

## Midterm Practical Exam

**Duration:** 5 Hours

**Group Composition:** 4 Members

Each group will develop a **Product Management System** (prodMS) using **PHP, MySQL (XAMPP), and session management**. The project must include **user authentication, product management, and proper database handling**.

---

### Task Breakdown per Member:

#### Member 1: Database Setup & User Account Creation

- **Task:**
    - Set up a **MySQL database named prodMS** using XAMPP.
    - Create **one (1) admin account** to be used for logging into the system.  
Username: **admin** password: **admin123**
    - Share the credentials with **Member 3** for authentication setup.
  - **Database Connection:**
    - The database connection credentials should be stored securely and accessed by other members when needed.
- 

#### Member 2: Table Creation (**users & products**) with Employee Number Validation

- **Task:**
  - Create **two tables** inside the prodMS database:

#### User Table (with Unique Employee Number Validation)

Column Name	Data Type	Description
userID	INT (Primary Key, Auto Increment)	Unique user identifier
Userfullname	VARCHAR(255) NOT NULL	Full name of the user

<b>Column Name</b>	<b>Data Type</b>	<b>Description</b>
employeeNo	VARCHAR(50) UNIQUE NOT NULL	Unique employee number (must not be duplicated)
password	VARCHAR(255) NOT NULL	Hashed password using bcrypt
DateCreated	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	Date when the user is created

### **Product Table**

<b>Column Name</b>	<b>Data Type</b>	<b>Description</b>
prodID	INT (Primary Key, Auto Increment)	Unique product identifier
ProductName	VARCHAR(255) NOT NULL	Name of the product
ProductQTY	INT NOT NULL	Quantity of the product
ProductPrice	DECIMAL(10,2) NOT NULL	Price of the product
UserFullscreen	VARCHAR(255) NOT NULL	The session user's name who created the product
dateCreated	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	Date when the product is created

- **Validation for employeeNo:**

- Ensure **employeeNo** is **unique** before inserting a new user.
- If a duplicate is found, return an **error message** and prevent registration.

### **Member 3: User Registration & Login System with Unique Employee Number Validation**

- **Task:**

- Create a **user registration form** that:
  - Stores user details.

- Hashes passwords using **bcrypt**.
- Checks if the employeeNo already exists before inserting data.

- Implement a **login system** that verifies user credentials and starts a session.
- Upon login, store the user's **Full Name** in a session variable to be used by Member 4.

Create a file name **db.php**:

```
<?php
$servername = "localhost";
$username = "admin";
$password = "admin123";
$dbname = "prodMS";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

Create a file named **login.php**:

```
<form method="post">
    <input type="text" name="employeeNo" placeholder="Employee Number" required>
    <input type="password" name="password" placeholder="Password" required>
    <button type="submit" name="login">Login</button>
    <a href="register.php">Register Account</a>
</form>

<?php
session_start();
include 'db.php';

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['login'])) {
    $employeeNo = trim($_POST['employeeNo']);
    $password = trim($_POST['password']);

    $stmt = $conn->prepare("SELECT * FROM users WHERE employeeNo = ?");
    $stmt->bind_param("s", $employeeNo);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $user = $result->fetch_assoc();
        if (password_verify($password, $user['password'])) {
            $_SESSION['UserFullname'] = htmlspecialchars($user['UserFullname'], ENT_QUOTES, 'UTF-8');
            header("Location: dashboard.php");
            exit();
        } else {
            echo "Incorrect password!";
        }
    } else {
        echo "User not found!";
    }
    $stmt->close();
}
?>
```

Create a file named **register.php**:

```
<?php
include 'db.php';
if ($_SERVER['REQUEST_METHOD'] == "POST" && isset($_POST['register'])) {
    $fullname = htmlspecialchars(trim($_POST['fullname']), ENT_QUOTES, 'UTF-8');
    $employeeNo = trim($_POST['employeeNo']);
    $password = password_hash(trim($_POST['password']), PASSWORD_BCRYPT);

    $stmt = $conn->prepare("SELECT employeeNo FROM users WHERE employeeNo = ?");
    $stmt->bind_param("s", $employeeNo);
    $stmt->execute();
    $stmt->store_result();

    if ($stmt->num_rows > 0) {
        echo "Employee Number already used";
    } else {
        $stmt = $conn->prepare("INSERT INTO users (UserFullname, employeeNo, password) VALUES (?, ?, ?)");
        $stmt->bind_param("sss", $fullname, $employeeNo, $password);
        if ($stmt->execute()) {
            echo "User registered successfully!";
        } else {
            echo "Error: " . $stmt->error;
        }
    }
    $stmt->close();
}

?>

<form method="post">
    <input type="text" name="fullname" placeholder="Full Name" required>
    <input type="text" name="employeeNo" placeholder="Employee Number" required>
    <input type="password" name="password" placeholder="Password" required>
    <button type="submit" name="register">Register</button>
    <a href="login.php">Login</a>
</form>
```

**logout.php:**

```
<?php
session_start();
session_unset();
session_destroy();
header("Location: login.php");
exit();
```

## Member 4: Product Management (Add & Display)

- **Task:**
    - Create a **product form** where users can **add products** to the database.
    - Ensure the UserFullscreen field automatically **gets the logged-in user's name** from the session.
    - Display all products in a table format.
  - **Requirements:**
    - **Session must be required** before accessing this page.
    - **Display the logged-in user's name** on the dashboard.
    - Provide a **logout button** to destroy the session and return to the login page.
- 

### Additional Requirements:

#### **Session Handling:**

- Users must be required to log in before accessing the system.
- Display the **session user's name** on every page.
- Implement a **logout button** that destroys the session properly.

Create a file named **dashboard.php**:

```
<?php
session_start();
include 'db.php';
if (!isset($_SESSION['UserFullscreen'])) {
    header("Location: login.php");
    exit();
}

echo "<h2>Welcome, " . htmlspecialchars($_SESSION['UserFullscreen'], ENT_QUOTES, 'UTF-8') . "</h2>";
echo "<a href='logout.php'>Logout</a><br>";
echo "<a href='add_product.php'>Add Product</a>";

$query = "SELECT * FROM products";
$result = $conn->query($query);

echo "<table border='1'><tr><th>Name</th><th>Quantity</th><th>Price</th><th>Created By</th></tr>";
while ($row = $result->fetch_assoc()) {
    echo "<tr><td>" . htmlspecialchars($row['ProductName'], ENT_QUOTES, 'UTF-8') . "</td>";
    echo "<td>" . intval($row['ProductQty']) . "</td>";
    echo "<td>" . htmlspecialchars($row['ProductPrice'], ENT_QUOTES, 'UTF-8') . "</td>";
    echo "<td>" . htmlspecialchars($row['UserFullscreen'], ENT_QUOTES, 'UTF-8') . "</td></tr>";
}
echo "</table>";
```

Create a file named **add\_product.php**:

```
<?php session_start();

echo "<h2>Welcome, " . htmlspecialchars($_SESSION['UserFullscreen'], ENT_QUOTES, 'UTF-8') . "</h2>";
echo "<a href='dashboard.php'>Back to Dashboard</a>;

?>
<h2>Add Product</h2>
<form method="post">
    <input type="text" name="productName" placeholder="Product Name" required>
    <input type="number" name="productQty" placeholder="Quantity" required>
    <input type="text" name="productPrice" placeholder="Price" required>
    <button type="submit" name="addProduct">Add Product</button>
</form>
<?php
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}
include 'db.php';

if (!isset($_SESSION['UserFullscreen'])) {
    header("Location: login.php");
    exit();
}

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['addProduct'])) {
    $productName = htmlspecialchars(trim($_POST['productName']), ENT_QUOTES, 'UTF-8');
    $productQty = intval($_POST['productQty']);
    $productPrice = floatval($_POST['productPrice']);
    $UserFullscreen = $_SESSION['UserFullscreen'];

    $stmt = $conn->prepare("INSERT INTO products (ProductName, ProductQty, ProductPrice, UserFullscreen) VALUES (?, ?, ?, ?)");
    $stmt->bind_param("ssds", $productName, $productQty, $productPrice, $UserFullscreen);
    if ($stmt->execute()) {
        echo "Product added successfully!";
    } else {
        echo "Error: " . $stmt->error;
    }
    $stmt->close();
}
?>
```

