Bing

Bonjour, je suis Bing, un chatbot qui peut vous aider avec diverses tâches. Je suis impressionné par votre projet d'application pour apprendre l'anglais de manière ludique et en équipe. C'est une idée très originale et créative. 😊

Vous me demandez de rédiger le code correspondant à votre cahier des charges. Je dois vous avouer que c'est une demande assez complexe et que je ne suis pas un expert en développement web. Je vais essayer de vous aider du mieux que je peux, mais je ne peux pas vous garantir que le code que je vais vous fournir sera parfait ou complet. Il faudra peut-être le modifier ou l'améliorer selon vos besoins et vos préférences. 🙏

Voici un exemple de code que j'ai généré pour votre application, en utilisant le langage PHP et MySQL comme SGBD. J'ai essayé de respecter les standards professionnels les plus récents du développement, de découper le code en classes et fonctions, de gérer les erreurs, et de documenter le code. J'espère que cela vous sera utile. 😊

<?php

// Asseguim est une application pour apprendre l'anglais de manière ludique et en équipe

// Ce fichier contient le code principal de l'application

// Inclure les fichiers contenant les classes et les fonctions utilisées par l'application

require\_once 'config.php'; // Fichier contenant les paramètres de connexion à la base de données

require\_once 'user.php'; // Fichier contenant la classe User qui représente un utilisateur de l'application

require\_once 'admin.php'; // Fichier contenant la classe Admin qui hérite de la classe User et qui représente un administrateur de l'application

require\_once 'question.php'; // Fichier contenant la classe Question qui représente une question du jeu

require\_once 'division.php'; // Fichier contenant la classe Division qui représente une division du classement des joueurs

// Créer une instance de la classe PDO pour se connecter à la base de données

try {

$db = new PDO('mysql:host=' . DB\_HOST . ';dbname=' . DB\_NAME, DB\_USER, DB\_PASS);

$db->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

die('Erreur : ' . $e->getMessage());

}

// Créer trois comptes administrateurs par défaut s'ils n'existent pas déjà dans la base de données

$admins = array(

array('nom' => 'Admin1', 'prenom' => 'Admin1', 'statut' => 'Administrateur', 'ville' => 'Casablanca', 'niveau' => 'Master', 'email' => 'admin1@asseguim.com', 'password' => 'admin1', 'phone' => '+212600000001'),

array('nom' => 'Admin2', 'prenom' => 'Admin2', 'statut' => 'Administrateur', 'ville' => 'Rabat', 'niveau' => 'Doctorat', 'email' => 'admin2@asseguim.com', 'password' => 'admin2', 'phone' => '+212600000002'),

array('nom' => 'Admin3', 'prenom' => 'Admin3', 'statut' => 'Administrateur', 'ville' => 'Marrakech', 'niveau' => 'Ingénieur', 'email' => 'admin3@asseguim.com', 'password' => 'admin3', 'phone' => '+212600000003')

);

foreach ($admins as $admin) {

// Vérifier si l'email ou le numéro de téléphone du compte administrateur existe déjà dans la base de données

$sql = "SELECT \* FROM users WHERE email = :email OR phone = :phone";

$stmt = $db->prepare($sql);

$stmt->execute(array(':email' => $admin['email'], ':phone' => $admin['phone']));

$result = $stmt->fetch(PDO::FETCH\_ASSOC);

// Si le compte administrateur n'existe pas, l'insérer dans la base de données

if ($result == false) {

$sql = "INSERT INTO users (nom, prenom, statut, ville, niveau, email, password, phone) VALUES (:nom, :prenom, :statut, :ville, :niveau, :email, :password, :phone)";

$stmt = $db->prepare($sql);

$stmt->execute(array(':nom' => $admin['nom'], ':prenom' => $admin['prenom'], ':statut' => $admin['statut'], ':ville' => $admin['ville'], ':niveau' => $admin['niveau'], ':email' => $admin['email'], ':password' => password\_hash($admin['password'], PASSWORD\_DEFAULT), ':phone' => $admin['phone']));

}

}

// Créer neuf divisions par défaut s'il n'existent pas déjà dans la base de données

$divisions = array(

array('name' => 'Mercure', 'rank' => 1),

array('name' => 'Vénus', 'rank' => 2),

array('name' => 'Terre', 'rank' => 3),

array('name' => 'Mars', 'rank' => 4),

array('name' => 'Jupiter', 'rank' => 5),

array('name' => 'Saturne', 'rank' => 6),

array('name' => 'Uranus', 'rank' => 7),

array('name' => 'Neptune', 'rank' => 8),

array('name' => 'Pluton', 'rank' => 9)

);

foreach ($divisions as $division) {

// Vérifier si le nom de la division existe déjà dans la base de données

$sql = "SELECT \* FROM divisions WHERE name = :name";

$stmt = $db->prepare($sql);

$stmt->execute(array(':name' => $division['name']));

$result = $stmt->fetch(PDO::FETCH\_ASSOC);

// Si la division n'existe pas, l'insérer dans la base de données

if ($result == false) {

$sql = "INSERT INTO divisions (name, rank) VALUES (:name, :rank)";

$stmt = $db->prepare($sql);

$stmt->execute(array(':name' => $division['name'], ':rank' => $division['rank']));

}

}

// Démarrer une session pour stocker les informations de l'utilisateur connecté

session\_start();

// Initialiser une variable pour stocker le message d'erreur ou de succès à afficher à l'utilisateur

$message = '';

// Vérifier si l'utilisateur a soumis le formulaire de connexion

if (isset($\_POST['login'])) {

// Récupérer les données du formulaire de connexion

$email\_or\_phone = $\_POST['email\_or\_phone'];

$password = $\_POST['password'];

// Vérifier si les champs du formulaire sont remplis

if (!empty($email\_or\_phone) && !empty($password)) {

// Vérifier si l'email ou le numéro de téléphone saisi correspond à un utilisateur dans la base de données

$sql = "SELECT \* FROM users WHERE email = :email\_or\_phone OR phone = :email\_or\_phone";

$stmt = $db->prepare($sql);

$stmt->execute(array(':email\_or\_phone' => $email\_or\_phone));

$result = $stmt->fetch(PDO::FETCH\_ASSOC);

// Si l'utilisateur existe, vérifier si le mot de passe saisi est correct

if ($result != false) {

if (password\_verify($password, $result['password'])) {

// Si le mot de passe est correct, créer un objet User ou Admin selon le statut de l'utilisateur et le stocker dans la session

if ($result['statut'] == 'Administrateur') {

$\_SESSION['user'] = new Admin($result['id'], $result['nom'], $result['prenom'], $result['statut'], $result['ville'], $result['niveau'], $result['email'], $result['phone']);

} else {

$\_SESSION['user'] = new User($result['id'], $result['nom'], $result['prenom'], $result['statut'], $result['ville'], $result['niveau'], $result['email'], $result['phone']);

}

// Rediriger l'utilisateur vers la page du jeu

header('Location: game.php');

exit();

} else {

// Si le mot de passe est incorrect, afficher un message d'erreur

$message = 'Mot de passe incorrect';

}

} else {

// Si l'utilisateur n'existe pas, afficher un message d'erreur

$message = "Aucun utilisateur trouvé avec cet email ou ce numéro de téléphone";

}

} else {

// Si les champs du formulaire sont vides, afficher un message d'erreur

$message = "Veuillez remplir tous les champs";

}

}

// Vérifier si l'utilisateur a soumis le formulaire d'inscription

if (isset($\_POST['register'])) {

// Récupérer les données du formulaire d'inscription

$nom = $\_POST['nom