

Développement d'une Application de Gestion Web

Gestion des réservations (hôtels, restaurants, salles).

Supervisé par:
AMAMOU Ahmed

Préparé par:
Adam Mentagui
Youssef Misaoui

Sommaire:

1-Introduction

2-Les étapes de développement

3-les difficultés rencontrées

4-les solutions apportées

Introduction

L'application de Gestion des Réservations a été développée pour simplifier et optimiser la gestion des réservations dans les hôtels, restaurants et salles de réception. Elle vise à répondre aux besoins des établissements en leur offrant un outil centralisé et facile à utiliser pour gérer leurs disponibilités, suivre les réservations et améliorer leur efficacité opérationnelle.

- **Objectifs de l'application :**

1. Centraliser les réservations : Permettre aux établissements de gérer les réservations pour les hôtels, restaurants et salles sur une seule plateforme.
2. Faciliter la gestion : Offrir aux administrateurs un outil intuitif pour suivre et organiser les réservations en temps réel.
3. Améliorer l'expérience client : Proposer une interface simple et rapide pour que les clients puissent effectuer des réservations sans difficulté.
4. Générer des statistiques : Fournir des données et des graphiques pour aider les gestionnaires à prendre des décisions éclairées.

Les étapes de développement

1-La logique de l'application:

Dans cette logique on aura deux acteurs qui ont des différentes fonctionnalités: utilisateur et administrateur.

Fonctionnalités d'administrateurs:

1. Gestion des Hôtels:

Ajouter un hôtel : L'administrateur peut ajouter un nouvel hôtel en renseignant des informations telles que le nom, l'adresse, la ville, le code postal, le téléphone, la description et la date de création.

Modifier un hôtel : L'administrateur peut modifier les informations d'un hôtel existant.

Supprimer un hôtel : L'administrateur peut supprimer un hôtel de la liste.

Afficher la liste des hôtels : L'administrateur peut consulter la liste des hôtels avec leurs informations détaillées.

2. Gestion des Restaurants:

Ajouter un restaurant : L'administrateur peut ajouter un nouveau restaurant en renseignant des informations telles que le nom, l'adresse, la ville, le code postal, le téléphone, la description et la date de création.

Modifier un restaurant : L'administrateur peut modifier les informations d'un restaurant existant.

Supprimer un restaurant : L'administrateur peut supprimer un restaurant de la liste.

Afficher la liste des restaurants : L'administrateur peut consulter la liste des restaurants avec leurs informations détaillées.

3. Gestion des Salles:

Ajouter une salle : L'administrateur peut ajouter une nouvelle salle en renseignant des informations telles que le nom, l'adresse, le code postal, le téléphone, la capacité, la description et la date de création.

Modifier une salle : L'administrateur peut modifier les informations d'une salle existante.

Supprimer une salle : L'administrateur peut supprimer une salle de la liste.

Afficher la liste des salles : L'administrateur peut consulter la liste des salles avec leurs informations détaillées.

4. Gestion des Réservations:

Afficher la liste des réservations : L'administrateur peut consulter la liste des réservations avec des informations telles que l'utilisateur ID, le type d'établissement, l'établissement ID, la date de réservation, l'heure de réservation, le statut et la date de création.

Supprimer une réservation : L'administrateur peut supprimer une réservation de la liste.

5. Statistiques des Réservations:

Afficher les statistiques des réservations : L'administrateur peut consulter un graphique en camembert (doughnut chart) qui montre le nombre de réservations par catégorie (hôtels, restaurants, salles).

6. Tableau de Bord Administrateur:

Accéder au tableau de bord : L'administrateur a accès à un tableau de bord qui lui permet de naviguer entre les différentes sections (hôtels, restaurants, salles, réservations, statistiques).

Se déconnecter : L'administrateur peut se déconnecter de la session administrateur.

7. Sécurité:

Vérification de la session administrateur : Chaque page vérifie si l'administrateur est connecté. Si ce n'est pas le cas, l'utilisateur est redirigé vers la page de connexion.

Protection contre les injections SQL : Les requêtes SQL utilisent des requêtes préparées pour éviter les injections SQL.

8. Interface Utilisateur:

Design moderne et responsive : L'interface utilisateur est conçue avec un design moderne et est responsive, ce qui la rend facile à utiliser sur différents appareils.

Messages de succès et d'erreur : Des messages de succès sont affichés après l'ajout, la modification ou la suppression d'un élément. Des messages d'erreur sont également affichés si nécessaire.

9. Navigation:

Navigation facile : L'administrateur peut facilement naviguer entre les différentes sections (hôtels, restaurants, salles, réservations, statistiques) via des boutons de navigation clairs.

10. Gestion des Dates:

Gestion des dates de création : L'administrateur peut spécifier la date de création des hôtels, restaurants et salles lors de leur ajout ou modification

Fonctionnalités d'utilisateurs:

1. Authentification et Gestion de Session

- **Connexion/Déconnexion :**
 - L'utilisateur doit être connecté pour accéder aux fonctionnalités.
 - Si l'utilisateur n'est pas connecté, il est redirigé vers la page de connexion (Login.php).
 - Une fois connecté, la session est maintenue pour permettre l'accès aux autres pages.
 - Option pour se déconnecter via un lien dans le tableau de bord.

2. Tableau de Bord

- **Accueil personnalisé :**
 - Affiche un message de bienvenue avec le nom et le prénom de l'utilisateur connecté.
- **Navigation :**
 - Liens pour accéder aux listes des hôtels, restaurants et salles.
 - Option pour se déconnecter.

3. Recherche et Affichage des Établissements

- **Recherche par nom :**
 - Fonctionnalité de recherche pour trouver des hôtels, restaurants ou salles par leur nom.
 - La recherche est dynamique et utilise des requêtes SQL avec LIKE pour filtrer les résultats.
- **Affichage des résultats :**
 - Les résultats sont affichés sous forme de cartes avec le nom et la description de chaque établissement.
 - Si aucun résultat n'est trouvé, un message approprié est affiché.

4. Réservation d'Établissements

- **Réservation d'hôtels, restaurants et salles :**
 - L'utilisateur peut réserver un établissement en cliquant sur un bouton "Réserver".
 - Un formulaire permet de sélectionner une date et une heure pour la réservation.
- **Validation des réservations :**
 - Les champs de date et d'heure sont obligatoires.
 - En cas de succès, la réservation est enregistrée dans la base de données avec un statut "confirmée".
 - En cas d'erreur, un message d'erreur est affiché.

5. Gestion des Réservations

- **Enregistrement en base de données :**
 - Les réservations sont stockées dans une table reservations avec des informations telles que l'ID de l'utilisateur, le type

d'établissement, l'ID de l'établissement, la date et l'heure de réservation, et le statut.

- **Redirection après réservation :**
 - Après une réservation réussie, l'utilisateur est redirigé vers le tableau de bord (`menu.php`).

6. Interface Utilisateur

- **Design moderne et responsive :**
 - Utilisation de CSS pour un design attrayant et adapté à tous les appareils.
 - Polices Google Fonts (`Poppins`) pour une meilleure lisibilité.
- **Expérience utilisateur intuitive :**
 - Boutons clairs et formulaires faciles à utiliser.
 - Messages d'erreurs et de succès bien visibles.

7. Sécurité

- **Protection contre les injections SQL :**
 - Utilisation de PDO et de requêtes préparées pour éviter les injections SQL.
- **Échappement des entrées utilisateur :**
 - Les entrées utilisateur sont échappées avec `htmlspecialchars` pour éviter les attaques XSS.
- **Vérification des sessions :**
 - Chaque page vérifie que l'utilisateur est connecté avant d'afficher du contenu.

8. Base de Données

- **Tables principales :**
 - `utilisateurs` : Stocke les informations des utilisateurs (nom, prénom, email, etc.).
 - `hotels, restaurants, salles` : stockent les informations des établissements (nom, description, etc.).
 - `reservations` : Stocke les réservations des utilisateurs avec les détails (date, heure, statut, etc.).
- **Relations :**
 - Les réservations sont liées aux utilisateurs et aux établissements via des clés étrangères.

9. Fonctionnalités supplémentaires

- **Retour au menu :**
 - Un bouton "Retour au menu" est disponible sur chaque page pour revenir au tableau de bord.
- **Messages d'erreur :**
 - Des messages d'erreur clairs sont affichés en cas de problème (ex : établissement non trouvé, champs manquants).

10. Extensibilité

- **Modularité :**
 - Le code est structuré de manière modulaire, ce qui permet d'ajouter facilement de nouveaux types d'établissements ou fonctionnalités.
- **Améliorations possibles :**
 - Ajout d'un système de gestion des utilisateurs (inscription, modification de profil).
 - Intégration d'un calendrier pour visualiser les réservations.
 - Notifications par email après une réservation.

2-La création de base de données:

La base de données de l'application gestion_reservation est structurée autour de six tables principales, chacune jouant un rôle spécifique dans la gestion des données de l'application:

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> administrateurs	★ Parcourir Structure Rechercher Insérer Vider Supprimer	1	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/> hotels	★ Parcourir Structure Rechercher Insérer Vider Supprimer	1	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> reservations	★ Parcourir Structure Rechercher Insérer Vider Supprimer	6	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> restaurants	★ Parcourir Structure Rechercher Insérer Vider Supprimer	1	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> salles	★ Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> utilisateurs	★ Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8mb4_general_ci	32,0 kio	-
6 tables	Somme	14	InnoDB	utf8mb4_general_ci	128,0 kio	0 o


```
SELECT * FROM `salles`
```

☐ Profilage
 [\[Éditer en ligne \]](#)
[\[Éditer \]](#)
[\[Expliquer SQL \]](#)
[\[Créer le code source PHP \]](#)
[\[Actualiser \]](#)

☐ Tout afficher
 Nombre de lignes : 25
 Filtrer les lignes:
 Trier par clé : Aucun(e)

Options supplémentaires

	id	nom	adresse	ville	code_postal	telephone	capacite	description	date_creation

```
SELECT * FROM `restaurants`
```

☐ Profilage
 [\[Éditer en ligne \]](#)
[\[Éditer \]](#)
[\[Expliquer SQL \]](#)
[\[Créer le code source PHP \]](#)
[\[Actualiser \]](#)

☐ Tout afficher
 Nombre de lignes : 25
 Filtrer les lignes:

Options supplémentaires

	id	nom	adresse	ville	code_postal	telephone	description	date_creation

```
SELECT * FROM `reservations` ORDER BY `utilisateur_id` DESC
```

☐ Profilage
 [\[Éditer en ligne \]](#)
[\[Éditer \]](#)
[\[Expliquer SQL \]](#)
[\[Créer le code source PHP \]](#)
[\[Actualiser \]](#)

☐ Tout afficher
 Nombre de lignes : 25
 Filtrer les lignes:
 Trier par clé : Aucun(e)

Options supplémentaires

	id	utilisateur_id	type_etablissement	etablissement_id	date_reservation	heure_reservation	statut	date_creation_res

```
SELECT * FROM `hotels`
```

☐ Profilage
 [\[Éditer en ligne \]](#)
[\[Éditer \]](#)
[\[Expliquer SQL \]](#)
[\[Créer le code source PHP \]](#)
[\[Actualiser \]](#)

☐ Tout afficher
 Nombre de lignes : 25
 Filtrer les lignes:

Options supplémentaires

	id	nom	adresse	ville	code_postal	telephone	description	date_creation

```
SELECT * FROM `salles`
```

☐ Profilage
 [\[Éditer en ligne \]](#)
[\[Éditer \]](#)
[\[Expliquer SQL \]](#)
[\[Créer le code source PHP \]](#)
[\[Actualiser \]](#)

☐ Tout afficher
 Nombre de lignes : 25
 Filtrer les lignes:
 Trier par clé : Aucun(e)

Options supplémentaires

	id	nom	adresse	ville	code_postal	telephone	capacite	description	date_creation

```
SELECT * FROM `administrateurs`
```

☐ Profilage
 [\[Éditer en ligne \]](#)
[\[Éditer \]](#)
[\[Expliquer SQL \]](#)
[\[Créer le code source PHP \]](#)
[\[Actualiser \]](#)

☐ Tout afficher
 Nombre de lignes : 25
 Filtrer les lignes:

Options supplémentaires

	id	nom	prenom	email	mot_de_passe

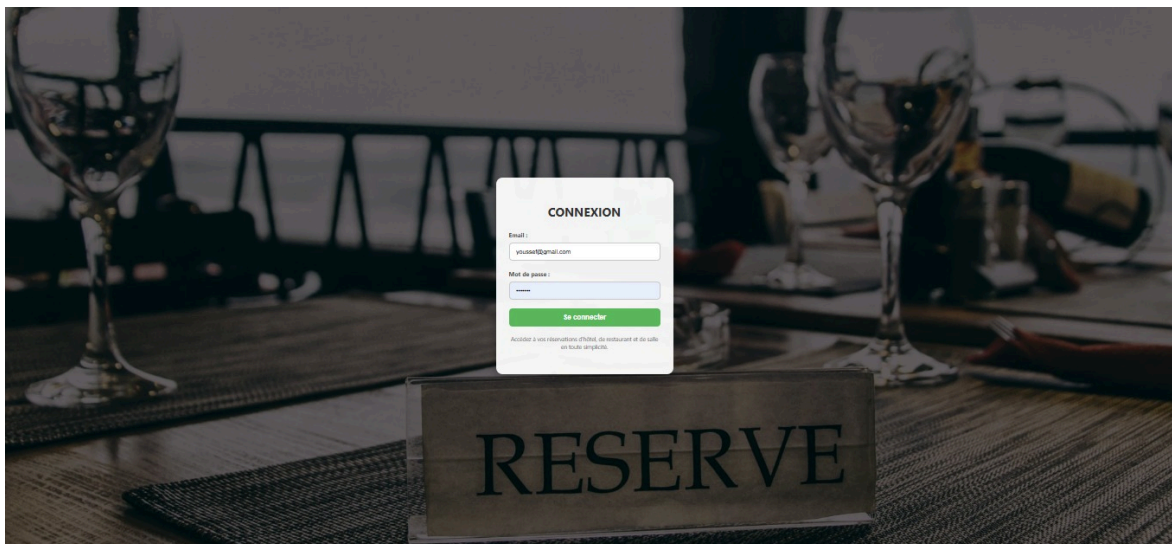
2-La connexion entre la base de données et php:

La connexion à la base de données est établie en PHP avec PDO. Les informations de connexion, comme l'hôte (localhost), le nom de la base (gestionreservations), l'utilisateur (root) et le mot de passe (vide), sont utilisées pour se connecter. En cas d'erreur, un message s'affiche. La connexion est configurée pour signaler les erreurs. Une fonction vérifie si un mot de passe est déjà haché, ce qui est utile pour la sécurité. Cette méthode assure une connexion simple et sécurisée à la base de données.

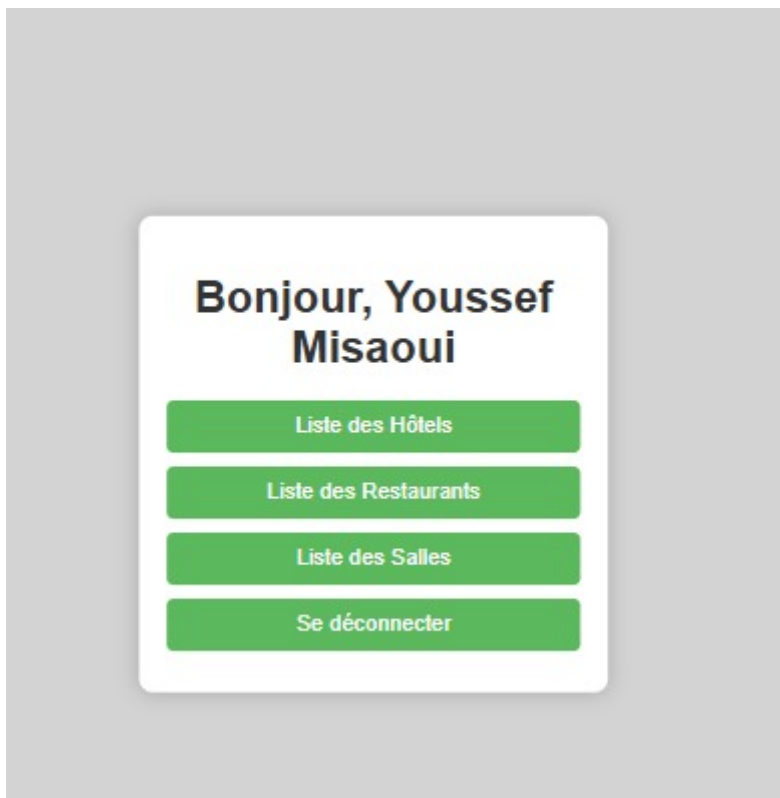
```
<?php
$host = 'localhost';
$db = 'gestionreservations';
$user = 'root';
$pass = '';

try {
    $pdo = new PDO( dsn: "mysql:host=$host;dbname=$db", $user, $pass);
    $pdo->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Erreur de connexion : " . $e->getMessage());
}

4 usages
function isPasswordHashed($password) {
    return (strlen($password) === 60 && (substr($password, offset: 0, length: 4) === '$2y$' || substr($password, offset: 0, length: 4) === '$2a$'));
}
?>
```



Menu utilisateur :



Menu d'administrateur:



Vous allez trouver ici , le menu des fonctionnalités pour admin et pour utilisateur, pour savoir les codes et l'exécution cliquez sur le github link : https://github.com/adamo349341/website_project/tree/master .

Problèmes et solutions

- La connexion et différenciation entre les données d'administrateur et d'utilisateur dans le login, aussi que le système n'arrive pas à trouver le mot de passe donné par l'utilisateur dans le database.

Solution: utiliser la fonction `isPasswordHashed` , cette dernière sert à hacher le mot de passe pour avoir la simplification de liaison par le système.

- Problème et difficulté de lier les deux tableaux d'utilisateurs et un tableau d'établissement , on a pensé de faire un tableau mais on a du mal à rester dans les informations de cet utilisateur.

Solution: Faire une condition en utilisant la fonction Trim qui nettoie tout ce qui est inutile.

- Pour le Graphe , on a du mal à appliquer notre logique qui est l'affichage de nombre de chaque établissement réservé et aussi l'implémentation a été compliquée.

Solution : Minimiser nos idées et utilisation un logiciel facile (Charte.js) et se concentrer à travailler seulement dans php

- La manipulation de xampp est un peu faible , chaque fois qu'on essaie d'ajouter quelques données dans notre base de données, l'application commence à bugger.

Solution : Minimiser les données et reformulation de la base de données.

Conclusion

Le développement de l'application de gestion des réservations pour hôtels, restaurants et salles a été une expérience enrichissante. L'application centralise les réservations, simplifie la gestion pour les administrateurs et améliore l'expérience client grâce à une interface intuitive et sécurisée. Malgré des défis techniques, comme la gestion des connexions et l'intégration de graphiques, des solutions ont été trouvées, notamment l'utilisation de PDO et Chart.js. Ce projet a renforcé nos compétences en développement web et en gestion de bases de données, tout en ouvrant des perspectives d'amélioration pour l'avenir.