

Rapport Projet Web

Fin de deuxième semestre

La thématique du site

Pour la création de notre site WEB il a bien sur fallu trouver un thème , nous avons tout d'abord pensé à faire un site ayant pour thème les jeux vidéos , cependant cette idée n'aurait pas été très originale et aurait été reprise de nombreuses fois. Notre seconde idée , idée gardée définitivement , se base sur le restaurant de la **Ch'tite Mamie** , le restaurant d'une mamie originaire du Nord préparant de super frites (bien sûr, ce restaurant n'existe pas réellement , il a été imaginé de toute pièce pour ce projet).

Concernant le choix des couleurs , nous avons premièrement décidé d'utiliser du rouge et du jaune rappelant les couleurs du nord et des restaurants anciens , voici le tableau des couleurs utilisées :

#e30a0a	Couleur principale	Fond menu footer, nav et footer
#ffd200	Couleur secondaire	Couleur police footer et nav
#8c0f0f	Contraste rouge	Contour menu footer, nav et footer
#ffd4b6	Couleur du fond	Fond du main
#f1aa78	Contraste couleur du fond	Contour du main
#8f6547	Contraste foncé	Contour sous-fenêtres
#615055	Couleurs contours	Contrastes
#ffffff	Couleurs de texte	Couleur texte footer

Ce site permettrait aux potentiels clients de prendre connaissance du menu, de réserver, d'en apprendre d'avantage concernant le magasin. Notre site se sépare en deux parties distinctes :

Coté Client/Non connecté :

Lorsque la personne est déconnectée ou connectée en tant que client, celle-ci n'a accès qu'aux pages accueil, menu, réservation, partenaires et connexion/profil selon si elle est connectée ou non.

Coté Admin

Lorsque la personne est connectée en tant qu'Admin, celle-ci a accès à l'accueil, à la page de modification du menu, à la liste des réservations et à son profil. Voici un tableau représentant si oui ou non, une page est accessible en fonction du type de compte :

Nom de la page	Description	Client	Admin
Accueil	L'accueil du site décrivant celui-ci et son histoire		
Menu	Le menu du restaurant ainsi que les ingrédients de chaque plats		
Reservation	La page pour réserver dans le restaurant de la Ch'tite Mamie		
Partenaires	La liste de nos partenaires et fournisseurs		
Profil	Le profil affichant les informations de la personne connectée		
Menu_admin	La page permettant de changer le menu		
Reservation_admin	La page affichant toutes les réservations du restaurant		

Le diagramme de classe d'UML

Nous avons, sur notre site, trois tables : reservation, profils et plats qui comme leurs noms l'indiquent, contiennent respectivement les : réservations faites sur le site, les profils des utilisateurs et les plats au menu du restaurant.

chtemamie_bd reservation
Id : int(11)
Nom : text
Prenom : text
Date : date
Heure : text
Nombre : int(11)
Event : text

chtemamie_bd profils
Id : int(11)
Nom : text
Prenom : text
Telephone : text
Email : text
MotDePasse : text
Admin : int(1)

chtemamie_bd plats
Id : int(11)
nom : varchar(64)
name : varchar(64)
recette : text
recipe : text
image : text
type : text

L'utilisation du PHP

La mise en forme :

Le PHP est utilisé sur chacune de nos pages, celui-ci sert par exemple à ne pas avoir à copier-coller la partie HTML gérant le header, le footer, ou le nav.

En effet , grâce à la fonction include('fichier.php'); , nous pouvons déplacer les parties répétitives dans des fichiers à part et les inclure dans la page au moment de l'affichage de celle-ci.

```
<?php
include "header.php";
?>
```

Il sert également à tester des valeurs, permettant ici d'afficher le bon menu de navigation en fonction de si la personne connectée utilise un compte admin ou non, pour cela on test la variable \$_SESSION["admin"] (réexpliqué plus loin ...)

```
?php
if(isset($_SESSION["admin"]) && $_SESSION["admin"]==true){
    echo<div class="conteneur-nav">;
    if($_SESSION["current_page"]=="accueil") echo<a href="../pages_php/accueil.php" class="lien-interne"><b><u>Accueil</u></b></a>;
    else echo<a href="../pages_php/accueil.php" class="lien-interne">Accueil</a>;
    if($_SESSION["current_page"]=="menuadmin") echo<a href="../pages_php/menu_admin.php" class="lien-interne"><b><u>Menu</u></b></a>;
    else echo<a href="../pages_php/menu_admin.php" class="lien-interne">Menu</a>;
    if($_SESSION["current_page"]=="reservadmin") echo<a href="../pages_php/reservations_admin.php" class="lien-interne"><b><u>Réservations</u></b></a>;
    else echo<a href="../pages_php/reservations_admin.php" class="lien-interne">Réservations</a>;
    echo<a href="logout.php" class="lien-interne" style="font-size: 2px;">Se déconnecter</a>;
}
```

Cela permet aussi de mettre en avant la page sur laquelle l'utilisateur se trouve actuellement :

Sur la page Accueil :

Accueil	Menu	Réservation	Partenaires	Se connecter
----------------	------	-------------	-------------	--------------

Sur la page Réservation

Accueil	Menu	Réservation	Partenaires	Se connecter
---------	------	--------------------	-------------	--------------

La Base de Données

Le PHP permet également une gestion automatique de la base de données. Pour notre cas on se connecte à la base de données dans la page "setting.php" que l'on ajoute dans chacune de nos pages PHP avec un include_once.

```
// Connexion à la base de données
$servername = 'localhost';
$username = 'root';
$password = 'root';
$dbname = 'chititemamie_bd';

$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
```

Les plats

Dans cette base de données, nous avons plusieurs tables dont une contenant les plats, par exemple, qui permet de pouvoir afficher dans la page menu, les différents plats de manière automatique.

Partie récupération des informations :

```
$reqPrep="SELECT * FROM plats WHERE type=?"; //La requête SQL SELECT
$req = $conn->prepare($reqPrep); //Préparation de la requête

$req->execute(['entree']); //Execution de la requête pour les entrées
$result_entree = $req->fetchAll(PDO::FETCH_ASSOC); //récupérer le résultat pour les entrées

$req->execute(['plat']); //Execution de la requête pour les plats
$result_plat = $req->fetchAll(PDO::FETCH_ASSOC); //récupérer le résultat pour les plats

$req->execute(['dessert']); //Execution de la requête pour les desserts
$result_dessert = $req->fetchAll(PDO::FETCH_ASSOC); //récupérer le résultat pour les desserts

$req->execute(['boisson']); //Execution de la requête pour les desserts
$result_boisson = $req->fetchAll(PDO::FETCH_ASSOC); //récupérer le résultat pour les desserts
```

```
$result_boisson = $req->fetchAll(PDO::FETCH_ASSOC); //récupérer le résultat pour les desserts
$req->execute(['boisson']); //Execution de la requête pour les desserts
```

Partie affichage :

```
<div id="liste-plats">
  <!--Debut de la liste des cartes des recettes -->
  <fieldset id="liste-recette-entree">
    <legend>Nos Entrées</legend> <!--Legende de chaque partie (ici les entrées) -->
    <?php
      for ($i=0;$i<count($result_entree);$i++) {
        echo '
        <div class="recette">
          <!-- Debut carte -->
          <div class="card-front">
            <!--Face avant de la carte -->
            <div>'.$result_entree[$i]['nom'].'</div>
            
          </div>

          <div class="card-back">
            <div class="lcarte">
              <!--Face arrière de la carte -->
              Ingrédients :<br />
              '.$result_entree[$i]['recette'].'
            </div>
          </div>
        </div> <!--Fin carte -->';
      }
    <?php
  </div>
</div>
```

Mais le PHP n'est pas seulement utile pour récupérer des informations. Il peut permettre d'ajouter, modifier et supprimer des informations dans la base de données, seulement à partir du site avec des formulaires (avec la méthode post), c'est-à-dire qu'il n'y a pas forcément besoin de modifier à la main la base de données. Si on reprend le cas du menu par exemple, notre site est doté d'une page menu administrateur, à partir de celle-ci l'admin, initialement connecté, peut réaliser les tâches citées plus tôt de manière simple et intuitive avec des formulaires.

Modifier :

```
if ($_POST['commande_plat'] == 'modifier_plat') {

  // On récupère les informations des plats
  $reqSQL = "SELECT * FROM plats";
  $req = $conn->prepare($reqSQL);
  $req->execute();

  $resultat = $req->fetchAll(PDO::FETCH_ASSOC);
  <?php

  <div class="formreserv">
    <form method="post" action="menu_admin.php"> <!-- formulaire pour savoir quel plat -->
      <fieldset>
        <legend>Modifier plat</legend>
        <select name="modifier_plat" required>
          <?php
            for ($i = 0; $i < count($resultat); $i++) {
              echo "<option value='".$resultat[$i]['Id']."'>".$resultat[$i]['nom']."'</option>";
            }
          <?php
        </select>
        <br><br>
        <input type="submit" value="modifier" id="reservation" name="choix_plat_modifier">
      </fieldset>
    </form>
  </div>
}
```

Modifier plat

Nom :

Nom en anglais :

Description :

Description en anglais :

Type de plat :

Image :

Supprimer :

```
if ($_POST['commande_plat'] == 'supprimer_plat') {
    // Récupération des informations par rapports aux plats
    $reqSQL = "SELECT * FROM plats";
    $req = $conn->prepare($reqSQL);
    $req->execute();

    $resultat = $req->fetchAll(PDO::FETCH_ASSOC);

    // Formulaire pour savoir quel aliment sera supprimer, en récupérant son Id
    echo "
    <div class='formreserv'>
        <form method='post' action='menu_admin.php'>
            <fieldset>
                <legend>Supprimer plat</legend>
                <select name='supprimer_plat' required>;

                for ($i = 0; $i < count($resultat); $i++) { // On met comme value l'id pour la récupérer
                    echo "<option value='".$resultat[$i]['Id']."'>".$resultat[$i]['nom']."</option>";
                }

                </select>
                <br><br>
                <input type='submit' value='valider' id='reservation' name='Soumettre_supprimer_plat'>
            </fieldset>
        </form>
    </div>;
```

Supprimer plat

Salade César

valider

Ajouter :

```
<div class='formreserv'>
    <form method='post' action='menu_admin.php' enctype='multipart/form-data'>
        <fieldset>
            <legend>Ajout plat</legend>
            <label >Nom :</label>
            <input type='text' name='nom_plat' placeholder='Entrez le nom du plat' required/>
            <br><br>
            <label >Nom en anglais :</label>
            <input type='text' name='name_plat' placeholder='Entrez le nom du plat en anglais' required/>
            <br><br>
            <label >Description :</label>
            <input type='text' name='ingredient_plat' placeholder='Entrez ses ingrédients (pensez à mettre br pour séparer)' required/>
            <br><br>
            <label >Description en anglais :</label>
            <input type='text' name='decipe_plat' placeholder='Entrez ses ingrédients en anglais (pensez à mettre br pour séparer)' required/>
            <br><br>
            <label >Type de plat :</label>
            <select name='type_plat' required>
                <option value='entree'>Entrees</option>
                <option value='plat'>Plat</option>
                <option value='dessert'>Dessert</option>
                <option value='boisson'>Boisson</option>
            </select>
            <br><br>
            <label >Image :</label>
            <input type='text' name='image' placeholder='Chemin du fichier' required/>
            <br><br>
            <input type='submit' value='valider' id='reservation' name='Soumettre_ajouter_plat'>
        </fieldset>
    </form>
```

Ajout plat

Nom : Entrez le nom du plat

Nom en anglais : Entrez le nom du plat en an

Description : Entrez ses ingrédients (pen

Description en anglais : Entrez ses ingrédients en a

Type de plat : Entree

Image : Chemin du fichier

valider

Les réservation

Bien sûr, les bases de données ne touchent pas que les plats. Nous en avons créé une pour sauvegarder et gérer les différentes réservations sur notre site :

[illegible]

Une fois entrées, et validées, les informations sont vérifiées grâce aux fonctions de sécurité ou aux pattern et required :

```
function valider_NomPrenom($NomPrenom){
    if(empty($NomPrenom) || strlen($NomPrenom) > 40 || !preg_match("/^[A-Za-z '-]+$/", $NomPrenom)) {
        return false;
    }
    return true;
}

function valider_Telephone($tel){
    if(empty($tel) || !preg_match("/^0[0-9]{9}/", $tel) || !preg_match("/[ ]*/", $tel)) {
        return false;
    }
    return true;
}

function valider_date($date_entree) {
    if (empty($date_entree) || strtotime($date_entree) > strtotime('-1 week')) {
        return false;
    }
    return true;
}
```


La réservation est ensuite envoyée à la base de données :

```
if(isset($_POST["nom"]) && isset($_POST["Réserver"])){
    $nom=nettoyer_donnees($_POST["nom"]);
    $prenom=nettoyer_donnees($_POST["prenom"]);
    $tel=nettoyer_donnees($_POST["tel"]);
    $date=nettoyer_donnees($_POST["date"]);
    $nbrparticipant=nettoyer_donnees($_POST["nbrparticipant"]);
    $time=nettoyer_donnees($_POST["time"]);
    if($_POST["nbrparticipant"]>15) $event=$_POST["event"];
    else $event="Table";

    if(valider_NomPrenom($prenom) && valider_NomPrenom($nom)){
        require_once("setting.php");
        $prereq=$conn->prepare("INSERT INTO 'reservation' ('Id', 'Nom', 'Prenom', 'Date', 'Heure', 'Nombre', 'Event') VALUES (NULL,?,?,?,?,?,?)");
        $prereq->execute([$nom, $prenom, $date, $time, $nbrparticipant, $event]);

        $req = $conn->prepare("SELECT * FROM reservation WHERE Nom=? AND Prenom=? AND Date=? AND Heure=? AND Nombre=? AND Event=?"); //Préparation de la requete
        $req->execute([$nom, $prenom, $date, $time, $nbrparticipant, $event]); //Execution de la requete pour les entrees
        $result = $req->fetch(PDO::FETCH_ASSOC); //récupérer le résultat pour les entrées
        $_SESSION['reserv'] = $result['Id'];
        header('location:confirmation.php');
    }
}
```

Une autre utilisation du php est de traiter des données, notamment les données d'un formulaire pour éviter les attaques XSS ou injection SQL. Ce traitement peut se faire de cette manière :

```
function nettoyer_donnees($donnees){
    $donnees = trim($donnees);
    $donnees = stripslashes($donnees);
    $donnees = htmlspecialchars($donnees);
    return $donnees;
}
```

```
if(isset($_POST["nom"]) && isset($_POST["Réserver"])){
    $nom=nettoyer_donnees($_POST["nom"]);
    $prenom=nettoyer_donnees($_POST["prenom"]);
    $tel=nettoyer_donnees($_POST["tel"]);
    $date=nettoyer_donnees($_POST["date"]);
    $nbrparticipant=nettoyer_donnees($_POST["nbrparticipant"]);
    $time=nettoyer_donnees($_POST["time"]);
    if($_POST["nbrparticipant"]>15) $event=$_POST["event"];
    else $event="Table";
```

Une fois ces réservation faites, nous les affichons sur la page “Reservation_admin.php” de la façon suivante :

```
<?php
try{
    require_once("setting.php");
    $reqPrep="SELECT * FROM reservation";//La requete SQL SELECT
    $req = $conn->prepare($reqPrep); //Préparation de la requete
    $req->execute(); //Execution de la requete pour les entrees
    $result = $req->fetchAll(PDO::FETCH_ASSOC);//récupérer le résultat pour les entrées
    $conn= NULL;
}
catch(Exception $e){
    die("Erreur : " . $e->getMessage());
}
if(isset($result)){
    echo'<div class="conteneur_collab">';
    for ($i=0; $i<count($result); $i++) {
        $date = new DateTimeImmutable($result[$i]["Date"]);
        echo'
        <fieldset id="liste-recette-entree" style="background-color: var(--marron-fond);width: 20%;">
        <legend style="background-color: var(--marron-fond);">Réservation n°' . ($i+1) . '</legend>
        ' . $result[$i]["Nom"] . ' ' . $result[$i]["Prenom"] . '<br>
        ' . $date->format('d-m-Y') . ' - ' . $result[$i]["Heure"] . '<br>
        ' . $result[$i]["Nombre"] . ' Personnes<br>
        ' . $result[$i]["Event"] . '
        </fieldset>';
    }
    echo'</div>';
}
?>
```

La première partie permet de se connecter à la base de données et d’y récupérer toutes les réservations sous forme de tableau, tableau que nous parcourons dans la seconde partie permettant de créer un champ rempli de “cartes” résumant chaque réservation.

Accueil	Menu	Réservations	Se déconnecter
Réservation n°1 NEVEU Theo 01-05-2024 - 23:30 64 Personnes Soirée	Réservation n°2 MARCANT Jeremie 29-04-2024 - 12:15 13 Personnes Table	Réservation n°3 EVIN Alexis 05-07-2024 - 13:00 4 Personnes Table	Réservation n°4 PAVY Adam 04-05-2024 - 20:00 8 Personnes Table
Réservation n°5 PAVY Adam 04-05-2024 - 13:00 8 Personnes Table	Réservation n°6 PAVY Adam 04-05-2024 - 13:00 8 Personnes Table	Réservation n°7 EVIN Alexis 11-05-2024 - 14:00 20 Personnes Mariage	Réservation n°8 MARCANT Jeremie 26-04-2024 - 12:00 24 Personnes Séminaire

Les comptes

Cette page est accessible seulement si l'utilisateur est admin, pour cela, il faut se connecter ! Voici donc la page de connexion :

Connexion

Email:

Mot de passe:

Se connecter

Pour se connecter, on demande à l'utilisateur d'entrer son adresse mail et son mot de passe, par la suite, on sélectionne dans la base de données un profil correspondant à l'Email entré :

```
if (isset($_POST['login'])) {
    $email = $_POST['email'];
    $mot_de_passe = $_POST['mot_de_passe'];

    //On récupère l'utilisateur depuis la base de données
    $stmt = $conn->prepare("SELECT * FROM profils WHERE Email = ?");
    $stmt->execute([$email]);
    $user = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

Grâce à ce profil, on teste le mot de passe entré, pour cela on le compare au mot de passe enregistré dans la base de données, si les deux correspondent, la personne est authentifiée.

```
if ($mot_de_passe == $user[0]['MotDePasse']) {
    //Authentification réussie : définir les variables de session
    $_SESSION['authentifie'] = true;
    $_SESSION['mail'] = $email;

    $testadminSQL = $conn->prepare("SELECT * FROM profils WHERE Email= ?");
    $testadminSQL->execute([$email]);
    $testadmin = $testadminSQL->fetch(PDO::FETCH_ASSOC);
    if($testadmin ['Admin'] == 1) $_SESSION['admin']=true;
    else $_SESSION['admin']=false;
    $_SESSION["TEST"]=$testadmin[0]['Admin'];
    header("location: account.php");
} else {
    //Authentification échouée : afficher un message d'erreur
    $_SESSION['MessageErreur'] = "Email ou mot de passe incorrect.";
}
```

Une fois authentifiée, on initialise les variables de session, avec les infos de l'utilisateur, notamment le caractère administrateur ou client. Pour cela, encore une fois, on cherche l'information dans la base de données, à la valeur correspondante à l'Email. Une fois authentifiée, on affiche le profil de la personne sur la page Profil nommée "account.php".

Une fois arrivé sur cette page, on regarde si l'utilisateur est un client, si c'est le cas, on crée des cookies contenant le nom, le prénom, et le numéro de téléphone de l'utilisateur :

```
if($_SESSION["admin"]==true)$typecompte="Admin";
else{
    setcookie("UserPrenom",$prenom,(time()+(3600*24*365)), '/', '', false, true);
    setcookie("UserNom",$nom,(time()+(3600*24*365)), '/', '', false, true);
    setcookie("UserTel",$telephone,(time()+(3600*24*365)), '/', '', false, true);
    $typecompte="Client";
}
```

Ces cookies serviront à pré-remplir les cases des formulaires avec des dites informations, permettant de gagner du temps, pour cela nous définissons la valeur à la valeur contenue dans le cookie, si celui-ci existe (onfocus sert ici à vider la case si celle-ci est sélectionnée pour écrire) :

```
<legend>Votre réservation : </legend>
<label for='nom'>Nom :</label>
<?php
if(isset($_COOKIE['UserNom']))
echo "<input type='text' name='nom' placeholder='Votre nom' required pattern='^[A-Za-z -]{$' maxlength='40' value='$_COOKIE['UserNom']' onfocus='\"this.value='\";\"/>";
else
echo "<input type='text' name='nom' placeholder='Votre nom' required pattern='^[A-Za-z -]{$' maxlength='40' />"; ?>
<img src='../images/Assiette.png' alt='LogoAssiette' id='Assiette'> <p style='display:inline;'></p>
<br><br />
<label for='prenom'>Prénom :</label>
<?php
if(isset($_COOKIE['UserPrenom']))
echo "<input type='text' name='prenom' placeholder='votre Prénom' required pattern='^[A-Za-z -]{$' maxlength='40' value='$_COOKIE['UserPrenom']' onfocus='\"this.value='\";\"/>";
else
echo "<input type='text' name='prenom' placeholder='votre Prénom' required pattern='^[A-Za-z -]{$' maxlength='40' />"; ?>
<br><br />
```

Répartition des tâches entre les membres

Jérémie	Adam	Théo	Alexis
<ul style="list-style-type: none">- Portage de l'html au php- Création et gestion de la base de données- Connexion à la page de données dans setting- Fonction de vérification de mdp, mail- Création de la page menu affichant les plats contenus dans la base de données- Création de la page de modification servant à modifier le menu	<ul style="list-style-type: none">- Création de la page réservation adaptée au nombre de personnes- Traitement et envoi des informations de la page réservation.	<ul style="list-style-type: none">- Création du menu nav en fonction du type d'utilisateur- Suppression des répétitions grâce aux include- Création de la page affichant toutes les réservations- Aide création page connexion/inscription- Création de cookies	<ul style="list-style-type: none">- Création de la page de connexion et inscription/connexion-Création de la page profil affichant les information de l'utilisateur