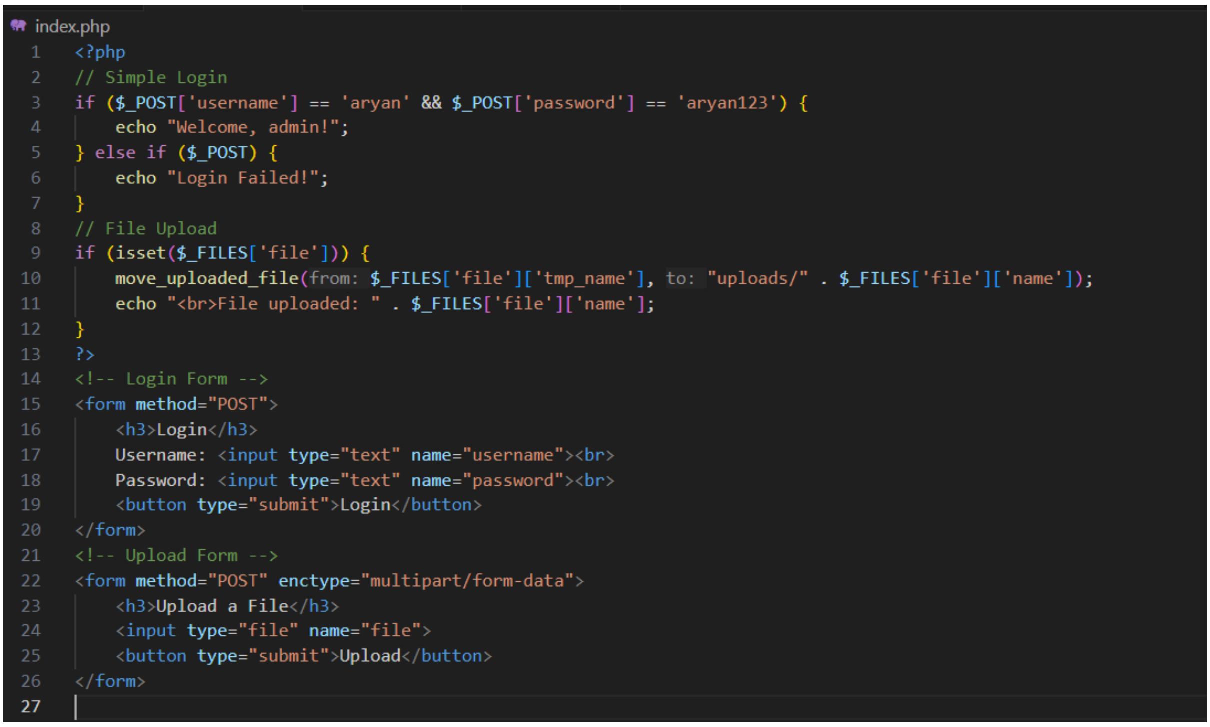
Cybersecurity Intern

TASKS

Task : Review a sample authentication & file upload code

Solution: I have Wrote a sample code in PHP .



index.php

```
<?php
// Simple Login
if ($_POST['username'] == 'aryan' && $_POST['password'] == 'aryan123') {
    echo "Welcome, admin!";
} else if ($_POST) {
    echo "Login Failed!";
}
// File Upload
if (isset($_FILES['file'])) {
    move_uploaded_file(from: $_FILES['file']['tmp_name'], to: "uploads/" . $_FILES['file']['name']);
    echo "<br>File uploaded: " . $_FILES['file']['name'];
}
?>
<!-- Login Form -->
<form method="POST">
    <h3>Login</h3>
    Username: <input type="text" name="username"><br>
    Password: <input type="text" name="password"><br>
    <button type="submit">Login</button>
</form>
<!-- Upload Form -->
<form method="POST" enctype="multipart/form-data">
    <h3>Upload a File</h3>
    <input type="file" name="file">
    <button type="submit">Upload</button>
</form>

```

Task : Identify 5 vulnerabilities And Provide fixes or mitigations

1. Not secure authentication

The username and password are written directly in the code.

-**Fix**: Use a database and store passwords securely using hashing.

2. No Input Cleaning(Cross-Site Scripting (XSS))

The uploaded filename is echoed directly back to the browser (echo "<br>File uploaded: " . $_FILES['file']['name'];). If someone uploads a file named with HTML or JavaScript code, it could lead to cross-site scripting (XSS).

-**Fix**: Use htmlspecialchars() to avoid script attacks.

3. Improper File Validation

Any file type can be uploaded, even dangerous ones like .php or .exe.

-**Fix**: Allow only safe files like .jpg, .png, .pdf.

4. Both Forms Use POST

Login and upload forms use the same method, which can cause confusion.
This mixing of logic can lead to unexpected behavior or bugs, especially as the code grows.

-**Fix**: Use a hidden input (like <input name="form" value="login">) to separate actions.

5. No HTTPS / Secure Communication

Passwords and files are sent without encryption ,which can be intercepted by attackers on public Wi-Fi or insecure networks . this can have user data.

-**Fix**: Use HTTPS to keep data safe and use use SSL/TLS certificates.