**Url du site :** [**http://www.kyufi-radhouan-clement.baycii.com/home.php**](http://www.kyufi-radhouan-clement.baycii.com/home.php)

**Url pour se connecter  :** [**http://www.kyufi-radhouan-clement.baycii.com/login.php**](http://www.kyufi-radhouan-clement.baycii.com/login.php)

**En tant que user :**

**Username :** [**titi@gmail.com**](mailto:titi@gmail.com)

**Mot de passe : titi**

**En tant que admin :**

**Username :** [**kyufi614@gmail.com**](mailto:kyufi614@gmail.com)

**Mot de passe : pise2023**

**Présentation de notre site web**

Notre site web est une plateforme dédiée à la vente de cafés et de gâteaux américains. Nous proposons une large variété de produits pour satisfaire les goûts de nos clients. Pour profiter pleinement de notre service, nous invitons les utilisateurs à créer un compte en tant qu'utilisateur. Une fois inscrit, ils auront la possibilité de passer des commandes et d'accéder à l'ensemble de nos produits.

Le site est structuré de manière à offrir une expérience conviviale. Tout commence par notre en-tête (header) qui comprend des éléments essentiels tels que notre logo distinctif. Les boutons "Home" permettent d'accéder à la page d'accueil, "Products" pour découvrir l'ensemble de nos produits, "Reviews" pour consulter les avis de nos clients et "Join Us" pour remplir un formulaire et postuler à un poste qui correspond à vos aspirations. Le bouton "About" vous permet d'en savoir plus sur notre concept.

Une barre de recherche est également disponible, vous permettant de trouver rapidement les produits de votre choix. Par exemple, en tapant simplement la première lettre d'un produit, vous obtiendrez une liste des produits commençant par cette lettre. De plus, un bouton de traduction est disponible pour traduire la page d’accueil du site.

Lorsque vous êtes connecté en tant qu'utilisateur, vous avez la possibilité de commander vos produits préférés et de visualiser vos informations personnelles en cliquant sur dans la barre de navigation. En revanche, si vous êtes administrateur, vous aurez accès à une page offrant la possibilité de gérer les données.

Le corps principal de la page est divisé en plusieurs sections qui présentent différents types de contenu. Tout d’abord, on a la section ou on a deux boutons : un bouton « shop now » pour aller directement aux produits et un bouton  « find us» pour voir l’emplacement de nos locaux. La section « Our Products »affiche une liste de tous nos produits disponibles, avec la possibilité de voir plus de produits en cliquant sur « View More Products » si le nombre dépasse une certaine limite. La section « Customers Reviews » met en avant les avis de nos clients et offre la possibilité à chacun de laisser son propre avis. Enfin, la section  « Our History » vous plonge dans l'histoire de notre entreprise et présente nos produits phares.

Nous mettons tout en œuvre pour offrir une expérience agréable et satisfaisante à nos clients, en proposant une gamme de produits, des fonctionnalités pratiques et un contenu informatif sur notre entreprise « Kyufi » et nos valeurs. N'hésitez pas à explorer notre site et à découvrir tout ce que nous avons à vous offrir.

Ici nous présentons l’utilité é de chaque fichier utilisé :

**Home.php:**

Tout d'abord, le code inclut le fichier header.php, qui doit contenir le code HTML commun à toutes les pages du site.Ensuite, le code vérifie si le formulaire de recherche a été soumis. Si c'est le cas, il appelle la fonction findProductsCriteria() avec la valeur du champ "criteria" du formulaire. Sinon, il appelle la fonction findAllProducts() pour récupérer tous les produits. Les résultats sont stockés dans la variable $products.

Ensuite, le code vérifie si le paramètre GET "upto" est présent dans l'URL. Si c'est le cas, sa valeur est assignée à la variable $upto. Sinon, la valeur par défaut est 9.

Le code HTML de la page d'accueil commence avec une section contenant du texte et des boutons d'action. Ensuite, il y a une section affichant une liste de produits. Le code utilise une boucle foreach pour parcourir les produits et afficher leurs détails, tels que l'image, le nom et le prix.

**Header.php :** représente le header de toutes nos pages

**Login.php :**

Il s’agit d’un formulaire de connexion où l'utilisateur peut entrer son nom d'utilisateur et son mot de passe. Si l'utilisateur n'est pas déjà connecté, le formulaire envoie les informations saisies à la fonction loginUser ().Elle vérifie ensuite si l'un de ces champs est vide à l'aide de la fonction. Si l'un des champs est vide, la fonction loginUser enregistre un message d'erreur dans la variable de session et redirige l'utilisateur vers la page de connexion avec un paramètre d'erreur "emptyinput" dans l'URL .Si l'utilisateur est déjà connecté, il est redirigé vers la page d'accueil.

**Signup.php :**

Le code PHP récupère les valeurs des champs du formulaire grâce à la superglobale $\_POST. Il utilise ensuite des fonctions par exemple, checkUid($uid) vérifie si l'ID utilisateur est déjà pris, checkTel($tel) vérifie si le numéro de téléphone est déjà utilisé, et checkEmail($email) vérifie si l'email est déjà enregistré. Si ces trois conditions sont respectées, il crée un nouvel utilisateur avec la fonction setUser ($uid, $pwd, $email, $tel, $address) et redirige l'utilisateur vers la page d'accueil.

Dans le cas contraire, il redirige l'utilisateur vers la page de connexion avec un message d'erreur dans l'URL (par exemple 'emailTaken', 'telTaken', 'uidTaken') qui est utilisé pour afficher un message d'erreur correspondant sur la page.

Le script vérifie également si l'utilisateur est connecté et s'il a des produits dans son panier. Si c'est le cas, il récupère le nombre de produits dans le panier et le stocke dans la variable $nproducts

**Product\_details** utilise différentes fonctions et variables pour afficher les détails d'un produit spécifique sur une page web. Tout d'abord, il inclut le fichier "header.php" qui contient le code HTML pour l'en-tête de la page. Ensuite, il établit une connexion à la base de données à l'aide de la fonction connect().

Ensuite, le code vérifie si l'ID du produit est présent dans l'URL. S'il est manquant, l'utilisateur est redirigé vers la page d'accueil avec un message d'erreur correspondant. Si l'ID du produit est présent, il est stocké dans la variable $product\_id. Le code utilise ensuite la fonction getProductDetails($con, $product\_id) pour récupérer les détails du produit à partir de la base de données, en passant la connexion $con et l'ID du produit $product\_id comme arguments.

Si le produit est introuvable dans la base de données, l'utilisateur est à nouveau redirigé vers la page d'accueil avec un message d'erreur approprié. Cependant, si le produit est trouvé, ses informations, y compris son nom, son image et sa description, sont affichées sur la page.

Le code vérifie également si le produit a plusieurs options en vérifiant la taille du tableau $product['options']. Si c'est le cas, il affiche les différentes options avec des boutons radio pour permettre à l'utilisateur de faire son choix. Chaque option est associée à un prix. Si le produit n'a qu'une seule option, le prix est affiché directement.

**Commander.php :** Ce fichier traite l'ajout d'articles au panier par l'utilisateur grâce à la fonction isset($\_POST['addtocart']), qui vérifie si le formulaire a été soumis. Si un utilisateur sélectionne une option de produit, cette option est ajoutée au panier de l'utilisateur qui est stocké dans la session. De plus, la fonction explode() est utilisée pour diviser les informations de l'option en plusieurs parties. Ensuite, le code vérifie si l'utilisateur a cliqué sur le bouton "checkout" avec isset($\_POST['checkout']) et si toutes les informations requises ont été fournies. Si c'est le cas, une commande est créée avec les informations fournies par l'utilisateur en utilisant la fonction create\_command(), et un message de confirmation est affiché.

**Rate.php :** Dans cette partie , l’utilisateur rempli un formulaire donné pour donner son avis.Si l'utilisateur a soumis un commentaire via un formulaire (détecté par isset($\_POST['comment'])), ce commentaire, ainsi que les informations de contact de l'utilisateur (nom, numéro de téléphone et adresse e-mail) sont passés à la fonction addRate() pour être traités. Si un commentaire a été soumis, une image de confirmation est affichée, et l'exécution du script est arrêtée avec exit().

**Account.php :**   
Le code propose plusieurs fonctionnalités pour l'ajout, la suppression et la modification des rôles des utilisateurs, ainsi que l'ajout et la suppression des produits dans une base de données.

Pour l'ajout de produits, le code vérifie si un formulaire d'ajout de produit a été soumis. Si c'est le cas, il récupère les détails du produit, tels que le nom, le prix, l'image, le type et la description, à partir du formulaire. Ensuite, ces informations sont transmises à la fonction "addProduct()" qui les traite et ajoute le produit à la table "products" de la base de données.

Pour la suppression de produits, le code vérifie si un formulaire de suppression de produit a été soumis. Lorsque le formulaire est soumis, l'ID du produit à supprimer est extrait du formulaire et passé à la fonction "deleteProduct()". Cette fonction se charge de supprimer le produit correspondant de la table "products" de la base de données.

Pour la modification du rôle d'un utilisateur, le code vérifie si un formulaire de changement de rôle a été soumis. Lorsque le formulaire est soumis, l'ID de l'utilisateur concerné et le nouveau rôle sélectionné sont récupérés. Ensuite, la fonction "changeUserRoleById()" est appelée avec ces paramètres pour mettre à jour le rôle de l'utilisateur dans la table "users" de la base de données.

En ce qui concerne l'ajout d'utilisateurs, le code vérifie si un formulaire d'ajout d'utilisateur a été soumis. Si c'est le cas, il extrait les détails de l'utilisateur, tels que le nom d'utilisateur, le mot de passe, le numéro de téléphone, l'adresse e-mail, l'adresse et le rôle, à partir du formulaire. Ensuite, ces informations sont passées à la fonction "addUser()" qui les utilise pour ajouter un nouvel enregistrement dans la table "users" de la base de données.

Enfin, pour la suppression d'utilisateurs, le code vérifie si un formulaire de suppression d'utilisateur a été soumis. Lorsque le formulaire est soumis, l'ID de l'utilisateur à supprimer est extrait du formulaire et passé à la fonction "deleteUserById()". Cette fonction supprime l'enregistrement de l'utilisateur correspondant de la table "users" de la base de données.

**Findjoinabout.php**

Ce code est une page PHP qui affiche différentes sections en fonction de la valeur de la variable $page.

Tout d'abord, le code inclut le fichier header.php, qui doit contenir le code HTML commun à toutes les pages du site.

Ensuite, la valeur de la variable $page est définie en fonction des paramètres de requête. Si le paramètre page est présent dans la requête, sa valeur est attribuée à $page. Sinon, la valeur par défaut est "findus".

Ensuite, le code vérifie la valeur de $page pour déterminer quelle section afficher.

Si $page est "addemployee", cela signifie que le formulaire pour ajouter un employé a été soumis. Dans ce cas, le code extrait les données du formulaire telles que le nom d'utilisateur, le mot de passe, le numéro de téléphone, l'âge, le poste, l'e-mail et le genre. Ces informations sont ensuite passées à la fonction addEmployee(), qui les utilise pour ajouter un nouvel employé à la base de données. Ensuite, une image et un lien de retour à la page d'accueil sont affichés.

Si $page est "findus", cela signifie que la section "Nous trouver" doit être affichée. Cette section peut contenir du code HTML et des scripts JavaScript pour afficher une carte. Le code HTML et JavaScript spécifique à cette section n'est pas inclus dans l'extrait de code fourni, mais il est référencé dans le code.

Si $page est "joinus", cela signifie que le formulaire pour rejoindre l'équipe a été soumis. Dans cette section, il y a un formulaire avec des champs pour le nom, le mot de passe, le numéro de téléphone, l'âge, l'e-mail, le genre, la position et le CV de l'utilisateur. Le code effectue une validation des champs et affiche les messages d'erreur appropriés s'il y a des erreurs de saisie. Lorsque le formulaire est soumis, une partie du code vérifie si un fichier a été téléchargé et le stocke dans un répertoire spécifié. Le formulaire est envoyé à la page findjoinabout.php?page=addemployee, où le traitement est effectué.

Si $page est "aboutus", cela signifie que la section "À propos de nous" doit être affichée. Cette section contient des informations sur l'entreprise et des liens vers d'autres pages du site.