**Projet IPTV php**

**Page 1: index.html**

Le fichier **index.html** est la page d'accueil du projet IPTV. Voici un aperçu de son contenu :

La page commence par les balises HTML et les balises de tête, où les méta-informations, les liens vers les styles CSS et les polices sont définis.

La section body contient la barre de navigation avec un logo et plusieurs liens vers d'autres pages du site. Ensuite, il y a une section divisée en deux parties : à gauche, il y a un conteneur avec du texte promotionnel et des boutons pour l'inscription et la lecture vidéo, et à droite, il y a une image illustrative.

En dessous, il y a une section avec des cartes présentant différentes offres d'abonnement à IPTV, chacune avec des détails sur les fonctionnalités incluses et un bouton "Get Started" pour accéder au processus de paiement.

Ensuite, il y a une section mettant en évidence les meilleures fonctionnalités de l'IPTV, avec des cartes présentant les avantages tels que le divertissement illimité, les économies de coûts et le support instantané.

Enfin, il y a un pied de page avec des informations sur l'entreprise, des liens rapides vers différentes sections du site et des informations sur le compte utilisateur.

**registration.php**

Il vérifie si le paramètre "id" est défini dans la chaîne de requête de l'URL ($\_GET["id"]). S'il l'est, le script suppose qu'il est utilisé pour mettre à jour les informations d'un utilisateur existant. Sinon, le script suppose qu'il est utilisé pour l'inscription des utilisateurs.

Si le paramètre "id" est défini, il récupère les informations de l'utilisateur à partir de la base de données en utilisant l'id fourni. Les informations de l'utilisateur sont stockées dans des variables telles que $firstName, $lastName, $phone, $email et $password.

Si le bouton de mise à jour est cliqué (c'est-à-dire que $\_POST["update"] est défini), il récupère les valeurs mises à jour à partir du formulaire soumis (variables $\_POST) et construit une requête SQL pour mettre à jour les informations de l'utilisateur dans la base de données.

La requête SQL est exécutée à l'aide de mysqli\_query, et si elle échoue, un message d'erreur est affiché. Si la requête est réussie, le script redirige l'utilisateur vers une page admin.php.

La structure HTML représente un formulaire d'inscription. L'attribut action du formulaire est défini sur "admin.php" si le paramètre "id" est défini, ce qui indique que le formulaire sera utilisé pour mettre à jour les informations de l'utilisateur. Sinon, l'attribut action du formulaire est laissé vide, ce qui indique que le formulaire sera utilisé pour l'inscription des utilisateurs.

Les champs du formulaire sont pré-remplis avec les informations de l'utilisateur si le paramètre "id" est défini, en utilisant des balises PHP (<?php ?>) et les valeurs récupérées précédemment.

Il y a du code PHP supplémentaire après le formulaire qui gère l'inscription des utilisateurs ($\_POST["register"]).

Il vérifie si les champs requis sont vides. Si l'un d'entre eux est vide, un message d'erreur est affiché.

Si tous les champs requis sont remplis, il établit une connexion à la base de données en utilisant le fichier connexion.php (non fourni), récupère les valeurs du formulaire et construit une requête SQL pour insérer les informations du nouvel utilisateur dans la base de données.

La requête SQL est exécutée à l'aide de mysqli\_query. Si la requête réussit, un message de succès est affiché. Sinon, un message d'erreur est affiché.

**login.php ( exemple :** [**admin@gmail.com**](mailto:admin@gmail.com)**// admin )**

La structure HTML représente un formulaire de connexion. Il se compose de deux champs de saisie pour l'e-mail et le mot de passe, ainsi que d'un bouton de soumission.

Lorsque le formulaire est soumis (c'est-à-dire que $\_POST["login"] est défini), le code **PHP** à l'intérieur du bloc if (isset($\_POST["login"])) est exécuté.

Il vérifie si les champs de l'e-mail et du mot de passe sont vides. Si l'un d'entre eux est vide, un message d'erreur est affiché.

Si les champs de l'e-mail et du mot de passe sont remplis, il récupère les valeurs du formulaire soumis (variables $\_POST) et établit une connexion à la base de données en utilisant le fichier connexion.php (non fourni).

Il construit une requête SQL pour sélectionner les enregistrements d'utilisateurs de la base de données en fonction de l'e-mail et du mot de passe fournis.

La requête SQL est exécutée à l'aide de mysqli\_query. S'il existe au moins un enregistrement d'utilisateur correspondant (mysqli\_num\_rows($query) > 0), l'utilisateur est redirigé vers une page index.html et le script se termine.

Si aucun enregistrement d'utilisateur correspondant n'est trouvé, un message "Utilisateur non trouvé" est affiché.

Les balises de fermeture **</form>** et **</body>** sont mal placées et doivent être déplacées juste au-dessus du code PHP, juste après le bouton de soumission.

**tv\_channels.php**

La structure HTML commence par la déclaration **<!DOCTYPE html>** et la balise d'ouverture **<html>.**

La section <head> inclut des balises meta pour le codage des caractères et les paramètres de l'viewport, ainsi que des liens vers des bibliothèques CSS externes et Bootstrap.

L'élément **<title>** définit le titre de la page Web comme "TV Channels".

Dans la section **<head>**, il y a également un bloc **<style>** qui définit des règles CSS personnalisées pour la page.

La section **<body>** contient le contenu de la page Web.

L'élément <nav> représente la barre de navigation en haut de la page. Il utilise les classes CSS de Bootstrap pour styliser la barre de navigation.

Dans la barre de navigation, il y a un logo affiché dans un élément <img>.

La barre de navigation contient également une liste de liens de navigation (<a>) vers différentes pages comme "Accueil", "Chaînes de télévision", "Contact", "Connexion" et "Inscription".

Le contenu principal de la page est enveloppé dans un élément <div> avec la classe "container content".

À l'intérieur de ce conteneur, il y a un autre élément <div> avec la classe "container-fluid content2" qui représente la section des informations de contact.

Les informations de contact sont affichées à l'aide d'éléments <p> et d'éléments <i> avec les icônes Material.

À la fin de la page, il y a une balise de fermeture </body> et une balise de fermeture </html>.

**delete.php**

Ce fichier contient du code PHP pour supprimer des utilisateurs de la base de données.

Il vérifie d'abord si l'ID de l'utilisateur à supprimer est passé en paramètre.

Ensuite, il se connecte à la base de données et exécute une requête **DELETE** pour supprimer l'utilisateur avec l'ID spécifié.

Le même processus est répété pour la suppression de cartes de crédit ou de comptes PayPal, en fonction de la méthode de paiement spécifiée.

Si la suppression réussit, l'utilisateur est redirigé vers la page "admin.php"

**payment.php**

Le formulaire comporte les champs suivants :

* ***Nom***
* ***Téléphone***
* ***Type (qui est un champ de texte)***
* ***Options de paiement (PayPal et carte de crédit)***

En fonction de la valeur du paramètre "promo" passé dans l'URL, le champ "Type" est pré-rempli avec une valeur spécifique et est désactivé (ce qui signifie que l'utilisateur ne peut pas le modifier). Cela semble être une fonctionnalité liée à une promotion.

Le code JavaScript présent dans la page est responsable de gérer l'affichage des champs pertinents en fonction de la méthode de paiement sélectionnée (PayPal ou carte de crédit). Lorsque l'utilisateur sélectionne l'une des options de paiement, les champs correspondants sont affichés, tandis que les autres sont masqués.

Le code PHP qui suit le formulaire gère le traitement des données envoyées lorsque le formulaire est soumis. Il se connecte à une base de données (à l'aide du fichier "connexion.php" qui n'est pas présent ici) et insère les données du formulaire dans les tables "paypal" et "credit\_card" en fonction de la valeur du paramètre "promo".

Il est important de noter que ce code est vulnérable aux attaques de sécurité, car il n'effectue pas de validation, d'échappement ou de nettoyage des données entrées par l'utilisateur. Il est fortement recommandé de mettre en place des mesures de sécurité appropriées, telles que la validation des données, l'utilisation de requêtes préparées et la protection contre les attaques par injection SQL, avant de déployer ce code dans un environnement de production.

En résumé, cette page HTML et PHP permet aux utilisateurs de saisir des informations de paiement et de les envoyer au serveur pour traitement. Le code gère également l'affichage des champs appropriés en fonction de la méthode de paiement sélectionnée.

**paym.php(c’est different au fichier payment.php)**

Tout d'abord, la page vérifie si les paramètres id et method sont présents dans l'URL via **$\_GET**. Ces paramètres sont utilisés pour déterminer quelle méthode de paiement a été sélectionnée et quel enregistrement de la base de données doit être récupéré.

Ensuite, la page inclut le fichier "connexion.php" qui est responsable de la connexion à la base de données.

En fonction de la valeur de method, la page construit une requête SQL pour récupérer les informations de paiement à partir de la table correspondante (paypal ou credit\_card).

Les informations récupérées sont stockées dans des variables, telles que $name\_aff, $phone\_aff, $type\_aff, $email\_aff, etc., pour les utiliser dans le formulaire.

La structure HTML du formulaire est définie, avec des champs de saisie pour les différentes informations de paiement. Les valeurs des champs sont pré-remplies en utilisant les variables contenant les données de paiement récupérées précédemment.

Le formulaire comporte également deux boutons radio pour sélectionner la méthode de paiement : PayPal ou Carte de crédit. Il y a également deux sections de champ de saisie distinctes pour chaque méthode de paiement (une pour PayPal et une pour la carte de crédit). Ces sections sont affichées ou masquées en fonction de la méthode de paiement sélectionnée à l'aide de JavaScript.

Le code JavaScript gère le changement de méthode de paiement en affichant ou masquant les champs de saisie appropriés en fonction de la sélection de l'utilisateur.

Enfin, lorsque le formulaire est soumis, le code PHP vérifie si le bouton de paiement (name="payer") a été soumis. Si c'est le cas, il récupère les valeurs des champs de saisie et met à jour la base de données avec les nouvelles informations de paiement pour l'enregistrement correspondant.

**contact.php**

Le fichier HTML fourni représente une page de contact pour un site Web. Voici une explication de ce que fait le code :

La structure HTML commence par la déclaration <!DOCTYPE html> et la balise d'ouverture <html> avec l'attribut lang="en".

La section <head> contient des balises meta pour le codage des caractères, les paramètres de compatibilité, et l'viewport. Elle inclut également des liens vers des bibliothèques de polices et du framework Bootstrap.

Le titre de la page est défini dans la balise <title> comme "TV Channels".

Dans la section <head>, il y a aussi un bloc <style> qui définit des règles CSS personnalisées pour la mise en page et le style de la page.

La section <body> contient le contenu de la page.

La navigation est représentée par l'élément <nav> avec la classe "navbar". Elle utilise des classes CSS de Bootstrap pour styliser la barre de navigation. La barre de navigation contient un logo et une liste de liens de navigation vers différentes pages du site.

Le contenu principal de la page est enveloppé dans un élément <div> avec la classe "container content". À l'intérieur de ce conteneur, il y a un autre <div> avec la classe "container-fluid content2" qui représente la section des informations de contact.

Les informations de contact sont affichées à l'aide d'éléments <p> et <i> avec des icônes Material. Les coordonnées de contact, telles que le numéro de téléphone, l'adresse e-mail et l'adresse, sont affichées avec des icônes correspondantes.

Enfin, il y a un lien vers les réseaux sociaux (Facebook) représenté par un élément <a> et une icône correspondante.

La balise de fermeture </body> et la balise de fermeture </html> terminent le fichier HTML.

***Admin.php (CRUD)***

La balise HTML comprend un titre, des styles CSS et du code **JavaScript**.

Il y a deux en-têtes avec les ID "users" et "sub" qui servent d'onglets pour basculer entre l'affichage des utilisateurs et des abonnés.

Les éléments "**users\_part" et "subscribers\_part"** sont des div qui contiennent les tableaux des utilisateurs et des abonnés, respectivement. Initialement, les deux div sont masqués.

Le code PHP récupère les données des utilisateurs à partir d'une base de données en utilisant des requêtes SQL et affiche les données dans les tableaux correspondants.

Le tableau des utilisateurs comprend des colonnes pour le prénom, le nom de famille, le numéro de téléphone, l'e-mail, le mot de passe et des boutons d'action pour mettre à jour et supprimer les utilisateurs.

Le tableau des abonnés se compose de deux sections : une pour les abonnés PayPal et l'autre pour les abonnés par carte de crédit. Les deux sections récupèrent les données à partir de tables distinctes de la base de données et les affichent dans des tableaux séparés.

Chaque tableau d'abonnés comprend des colonnes pour le nom, le numéro de téléphone, l'e-mail, le mot de passe (pour les abonnés PayPal), le numéro de carte, la date d'expiration, le code CVV (pour les abonnés par carte de crédit), le type de IPTV et **des boutons d'action pour mettre à jour et supprimer les abonnés.**