**DESIGN/DEVELOPMENT OF COURSE REGISTRATION WEB APPLICATION**

|  |  |  |
| --- | --- | --- |
| **S/N** | **NAME** | **MATRIC NUMBER** |
| **1.** | **AL-AMEEN EMIGI MUSTAPHA** | **2019/1/76004CS** |
| **2.** | **NUHUDEEN AMODU** | **2021/2/79850CS** |
| **3.** | **SALIM MUHAMMAD MIKHAIL** | **2019/1/75975CS** |
| **4.** | **OBINNA JEREMIAH NNAJI** | **2019/1/76035CS** |
| **5.** | **KELVIN TERFA AGBE** | **2019/1/75031CS** |
| **6.** | **ISMAIL IBRAHIM** | **2019/1/75635CS** |

**GROUP ELEVEN (11)**

**MARCH 27TH, 2024**

Overview of Website Payment for Course Registration

Functionality

This website facilitates course registration for students at the Federal University of Technology, Minna. Before accessing course registration, students are required to log in using their student ID and password.

Workflow

1. \*\*Login Page (`login.html`):\*\*

- Displays a form for students to enter their credentials (student ID and password).

- Validates the entered credentials against the database.

- Redirects authenticated users to the dashboard upon successful login.

2. \*\*Database Interaction (`slogin.php`):\*\*

- Connects to the database (`group7`) on the localhost server.

- Retrieves student details from the `group7css418` table based on the provided student ID.

- Compares the entered password with the password stored in the database.

- Authenticates the user and starts a session upon successful login.

#### Security Mechanisms

- \*\*Password Hashing:\*\* Ensure passwords are stored securely in the database using cryptographic hashing functions.

- \*\*SQL Injection Prevention:\*\* Use prepared statements (`mysqli\_prepare`) to protect against SQL injection attacks.

### Login Page (`login.html`)

FRONTEND CODE

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="indexstyle.css">

<title>Login</title>

</head>

<body>

<main>

<section>

<img src="fut.png" alt="futninna" id="futpng">

<div id="FEDERAL">FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA</div>

</section>

<div class="container">

<div class="row justify-content-center">

<div class="col-sm-6">

<form action="slogin.php" method="POST" id="login-form">

<div class="form-group">

<label for="matric-number">Student ID</label>

<input type="text" class="form-control" id="matric-number" name="matric-number" required>

</div>

<div class="form-group">

<label for="password">Password</label>

<input type="password" class="form-control" id="password" name="password" required>

</div>

<button type="submit" class="btn btn-primary btn-block">Login</button>

</form>

<div id="message"></div>

<div class="text-center mt-3">

<a href="forgot-password.html" id="forgot-password-link">Forgot Password?</a>

</div>

</div>

</div>

</div>

</main>

</body>

</html>

```

BACKEND CODE

<?php

session\_start();

$host = "localhost";

$username = "root";

$password = "";

$database = "group7";

$connect = mysqli\_connect($host, $username, $password, $database);

if (isset($\_POST['matric-number'], $\_POST['password'])) {

$matricNumber = $\_POST['matric-number'];

$password = $\_POST['password'];

$query = "SELECT \* FROM group7css418 WHERE STUDENT\_ID = ?";

$stmt = mysqli\_prepare($connect, $query);

if ($stmt) {

mysqli\_stmt\_bind\_param($stmt, "s", $matricNumber);

mysqli\_stmt\_execute($stmt);

$result = mysqli\_stmt\_get\_result($stmt);

if ($result && $row = mysqli\_fetch\_assoc($result)) {

$dbPassword = $row['PASSWORD'];

if ($password === $dbPassword) {

$\_SESSION['STUDENT\_ID'] = $matricNumber;

header('Location: dashboard.php'); // Redirect to dashboard page

exit();

} else {

echo '<div class="alert alert-danger" role="alert">Invalid password.</div>';

}

} else {

echo '<div class="alert alert-danger" role="alert">Invalid Student ID.</div>';

}

} else {

echo '<div class="alert alert-danger" role="alert">Database query error.</div>';

}

}

?>

#### Notes:

- proper error handling and validation mechanisms are implemented to enhance security and user experience.

- HTTPS for secure communication, especially when transmitting sensitive information like passwords.

### Secure Payment Confirmation Page Overview

#### Functionality

The payment confirmation page (`confirm\_payment.html`) allows users to confirm their payment by entering their matriculation number. Upon submission, if the matriculation number is valid, the page displays a success message and provides options to register for courses or return to the homepage.

#### Workflow

1. \*\*Payment Confirmation Form (`confirm\_payment.html`):\*\*

- Presents a form for users to input their matriculation number.

- Displays a message area (`<div id="message"></div>`) to show payment status and actions.

2. \*\*JavaScript Logic (`script.js`):\*\*

- Handles form submission using event listeners to prevent default form behavior.

- Validates the entered matriculation number against a predefined valid number (`'2021/2/2354cs'` in this example).

- Displays appropriate messages and actions based on the validation result:

- If the matriculation number is valid, shows a success message with options to register for courses or return home.

- If the matriculation number is invalid, displays an error message indicating the number is invalid.

#### Security Mechanisms

1. \*\*Input Validation\*\*

- Implements client-side validation to ensure that the entered matriculation number is in the correct format and length.

- Performs server-side validation to verify the matriculation number against authorized records in the backend database.

2. \*\*Secure Communication\*\*

- Utilizes HTTPS (HTTP over SSL/TLS) to encrypt data transmitted between the client's browser and the server, preventing eavesdropping and man-in-the-middle attacks.

3. \*\*Error Handling\*\*

- Implements robust error handling to gracefully manage invalid inputs and prevent potential information leakage (e.g., generic error messages).

4. \*\*Access Control\*\*

- Ensures that only authorized users with valid matriculation numbers can proceed with the payment confirmation process.

- Implements proper session management and authentication mechanisms to validate user identities before processing sensitive actions.

5. \*\*Prevention of Brute Force Attacks\*\*

- Implements rate-limiting and CAPTCHA challenges to prevent automated or brute-force attacks on the payment confirmation form.

6. \*\*Data Privacy and Confidentiality\*\*

- Adheres to data protection regulations (e.g., GDPR, HIPAA) to safeguard user information and ensure confidentiality of matriculation numbers and payment details.

7. \*\*Regular Security Audits and Updates\*\*

- Conducts regular security audits and vulnerability assessments to identify and mitigate potential security vulnerabilities in the application.

- Ensures all software components (libraries, dependencies) are kept up to date with the latest security patches and updates.

By incorporating these security mechanisms, the payment confirmation page can provide a secure and reliable user experience, safeguarding user data and preventing unauthorized access or manipulation of payment-related actions. Security should be a foundational consideration in the design, development, and deployment of web applications to maintain user trust and compliance with security standards.

FRONTEND CODE

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Confirm Payment</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="container">

<h1>Confirm Payment</h1>

<form id="payment-form">

<label for="matric-number">Matric Number:</label>

<input type="text" id="matric-number" name="matric-number" required>

<br>

<button type="submit">Submit</button>

</form>

<div id="message"></div>

</div>

<script src="script.js"></script>

</body>

</html>

BACKEND CODE

const form = document.getElementById('payment-form');

const message = document.getElementById('message');

form.addEventListener('submit', (event) => {

event.preventDefault();

const matricNumber = document.getElementById('matric-number').value;

if (matricNumber === '2021/2/2354cs') {

message.innerHTML = `

<p class="success">Payment Successful!</p>

<p>Would you like to:</p>

<button id="register-course">Register Course</button>

<button id="go-home">Go to Home</button>

`;

const registerCourse = document.getElementById('register-course');

const goHome = document.getElementById('go-home');

registerCourse.addEventListener('click', () => {

window.location.href = '../verify/verify.html'; // Replace with your course registration page

});

goHome.addEventListener('click', () => {

window.location.href = '../dashboard.html'; // Assuming this is your home page

});

} else {

message.textContent = 'Invalid Matric Number'; // Display error for invalid matric number

}

});

SCHOOL FEE PAYMENT PAGE

To document the HTML and JavaScript code for the school fee payment form and associated functionality, I'll provide structured documentation following a clear format. We'll cover both the HTML structure and the JavaScript functionality with appropriate comments.

### HTML Structure

#### Purpose

This HTML document represents a school fee payment form where students can enter their matriculation number to initiate the payment process.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>School Fee Payment</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="container">

<h2>School Fee Payment</h2>

<form id="paymentForm">

<label for="matricNumber">Matric Number:</label>

<input type="text" id="matricNumber" name="matricNumber" required><br><br>

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

</form>

</div>

<script src="script.js"></script>

</body>

</html>

BACKEND CODE

document.getElementById('paymentForm').addEventListener('submit', function(event) {

event.preventDefault();

const matricNumber = document.getElementById('matricNumber').value;

if (matricNumber === '2021/2/2354cs') {

// Successful payment message and course registration prompt

alert('Payment Successful!\n\nName: Nuhu Amodu\nDepartment: Cyber Security Science');

const registerCourse = confirm('Do you want to register for courses?');

if (registerCourse) {

window.location.href = '../verify/verify.html';

} else {

window.location.href = '../dashboard.html';

}

} else if (matricNumber === '2021/2/6798cs') {

// Unsuccessful payment message and RRR generation

const rrr = generateRRR();

alert(`Payment Unsuccessful!\n\nRRR: ${rrr}`);

const proceedToPay = confirm('Do you want to proceed to pay online?');

if (proceedToPay) {

window.location.href = 'payment\_link.html';

} else {

window.location.href = '../dashboard.html';

}

} else {a

alert('Invalid Matric Number');

}

});

function generateRRR() {

// Generate a random 12-digit number

const rrr = Math.floor(Math.random() \* 1000000000000).toString().padStart(12, '0');

return rrr;

}

#### Comments Explanation

- \*\*Event Listener\*\*: Listens for form submission (`submit` event) and prevents the default form submission behavior.

- \*\*Retrieve Matriculation Number\*\*: Fetches the entered matriculation number from the form input field.

- \*\*Payment Flow\*\*:

- If the matriculation number matches a predefined value (`2021/2/2354cs`), a successful payment message is shown, and the user is prompted to register for courses or redirected to the dashboard.

- If the matriculation number matches another predefined value (`2021/2/6798cs`), an unsuccessful payment message is shown with a randomly generated RRR. The user is prompted to proceed to pay online or redirected to the dashboard.

- If the matriculation number doesn't match any predefined value, an alert notifies the user of an invalid matriculation number.

**### Secure Course Selection and Submission System Overview**

#### Functionality

The course selection system allows students to choose courses for the academic session and submit their selections securely. The system includes a web interface (`index.html`) for course selection, JavaScript (`script.js`) for handling user interactions, and a server-side PHP script (`submit\_selected\_courses.php`) to process and store selected courses in a database.

#### Workflow

1. \*\*Course Selection Interface (`index.html`):\*\*

- Displays available courses organized by semester in HTML tables.

- Allows students to select desired courses using checkboxes.

- Calculates and displays the total credit units of selected courses.

- Provides a button to submit selected courses.

2. \*\*JavaScript Functionality (`script.js`):\*\*

- Computes the total credits of selected courses based on checkbox states.

- Gathers selected courses' data (code, title, unit) into an array.

- Sends the selected courses data to the server-side script (`submit\_selected\_courses.php`) using AJAX for processing and storage.

3. \*\*Server-side Processing (`submit\_selected\_courses.php`):\*\*

- Establishes a database connection to store selected courses securely.

- Receives and processes selected courses data sent from the client-side.

- Performs data validation and sanitization to prevent SQL injection and other security vulnerabilities.

- Inserts validated course data into the database table (`selected\_courses`).

#### Security Mechanisms

1. \*\*Secure Communication\*\*

- Utilizes HTTPS to encrypt data transmitted between the client and the server, ensuring confidentiality and integrity of information.

2. \*\*Input Validation and Sanitization\*\*

- Implements client-side and server-side validation to ensure that user inputs (e.g., checkbox states, course data) are valid and safe for processing.

- Uses parameterized queries in PHP to prevent SQL injection attacks when interacting with the database.

3. \*\*Data Privacy and Storage\*\*

- Stores sensitive data (e.g., course details) in a secure database using proper encryption and access control measures.

- Avoids storing unnecessary or sensitive user information beyond what is required for course selection and processing.

4. \*\*Session Management\*\*

- Implements secure session management techniques to authenticate and authorize users before allowing course selection and submission.

- Uses CSRF (Cross-Site Request Forgery) tokens to prevent unauthorized requests from malicious websites.

5. \*\*Error Handling and Logging\*\*

- Implements robust error handling mechanisms in both client-side and server-side code to gracefully handle unexpected scenarios and prevent information disclosure.

- Logs security-relevant events and errors for auditing and monitoring purposes.

6. \*\*Code Security Practices\*\*

- Adheres to secure coding practices to mitigate common vulnerabilities such as XSS (Cross-Site Scripting) and CSRF.

7. \*\*Regular Security Audits and Updates\*\*

- Conducts periodic security audits and vulnerability assessments to identify and remediate potential security weaknesses in the application.

- Ensures all software components (frameworks, libraries, server software) are kept up to date with the latest security patches and updates.

SOURCE CODE

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

<title>CSS418PROJECT</title>

<link href="styles.css" rel="stylesheet">

</head>

<body>

<ul>

<li><a href="index.html">DASHBOARD</a></li>

<li><a href="login.html">LOGOUT</a></li>

</ul>

<center><h2>SESSION: 2022/2023</h2></center>

<center><h4>1ST SEMESTER COURSES</h4></center>

<center>

<table id="semester1" border="1" cellspacing="5" cellpadding="5">

<tr bgcolor="gray">

<th>COURSE CODE</th>

<th>COURSE TITLE</th>

<th>UNIT</th>

<th>SEMESTER</th>

<th>NATURE</th>

<th>Select</th>

</tr>

<tr>

<td>CPT141</td>

<td>Information Security Engineering</td>

<td>3</td>

<td>1ST</td>

<td>Core</td>

<td><input type="checkbox" class="course-checkbox" data-code="CPT141" data-title="Systems Analysis and Design" data-unit="3"></td>

</tr>

<tr bgcolor="gray">

<th>COURSE CODE</th>

<th>COURSE TITLE</th>

<th>UNIT</th>

<th>SEMESTER</th>

<th>NATURE</th>

<th>Select</th>

</tr>

<tr>

<td>CPT141</td>

<td>Algorithms and Complexity ANalysis</td>

<td>3</td>

<td>1ST</td>

<td>Core</td>

<td><input type="checkbox" class="course-checkbox" data-code="CPT141" data-title="Systems Analysis and Design" data-unit="3"></td>

</tr> <tr bgcolor="gray">

<th>COURSE CODE</th>

<th>COURSE TITLE</th>

<th>UNIT</th>

<th>SEMESTER</th>

<th>NATURE</th>

<th>Select</th>

</tr>

<tr>

<td>CPT141</td>

<td>Entrepreneurship I</td>

<td>3</td>

<td>1ST</td>

<td>Core</td>

<td><input type="checkbox" class="course-checkbox" data-code="CPT141" data-title="Systems Analysis and Design" data-unit="3"></td>

</tr> <tr bgcolor="gray">

<th>COURSE CODE</th>

<th>COURSE TITLE</th>

<th>UNIT</th>

<th>SEMESTER</th>

<th>NATURE</th>

<th>Select</th>

</tr>

<tr>

<td>CPT141</td>

<td>Enterprise and Perimeter Security </td>

<td>3</td>

<td>1ST</td>

<td>Core</td>

<td><input type="checkbox" class="course-checkbox" data-code="CPT141" data-title="Systems Analysis and Design" data-unit="3"></td>

</tr> <tr bgcolor="gray">

<th>COURSE CODE</th>

<th>COURSE TITLE</th>

<th>UNIT</th>

<th>SEMESTER</th>

<th>NATURE</th>

<th>Select</th>

</tr>

<tr>

<td>CPT141</td>

<td>Forensic Analysis</td>

<td>3</td>

<td>1ST</td>

<td>Core</td>

<td><input type="checkbox" class="course-checkbox" data-code="CPT141" data-title="Systems Analysis and Design" data-unit="3"></td>

</tr>

</table>

<center><h4>2 SEMESTER COURSES</h4></center>

<center>

<table id="semester1" border="1" cellspacing="5" cellpadding="5">

</tr> <tr bgcolor="gray">

<th>COURSE CODE</th>

<th>COURSE TITLE</th>

<th>UNIT</th>

<th>SEMESTER</th>

<th>NATURE</th>

<th>Select</th>

</tr>

<tr>

<td>CPT141</td>

<td>threat Exploits and Countermeasures</td>

<td>3</td>

<td>1ST</td>

<td>Core</td>

<td><input type="checkbox" class="course-checkbox" data-code="CPT141" data-title="Systems Analysis and Design" data-unit="3"></td>

</tr>

<!-- Add other course rows similarly -->

<tr bgcolor="gray">

<th>COURSE CODE</th>

<th>COURSE TITLE</th>

<th>UNIT</th>

<th>SEMESTER</th>

<th>NATURE</th>

<th>Select</th>

</tr>

<tr>

<td>CPT141</td>

<td>computer and Network Security</td>

<td>3</td>

<td>1ST</td>

<td>Core</td>

<td><input type="checkbox" class="course-checkbox" data-code="CPT141" data-title="Systems Analysis and Design" data-unit="3"></td>

</tr><tr bgcolor="gray">

<th>COURSE CODE</th>

<th>COURSE TITLE</th>

<th>UNIT</th>

<th>SEMESTER</th>

<th>NATURE</th>

<th>Select</th>

</tr>

<tr>

<td>CPT141</td>

<td>web Hacking </td>

<td>3</td>

<td>1ST</td>

<td>Core</td>

<td><input type="checkbox" class="course-checkbox" data-code="CPT141" data-title="Systems Analysis and Design" data-unit="3"></td>

</tr><tr bgcolor="gray">

<th>COURSE CODE</th>

<th>COURSE TITLE</th>

<th>UNIT</th>

<th>SEMESTER</th>

<th>NATURE</th>

<th>Select</th>

</tr>

<tr>

<td>CPT141</td>

<td>Computational Science and Numeric Methods</td>

<td>3</td>

<td>1ST</td>

<td>Core</td>

<td><input type="checkbox" class="course-checkbox" data-code="CPT141" data-title="Systems Analysis and Design" data-unit="3"></td>

</tr><tr bgcolor="gray">

<th>COURSE CODE</th>

<th>COURSE TITLE</th>

<th>UNIT</th>

<th>SEMESTER</th>

<th>NATURE</th>

<th>Select</th>

</tr>

<tr>

<td>Css441</td>

<td>Biometrics Security</td>

<td>3</td>

<td>1ST</td>

<td>Core</td>

<td><input type="checkbox" class="course-checkbox" data-code="CPT141" data-title="Systems Analysis and Design" data-unit="3"></td>

</tr>

</table>

</center>

<center><p>Total Credit Units: <span id="totalCredits">0</span></p></center>

<br>

<button onclick="calculateTotalCredits()">Total Credits</button>

<button onclick="submitSelectedCourses()">Submit</button>

<script>

function calculateTotalCredits() {

let totalCredits = 0;

const checkboxes = document.querySelectorAll('.course-checkbox:checked');

checkboxes.forEach((checkbox) => {

const unit = parseInt(checkbox.getAttribute('data-unit'), 10);

if (!isNaN(unit)) {

totalCredits += unit;

}

});

document.getElementById('totalCredits').textContent = totalCredits;

}

function submitSelectedCourses() {

const selectedCourses = [];

const checkboxes = document.querySelectorAll('.course-checkbox:checked');

checkboxes.forEach((checkbox) => {

const code = checkbox.getAttribute('data-code');

const title = checkbox.getAttribute('data-title');

const unit = parseInt(checkbox.getAttribute('data-unit'));

selectedCourses.push({ code, title, unit });

});

// Send selectedCourses data to the server-side PHP script using AJAX

const xhr = new XMLHttpRequest();

xhr.open('POST', 'submit\_selected\_courses.php');

xhr.setRequestHeader('Content-Type', 'application/json');

xhr.onload = function() {

if (xhr.status === 200) {

// Redirect to a success page or handle response as needed

window.location.href = 'success.html';

} else {

// Handle error response

console.error('Error submitting selected courses:', xhr.statusText);

alert('Failed to submit selected courses. Please try again.');

}

};

xhr.onerror = function() {

console.error('Request failed:', xhr.statusText);

alert('Failed to submit selected courses. Please check your internet connection.');

};

xhr.send(JSON.stringify(selectedCourses));

}

</script>

</body>

</html>

BACKEND CODE

<?php

session\_start();

// Database connection parameters

$host = "localhost";

$username = "root";

$password = "";

$database = "group7";

// Establish database connection

$connect = mysqli\_connect($host, $username, $password, $database);

// Check if the request contains JSON data

$postData = file\_get\_contents("php://input");

$selectedCourses = json\_decode($postData, true);

if ($selectedCourses && is\_array($selectedCourses)) {

foreach ($selectedCourses as $course) {

$code = mysqli\_real\_escape\_string($connect, $course['code']);

$title = mysqli\_real\_escape\_string($connect, $course['title']);

$unit = (int)$course['unit'];

// Insert selected course into the database

$query = "INSERT INTO selected\_courses (course\_code, course\_title, credit\_unit) VALUES ('$code', '$title', $unit)";

$result = mysqli\_query($connect, $query);

if (!$result) {

echo "Error inserting course: " . mysqli\_error($connect);

exit;

}

}

// Respond with success status (HTTP 200)

http\_response\_code(200);

echo "Selected courses submitted successfully.";

} else {

// Respond with error status (HTTP 400)

http\_response\_code(400);

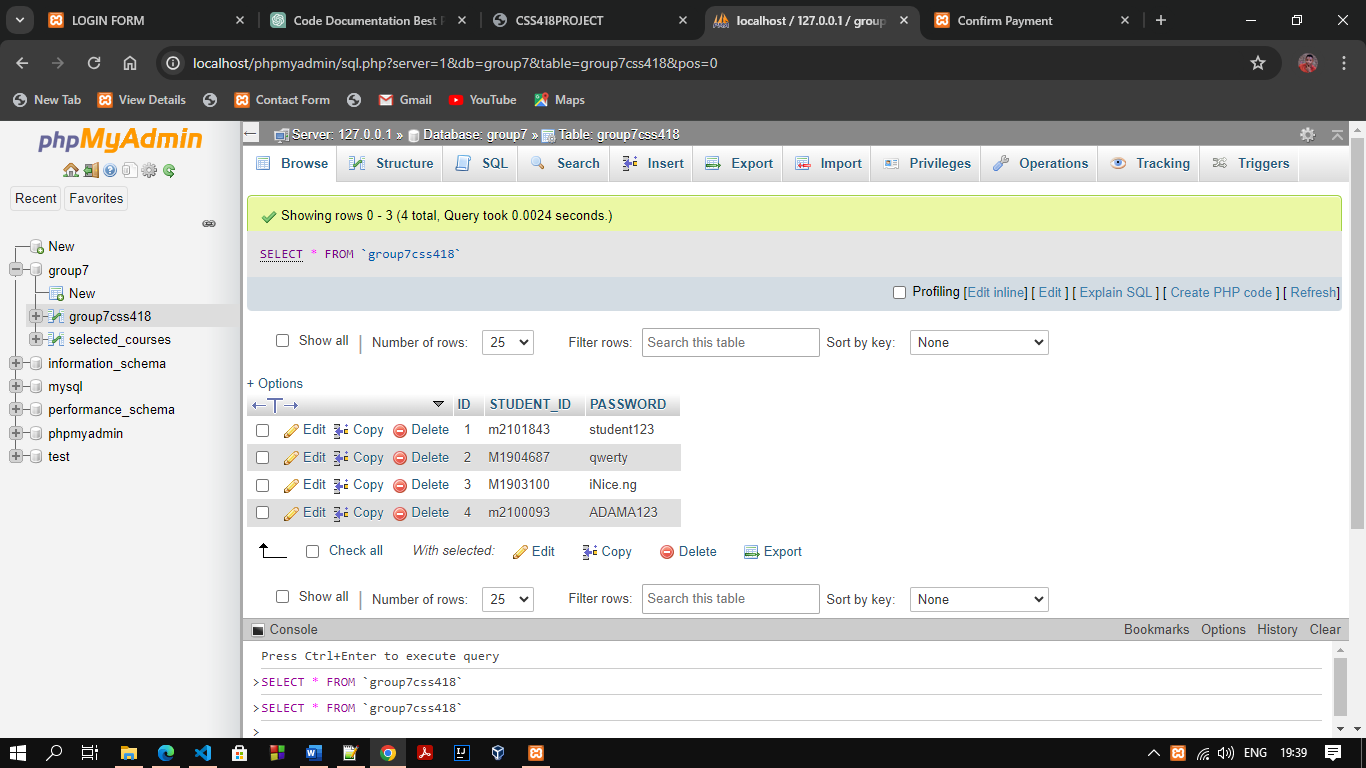
echo "Invalid request data.";

}

?>

‘

STUDENT USERNAME AND PASSWORD



COURSE REGISTRATION IN THE DATA BASE 