Sessió 8 - Cas pràctic d'Autenticació d'usuaris i seguretat web.

Aquest cas pràctic consisteix a desenvolupar una aplicació bàsica que gestioni usuaris amb diferents rols. Aquest projecte et permetrà practicar el que has après durant les sessions anteriors, com ara formularis, processament de dades, gestió de sessions i mesures de seguretat.

Objectiu General

Crear un sistema d'accés amb login i contrasenya que classifiqui els usuaris segons dos rols principals (*Administrador* i *Usuari*) i mostri per pantalla missatges diferents en funció del rol.

Requisits:

1. **Formulari d'Accés:**
    - Permetre als usuaris introduir el seu nom d'usuari i contrasenya.
    - Validar les credencials amb una base de dades MySQL.
    - Gestionar sessions per mantenir l'estat de l'usuari.
2. **Classificació per Rols:**
    - Assignar a cada usuari un dels dos rols disponibles:
        - **Administrador**
        - **Usuari**
3. **Funcionalitats segons el Rol:**
    - El rol determinarà quin missatge mostrem per pantalla. Per a l'Administrador mostrarem el missatge "Permís de gestió" i per a l'usuari mostrarem el missatge "Permís de visualització".
4. **Seguretat:**
    - Xifrat de contrasenyes utilitzant password_hash() i verificació amb password_verify().
    - Protecció contra *SQL Injection*.
    - Restricció d'accés segons el rol.

Pautes per al Desenvolupament

1. **Inicialització de la base de dades i inserció d'usuaris amb contrasenyes xifrades (***setup_database.php***):**
    - Crea una base de dades anomenada *sistema_acces* amb una taula usuaris que contingui els camps:

- id (int, clau primària, autoincremental).
- nom_usuari (varchar).
- contrasenya (varchar).
- rol (varchar, valors possibles: 'Administrador', 'Usuari').

Per exemple:

```
CREATE TABLE usuaris (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nom_usuari VARCHAR(50) NOT NULL UNIQUE,
  contrasenya VARCHAR(255) NOT NULL,
  rol ENUM('Administrador', 'Usuari') NOT NULL
);
```

- Insereix alguns usuaris de prova, amb contrasenyes xifrades des del codi PHP, per inicialitzar els usuaris. Per exemple:

```
// Insereix usuaris inicials
$passwordAdmin = password_hash("admin123", PASSWORD_DEFAULT);
$passwordUser = password_hash("user123", PASSWORD_DEFAULT);

$conn->query("

INSERT INTO usuaris (nom_usuari, contrasenya, rol)
VALUES
('admin', '$passwordAdmin', 'Administrador'), ('usuari', '$passwordUser',
'Usuari')

");
```

Aquest fitxer *setup_database.php* **ha de ser executat una sola vegada** per preparar la base de dades i, en cas d'èxit, ens retornarà un missatge dient que la base de dades, les taules i el contingut s'han creat correctament.

2. **Implementació del Formulari d'Accés:**
   - Dissenya un formulari senzill amb camps per a nom d'usuari i contrasenya *(login.php)*.
   - Implementa el backend en PHP per autenticar les credencials i iniciar sessió *(process_login.php)*.

3. **Pàgines per a funcionalitats:**
   - **Pàgina Administrativa** (*admin.php*)**:** Accessible només per a administradors. Ha de mostrar un missatge de benvinguda al rol corresponent (Administrador) i el missatge "Permís de gestió". Exemple: **"Benvingut, [Nom d'usuari]! Permís de gestió."** A més, inclou un enllaç per tancar la sessió i redirigir al formulari de login.

- **Pàgina d'Usuari** *(usuari.php)***:** Accessible per a tots els usuaris. Ha de mostrar un missatge de benvinguda al rol corresponent (Usuari) i el missatge “Permís de visualització”. Exemple: **"Benvingut, [Nom d'usuari]! Permís de visualització."** A més, inclou un enllaç per tancar la sessió i redirigir al formulari de login.
- **Tancament de sessió** *(logout.php)***:** Tanca la sessió activa i redirigeix al formulari de login. Aquesta funcionalitat es pot accedir a través de l’enllaç de tancament disponible a cada pàgina segons el rol de l’usuari.

4. **Seguretat:**
    - Assegura't que només usuaris autenticats accedeixin al sistema.
    - Gestió de sessions: S'ha d'utilitzar session_start() a l'inici de cada fitxer PHP que manipuli dades de sessió.
    - Implementa proteccions contra vulnerabilitats comunes (*SQL Injection* i accés no autoritzat).

5. **Proves:**
    - Prova que el sistema reconegui correctament el rol de l'usuari.
    - Verifica que es respecten les restriccions d'accés segons el rol.