to government agencies such as the Department of Information and Communications Technology (DICT) and the Cybercrime Investigation and Coordinating Center (CICC) to investigate the attack. The House has been dedicated to resolving the issue and improving its security. The event made clear the ongoing cybersecurity challenges faced by Philippine governmental entities, underscoring the necessity for robust protective measures for digital infrastructure and sensitive data.

Another article written by Heland Ortega (2016) reports on cyber attacks affecting Philippine government websites and other local sites, following the recent defacement of Comelec's website. On April 27, the Indonesian Defacer Team successfully hacked and defaced several of these websites. Anonymous Philippines promptly issued a warning on their Facebook Page regarding the lack of security measures on government websites. The group clarified that they were not responsible for these defacements and declared their intention not to retaliate by attacking Indonesian sites in response to the hacking incident.

Mitigation Strategies

The proposed mitigation strategies apply to most websites that are created by freshly graduated students or people who are not knowledgeable about website security and how vulnerable their databases are.

Commented-out Credentials: We will remove the commented-out credentials found on the bottom of the products.php.

- We can simply remove the commented-out credentials on the HTML code of the website and double-check if there are any more credentials commented on the source codes of the website.

Public File Upload Access: We will store the file uploads on the Admin.php.

- We will first transfer the file upload button on the Admin.php and remove it on the products.php so that normal users will not have access to the file upload button of the website, we then proceed to clean the products.php file to make sure that no form of SQLi and XSS can be executed

PHP Reverse Shell: We will lock the Admin.PHP so that the attackers are unable to upload files freely and change the file extension to .bin to disable .bat executable files when downloading the submissions.

- It has the same mitigation technique as the file upload access as we put the file upload button on the Admin.php part of the website, we also double-check if there are any more forms of free upload on the website via the source codes.

Sign-up allows a Similar name to Admin: We will change the code structure and make it so that users can only put unique usernames and cannot input Admin as their username.

- We will make use of a *strtolower* IF sentence, $_POST, and $result which indicates new users to make a unique name and that the admin username is not allowed for use regardless of the letters if it's in uppercase or lowercase.
- The strtolower fully converts a string to lowercase.
- An IF sentence is required so that we can tell the code to allow other usernames except the 'admin' username.
- The $_POST is a PHP query function for sending a request to the database which then decides if the user can input and sign in with their desired username.

Directory Traversal: We will make use of sessions so that only the admin can access the website.

- Sessions paired with an if and else statement allow the code to detect if the admin is logged in and the user is not. If the admin logs in, the code directs the admin to the admin.php and locks other users from

accessing the admin.php part of the page, redirecting them to the homepage of the website.

Hacker knows admin credentials: We will simply change the Admin's password to a stronger one.

- We simply change the password of the admin to a stronger one containing numbers and special characters so that we can prevent the attacker's use of brute force and cleanse the database of the previous one.

Cross-site scripting (XSS): We removed the error part of the php that causes the url to echo an alert message..

- The php part of the index.php has a php code that shows an error if the user uploaded no credentials. We entirely removed the php script and tried the XSS again and this time no alerts are shown whenever you try a script on the url.

SQL Injection: The website is vulnerable to one because the database on the first version was not secured and is connected to the login.php.

- We implemented a secure and clean signup form so that even if hackers want to input a payload like ' OR 1=1;-- on the username field or ' OR '1'='1 on the password field, the database allows them to signup but if they try to log in with the characters specified, they are

unable to log in. Please see technical explanation for more information on how the SQL Injection was mitigated.

Due to the implementation of unique usernames and that the code detects if the user is the admin and the password is what is designated for the admin, we also prevent the use of payloads which is a branch of the signup and log-in vulnerability.

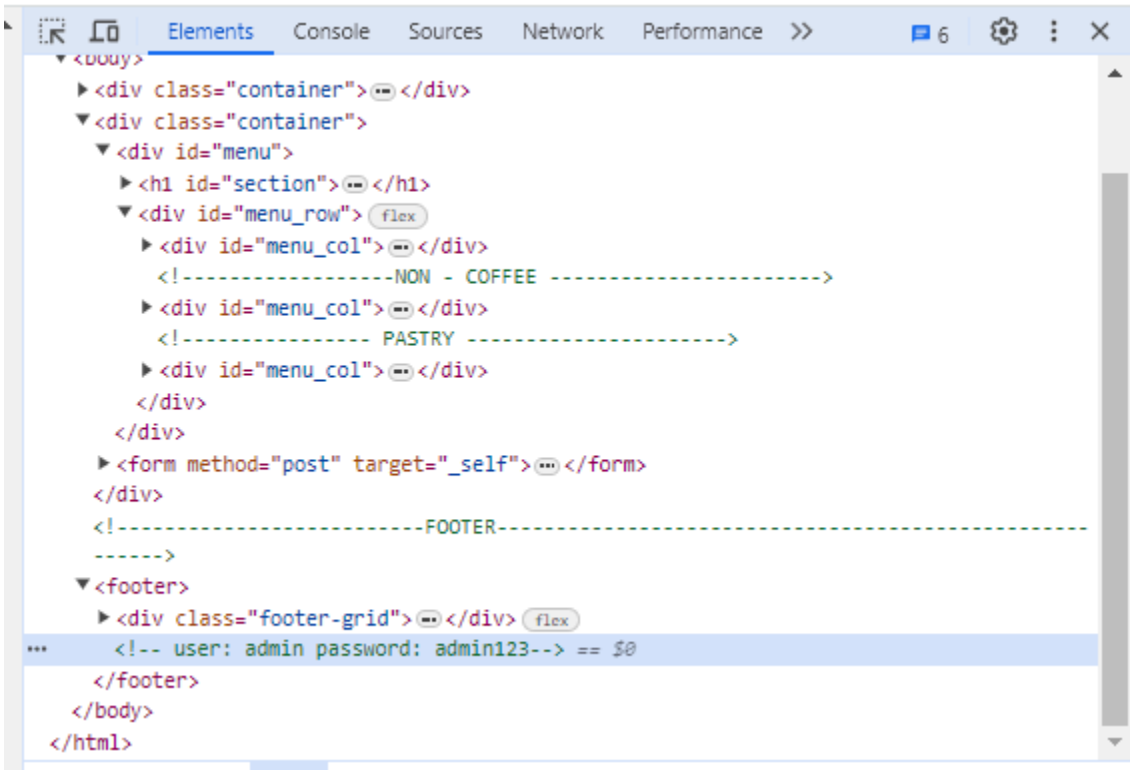
Innovation and Depth

We discussed that a website mitigation strategy would be better research to implement modes of defense. The proposed website was a previous coursework done last school year that was vulnerable to attacks. We have researched the matter and remembered one particular scenario: The attack on the House of Phillippine Representatives' website.

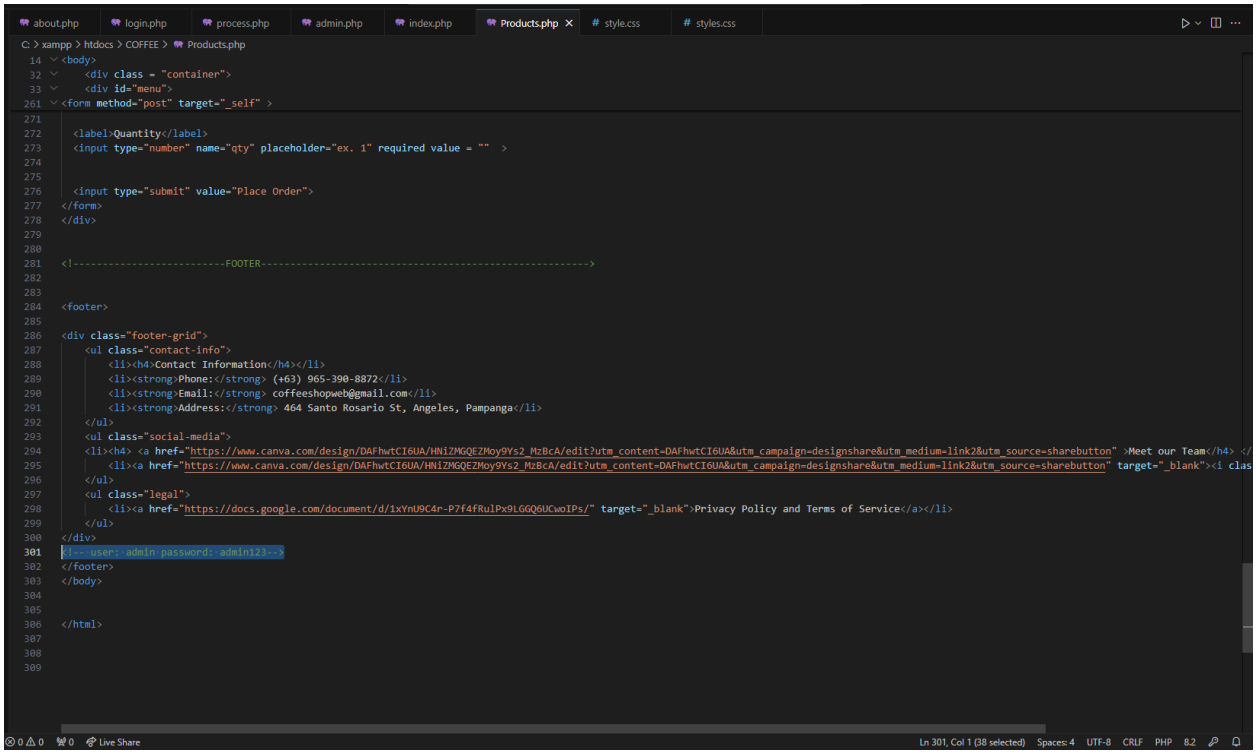
Website scanners were used in the strategies like Nessus, Nikto, and Nmap. We also thought of implementing various modes of defense like the use of cookies but the website is already working fine without needing the cookies. So, we discarded the idea, we also wanted to put a notification system that alerts the database if someone is tampering with the cookies or the sign-up and login form. Still, it was unnecessary as the website was already secured. The database was locked and would cost us a large amount of time to implement something that even professionals struggle with or haven’t implemented yet like an idea that we thought of which is an automated bot that scans the database daily, monitors the visitors, encrypts the source code daily with a different encryption each day, a 2FA on the login requirements, and an automatic banning bot that logs customers out and deletes their account ones tampering with secured files are detected.

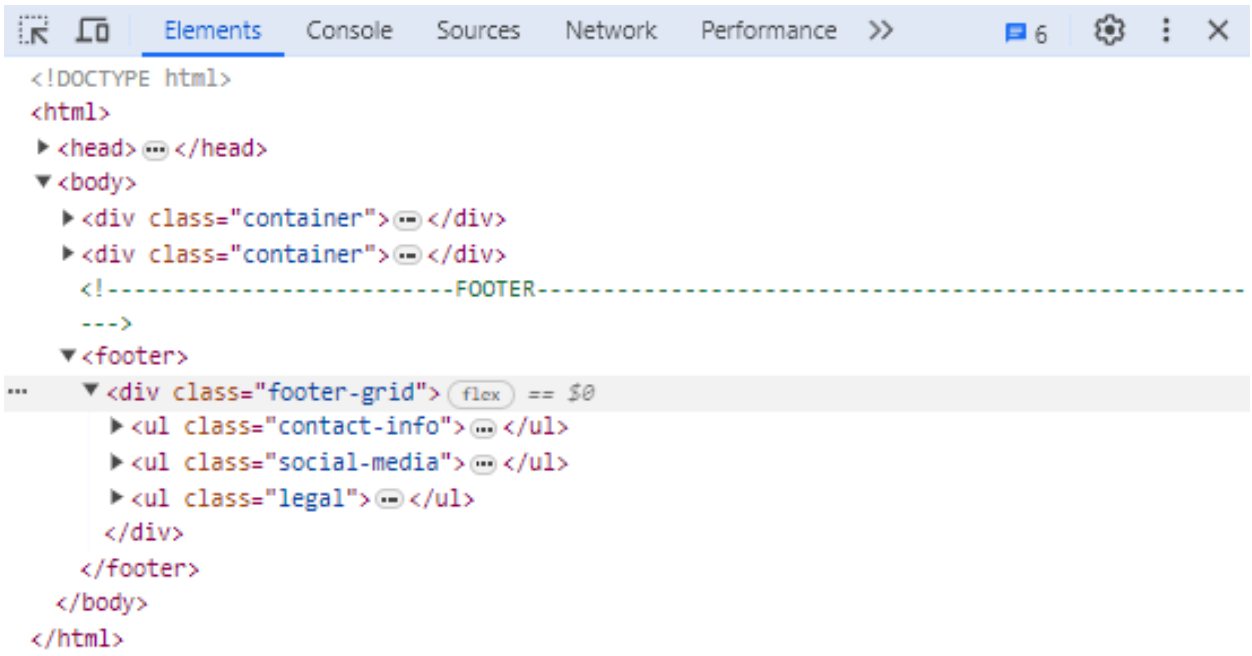
Technical Explanation

Commented out Credentials: There were credentials on the Coffee Shop website commented out on the bottom of products.php.



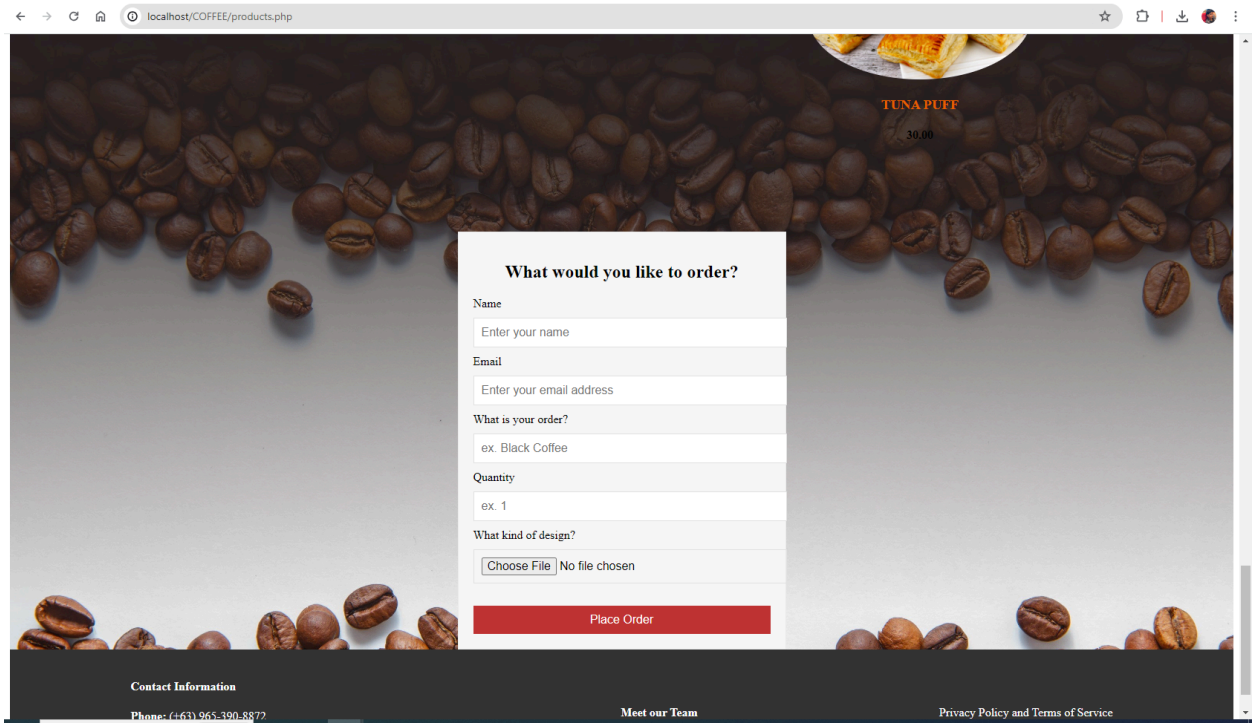
- We removed the commented credentials



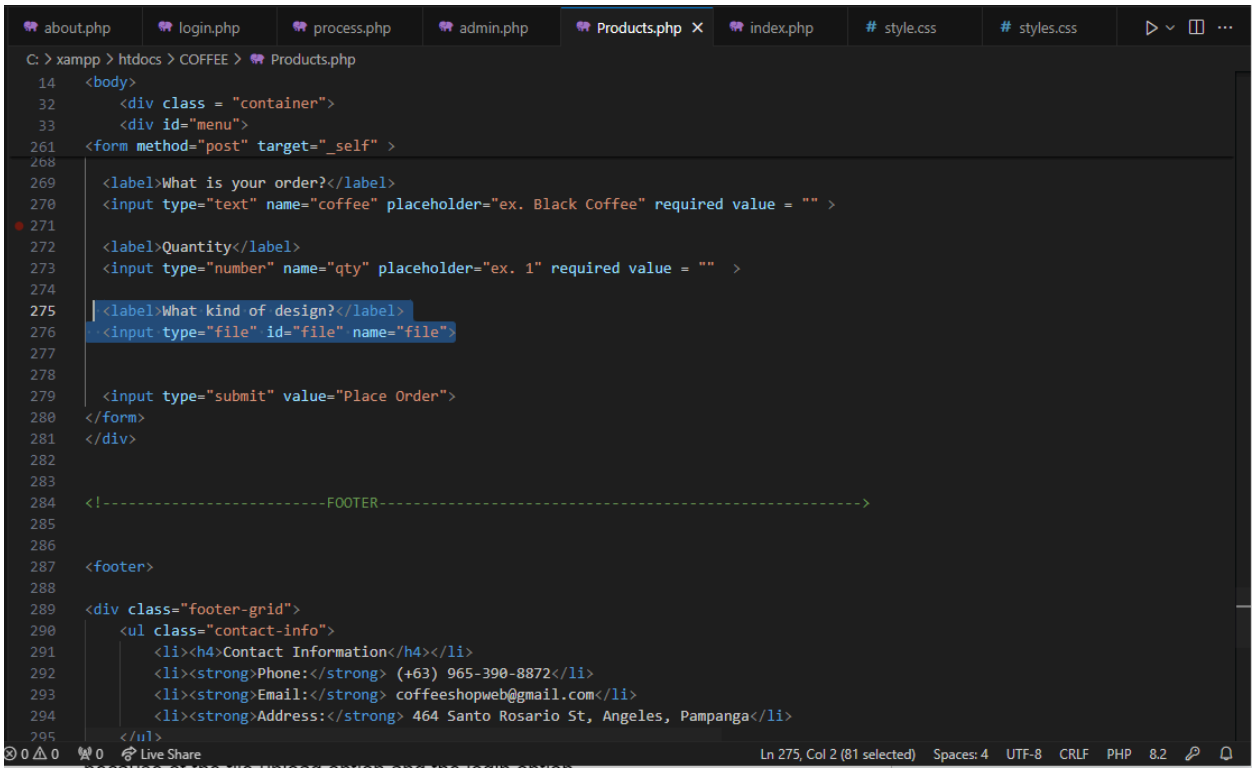


- We now have successfully removed the commented credentials.

Public File Upload Access: An upload file button is open for public use on the products.php section of the website.



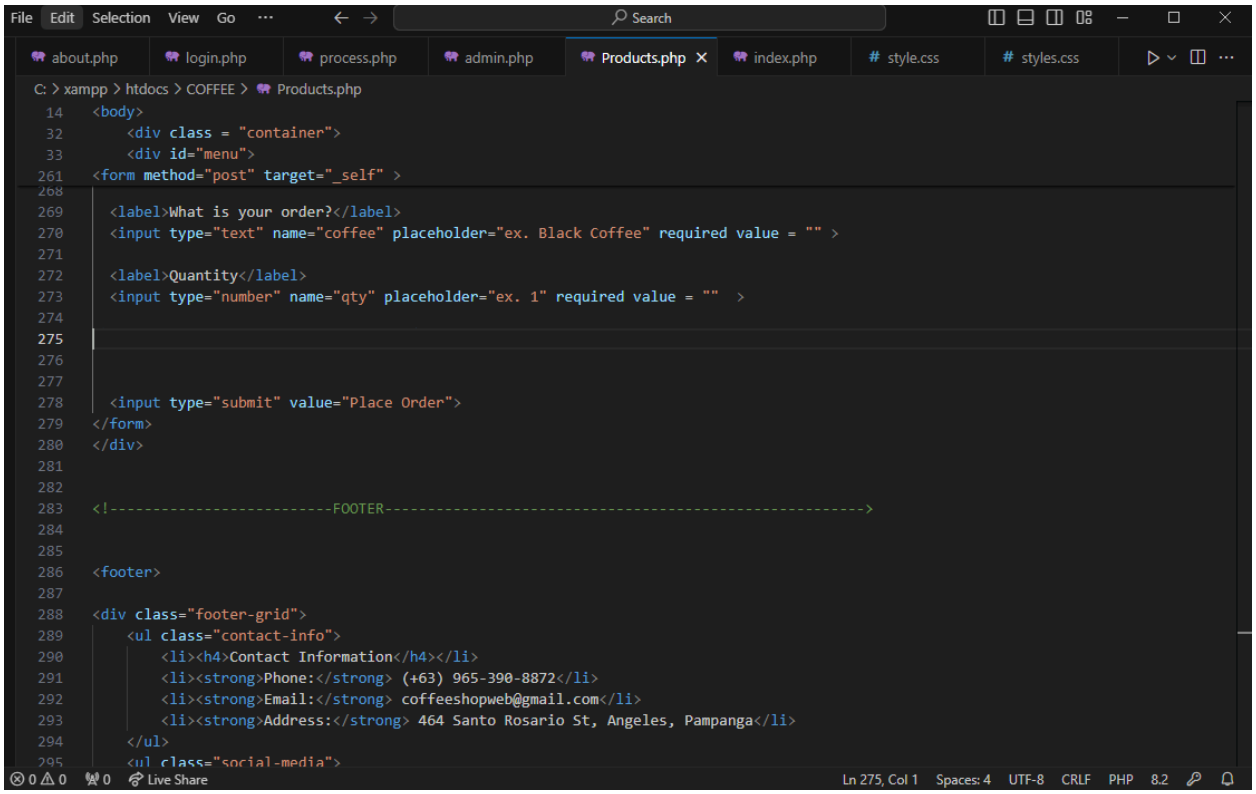
- We will now remove the Public File Upload Button and transfer it to the Admin.php


about.php
login.php
process.php
admin.php
Products.php
index.php
style.css
styles.css
C: > xampp > htdocs > COFFEE > Products.php
<body>

<form method="post" target="_self" >
<label>What is your order?</label>
<input type="text" name="coffee" placeholder="ex. Black Coffee" required value = "" >
<label>Quantity</label>
<input type="number" name="qty" placeholder="ex. 1" required value = "" >
<label>What kind of design?</label>
<input type="file" id="file" name="file">
<input type="submit" value="Place Order">
</form>

<!-------------------------FOOTER-->
<footer>

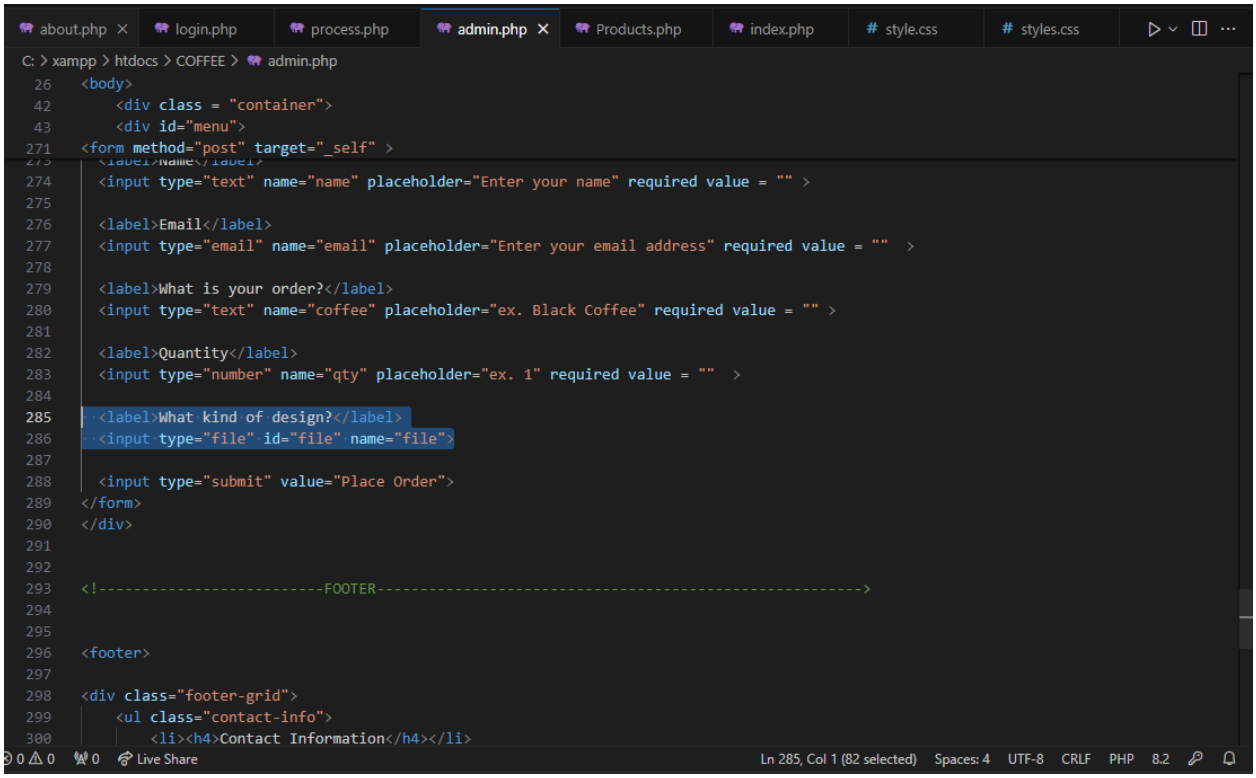
<ul class="contact-info">
<h4>Contact Information</h4>
Phone: (+63) 965-390-8872
Email: coffeeshopweb@gmail.com
Address: 464 Santo Rosario St, Angeles, Pampanga
Live Share
Ln 275, Col 2 (81 selected)
Spaces: 4
UTF-8
CRLF
PHP
8.2


File
Edit
Selection
View
Go
Search
about.php
login.php
process.php
admin.php
Products.php
index.php
style.css
styles.css
C: > xampp > htdocs > COFFEE > Products.php
<body>

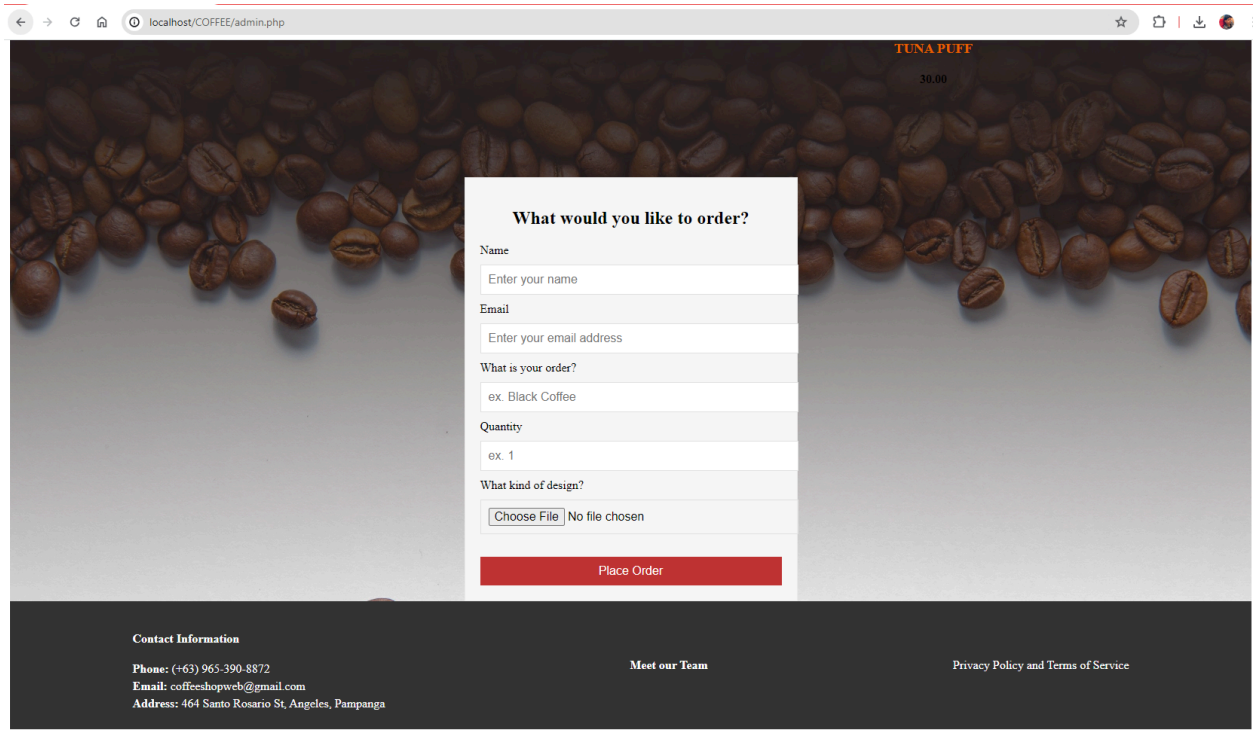
<form method="post" target="_self" >
<label>What is your order?</label>
<input type="text" name="coffee" placeholder="ex. Black Coffee" required value = "" >
<label>Quantity</label>
<input type="number" name="qty" placeholder="ex. 1" required value = "" >
<input type="submit" value="Place Order">
</form>

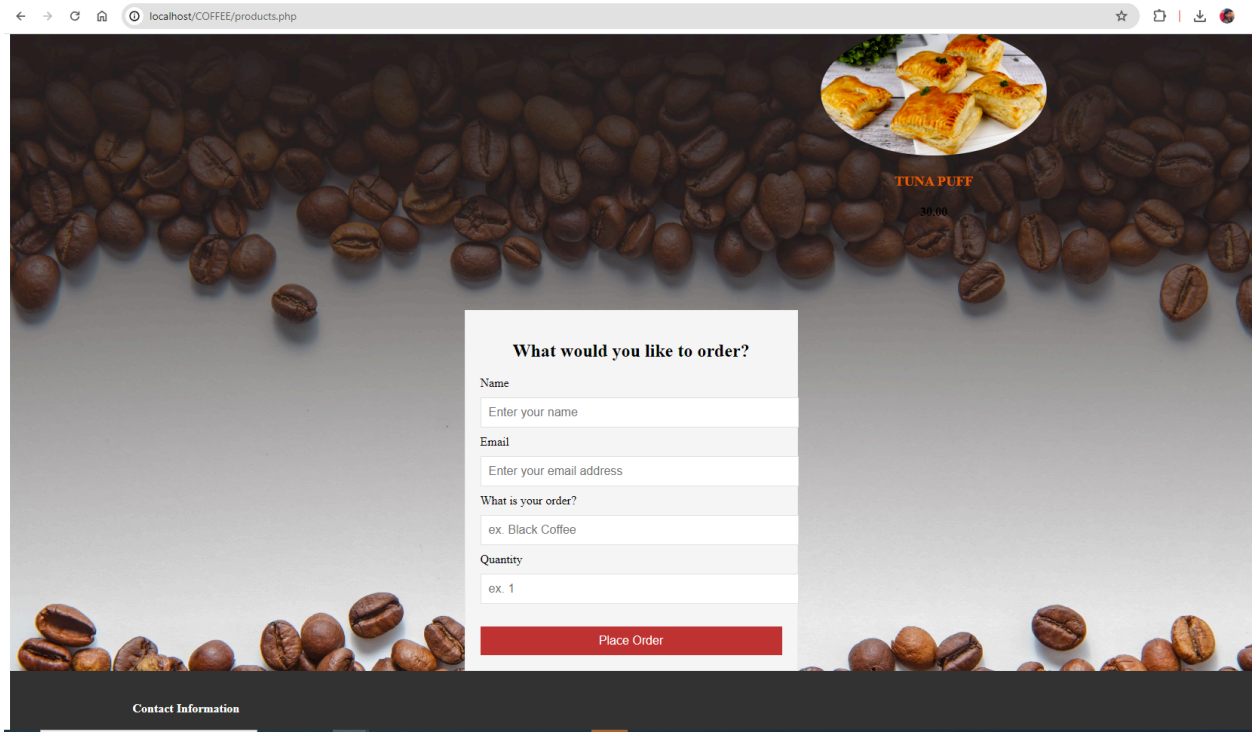
<!-------------------------FOOTER-->
<footer>

<ul class="contact-info">
<h4>Contact Information</h4>
Phone: (+63) 965-390-8872
Email: coffeeshopweb@gmail.com
Address: 464 Santo Rosario St, Angeles, Pampanga
Live Share
Ln 275, Col 1
Spaces: 4
UTF-8
CRLF
PHP
8.2



- We can now see that the File Upload Button is gone from the products.php and is now transferred to the admin.php.

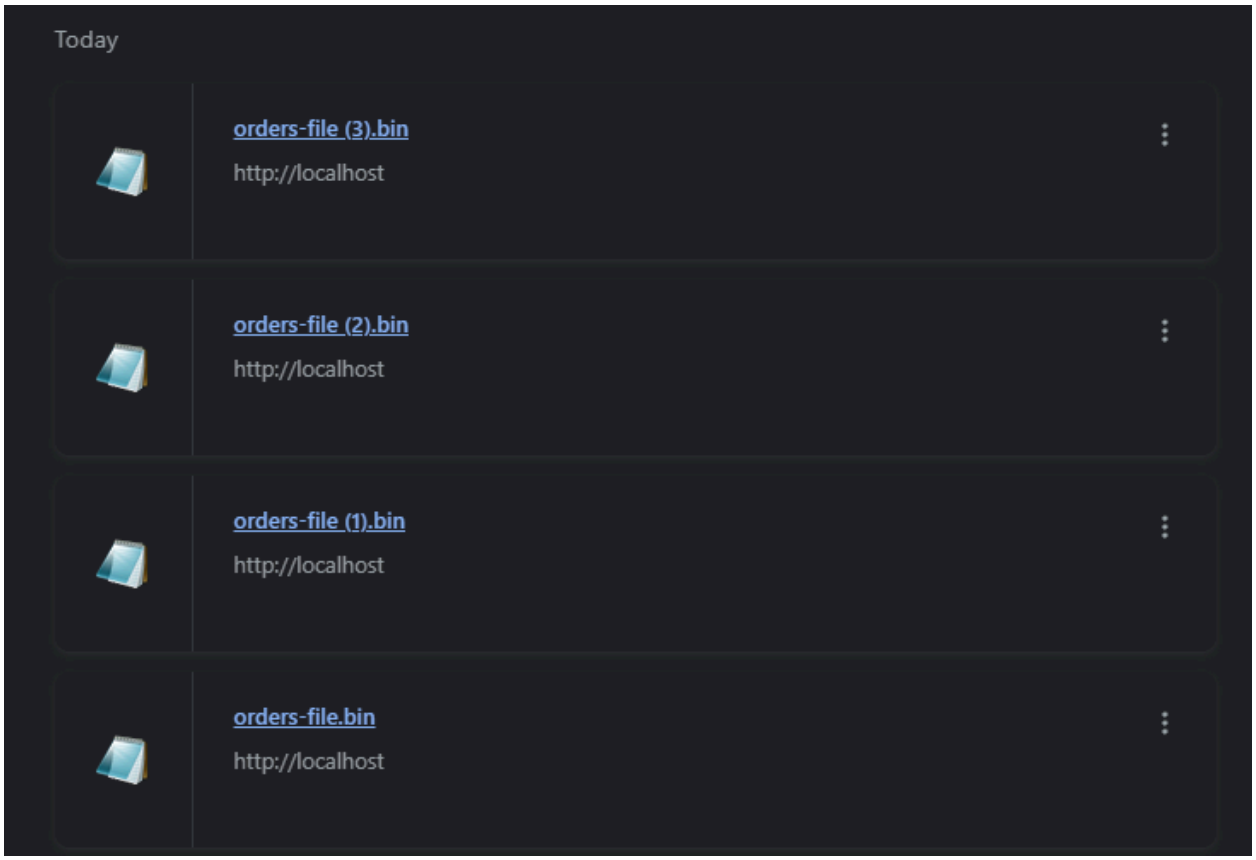

localhost/COFFEE/admin.php
TUNA PUFF
30.00
What would you like to order?
Name
Enter your name
Email
Enter your email address
What is your order?
ex. Black Coffee
Quantity
ex. 1
What kind of design?
Choose File No file chosen
Place Order
Contact Information
Phone: (+63) 965-390-8872
Email: coffeeshopweb@gmail.com
Address: 464 Santo Rosario St, Angeles, Pampanga
Meet our Team
Privacy Policy and Terms of Service


localhost/COFFEE/products.php
TUNA PUFF
30.00
What would you like to order?
Name
Enter your name
Email
Enter your email address
What is your order?
ex. Black Coffee
Quantity
ex. 1
Place Order
Contact Information

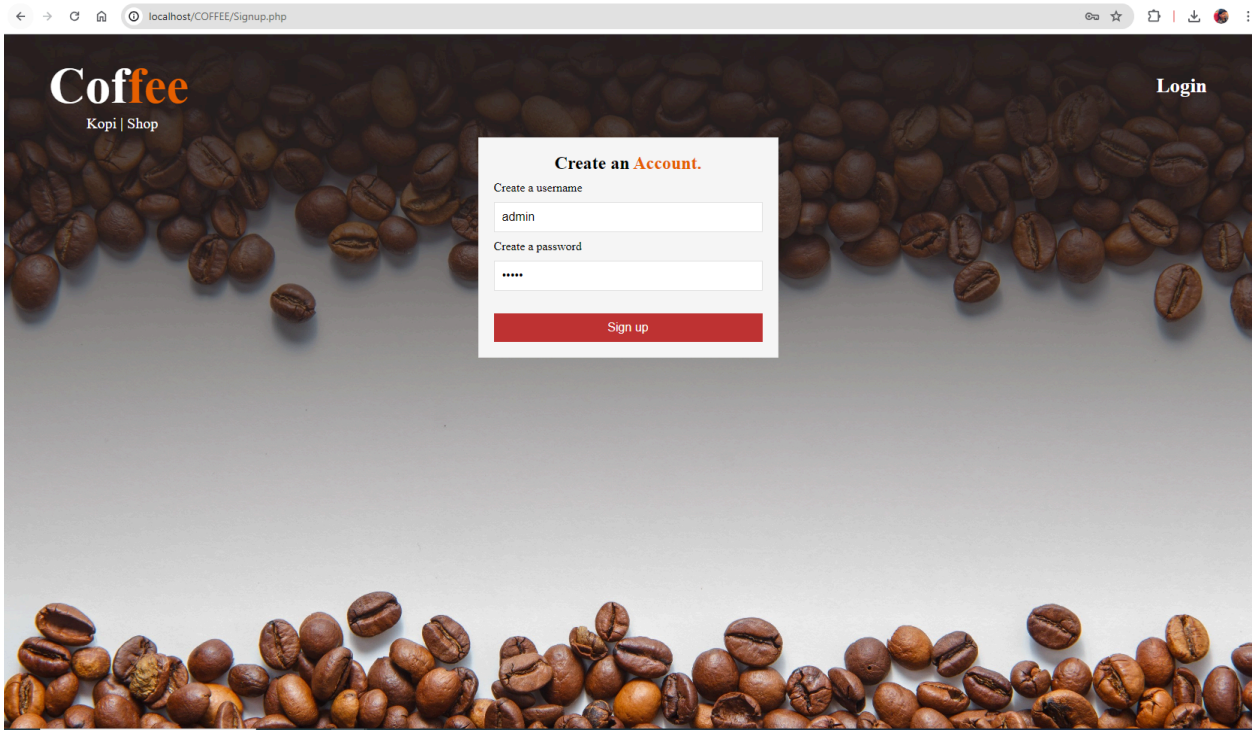
PHP Reverse Shell: The website is vulnerable to a PHP reverse shell because of the file upload option and the login option.

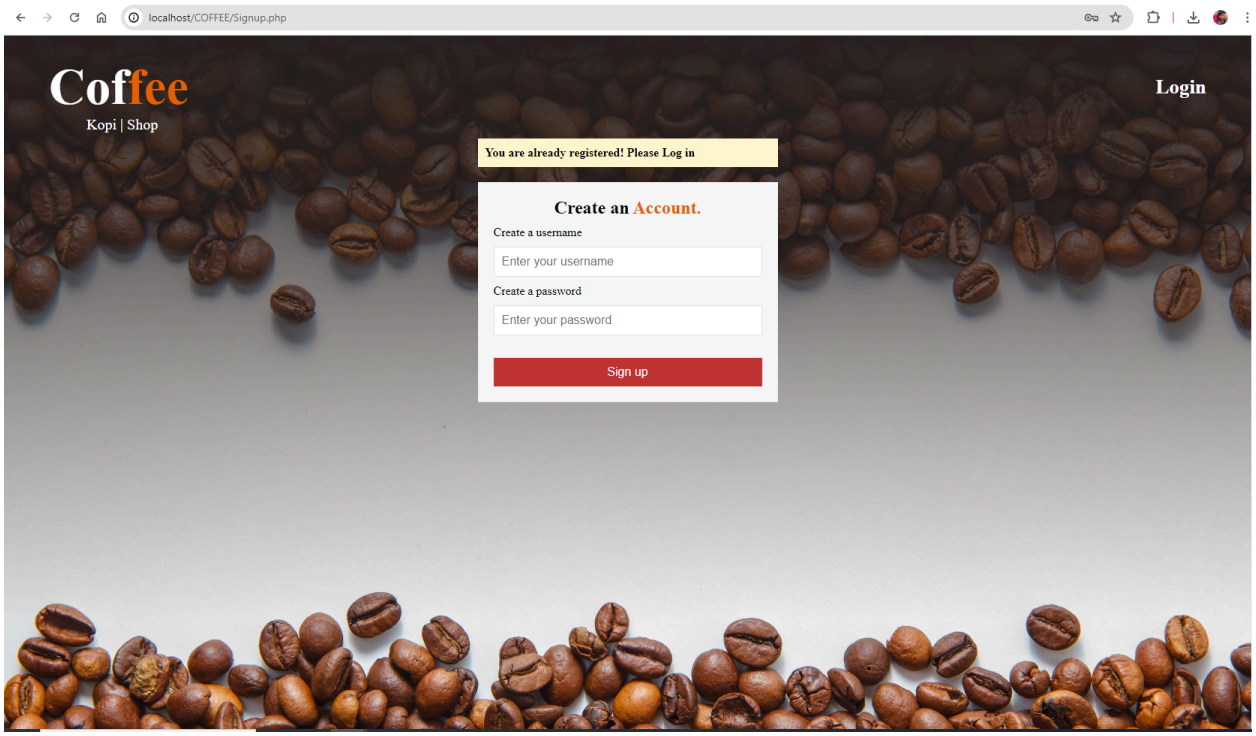
	order_id	dop	name	email	Coffee	qty	file
Edit Copy Delete	91	2024-04-16 00:22:47	Reinel Garcia	rggarcia2@student.hau.edu.ph	black coffee	2	[BLOB - 5 B]
Edit Copy Delete	92	2024-04-16 00:24:00	Florence Singson	jmsingson@student.hau.edu.ph	Affogato	3	[BLOB - 5 B]
Edit Copy Delete	93	2024-04-16 00:24:52	Iñigo Razon	icrazon@student.hau.edu.ph	Americano	2	[BLOB - 5 B]

- We changed the uploaded file on the database to BLOB so then when downloaded they will not be automatically executed if they are a .bat or .exe file, instead, they will appear as a bin file with an extended extension that displays the contents via a text file.



Sign-up allows a Similar name to Admin: The website allows you to use the username Admin for signing up, which allows you to access the website and the database.



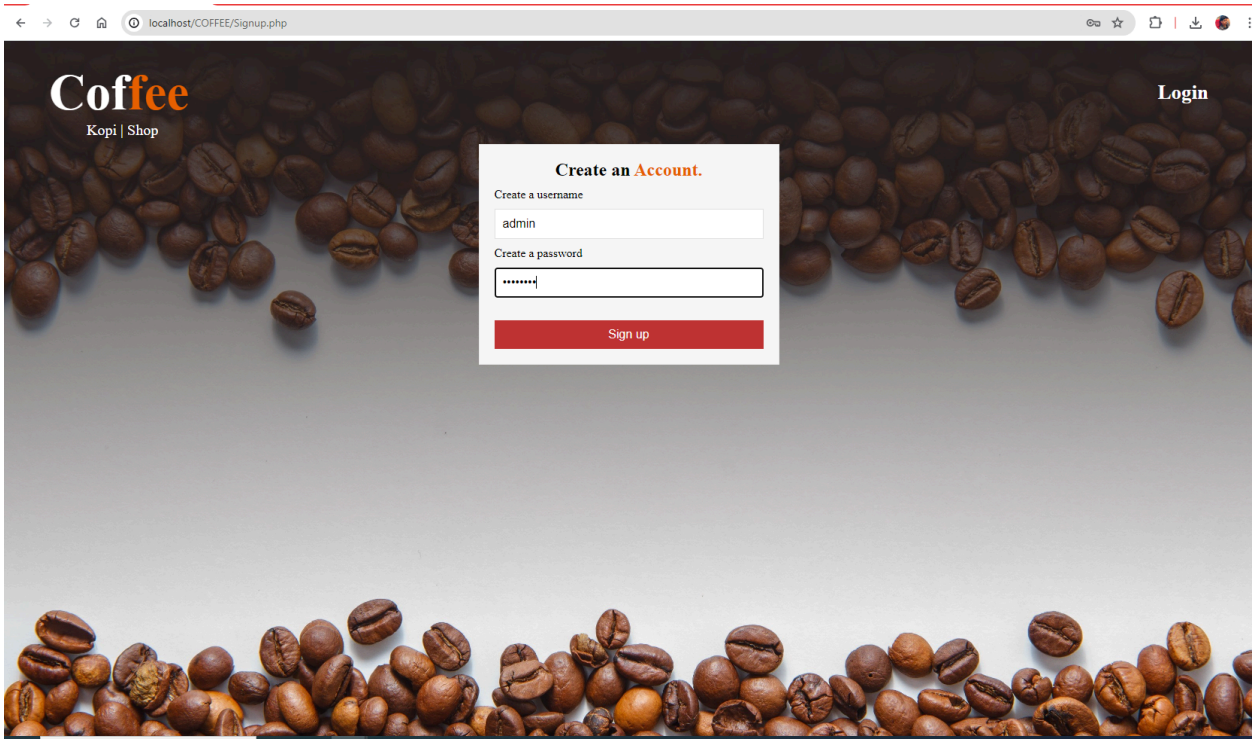


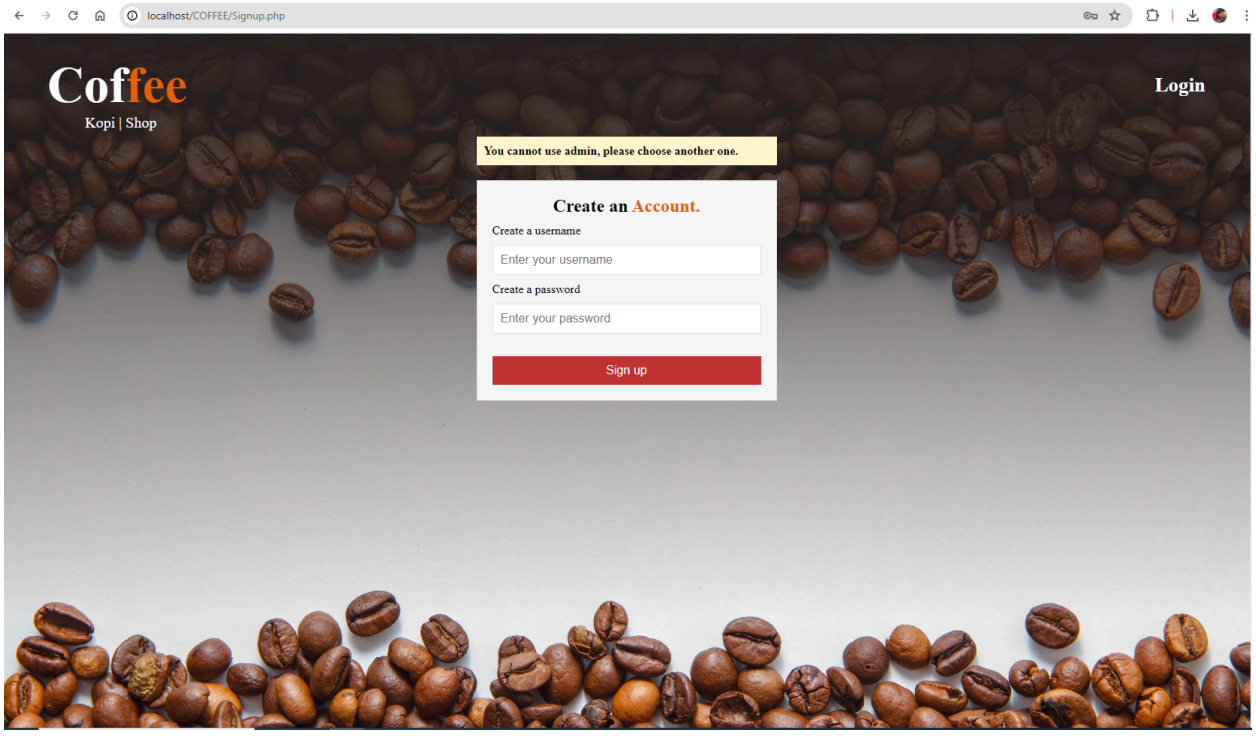
	id	user_name	password	privelege
Edit Copy Delete	1	admin	admin123	admin
Edit Copy Delete	3	adminCF	admin321	
Edit Copy Delete	4	coffeelover13	pacquiao3in1	
Edit Copy Delete	5	Jonel	jonel1	
Edit Copy Delete	6	Reinel	1234	
Edit Copy Delete	12	Florence	1234	
Edit Copy Delete	34	admin	iamhacker	admin

- The hacker was able to create an admin account with admin privileges with his choice of password.
- We proceed to remove the allowance of the ‘admin’ username in the website account creation with the use of this modified code.

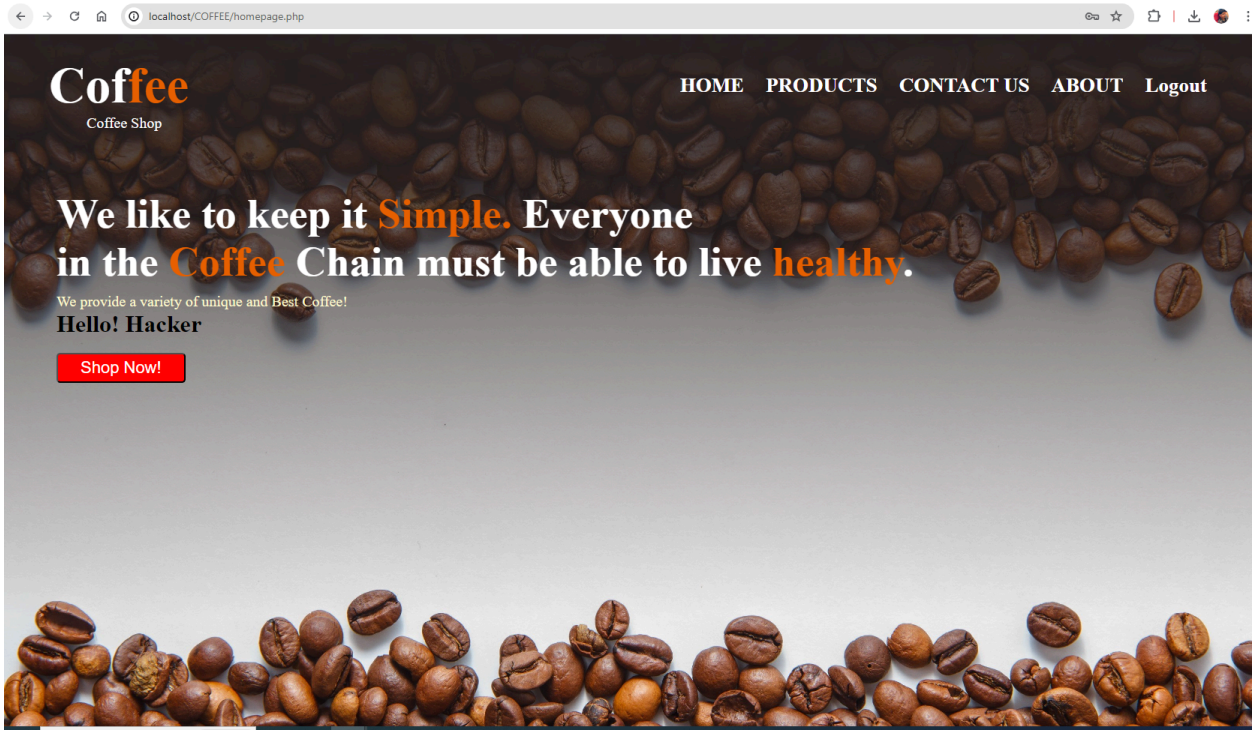
```php
<?php
if (isset($_POST["user_name"])) {
  if (strtolower($_POST["user_name"]) === "admin") {
    echo $result = "<div class='notify'>You cannot use admin, please choose another one.</div>";
  } else {
    require "process3.php";
    echo $result = $result == ""
      ? "<div class='notify'>You are already registered! Please Log in</div>"
      : "<div class='notify'>$result</div>";
  }
}
?>
```

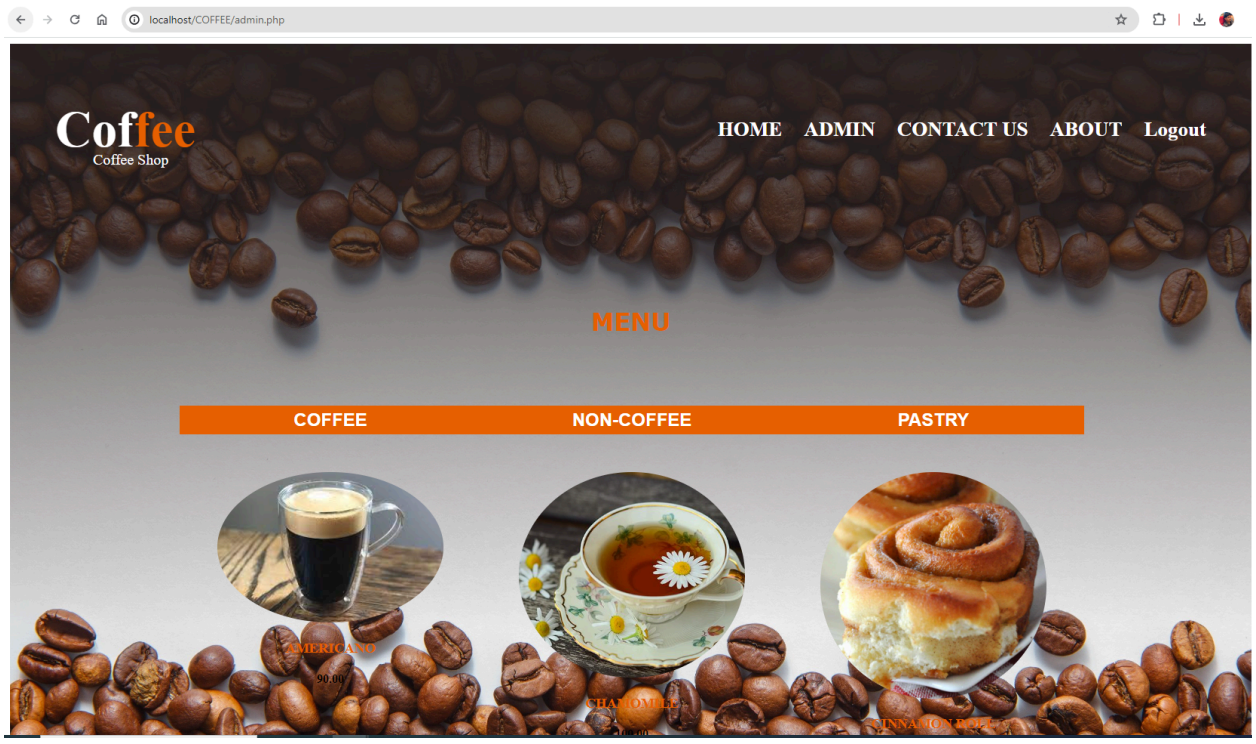
- The strtolower converts the variable names to lowercase letters.
- The $_POST is a PHP query that sends a request to the database.
- Paired with the $_POST query is the === ‘admin’ which specifies that the users cannot create an account with the name admin on it
- The users are also alerted that they cannot use the username admin, and they are also notified if they successfully register.


localhost/COFFEE/Signup.php
Coffee
Kopi | Shop
Login
Create an Account.
Create a username
admin
Create a password
Sign up


localhost/COFFEE/Signup.php
Coffee
Kopi | Shop
Login
You cannot use admin, please choose another one.
Create an Account.
Create a username
Enter your username
Create a password
Enter your password
Sign up

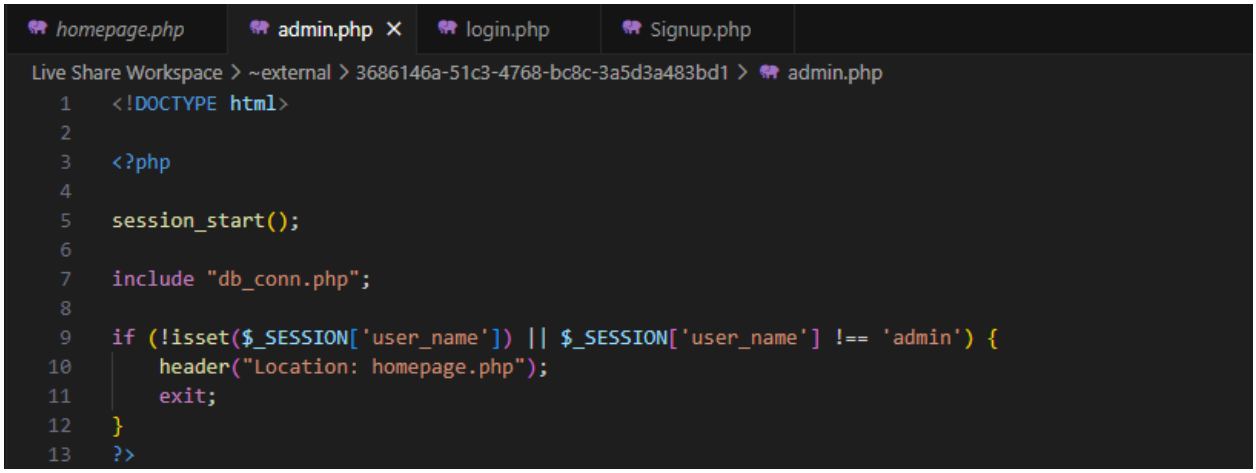
Directory Traversal: Users can access the admin.php directly.



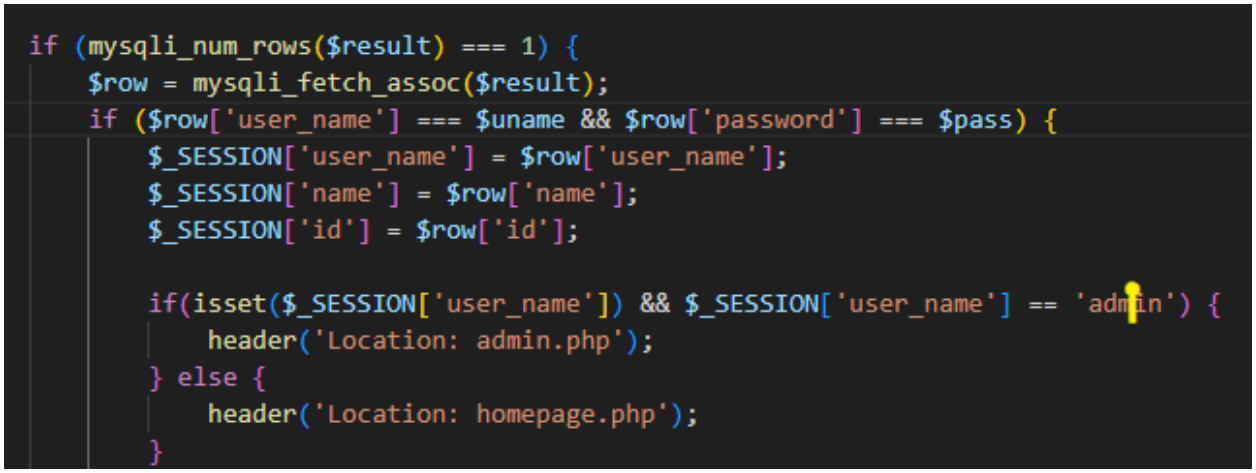


- We can see that the hackers can access the admin page directly from the URL.

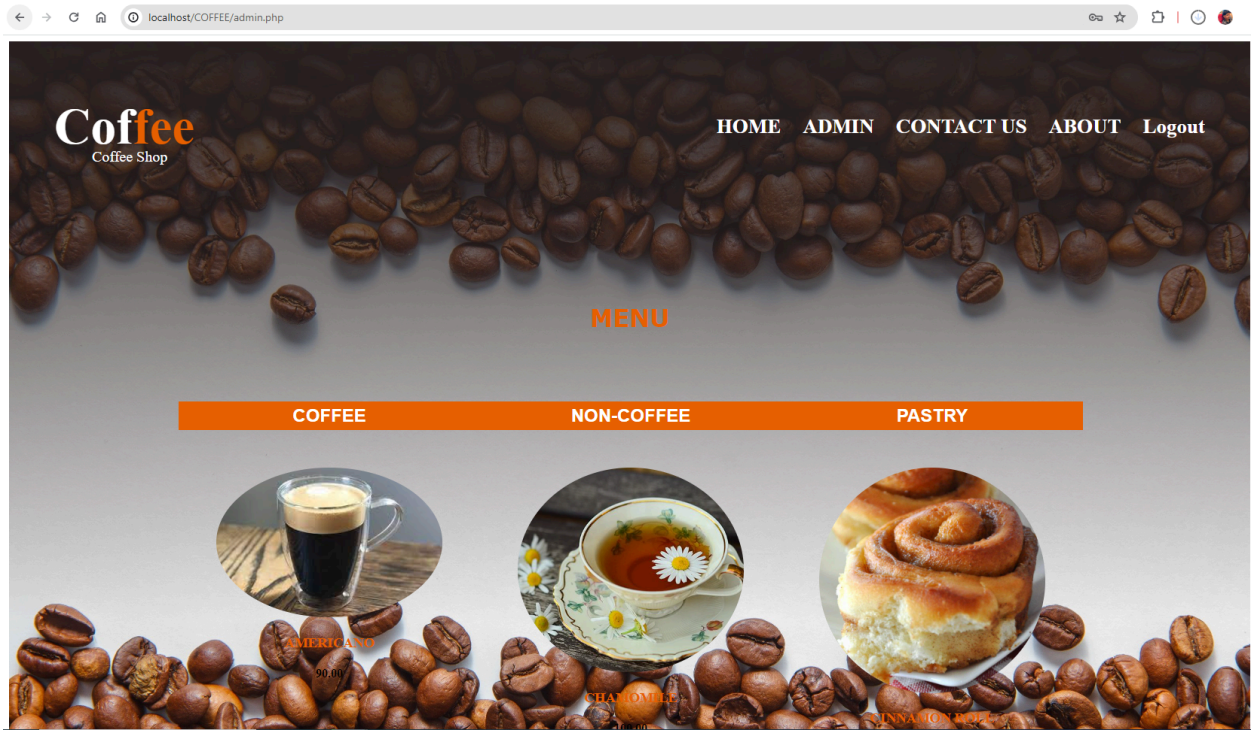
- We will make use of sessions so that the code can detect if the user who logged in is the admin.



- The picture above is taken from the admin.php
- The picture below is taken from the login.php



- The website now redirects the users to the homepage.php if the user tries to access the admin.php, but if the admin logs in, the PHP will automatically redirect the admin to the admin.php with the GUI elements of the admin's tools.

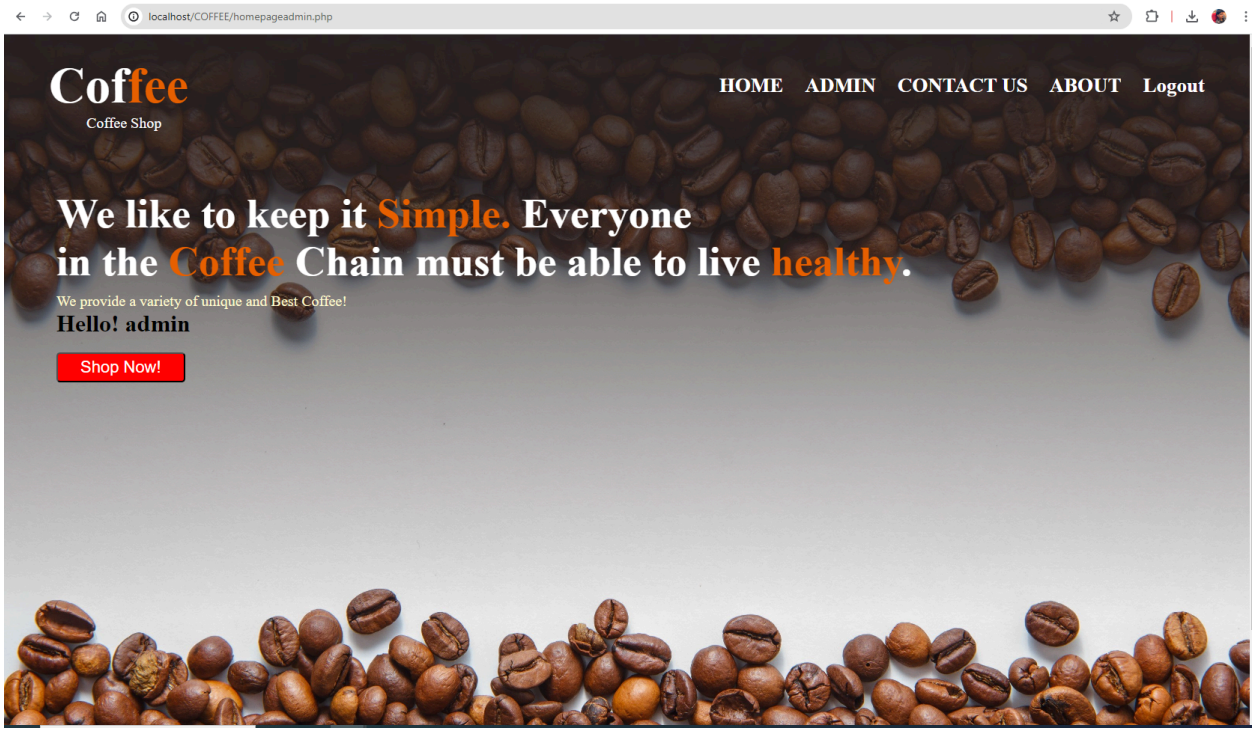


Hacker knows admin credentials: The previous hacker knows the previous admin credentials.

- We just simply change the credentials on the database but with a better password, we will include 2 special characters, a minimum of 8 words paired with 4 numbers.
- We used unexpected#2000#wordpass

	id	user_name	password	privelege
Edit Copy Delete	1	admin	admin123	admin

	id	user_name	password	privelege
Edit Copy Delete	1	admin	unexpected#2000#wordpass	admin

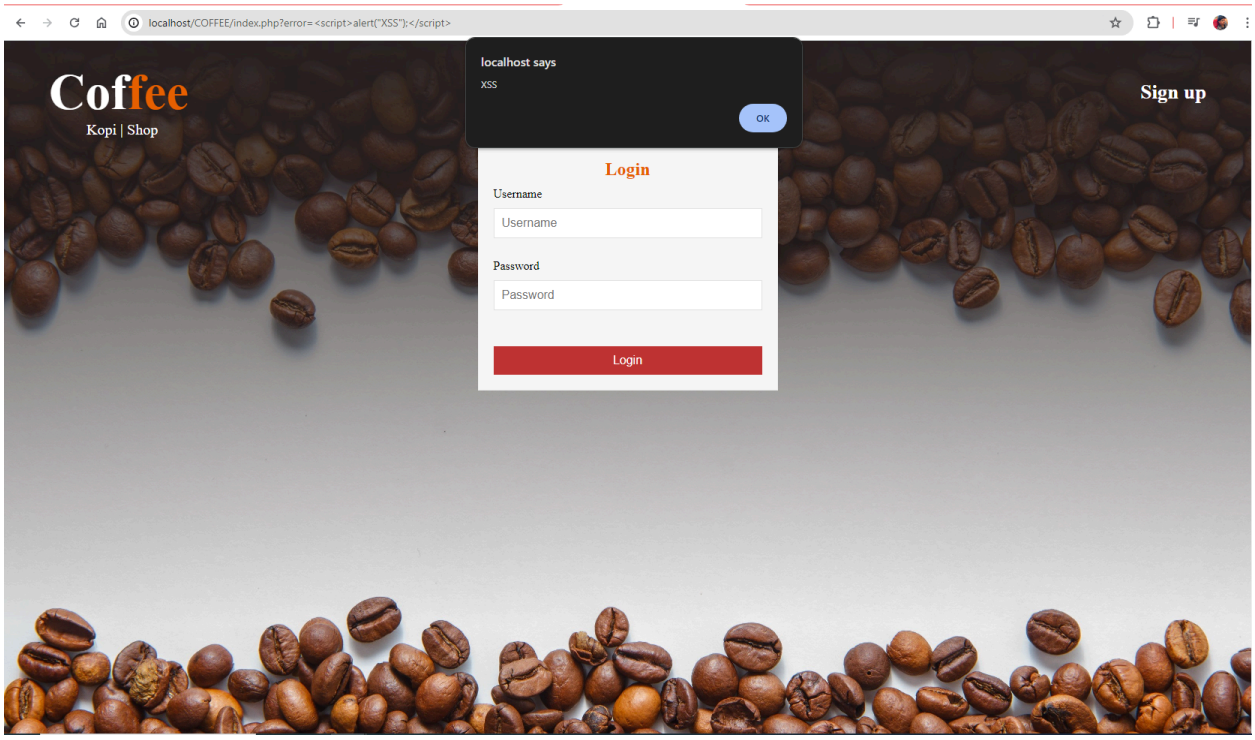


Cross-site scripting (XSS): We removed the error part of the php that causes the url to echo an alert message..

XSS script that we used:

http://localhost/COFFEE/index.php?error=<script>alert("XSS");</script>

The picture below was taken from before the mitigation process.

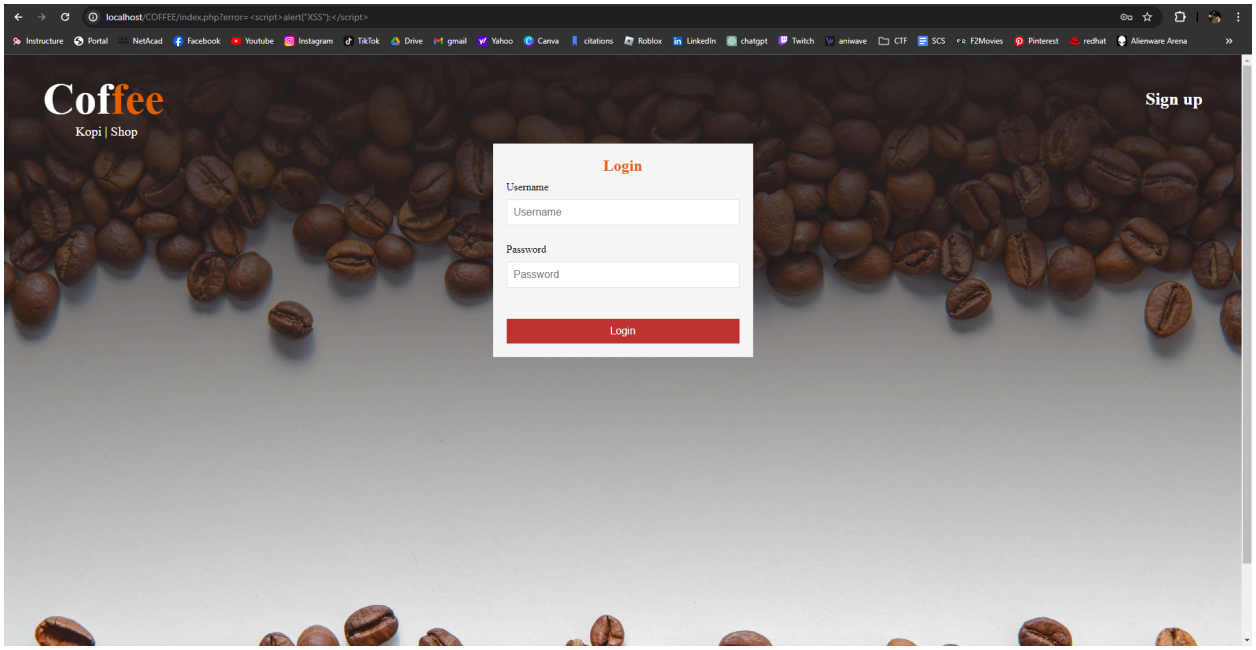


This is the code that we removed.

```
<?php
// (A) PROCESS ORDER FORM
if (isset($_GET['error'])) { ?>

    <p class="error"><?php echo $_GET['error']; ?></p>

<?php } ?>

<input type="submit" value="Login">

</form>
```

It was an unnecessary part of the code as it only echos the error giving an open vulnerability. Removing it does not affect uploading on the website as an admin and it also does not affect the login in any form.

The picture below shows that when the XSS is executed on the url, no alert sign shows anymore.



SQL Injection: The website is vulnerable to one because the database on the first version was not secured and is connected to the login.php.

- We implemented a signup form that allows any kind of signup for the username and password but also strings them to lowercase so that they may not use any payloads that are sensitive to characters, the signup also stops the users from using the username admin, so even if they use or bypass the login they would still be logged in as a user and not as the admin.

Payloads used:

' OR 1=1;--
Evaluates to true bypassing the login authentication
SELECT * FROM users WHERE user_name='' OR 1=1;--' AND password='$pass'

Or you can try another payload which is:

' OR '1'='1

SELECT * FROM users WHERE user_name='$uname' AND password='' OR '1'='1'

The user inputs are already validated so no one can access the admin except the administrator user only and only allows the specific password set by the admin.

Real-World Application and Impact

There are practical applications and effects for the login system found on coffee shop websites. The business promotes customer engagement with loyalty programs and its experience based on before-placed orders and stored preferences. It mitigates a risk associated with unauthorized access by employing encryption techniques for sensitive data like passwords and implementing account lockout policies to defend against brute-force attacks. Additionally, the customer’s data is secure to protect privacy and comply with regulations. Ensuring the integrity and confidentiality of customer information.

According to the South China Morning Post, the exposure of confidential data due to security weaknesses raises concerns about the adequacy of cybersecurity practices within the local government. The hackers gained access because of weak passwords for example “Admin123”. However, the importance of implementing strong authentication methods is very essential due to the generation today.

As reported by GMA News Online on March 31, 2013. Website hackers claim Saudi Arabia was behind the attack. In the edited current of the LTO government website, the hacker group posted two Hotmail addresses with the image "Hacked by Bin Laden Hacker of Saudi Arabia Hacker" on a black background. This incident points out how

important it is that all local governments maintain cybersecurity defenses, including putting cybersecurity protocols in place. By taking the appropriate measures, web applications can be better secured against cyberattacks and the integrity of online platforms can be maintained.

On the other hand, sensitive data was breached by hackers in the case of the Land Transportation Office (LTO). In the LTO's web application. Hackers may have taken advantage of these vulnerabilities which could have included coding vulnerabilities in the website or lacking security measures to extract sensitive personal data from the application including vehicle registration information and other sensitive information. The impact of the breach goes beyond simply compromising data to include the local government systems and exposing people to risks like fraud and identity theft.

It Is important to prioritize cybersecurity in protecting a website from potential attacks and to show that an organization is committed to protecting the security and privacy of its users or customers. Maintaining the integrity of online interactions and developing user trust rely on providing a basic level of security through measures like password encryption and account lockout policies.

Conclusion

Limiting user access and interactions on a website, sessions act as gatekeepers and implement techniques such as strict authentication protocols, session timeouts, etc. Securing the session tokens is necessary for proper session management and to lessen the possibility of unauthorized access, session hijacking, and other malicious exploits by thoroughly monitoring session lifecycles and activity. The code is like the foundation that holds everything therefore it is essential to keep it safe if the code is weak and the hackers can steal data or take over the website it can be very harmful. Using security measures. Such as testing and scanning the code to find vulnerabilities inspecting it for errors or modifications and regularly updating it to keep it current and stronger with the newest cybersecurity innovations available. These procedures help ensure that the website can fend off online attacks and remain secure for users to access.

References

https://newsinfo.inquirer.net/1846178/hackers-deface-website-of-house-of-representatives-2

https://www.yugatech.com/news/breaking-government-websites-hacked-and-defaced-by-indonesian-defacer-team/

https://www.scmp.com/week-asia/politics/article/3238687/philippines-cybersecurity-failures-display-hackers-expose-state-secrets-peoples-data?campaign=3238687&module=perpetual_scroll_0&pgtype=article

https://www.gmanetwork.com/news/scitech/technology/301770/saudi-hackers-deface-lto-website/story/

https://mb.com.ph/2020/11/14/how-hackers-collected-sensitive-data-from-the-land-transportation-office/

https://stackoverflow.com/questions/59331942/prevent-user-from-accessing-admin-php

https://stackoverflow.com/questions/22069398/php-session-multi-user-and-admin-privileges

Appendices

Not Applicable.