

## Matte ton stream

Matte ton stream.....	1
<b>I. Introduction.....</b>	<b>3</b>
<b>Objectif du Site de Streaming .....</b>	<b>3</b>
<b>Principales Fonctionnalités .....</b>	<b>3</b>
<b>II. Structure du Code.....</b>	<b>4</b>
1. Répertoires Principaux .....	4
2. Fichiers Principaux.....	4
3. Connexion à la Base de Données .....	4
<b>III. Présentation des Principales Fonctionnalités .....</b>	<b>5</b>
<b>1. Gestion de l'Authentification .....</b>	<b>5</b>
<b>2. Catalogue de Films et Séries .....</b>	<b>5</b>
<b>3. Lecture de Films et Séries .....</b>	<b>6</b>
<b>IV. Gestion des Profils Utilisateurs et Administration .....</b>	<b>7</b>
1. Profil Utilisateur.....	7
2. Page d'Inscription et Succès d'Inscription .....	7
3. Déconnexion .....	8
4. Administration des Films et Séries .....	8
<b>V. Pages de Connexion et Gestion des Sessions .....</b>	<b>10</b>
1. Page de Connexion .....	10
2. Déconnexion .....	10
<b>VI. Gestion des Utilisateurs et Profils .....</b>	<b>12</b>
1. Inscription d'Utilisateur.....	12
2. Modification du Mot de Passe et Suppression de Compte .....	12
3. Affichage du Profil Utilisateur.....	13
<b>VII. Fonctionnalités d'Authentification et Sécurité.....</b>	<b>15</b>
1. Connexion d'Utilisateur.....	15
2. Déconnexion d'Utilisateur .....	15
3. Sécurité du Mot de Passe et Validation du Formulaire .....	16
4. Protection contre l'Accès Non Autorisé .....	16
<b>VIII. Gestion des Profils Utilisateurs .....</b>	<b>17</b>
1. Affichage du Profil .....	17
2. Changement de Mot de Passe.....	17

3. Suppression de Compte.....	18
IX. Gestion des Utilisateurs (Section Admin).....	19
1. Menu d'Administration .....	19
2. Suppression de Films/Séries.....	19
3. Administration des Utilisateurs .....	20
X. Gestion des Profils Utilisateurs.....	21
1. Profil Utilisateur.....	21
2. Changement de Mot de Passe.....	21
3. Suppression de Compte.....	22
4. Traitement de l'Action .....	22
XI. Gestion des Utilisateurs (Partie Admin) .....	24
1. Page d'Administration des Utilisateurs .....	24
2. Ajout d'un Nouvel Utilisateur.....	24
3. Menu Admin (Partie du Header) .....	24
4. Suppression d'Utilisateur (Partie Admin) .....	25
XII. Sécurité et Bonnes Pratiques .....	27
1. Sécurité des Mot de Passe.....	27
2. Validation des Données du Formulaire .....	27
3. Protection Contre les Attaques par Injection SQL.....	27
4. Protection Contre les Attaques CSRF.....	28
5. Contrôle d'Accès et Gestion des Sessions .....	28
6. Redirection Après Opérations Sensibles .....	29
<b>XIII. Guide de Déploiement.....</b>	<b>30</b>
<b>Prérequis.....</b>	<b>30</b>
<b>Instructions de Déploiement .....</b>	<b>30</b>
<b>XIV. Résolution de Problèmes Courants.....</b>	<b>32</b>
<b>1. Problème de Connexion.....</b>	<b>32</b>
<b>2. Erreur de Suppression de Film ou de Série.....</b>	<b>32</b>
<b>3. Problème d'Affichage du Lecteur Vidéo .....</b>	<b>32</b>
<b>4. Problème de Personnalisation .....</b>	<b>32</b>
<b>5. Problèmes de Performance.....</b>	<b>32</b>
<b>6. Erreur de Configuration du Serveur Web .....</b>	<b>33</b>
<b>7. Erreur de Mise en Page .....</b>	<b>33</b>
XV. Conclusion .....	34

## I. Introduction

Bienvenue dans la documentation du code du site de streaming "Matte en stream". Cette documentation vise à fournir une compréhension approfondie de la structure, de la logique et des fonctionnalités clés de l'application.

### Objectif du Site de Streaming

"Matte en stream" a pour objectif de permettre aux utilisateurs de visionner des films et des séries en streaming de manière pratique et conviviale. Il offre une plateforme centralisée pour la gestion de contenu vidéo, permettant aux utilisateurs de parcourir un catalogue de films et de séries, de sélectionner des titres spécifiques, et de les regarder en streaming.

### Principales Fonctionnalités

1. **Catalogue Étendu** : La plateforme propose une vaste collection de films et de séries, organisée de manière accessible.
2. **Gestion des Utilisateurs** : Les utilisateurs peuvent créer des comptes, se connecter, et personnaliser leur profil.
3. **Streaming Intégré** : Le lecteur intégré permet la lecture fluide des films et épisodes de séries sélectionnés.
4. **Administration** : Les administrateurs ont des fonctionnalités supplémentaires pour gérer le catalogue et les utilisateurs.
5. **Sécurité** : Des mesures de sécurité sont mises en place pour protéger l'accès au contenu et les données des utilisateurs.

À travers cette documentation, nous explorerons la structure du code, les fichiers clés, la gestion de base de données, les mesures de sécurité, et d'autres aspects cruciaux de "Matte en stream". Prenez le temps de vous plonger dans chaque section pour comprendre pleinement le fonctionnement de l'application. Si des questions surviennent, n'hésitez pas à vous référer à la section de contact à la fin de la documentation.

## II. Structure du Code

La structure du code de "Matte en stream" est organisée de manière à faciliter la gestion, la lisibilité et la maintenance du système. Cette section détaille les principaux éléments de la structure du code, y compris les fichiers essentiels et les répertoires clés.

### 1. Répertoires Principaux

- **css/** : Contient les fichiers de style CSS pour la présentation visuelle.
- **film/** : Comprend les fichiers liés aux fonctionnalités de streaming des films.
- **img/** : Stocke les images et autres ressources graphiques.
- **js/** : Contient les scripts JavaScript utilisés pour la manipulation dynamique du contenu.
- **style/** : Inclut les fichiers de style CSS spécifiques à chaque page.
- **util/** : Rassemble des fichiers utilitaires partagés entre plusieurs parties de l'application.

### 2. Fichiers Principaux

- **index.php** : Redirige vers la page du catalogue. Point d'entrée de l'application.
- **login.php et register.php** : Gèrent l'authentification des utilisateurs.
- **catalogue.php** : Affiche la liste des films et séries disponibles.
- **film.php et serie.php** : Gèrent l'affichage et la lecture de films et d'épisodes de séries.
- **profil.php** : Permet aux utilisateurs de gérer leur profil, changer de mot de passe, et supprimer leur compte.
- **suppression\_serie.php** : Gère la suppression de films ou séries par les administrateurs.
- **preset\_header.php** : Inclut l'en-tête commun à plusieurs pages, gère la navigation et l'affichage utilisateur.
- **logout.php** : Gère la déconnexion des utilisateurs.
- **ajt.php** : Permet l'ajout de nouveaux films ou séries par les administrateurs.
- **register\_success.php** : Confirme la réussite de l'inscription.

### 3. Connexion à la Base de Données

Le fichier **cnx.php** gère la connexion à la base de données. Les requêtes SQL sont exécutées pour interagir avec les données utilisateur et le catalogue de films/séries.

La structure modulaire du code facilite l'ajout de fonctionnalités et la maintenance continue de l'application. Explorez chaque répertoire et fichier pour comprendre comment les différentes parties de l'application interagissent entre elles.

### III. Présentation des Principales Fonctionnalités

La section précédente a couvert la structure générale du code. Cette section se concentre sur les principales fonctionnalités de l'application "Matte en stream". Chaque fonctionnalité est présentée avec des extraits de code pertinents pour fournir un aperçu de son implémentation.

#### 1. Gestion de l'Authentification

L'authentification des utilisateurs est gérée par les fichiers **login.php** et **register.php**. Le hashage sécurisé des mots de passe est effectué via la bibliothèque **password.php**. Après la connexion réussie, les utilisateurs sont redirigés vers le catalogue.

```
php
// login.php

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Recherche de l'utilisateur dans la base de données
    $stmt = $pdo->prepare("SELECT * FROM users WHERE username = ?");
    $stmt->execute([$username]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    // Vérification du mot de passe
    if ($user && password_verify($password, $user['password'])) {
        // Connexion réussie
        session_start();
        $_SESSION['user_id'] = $user['id'];
        $_SESSION['username'] = $user['username'];
        $_SESSION['role'] = $user['role']; // Assurez-vous que le champ
        role existe dans votre base de données
        header("Location: catalogue.php"); // Rediriger vers la page du
        catalogue
        exit();
    } else {
        echo "Nom d'utilisateur ou mot de passe incorrect.";
    }
}
```

#### 2. Catalogue de Films et Séries

Le fichier **catalogue.php** affiche la liste des films et séries disponibles. Les données sont extraites de la base de données et affichées dynamiquement.

```
php
// catalogue.php

// Récupérer la liste des films et séries depuis la base de données
$stmt = $pdo->query("SELECT * FROM films");
$films = $stmt->fetchAll(PDO::FETCH_ASSOC);

// Affichage des films dans une table HTML
foreach ($films as $film) {
    // Affichage des détails du film
    echo "<td>{$film['id']}</td>";
}
```

```

        echo "<td>{$film['type']}</td>";
        echo "<td>{$film['title']}</td>";
        echo "<td>... <!-- Autres détails du film --></td>";
    }

```

### 3. Lecture de Films et Séries

Les fichiers **film.php** et **serie.php** gèrent la lecture de films et d'épisodes de séries. Ils intègrent un lecteur vidéo et permettent la navigation entre les épisodes.

```

php
// film.php

// Récupère le titre du film à partir du paramètre de requête
$selectedFilm = isset($_GET['title']) ? $_GET['title'] : '';

// Charge les informations sur les films à partir du fichier PHP
include("../film/films.php");

// Utilise le titre du film pour obtenir les informations correspondantes
$filmData = isset($films[$selectedFilm]) ? $films[$selectedFilm] : null;

// Modification pour utiliser la valeur de l'URL directement dans le code
// JavaScript
$filmURL = isset($filmData['url']) ? json_encode($filmData['url']) : '';

```

Ces extraits de code illustrent certaines des fonctionnalités clés de l'application. La suite de cette documentation explorera d'autres aspects tels que la gestion de profil utilisateur, l'administration, et l'ajout/suppression de films.

## IV. Gestion des Profils Utilisateurs et Administration

Cette section met en lumière la gestion des profils utilisateurs et les fonctionnalités administratives, y compris la possibilité de changer de mot de passe, de supprimer un compte, et d'administrer le contenu.

### 1. Profil Utilisateur

Le fichier **profil.php** permet aux utilisateurs de consulter leur profil, de changer leur mot de passe, et de supprimer leur compte. Il utilise des sessions pour s'assurer que l'utilisateur est connecté.

```
php
// profil.php

// Check if the user is logged in
session_start();

if (!isset($_SESSION['user_id'])) {
    header("Location: login.php"); // Redirect to login page if not logged
    in
    exit();
}

// Retrieve user information from the database (replace with your actual
database query)
$user_id = $_SESSION['user_id'];
$stmt = $pdo->prepare("SELECT * FROM users WHERE id = ?");
$stmt->execute([$user_id]);
$user = $stmt->fetch(PDO::FETCH_ASSOC);

// Handle password change if the form is submitted
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['changePassword'])) {
    // ... (Voir le code complet dans la documentation)
}

// Handle account deletion if the form is submitted
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['deleteAccount'])) {
    // ... (Voir le code complet dans la documentation)
}
```

### 2. Page d'Inscription et Succès d'Inscription

Les fichiers **register.php** et **register\_success.php** gèrent l'inscription des utilisateurs. Le mot de passe est haché avant d'être stocké dans la base de données.

```
php
// register.php

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $confirmPassword = $_POST['confirm_password'];
```

```

// Vérifier si le mot de passe et la confirmation sont identiques
if ($password !== $confirmPassword) {
    echo "Les mots de passe ne correspondent pas.";
    exit();
}

// Hacher le mot de passe
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);

// Ajouter l'utilisateur à la base de données
$stmt = $pdo->prepare("INSERT INTO users (username, email, password)
VALUES (?, ?, ?)");
$stmt->execute([$username, $email, $hashedPassword]);

// Redirection vers la page de connexion après l'inscription
header("Location: login.php");
exit();
}

```

### 3. Déconnexion

Le fichier **logout.php** se charge de détruire toutes les données de session lors de la déconnexion de l'utilisateur.

```

php
// logout.php

session_start();

// Détruit toutes les données de session
session_destroy();

// Redirige l'utilisateur vers la page d'accueil ou une autre page après la
déconnexion
header("Location: index.php");
exit;

```

### 4. Administration des Films et Séries

Les fichiers **ajt.php** et **suppression\_serie.php** sont dédiés à l'administration du contenu. Ils permettent respectivement l'ajout de nouveaux films ou séries et la suppression de films ou séries existants.

```

php
// ajt.php (Extrait)

// Vérifier si l'utilisateur est connecté et a le rôle d'administrateur
if (!isset($_SESSION['user_id']) || $_SESSION['role'] !== 'admin') {
    header("Location: login.php");
    exit;
}

// ... (Code complet dans la documentation)
php
// suppression_serie.php (Extrait)

// Vérifier si l'utilisateur est connecté
if (!isset($_SESSION['user_id'])) {

```



```
        header("Location: login.php");  
        exit;  
    }  
  
    // ... (Code complet dans la documentation)
```

Ces extraits couvrent la gestion des profils utilisateurs, l'inscription, la déconnexion, et l'administration des films et séries. La documentation complète fournit des détails supplémentaires et des extraits de code pour chaque fonctionnalité.

## V. Pages de Connexion et Gestion des Sessions

Cette section examine les pages et fonctionnalités liées à l'authentification, à la connexion des utilisateurs, et à la gestion des sessions.

### 1. Page de Connexion

Le fichier **login.php** permet aux utilisateurs de saisir leurs informations de connexion. Il utilise le hachage de mot de passe pour vérifier l'authenticité de l'utilisateur.

```
php
// login.php

include("cnx.php");
require 'password.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Recherche de l'utilisateur dans la base de données
    $stmt = $pdo->prepare("SELECT * FROM users WHERE username = ?");
    $stmt->execute([$username]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    // Vérification du mot de passe
    if ($user && password_verify($password, $user['password'])) {
        // Connexion réussie
        session_start();
        $_SESSION['user_id'] = $user['id'];
        $_SESSION['username'] = $user['username'];
        $_SESSION['role'] = $user['role']; // Assurez-vous que le champ
        // rôle existe dans votre base de données
        header("Location: catalogue.php"); // Rediriger vers la page du
        catalogue
        exit();
    } else {
        echo "Nom d'utilisateur ou mot de passe incorrect.";
    }
}
```

### 2. Déconnexion

Le fichier **logout.php** gère la déconnexion en détruisant toutes les données de session et en redirigeant l'utilisateur.

```
php
// logout.php

session_start();

// Détruit toutes les données de session
session_destroy();

// Redirige l'utilisateur vers la page d'accueil ou une autre page après la
déconnexion
header("Location: index.php");
exit;
```

Ces extraits illustrent la mise en œuvre de la page de connexion et de déconnexion, ainsi que la manière dont les sessions sont utilisées pour maintenir l'état de l'utilisateur. La documentation complète fournit des détails approfondis et des extraits de code supplémentaires.

## VI. Gestion des Utilisateurs et Profils

Cette section explore les fonctionnalités de gestion des utilisateurs, y compris l'inscription, la modification du profil, la suppression de compte, et l'affichage du profil utilisateur.

### 1. Inscription d'Utilisateur

Le fichier **register.php** gère le processus d'inscription des utilisateurs. Il inclut la vérification des champs de formulaire et le hachage du mot de passe.

```
php
// register.php

include("cnx.php");
require("password.php");

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $confirmPassword = $_POST['confirm_password'];

    // Vérifier si le mot de passe et la confirmation sont identiques
    if ($password !== $confirmPassword) {
        echo "Les mots de passe ne correspondent pas.";
        exit();
    }

    // Hacher le mot de passe
    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);

    // Ajouter l'utilisateur à la base de données
    $stmt = $pdo->prepare("INSERT INTO users (username, email, password)
VALUES (?, ?, ?)");
    $stmt->execute([$username, $email, $hashedPassword]);

    // Redirection vers la page de connexion après l'inscription
    header("Location: login.php");
    exit();
}
```

### 2. Modification du Mot de Passe et Suppression de Compte

Le fichier **profil.php** gère la modification du mot de passe et offre la possibilité de supprimer le compte utilisateur. Il utilise également JavaScript pour améliorer l'expérience utilisateur.

```
php
// profil.php

// ...

// Handle password change if the form is submitted
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['changePassword'])) {
    // ...
}
```

```

}

// Handle account deletion if the form is submitted
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['deleteAccount'])) {
    // Demander une confirmation avant de supprimer le compte
    echo '<script language="javascript">';
    echo 'if(confirm("Êtes-vous sûr de vouloir supprimer votre compte?
Cette action est irréversible.")) {';
    echo 'window.location.href = "delete_account.php";';
    echo '}'';
    echo '</script>';
}

// ...

```

### 3. Affichage du Profil Utilisateur

La page **profil.php** affiche les informations du profil de l'utilisateur, y compris le nom d'utilisateur, l'e-mail et offre des fonctionnalités pour modifier le mot de passe et supprimer le compte.

```

php
// profil.php

include("cnx.php");
include("password.php");

// Check if the user is logged in
session_start();

// ...

<h1>Profil de <?php echo $user['username']; ?></h1>

<div>
    <p>Nom d'utilisateur: <?php echo $user['username']; ?></p>
    <p>Email: <?php echo $user['email']; ?></p>
    <!-- Add more user information as needed -->
</div>

<h2>Changer le Mot de Passe</h2>

<!-- Form for changing the password -->
<form method="POST" action="profil.php">
    <!-- ...
        Fields for entering old and new passwords
    ... -->
    <input type="submit" name="changePassword" value="Changer le Mot de
Passe">
</form>

<h2>Supprimer le Compte</h2>

<!-- Form for deleting the account -->
<form method="POST" action="profil.php">
    <input type="submit" name="deleteAccount" value="Supprimer le Compte">
</form>

// ...

```

Ces extraits de code présentent la logique associée à l'inscription, à la modification du mot de passe, à la suppression du compte, et à l'affichage du profil utilisateur. La documentation complète fournit une explication détaillée de ces fonctionnalités.

## VII. Fonctionnalités d'Authentification et Sécurité

Cette section explore les fonctionnalités liées à l'authentification, à la sécurité, et à la déconnexion des utilisateurs.

### 1. Connexion d'Utilisateur

Le fichier **login.php** gère le processus d'authentification des utilisateurs. Il vérifie les informations d'identification et crée une session si l'authentification réussit.

```
php
// login.php

include("cnx.php");
require 'password.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Recherche de l'utilisateur dans la base de données
    $stmt = $pdo->prepare("SELECT * FROM users WHERE username = ?");
    $stmt->execute([$username]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    // Vérification du mot de passe
    if ($user && password_verify($password, $user['password'])) {
        // Connexion réussie
        session_start();
        $_SESSION['user_id'] = $user['id'];
        $_SESSION['username'] = $user['username'];
        $_SESSION['role'] = $user['role']; // Assurez-vous que le champ
        // rôle existe dans votre base de données
        header("Location: catalogue.php"); // Rediriger vers la page du
        // catalogue
        exit();
    } else {
        echo "Nom d'utilisateur ou mot de passe incorrect.";
    }
}
```

### 2. Déconnexion d'Utilisateur

Le fichier **logout.php** gère la déconnexion des utilisateurs. Il détruit toutes les données de session et redirige l'utilisateur vers la page d'accueil.

```
php
// logout.php

session_start();

// Détruit toutes les données de session
session_destroy();

// Redirige l'utilisateur vers la page d'accueil ou une autre page après la
// déconnexion
header("Location: index.php");
exit;
```

### 3. Sécurité du Mot de Passe et Validation du Formulaire

Le fichier **register.php** inclut des vérifications de sécurité telles que la correspondance des mots de passe, le hachage du mot de passe, et la validation des critères de mot de passe.

```
php
// register.php

// ...

// Vérifier si le mot de passe et la confirmation sont identiques
if ($password !== $confirmPassword) {
    echo "Les mots de passe ne correspondent pas.";
    exit();
}

// Hacher le mot de passe
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);

// ...
```

### 4. Protection contre l'Accès Non Autorisé

Les fichiers tels que **profil.php** vérifient si l'utilisateur est connecté. S'il ne l'est pas, l'utilisateur est redirigé vers la page de connexion.

```
php
// profil.php

include("cnx.php");
include("password.php");

// Check if the user is logged in
session_start();

if (!isset($_SESSION['user_id'])) {
    header("Location: login.php"); // Redirect to login page if not logged
in
    exit();
}

// ...
```

Ces extraits de code illustrent la gestion de l'authentification, de la déconnexion, de la sécurité des mots de passe, et de la protection contre l'accès non autorisé. La documentation complète fournit des détails approfondis sur ces aspects.



## VIII. Gestion des Profils Utilisateurs

Cette section aborde la gestion des profils utilisateurs, y compris la modification du mot de passe et la suppression de compte.

### 1. Affichage du Profil

Le fichier **profil.php** affiche les informations du profil de l'utilisateur connecté, telles que le nom d'utilisateur et l'adresse e-mail.

```
php
// profil.php

include("cnx.php");
include("password.php");

// Check if the user is logged in
session_start();

if (!isset($_SESSION['user_id'])) {
    header("Location: login.php"); // Redirect to login page if not logged
    in
    exit();
}

// Retrieve user information from the database
$user_id = $_SESSION['user_id'];
$stmt = $pdo->prepare("SELECT * FROM users WHERE id = ?");
$stmt->execute([$user_id]);
$user = $stmt->fetch(PDO::FETCH_ASSOC);

// Display user information
// ...
```

### 2. Changement de Mot de Passe

Le fichier **profil.php** gère la modification du mot de passe. Il vérifie l'ancien mot de passe, valide le nouveau mot de passe, le hache, puis le met à jour dans la base de données.

```
php
// profil.php

// ...

// Handle password change if the form is submitted
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
    isset($_POST['changePassword'])) {
    $old_password = $_POST['oldPassword'];
    $new_password = $_POST['newPassword'];

    // Check if the old password matches the one in the database
    if (password_verify($old_password, $user['password'])) {
        // Check if the new password meets security criteria
        if (
            strlen($new_password) >= 10 &&
            preg_match('/[A-Z]/', $new_password) &&
```

```

        preg_match('/[a-z]/', $new_password) &&
        preg_match('/[0-9]/', $new_password) &&
        preg_match('/[^a-zA-Z0-9]/', $new_password)
    ) {
        // Hash the new password
        $new_password_hashed = password_hash($new_password,
PASSWORD_DEFAULT);

        // Update the password in the database
        $stmt = $pdo->prepare("UPDATE users SET password = ? WHERE id =
?");
        $stmt->execute([$new_password_hashed, $user_id]);

        // Redirect to prevent form resubmission on page refresh
        header("Location: profil.php?success=1");
        exit;
    } else {
        // Display an error message if the password does not meet
criteria
        $password_error = "Le mot de passe doit contenir au moins 10
caractères, dont au moins une majuscule, une minuscule, un chiffre et un
caractère spécial.";
    }
    } else {
        // Display an error message if the old password does not match
        $password_error = "L'ancien mot de passe est incorrect.";
    }
}

// ...

```

### 3. Suppression de Compte

Le fichier **profil.php** permet à l'utilisateur de demander la suppression de son compte. Il affiche une boîte de dialogue de confirmation avant de procéder à la suppression.

```

php
// profil.php

// ...

// Handle account deletion if the form is submitted
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['deleteAccount'])) {
    echo '<script language="javascript">';
    echo 'if(confirm("Êtes-vous sûr de vouloir supprimer votre compte?
Cette action est irréversible.")) {';
    echo 'window.location.href = "delete_account.php";';
    echo '}'';
    echo '</script>';
}

// ...

```

Ces extraits de code montrent comment afficher les profils utilisateur, gérer les modifications de mot de passe et traiter les demandes de suppression de compte. La documentation complète fournit des informations détaillées sur ces fonctionnalités.

## IX. Gestion des Utilisateurs (Section Admin)

Cette section concerne la gestion des utilisateurs, spécifiquement pour les administrateurs.

### 1. Menu d'Administration

Le fichier **preset\_header.php** inclut un menu d'administration accessible aux utilisateurs ayant le rôle d'administrateur. Ce menu offre des fonctionnalités telles que l'ajout de films, la suppression de films, et l'administration des utilisateurs.

```
php
// preset_header.php

// ...

echo '<nav>';
if (isset($_SESSION['user_id'])) {
    // ...
    // Admin dropdown menu
    if (isset($_SESSION['role']) && $_SESSION['role'] === 'admin') {
        echo '<div class="admin-dropdown">';
        echo '<a class="admin-btn" href="#">Admin</a>';
        echo '<div class="admin-dropdown-content">';
        echo '<a href="ajt.php">Ajouter un film ou une série</a>';
        echo '<a href="suppression_serie.php">Supprimer un film ou une
série</a>';
        echo '<a href="admin/users.php">Administration des
Utilisateurs</a>';
        echo '</div>';
        echo '</div>';
    }
    // ...
}
echo '</nav>';

// ...
```

### 2. Suppression de Films/Séries

Le fichier **suppression\_serie.php** permet aux administrateurs de supprimer des films ou séries de la base de données.

```
php
// suppression_serie.php

// ...

// Traitement de la suppression
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['delete_id'])) {
    $deleteId = $_POST['delete_id'];

    // Supprimer le film de la base de données
    $stmt = $pdo->prepare("DELETE FROM films WHERE id = ?");
    $stmt->execute([$deleteId]);

    echo "Le film ou la série a été supprimé(e) avec succès.";
}
```

```
}  
  
// ...
```

### 3. Administration des Utilisateurs

Le fichier **admin/users.php** permet aux administrateurs de gérer les utilisateurs en affichant une liste et en offrant des fonctionnalités telles que la suppression de comptes.

```
php  
// admin/users.php  
  
// ...  
  
// Retrieve user data from the database  
$stmt = $pdo->query("SELECT * FROM users");  
$users = $stmt->fetchAll(PDO::FETCH_ASSOC);  
  
// Display user information and options for administrators  
foreach ($users as $user) {  
    echo '<div>';  
    echo '<p>ID: ' . $user['id'] . '</p>';  
    echo '<p>Nom d\'utilisateur: ' . $user['username'] . '</p>';  
    echo '<p>Email: ' . $user['email'] . '</p>';  
    echo '<form method="POST" action="admin/delete_user.php">';  
    echo '<input type="hidden" name="delete_id" value="' . $user['id'] .  
'">';  
    echo '<input type="submit" value="Supprimer le Compte">';  
    echo '</form>';  
    echo '</div>';  
}  
  
// ...
```

Ces extraits mettent en évidence la gestion des utilisateurs dans un contexte d'administration, y compris la suppression de films/séries et l'administration des utilisateurs. Consultez la documentation complète pour des détails approfondis sur ces fonctionnalités.

## X. Gestion des Profils Utilisateurs

Cette section concerne la gestion des profils utilisateurs, y compris la modification du mot de passe et la suppression de compte.

### 1. Profil Utilisateur

Le fichier **profil.php** permet aux utilisateurs de visualiser leur profil, affichant des informations telles que le nom d'utilisateur et l'e-mail.

```
php
// profil.php

// ...

<h1>Profil de <?php echo $user['username']; ?></h1>

<div>
    <p>Nom d'utilisateur: <?php echo $user['username']; ?></p>
    <p>Email: <?php echo $user['email']; ?></p>
    <!-- Add more user information as needed -->
</div>

// ...
```

### 2. Changement de Mot de Passe

Les utilisateurs peuvent changer leur mot de passe en utilisant le formulaire dans le fichier **profil.php**.

```
php
// profil.php

// ...

<h2>Changer le Mot de Passe</h2>

<?php if (isset($successMessage)) : ?>
    <p style="color: green;"><?php echo $successMessage; ?></p>
<?php endif; ?>

<?php if (isset($password_error)) : ?>
    <p style="color: red;"><?php echo $password_error; ?></p>
<?php endif; ?>

<form method="POST" action="profil.php">
    <label for="oldPassword">Ancien Mot de Passe:</label>
    <input type="password" name="oldPassword" required>

    <label for="newPassword">Nouveau Mot de Passe:</label>
    <input type="password" name="newPassword" required>

    <label for="confirmPassword">Confirmer le Nouveau Mot de Passe:</label>
    <input type="password" name="confirmPassword" required>

    <input type="submit" name="changePassword" value="Changer le Mot de Passe">
</form>
```

```
// ...
```

### 3. Suppression de Compte

Les utilisateurs peuvent choisir de supprimer leur compte via le formulaire dans le fichier **profil.php**.

```
php
// profil.php

// ...

<h2>Supprimer le Compte</h2>

<form method="POST" action="profil.php">
    <input type="submit" name="deleteAccount" value="Supprimer le Compte">
</form>

// ...
```

### 4. Traitement de l'Action

Le traitement des actions de modification de mot de passe et de suppression de compte se fait dans le fichier **profil.php**.

```
php
// profil.php

// ...

// Handle password change if the form is submitted
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
    isset($_POST['changePassword'])) {
    // ...

    // Redirect to avoid form resubmission on page refresh
    header("Location: profil.php?success=1");
    exit;
}

// Handle account deletion if the form is submitted
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
    isset($_POST['deleteAccount'])) {
    // Confirmation prompt handled in JavaScript
    echo '<script language="javascript">';
    echo 'if(confirm("Are you sure you want to delete your account? This
action is irreversible.")) {';
    echo 'window.location.href = "delete_account.php";';
    echo '}'';
    echo '</script>';
}

// ...
```

Cette section met en évidence la gestion des profils utilisateurs, y compris le changement de mot de passe et la suppression de compte. Consultez la

documentation complète pour une compréhension approfondie de ces fonctionnalités.

## XI. Gestion des Utilisateurs (Partie Admin)

Cette section se concentre sur la gestion des utilisateurs, en mettant l'accent sur les fonctionnalités d'administration disponibles pour les utilisateurs ayant le rôle d'administrateur.

### 1. Page d'Administration des Utilisateurs

La page d'administration des utilisateurs est accessible depuis le menu Admin, réservé aux utilisateurs ayant le rôle d'administrateur. Le fichier **admin/users.php** est responsable de cette fonctionnalité.

```
php
// admin/users.php

// ...

<h1>Administration des Utilisateurs</h1>

<!-- User management functionality goes here -->

// ...
```

### 2. Ajout d'un Nouvel Utilisateur

L'ajout d'un nouvel utilisateur est pris en charge par le fichier **register.php**. Les administrateurs peuvent utiliser la même page que les utilisateurs réguliers pour créer de nouveaux comptes.

```
php
// register.php

// ...

<h1>Inscription</h1>

<form method="POST" action="register.php">
    <!-- Registration form for both regular and admin users -->
</form>

// ...
```

### 3. Menu Admin (Partie du Header)

Le menu Admin est inclus dans le fichier **preset\_header.php** et apparaît dans le header pour les utilisateurs ayant le rôle d'administrateur. Il comprend des liens vers différentes fonctionnalités d'administration.

```
php
// preset_header.php

// ...

// Check if the user is logged in
if (isset($_SESSION['user_id'])) {
```



```

echo '<a href="catalogue.php">Catalogue</a>';

// Display the "Modify my profile" button
echo '<a class="profile-btn" href="profil.php">Mon profil</a>';

// Check if the user has the administrator role
if (isset($_SESSION['role']) && $_SESSION['role'] === 'admin') {
    // Display the Admin button with the dropdown menu
    echo '<div class="admin-dropdown">';
    echo '<a class="admin-btn" href="#">Admin</a>';
    echo '<div class="admin-dropdown-content">';
    echo '<a href="ajt.php">Ajouter un film ou une série</a>';
    echo '<a href="suppression_serie.php">Supprimer un film ou une
série</a>';
    echo '<a href="admin/users.php">Administration des
Utilisateurs</a>';
    echo '</div>';
    echo '</div>';
}

// Display the "Logout" link aligned to the right
echo '<a class="logout-link" href="logout.php">Déconnexion</a>';
} else {
    // If not, display links to the login and registration page
    echo '<a href="register.php">Inscription</a>';
    echo '<a href="login.php">Connexion</a>';
}

// ...

```

#### 4. Suppression d'Utilisateur (Partie Admin)

La suppression d'un utilisateur par un administrateur est gérée dans la page **admin/users.php**. Elle implique la sélection d'un utilisateur dans la liste et l'utilisation d'un formulaire pour déclencher l'action de suppression.

```

php
// admin/users.php

// ...

<!-- User deletion form -->
<form method="POST" action="admin/users.php">
    <label for="deleteUser">Supprimer un Utilisateur:</label>
    <select id="deleteUser" name="deleteUser">
        <!-- User options will be dynamically generated from the database -
->
    </select>
    <input type="submit" name="deleteUserBtn" value="Supprimer">
</form>

// ...

// Handling user deletion
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['deleteUserBtn'])) {
    $selectedUserId = $_POST['deleteUser'];

    // Perform the user deletion based on the selected user ID
    // ...

```

```
}
```

```
// ...
```

Cette section couvre les fonctionnalités d'administration des utilisateurs, notamment l'accès à la page d'administration des utilisateurs, l'ajout de nouveaux utilisateurs et la suppression d'utilisateurs existants. Consultez la documentation complète pour obtenir des détails plus approfondis sur ces fonctionnalités.

## XII. Sécurité et Bonnes Pratiques

Dans cette section, nous examinerons les aspects de sécurité et les bonnes pratiques à prendre en compte dans le développement de l'application de gestion de films et de séries.

### 1. Sécurité des Mot de Passe

La sécurité des mots de passe est gérée dans les fichiers **password.php** et **profil.php**. La fonction **password\_hash** est utilisée pour hacher les mots de passe avant de les stocker en base de données, et la vérification se fait avec **password\_verify** lors de la connexion et du changement de mot de passe.

```
php
// password.php

// ...

// Hashing the password
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);

// ...

// Verifying the password
if ($user && password_verify($old_password, $user['password'])) {
    // Password change logic
    // ...
}
```

### 2. Validation des Données du Formulaire

La validation des données du formulaire est effectuée dans le fichier **register.php** pour assurer que les données saisies par l'utilisateur sont conformes aux critères requis.

```
php
// register.php

// ...

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // ...

    // Validate username, email, and password
    // ...

    // Add the user to the database
    // ...
}
```

### 3. Protection Contre les Attaques par Injection SQL

Les requêtes SQL utilisées dans l'application sont préparées pour éviter les attaques par injection SQL. Par exemple, dans **suppression\_serie.php**, la suppression d'un film ou d'une série est effectuée en utilisant des requêtes préparées.

```

php
// suppression_serie.php

// ...

if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['delete_id'])) {
    $deleteId = $_POST['delete_id'];

    // Prepared statement to delete the film or series
    $stmt = $pdo->prepare("DELETE FROM films WHERE id = ?");
    $stmt->execute([$deleteId]);

    echo "Le film ou la série a été supprimé(e) avec succès.";
}

// ...

```

#### 4. Protection Contre les Attaques CSRF

La protection contre les attaques CSRF est mise en place avec l'utilisation de jetons CSRF dans le formulaire, comme on peut le voir dans **profil.php** lors de la modification du mot de passe.

```

php
// profil.php

// ...

<form method="POST" action="profil.php">
    <!-- CSRF token field -->
    <input type="hidden" name="csrf_token" value="<?php echo
$_SESSION['csrf_token']; ?>">

    <!-- Other form fields -->
    <!-- ...

    <input type="submit" name="changePassword" value="Changer le Mot de
Passe">
</form>

// ...

```

#### 5. Contrôle d'Accès et Gestion des Sessions

La gestion des sessions est utilisée pour contrôler l'accès aux pages restreintes. Par exemple, dans **profil.php**, la page est protégée et redirigée vers la page de connexion si l'utilisateur n'est pas authentifié.

```

php
// profil.php

// ...

// Check if the user is logged in
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php"); // Redirect to login page if not logged
in
    exit();
}

```

```
}  
  
// ...  
  
// ...
```

## 6. Redirection Après Opérations Sensibles

Après des opérations sensibles, comme la modification du mot de passe, une redirection est effectuée pour éviter le renvoi du formulaire lors du rafraîchissement de la page.

```
php  
// profil.php  
  
// ...  
  
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&  
isset($_POST['changePassword'])) {  
    // ...  
  
    // Redirect to prevent form resubmission on page refresh  
    header("Location: profil.php?success=1");  
    exit;  
}  
  
// ...
```

Cette section met en lumière les pratiques de sécurité mises en œuvre dans l'application, notamment la gestion des mots de passe, la validation des données, la protection contre les attaques SQL injection et CSRF, le contrôle d'accès, et les redirections après des opérations sensibles. Veillez à examiner attentivement ces aspects dans le cadre du développement de votre application.

## XIII. Guide de Déploiement

Pour déployer cette application de gestion de films et de séries sur votre propre serveur, suivez ces étapes. Assurez-vous que votre serveur répond aux exigences nécessaires.

### Prérequis

1. **Serveur Web** : Apache, Nginx, ou tout autre serveur web compatible.
2. **Base de Données** : MySQL, PostgreSQL, ou tout autre système de gestion de base de données pris en charge.
3. **PHP** : Version 7.0 ou supérieure.
4. **Composer** : Pour l'installation des dépendances PHP.

### Instructions de Déploiement

1. **Téléchargement du Code Source:**
  - Téléchargez le code source depuis le référentiel GitHub ou copiez-le sur votre serveur.
2. **Configuration de la Base de Données:**
  - Importez le fichier de schéma de base de données fourni (database\_schema.sql) dans votre base de données.
3. **Configuration du Fichier cnx.php:**
  - Dans le fichier cnx.php, assurez-vous que les informations de connexion à la base de données sont correctes.
4. **Installation des Dépendances avec Composer:**
  - Exécutez `composer install` pour installer les dépendances PHP.
5. **Configuration du Serveur Web:**
  - Configurez le serveur web pour pointer vers le répertoire où le code source est situé.
  - Assurez-vous que les permissions sont correctement définies pour les fichiers et répertoires.
6. **Lancement de l'Application:**
  - Accédez à l'URL de votre application via un navigateur web.
  - Vous devriez être redirigé vers la page de connexion.
7. **Création d'un Compte Administrateur:**
  - Manuellement, dans la base de données, attribuez le rôle 'admin' à un utilisateur pour lui donner des privilèges d'administration.
8. **Personnalisation Additionnelle:**
  - Personnalisez l'application en fonction de vos besoins, tels que le logo, le thème, etc.
9. **Sécurisation:**
  - Mettez en œuvre des mesures de sécurité telles que SSL, pare-feu, et assurez-vous que le serveur est à jour.
10. **Maintenance:**
  - Surveillez les journaux d'erreur, appliquez les mises à jour de sécurité et sauvegardez régulièrement la base de données.

**Note Importante:** Assurez-vous de respecter les licences des composants tiers utilisés dans l'application et de mettre à jour régulièrement les dépendances pour corriger les éventuelles vulnérabilités de sécurité.

## XIV. Résolution de Problèmes Courants

Au cours de l'utilisation ou du déploiement de l'application, vous pourriez rencontrer certains problèmes courants. Voici quelques solutions pour les résoudre.

### 1. Problème de Connexion

- **Symptôme** : Vous ne pouvez pas vous connecter à l'application.
- **Solution** :
  1. Assurez-vous que les informations de connexion dans le fichier `cnx.php` sont correctes.
  2. Vérifiez que la base de données est accessible et que le schéma a été correctement importé.
  3. Consultez les journaux d'erreur du serveur web pour des indications sur les erreurs.

### 2. Erreur de Suppression de Film ou de Série

- **Symptôme** : Lors de la suppression d'un film ou d'une série, une erreur se produit.
- **Solution** :
  1. Assurez-vous que l'ID du film ou de la série à supprimer est transmis correctement.
  2. Vérifiez les autorisations de base de données pour l'utilisateur utilisé par l'application.

### 3. Problème d'Affichage du Lecteur Vidéo

- **Symptôme** : Le lecteur vidéo ne s'affiche pas correctement.
- **Solution** :
  1. Vérifiez que les fichiers vidéo spécifiés dans la base de données existent réellement sur le serveur.
  2. Assurez-vous que le serveur web a les permissions nécessaires pour accéder aux fichiers vidéo.

### 4. Problème de Personnalisation

- **Symptôme** : Vous ne parvenez pas à personnaliser l'application selon vos besoins.
- **Solution** :
  1. Assurez-vous que les fichiers de style CSS sont accessibles et correctement référencés.
  2. Consultez la documentation du framework ou de la bibliothèque utilisé pour comprendre comment personnaliser le thème.

### 5. Problèmes de Performance

- **Symptôme** : L'application est lente ou non réactive.
- **Solution** :



1. Optimisez les requêtes SQL pour améliorer les performances de la base de données.
2. Vérifiez la charge du serveur et considérez l'optimisation du code PHP.
3. Utilisez des outils de profilage pour identifier les goulots d'étranglement.

## 6. Erreur de Configuration du Serveur Web

- **Symptôme** : L'application ne s'affiche pas correctement ou retourne une erreur 404.
- **Solution** :
  1. Assurez-vous que le serveur web est correctement configuré pour rediriger le trafic vers le bon répertoire.
  2. Vérifiez les fichiers de configuration du serveur web pour les erreurs de syntaxe.

## 7. Erreur de Mise en Page

- **Symptôme** : La mise en page de l'application semble incorrecte.
- **Solution** :
  1. Assurez-vous que les fichiers de style CSS sont correctement chargés.
  2. Vérifiez la validité du code HTML pour détecter des erreurs.

Ces solutions couvrent certains des problèmes courants, mais en cas de difficultés persistantes, consultez la documentation du framework, recherchez des solutions en ligne, ou demandez de l'aide à la communauté de développement.

## XV. Conclusion

Le projet "Matte en stream" offre une plateforme de gestion de films et séries qui vise à offrir une expérience de streaming personnalisée et conviviale. À travers une combinaison de fonctionnalités telles que la gestion des utilisateurs, la visualisation de contenu en streaming, et l'administration du catalogue, l'application aspire à répondre aux besoins diversifiés des amateurs de films et séries.

L'utilisation de PHP, HTML, et JavaScript constitue la base technologique de l'application, avec une gestion des données assurée par une base de données MySQL. L'architecture modulaire permet une évolutivité facile, offrant des opportunités d'expansion et d'amélioration continue.

L'intégration d'une interface utilisateur responsive, la gestion des utilisateurs avec différentes autorisations, et la fonctionnalité de streaming dynamique sont quelques-unes des caractéristiques clés. De plus, l'application prend en charge les fonctionnalités d'administration, permettant aux administrateurs de gérer le catalogue de films et séries de manière efficace.

Les évolutions futures suggérées visent à enrichir davantage l'expérience utilisateur en explorant des aspects tels que l'intelligence artificielle pour les recommandations, l'intégration de services externes, et des fonctionnalités sociales. La sécurité et la compatibilité multi-appareils sont également prioritaires pour garantir une expérience utilisateur sûre et sans heurts.

En résumé, "Matte en stream" sert de base solide pour une plateforme de streaming évolutive, offrant des fonctionnalités avancées et un potentiel continu d'améliorations pour s'adapter aux tendances changeantes et aux besoins des utilisateurs. Que ce soit pour les amateurs de cinéma ou les administrateurs de contenu, cette application aspire à créer une communauté dynamique autour du plaisir du streaming.