



08/04/2025

V.1.7

# Documentation d'installation Serveur Web

PROJET « SYVAULT »

## Sommaire

<b>Machine Serveur Web</b> .....	2
Configuration générale :.....	2
Configuration réseau de la machine :.....	2
Configuration système :.....	2
Langue :.....	2
Emplacement des fichiers du site web :.....	2
Emplacement du script de sauvegarde :.....	2
Emplacement des sauvegardes.....	2
Emplacement du fichier journal (log) :.....	2
Nom de la base de données du site :.....	2
Tables de la base de données :.....	2
Arborescence du site web (code source en annexe):.....	3
Installation :.....	3
<b>Annexe :</b> .....	7
Code source index.php :.....	7
Code source login.php :.....	7
Code source accueil.php :.....	10
Code source admin.php (CSS à ajouter) :.....	13
Code source auth.php :.....	15
Code source logout.php :.....	15
Code source ajouter_article.php :.....	15
Code source modifier_article.php :.....	17

# Machine Serveur Web

## Configuration générale :

OS : Debian 12.8

Nom : Projet Serveur Web

Nom d'utilisateur : amin

Mot de passe : 20064002h (en cas de problème, utiliser « 78200 »)

## Configuration réseau de la machine :

IP : 192.168.0.5 /24

Passerelle : 192.168.0.1 /24

DNS : vide

Mode d'accès réseau : Réseau NAT

Configuration NAT (VirtualBox) : « VLAN 10 » 192.168.0.0/24

## Configuration système :

Mémoire vive : 2048 Mo

## Langue :

Français (AZERTY)

## Emplacement des fichiers du site web :

/var/www/html/

## Emplacement du script de sauvegarde :

/usr/local/bin/sauvegarde.sh

## Emplacement des sauvegardes

/backup/

## Emplacement du fichier journal (log) :

/var/www/html/journal.log

## Nom de la base de données du site :

Syvault

## Tables de la base de données :

Utilisateurs (pseudo, mdp et roles)

Articles (titre, contenu et date)

## Arborescence du site web (code source en annexe):

-index.php(1) (Page par défaut) : Cette page est la première à laquelle on accède lorsque qu'on tape l'adresse ip, elle sert à la redirection des usagers sur la page « login »

-login.php(2) (Page de connexion/inscription) : Cette page permet aux utilisateurs de se connecter ou de s'inscrire en créant un nouveau compte. Toute tentative d'accès aux autres pages en étant non-authentifié résulte en une redirection vers cette page.

-accueil.php(3) (Page principale du site) : Cette page est la page d'accueil après la connexion. Les utilisateurs peuvent voir les articles.

-admin.php(4) (Page réservée à l'administration) : Cette page permet aux administrateurs de gérer les utilisateurs, valider et dévalider les comptes. Elle est protégée et accessible uniquement aux utilisateurs avec un rôle d'admin.

-auth.php(5) (Page de vérification de l'authentification) : Cette page php sert à valider l'utilisateur après qu'il ait soumis ses informations de connexion via login.php. Elle est intégrée à chaque page pour vérifier le statut « connecté » des utilisateurs et afficher ou non les pages.

-logout.php(6) : Fonction déconnexion

-ajouter\_article.php(7): Page d'ajout d'articles. Accessible uniquement aux administrateurs

-modifier\_article.php(8) : Page de modification d'articles. Accessible uniquement aux administrateurs

-style.css : Feuille de style CSS (nécessite le téléchargement de la police d'écriture « minecraft.ttf »)

-Répertoire « fonts » : contient « minecraft.ttf »

-journal.log : fichier log

## Installation :

### *1. Service distanciel : SSH (OpenSSH)*

- **Installation :**

```
sudo apt update  
sudo apt install openssh-server
```

- **Démarrage et activation :**

```
sudo systemctl start ssh  
sudo systemctl enable ssh  
sudo systemctl status ssh
```

- **Configuration :** Dans le fichier : /etc/ssh/sshd\_config, passer PermitRootLogin en « no »

### *2. Daemon installé : SSHD (OpenSSH-Server)*

- **Installation :** Le service sshd est inclus dans le package openssh-server.
- **Vérification du service :**

```
sudo systemctl status sshd
```

- **Redémarrage :**

```
sudo systemctl restart sshd
```

### *3. Serveur Web Apache2 (+module PHP)*

- **Installation d'Apache2 :**

```
sudo apt update  
sudo apt install apache2
```

- **Démarrage et activation :**

```
sudo systemctl start apache2  
sudo systemctl enable apache2  
sudo systemctl status apache2
```

- **Démarrage et activation :**

```
sudo apt install libapache2-mod-php
```

### *4. MySQL/MariaDB*

- **Installation de MySQL (ou MariaDB) :**

```
sudo apt update  
sudo apt install mysql-server
```

- **Ou pour MariaDB :**

```
sudo apt install mariadb-server
```

- **Démarrage et activation :**

```
sudo systemctl start mysql  
sudo systemctl enable mysql  
sudo systemctl status mysql
```

### *5. PHP*

- **Installation de PHP :**

```
sudo apt update  
sudo apt install php
```

### *6. phpMyAdmin*

- **Installation de phpMyAdmin :**

```
sudo apt update
sudo apt install phpmyadmin
```

### 7. Installation et configuration de l'agent GLPI :

- On installe le fichier d'installation de l'agent glpi depuis le Github officiel. On se place dans un répertoire (dans notre cas : /tmp) et on exécute la commande :

```
wget https://github.com/glpiproject/glpiagent/releases/download/1.13/glpi-agent-1.13-linux-installer.pl
```

- On obtiens un fichier d'installation à exécuter via la commande perl :

```
perl glpi-agent-1.13-linux-installer.pl -s
```

- On nous demande l'url du serveur GLPI :

```
http://192.168.0.2/glpi/plugins/glpiinventory/
```

- On vérifie l'état du service agent glpi et relance le service si nécessaire :

```
systemctl status glpi-agent
sudo glpi-agent
```

### 8. Création du script de sauvegarde :

- Créer un fichier « sauvegarde.sh » dans le repertoire /usr/local/bin/
- Entrer le script (disponible en annexe)
- Taper la commande « crontab -e »
- Entrer cette ligne (non commenté) dans le fichier :  
0 3 \*/3 \* \* /backup/sauvegarde.sh >> /backup/cron\_log.txt 2>&1

### 9. Création de la base de données et des tables :

- Taper la commande « mysql » (sudo si necessaire)
- Taper : « Create database syvault ; »
- Pour créer la table utilisateurs :

```
CREATE TABLE utilisateurs(
    id INT AUTO_INCREMENT PRIMARY KEY,
    pseudo VARCHAR(50) NOT NULL UNIQUE,
    mdp VARCHAR(255) NOT NULL,
    role ENUM('admin', 'user', 'attente') NOT NULL DEFAULT 'attente'
);
```

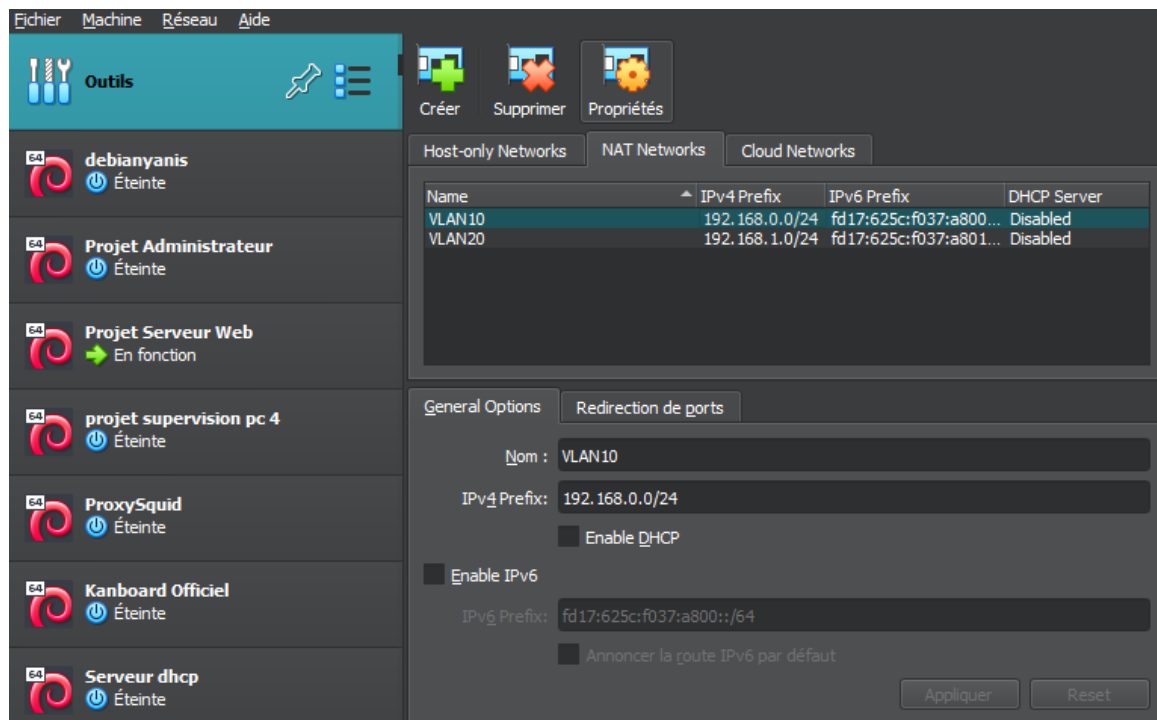
- Pour créer la table article :

```
CREATE TABLE articles(
    id INT AUTO_INCREMENT PRIMARY KEY,
    titre VARCHAR(255) NOT NULL,
    contenu TEXT NOT NULL,
```

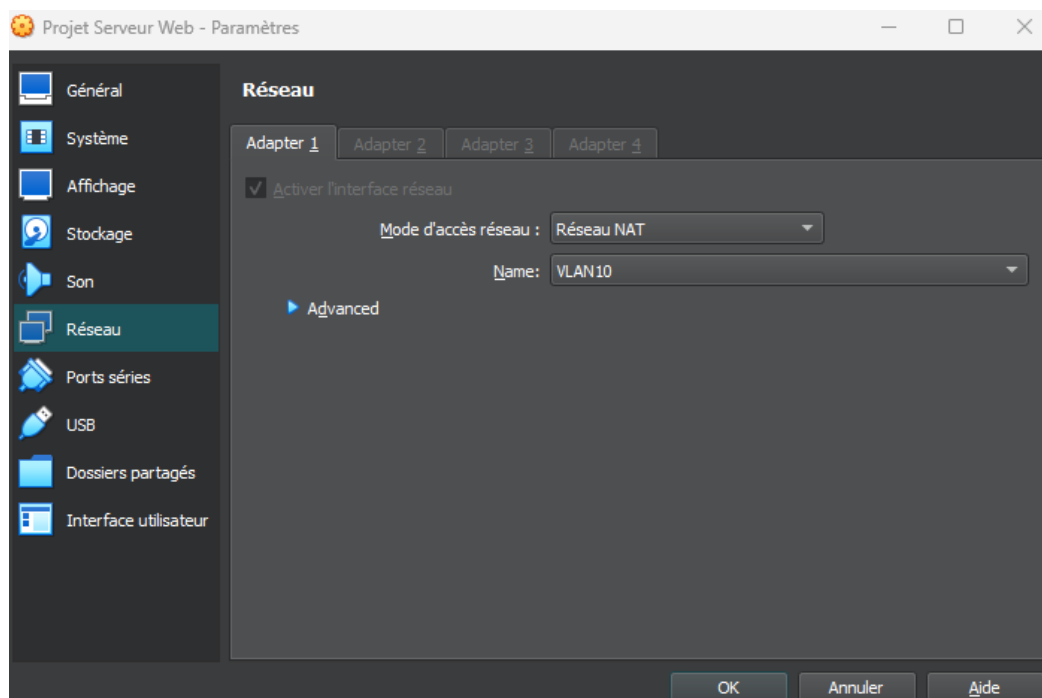
```
date_publication TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

## 10. Configuration de VirtualBox:

- Aller dans Fichier -> Outils -> Network Manager



- Configurer 2 réseaux NAT :  
VLAN10: 192.168.0.0/24 et VLAN20: 192.168.1.0/24
- Dans les paramètres de la machine, la configurer au Réseau NAT correspondant



## Annexe :

### Code source index.php :

```
<?php  
require 'auth.php';  
?>
```

### Code source login.php :

```
<?php  
  
session_start();  
  
date_default_timezone_set('Europe/Paris'); // Définir le fuseau horaire  
  
  
$host = 'localhost';  
$dbname = 'syvault';  
$username = 'root';  
$password = '';  
  
try {  
    $pdo = new PDO("mysql:host={$host};dbname={$dbname};charset=utf8mb4",  
$username, $password);  
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
} catch (PDOException $e) {  
    die("Erreur de connexion : " . $e->getMessage());  
}  
  
  
// Fonction pour enregistrer les logs  
function logEvent($message) {  
    $logFile = __DIR__ . "/journal.log";  
    $date = date("Y-m-d H:i:s");  
    $ip = $_SERVER['REMOTE_ADDR'];  
    file_put_contents($logFile, "[$date] - $ip - $message\n", FILE_APPEND);  
}  
  
  
// Traitement du formulaire de connexion  
if (isset($_POST['connexion'])) {  
    $pseudo = htmlspecialchars(trim($_POST['pseudo']));
```



```

$mdp = trim($_POST['mdp']);

$stmt = $pdo->prepare("SELECT * FROM utilisateurs WHERE pseudo = :pseudo");
$stmt->bindParam(':pseudo', $pseudo);
$stmt->execute();
$user = $stmt->fetch(PDO::FETCH_ASSOC);

if ($user && password_verify($mdp, $user['mdp'])) {
    if ($user['role'] == 'attente') {
        $error = "Votre compte est en attente de validation.";
        logEvent("Tentative de connexion refusée - $pseudo (en attente)");
    } else {
        $_SESSION['pseudo'] = $user['pseudo'];
        $_SESSION['role'] = $user['role'];
        logEvent("Connexion réussie - $pseudo");
        header('Location: accueil.php');
        exit();
    }
} else {
    $error = "Identifiants incorrects.";
    logEvent("Échec de connexion - $pseudo (mot de passe incorrect)");
}
}

// Traitement du formulaire d'inscription
if (isset($_POST['inscription'])) {
    $pseudo = htmlspecialchars(trim($_POST['pseudo']));
    $mdp = trim($_POST['mdp']);
    $mdp_confirmation = trim($_POST['mdp_confirmation']);

    $stmt = $pdo->prepare("SELECT * FROM utilisateurs WHERE pseudo = :pseudo");
    $stmt->bindParam(':pseudo', $pseudo);
    $stmt->execute();
    $existingUser = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($existingUser) {
        $error = "Le pseudo est déjà pris.";
    }
}

```

```

    } elseif ($mdp !== $mdp_confirmation) {
        $error = "Les mots de passe ne correspondent pas.";
    } else {
        $mdp_hash = password_hash($mdp, PASSWORD_DEFAULT);

        $stmt = $pdo->prepare("INSERT INTO utilisateurs (pseudo, mdp, role) VALUES
(:pseudo, :mdp, 'attente')");

        $stmt->bindParam(':pseudo', $pseudo);
        $stmt->bindParam(':mdp', $mdp_hash);
        $stmt->execute();

        logEvent("Nouvelle inscription - $pseudo (en attente)");

        header("Location: login.php?success=1");
        exit();
    }
}
?>

```

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Connexion / Inscription</title>
</head>
<body>
    <header>
        <h1>SyVault</h1>
    </header>

    <main class="center-container">
        <div class="login-box">
            <h2>Se connecter</h2>
            <form method="POST">
                <label for="pseudo">Pseudo :</label>
                <input type="text" name="pseudo" required>
                <label for="mdp">Mot de passe :</label>
            </form>
        </div>
    </main>
</body>
</html>

```

```

        <input type="password" name="mdp" required>

        <button type="submit" name="connexion">Se connecter</button>

    </form>

    <?php if (isset($error)) { echo "<p class='error'>$error</p>"; } ?>

    <?php if (isset($_GET['success'])) { echo "<p
class='success'>Inscription réussie. Connectez-vous maintenant.</p>"; } ?>

    <hr>

    <h2>S'inscrire</h2>

    <form method="POST">

        <label for="pseudo">Pseudo :</label>

        <input type="text" name="pseudo" required>

        <label for="mdp">Mot de passe :</label>

        <input type="password" name="mdp" required>

        <label for="mdp_confirmation">Confirmer le mot de passe :</label>

        <input type="password" name="mdp_confirmation" required>

        <button type="submit" name="inscription">S'inscrire</button>

    </form>

    <?php if (isset($error)) { echo "<p class='error'>$error</p>"; } ?>

</div>

</main>

</body>

</html>

```

### Code source accueil.php :

```

<?php

session_start();

require 'auth.php';

$pdo = new PDO("mysql:host=localhost;dbname=sylvault;charset=utf8mb4",
"root", "");

$articles = $pdo->query("SELECT * FROM articles ORDER BY date_publication
DESC")->fetchAll(PDO::FETCH_ASSOC);

$isAdmin = isset($_SESSION['role']) && $_SESSION['role'] === 'admin';

?>

<!DOCTYPE html>

<html lang="fr">

<head>

    <meta charset="UTF-8">

```

```

    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Accueil</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

<header>
    <h1>SyVault</h1>
</header>

<nav>
    <a href="accueil.php">Accueil</a>
    <?php
    if (isset($_SESSION['pseudo'])) {
        echo '<a href="logout.php">Se déconnecter</a>';
    }
    if (isset($_SESSION['role']) && $_SESSION['role'] === 'admin') {
        echo '<a href="admin.php">Gestion des utilisateurs</a>';
    }
    ?>
</nav>

<div class="sidebar">
    <h2>Articles</h2>
    <?php foreach ($articles as $article): ?>
        <a href="accueil.php?id=<?= $article['id'] ?>"><?=
htmlspecialchars($article['titre']) ?></a>
    <?php endforeach; ?>

    <?php if ($isAdmin): ?>
        <a href="ajouter_article.php" class="admin-btn">+ Ajouter un
article</a>
    <?php endif; ?>
</div>

<div class="content">

```

```

<?php
if (isset($_GET['id'])) {
    $id = (int) $_GET['id'];
    $stmt = $pdo->prepare("SELECT * FROM articles WHERE id = ?");
    $stmt->execute([$id]);
    $article = $stmt->fetch();

    if ($article) {
        echo "<h2>" . htmlspecialchars($article['titre']) .
"</h2>";

        echo "<p>" . nl2br(htmlspecialchars($article['contenu'])) .
"</p>";

        if ($isAdmin) {
            echo '<a href="modifier_article.php?id=' .
$article['id'] . '" class="admin-btn">✎ Modifier l\'article</a>';
        }
    } else {
        echo "<p>Article non trouvé.</p>";
    }
} else {
    echo "<p>Veuillez sélectionner un article dans le menu.</p>";
}
?>
</div>

<footer>
    <p>&copy; 2025 SyVault. Tous droits réservés.</p>
</footer>

</body>
</html>

```

### Code source admin.php (CSS à ajouter) :

```
<?php

session_start();

require 'auth.php';

// Vérifier si l'utilisateur est bien admin
if (!isset($_SESSION['role']) || $_SESSION['role'] !== 'admin') {
    echo "Accès interdit.";
    exit();
}

// Connexion à la base de données
$host = 'localhost';
$dbname = 'syvault';
$user = 'root';
$pass = '';

try {
    $bdd = new PDO("mysql:host={$host};dbname={$dbname};charset=utf8mb4", $user,
    $pass);
    $bdd->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Erreur : " . $e->getMessage());
}

// Gestion des validations / changements de rôle
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST['valider'])) {
        $user_id = $_POST['user_id'];
        $req = $bdd->prepare("UPDATE utilisateurs SET role = 'user' WHERE id = ?");
        $req->execute([$user_id]);
    } elseif (isset($_POST['remettre_attente'])) {
        $user_id = $_POST['user_id'];
        $req = $bdd->prepare("UPDATE utilisateurs SET role = 'attente' WHERE id
= ?");
        $req->execute([$user_id]);
    }
}
```

```
// Récupérer la liste des utilisateurs
$users = $bdd->query("SELECT id, pseudo, role FROM utilisateurs ORDER BY role
DESC");

?>

<!DOCTYPE html>

<html lang="fr">

<head>

    <meta charset="UTF-8">

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

    <title>Gestion des utilisateurs - Admin</title>

</head>

<body>

    <h2>Gestion des utilisateurs</h2>

    <table border="1">

        <tr>

            <th>ID</th>

            <th>Pseudo</th>

            <th>Rôle</th>

            <th>Actions</th>

        </tr>

        <?php while ($user = $users->fetch()) { ?>

            <tr>

                <td><?= $user['id'] ?></td>

                <td><?= htmlspecialchars($user['pseudo']) ?></td>

                <td><?= $user['role'] ?></td>

                <td>

                    <form method="post" style="display:inline;">

                        <input type="hidden" name="user_id" value="<?=
$user['id'] ?>">

                        <?php if ($user['role'] == 'attente') { ?>

                            <button type="submit" name="valider">✓

Valider</button>

                            <?php } elseif ($user['role'] == 'user') { ?>

                                <button type="submit" name="remettre_attente">🕒

Remettre en attente</button>

                                <?php } ?>

                            </form>
```

```

        </td>
    </tr>
<?php } ?>
</table>
<br>
<a href="index.php">Retour au site</a>
</body>
</html>

```

### Code source auth.php :

```

<?php
session_start();

// Vérifier si l'utilisateur est connecté
if (!isset($_SESSION['pseudo'])) {
    header("Location: login.php");
    exit();
}
?>

```

### Code source logout.php :

```

<?php
session_start();
session_destroy(); // Supprime toutes les sessions
header("Location: login.php");
exit();
?>

```

### Code source ajouter\_article.php :

```

<?php
session_start();
require 'auth.php';

if ($_SESSION['role'] !== 'admin') {
    die("Accès refusé.");
}

$pdo = new PDO("mysql:host=localhost;dbname=sylvault;charset=utf8mb4", "root", "");

```



```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $titre = htmlspecialchars($_POST['titre']);
    $contenu = htmlspecialchars($_POST['contenu']);

    if (!empty($titre) && !empty($contenu)) {
        $stmt = $pdo->prepare("INSERT INTO articles (titre, contenu) VALUES
(?, ?)");
        $stmt->execute([$titre, $contenu]);
        header("Location: accueil.php");
        exit();
    } else {
        $error = "Veuillez remplir tous les champs.";
    }
}
?>

```

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <title>Ajouter un article</title>
</head>
<body>
    <header>
        <h1>SyVault</h1>
    </header>

    <main class="center-container">
        <div class="form-box">
            <h2>Ajouter un nouvel article</h2>
            <?php if (isset($error)) echo "<p class='error'>$error</p>"; ?>
            <form method="POST">
                <label>Titre :</label>
                <input type="text" name="titre" required>
                <label>Contenu :</label>
                <textarea name="contenu" required></textarea>
            </form>
        </div>
    </main>
</body>
</html>

```

```

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

    </form>

</div>

</main>

</body>

</html>

```

### Code source modifier\_article.php :

```

<?php

session_start();

require 'auth.php';

if ($_SESSION['role'] !== 'admin') {

    die("Accès refusé.");

}

$pdo = new PDO("mysql:host=localhost;dbname=sylvault;charset=utf8mb4", "root", "");

if (isset($_GET['id'])) {

    $id = (int) $_GET['id'];

    $stmt = $pdo->prepare("SELECT * FROM articles WHERE id = ?");

    $stmt->execute([$id]);

    $article = $stmt->fetch();

} else {

    die("Article introuvable.");

}

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    $titre = htmlspecialchars($_POST['titre']);

    $contenu = htmlspecialchars($_POST['contenu']);

    if (!empty($titre) && !empty($contenu)) {

        $stmt = $pdo->prepare("UPDATE articles SET titre = ?, contenu = ? WHERE id
= ?");

        $stmt->execute([$titre, $contenu, $id]);

        header("Location: accueil.php?id=$id");

        exit();

    } else {

```

```

        $error = "Veuillez remplir tous les champs.";
    }
}
?>

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Modifier l'article</title>
</head>
<body>
    <h2>Modifier l'article</h2>
    <?php if (isset($error)) echo "<p style='color:red;'>$error</p>"; ?>
    <form method="POST">
        <label>Titre :</label>
        <input type="text" name="titre" value="<?= htmlspecialchars($article['titre']) ?>" required>
        <br>
        <label>Contenu :</label>
        <textarea name="contenu" required><?= htmlspecialchars($article['contenu'])
?></textarea>
        <br>
        <button type="submit">Modifier</button>
    </form>
</body>
</html>

```

## Script sauvegarde.sh :

```

#!/bin/bash

# Définir les variables
DATE=$(date +"%Y-%m-%d")
BACKUP_DIR="/backup"
SITE_DIR="/var/www/html"
DB_NAME="syvaul"
DB_USER="root"

```

```

DB_PASSWORD="" # Mettre ton mot de passe MySQL si nécessaire
# Création du dossier de sauvegarde du jour
mkdir -p $BACKUP_DIR/$DATE
# Sauvegarde des fichiers du site
tar -czf $BACKUP_DIR/$DATE/site_backup.tar.gz $SITE_DIR
# Sauvegarde de la base de données
mysqldump -u $DB_USER -p$DB_PASSWORD $DB_NAME >
$BACKUP_DIR/$DATE/db_backup.sql
echo "Sauvegarde terminée : $DATE"

```

## Style.css :

```

@font-face {
    font-family: "Minecraft";
    src: url("fonts/minecraft.ttf") format("truetype");
}
/* Styles généraux */
body {
    margin: 0;
    padding: 0;
    background-color: #111;
    color: white;
    font-family: "Minecraft", Arial, sans-serif;
    display: flex;
    flex-direction: column;
    min-height: 100vh;
}
/* Header */
header {
    text-align: center;
    padding: 20px;
    font-size: 28px;
    font-weight: bold;
    background: #222;
}
/* Navigation */
nav {

```

```

    display: flex;
    justify-content: center;
    background: #333;
    padding: 10px;
    gap: 20px;
}
nav a {
    text-decoration: none;
    color: white;
    font-size: 18px;
    padding: 8px 15px;
    transition: 0.3s;
}
nav a:hover {
    background: #444;
    border-radius: 5px;
}
/* Conteneur principal */
main {
    display: flex;
    flex: 1;
}
/* Sidebar (Menu des articles) */
.sidebar {
    width: 250px;
    background: #222;
    padding: 15px;
    height: 100vh;
    position: fixed;
    top: 80px; /* Ajustement après le menu */
    left: 0;
    overflow-y: auto;
}
.sidebar h2 {
    text-align: center;
    font-size: 22px;
}

```

```

.sidebar a {
    display: block;
    color: white;
    text-decoration: none;
    padding: 10px;
    border-bottom: 1px solid #333;
    transition: 0.3s;
}

.sidebar a:hover {
    background: #444;
}

/* Bouton Admin */

.admin-btn {
    display: block;
    background: #ff9800;
    color: white;
    text-align: center;
    padding: 10px;
    margin-top: 15px;
    text-decoration: none;
    border-radius: 5px;
}

.admin-btn:hover {
    background: #e68a00;
}

/* Contenu principal */

.content {
    flex: 1;
    margin-left: 270px; /* Décalage pour éviter la sidebar */
    padding: 20px;
}

/* Footer */

footer {
    text-align: center;
    padding: 10px;
    background: #222;
    position: fixed;

```

```

        bottom: 0;

        width: 100%;
    }

/* Centre les formulaires de connexion et d'ajout d'article */
.center-container {
    display: flex;
    justify-content: center;
    align-items: center;
    height: calc(100vh - 100px);
}

/* Style des boîtes de formulaire */
.login-box, .form-box {
    background: #222;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0px 0px 10px rgba(255, 255, 255, 0.2);
    width: 400px;
    text-align: center;
}

/* Champs de formulaire */
input, textarea {
    width: 100%;
    padding: 8px;
    margin: 10px 0;
    background: #333;
    border: 1px solid #444;
    color: white;
    border-radius: 5px;
}

/* Boutons */
button {
    width: 100%;
    padding: 10px;
    background: #ff9800;
    border: none;
    color: white;
    font-size: 16px;
}

```

```
        cursor: pointer;
        border-radius: 5px;
    }
    button:hover {
        background: #e68a00;
    }
    /* Messages d'erreur et de succès */
    .error {
        color: red;
        font-weight: bold;
    }
    .success {
        color: green;
        font-weight: bold;
    }
}
```