DOCUMENTATION

Les fonctions de la classe Facture

CONSTRUCTEUR

```php
public function __construct($id = 0, $dateFacture = '', $idCong = 0, $idHotel
= 0, $idAct = 0, $idSession = 0) {
        $this->id = $id;
        $this->dateFacture = $dateFacture;
        $this->idCong = $idCong;
        $this->idHotel = $idHotel;
        $this->idAct = $idAct;
        $this->idSession = $idSession;
        $this->pdo = connexionPDO();
    }
```

Constructeur (`__construct`) : Le constructeur est une méthode spéciale dans une classe PHP qui est automatiquement appelée lorsqu'un nouvel objet est créé à partir de cette classe. Dans ce cas, le constructeur prend plusieurs paramètres (id, dateFacture, idCong, idHotel, idAct, idSession) et initialise les propriétés de la classe avec ces valeurs. Il utilise également une instance de PDO (PHP Data Objects) stockée dans la propriété **`$pdo`** pour la connexion à la base de données.

GETTERS

```php
public function getId() {
        return $this->id;
    }

    public function getDateFacture() {
        return $this->dateFacture;
    }

    public function getIdCong() {
        return $this->idCong;
    }

    public function getIdHotel() {
        return $this->idHotel;
    }

    public function getIdAct() {
        return $this->idAct;
    }

    public function getIdSession() {
        return $this->idSession;
```

```
    }
```

Méthodes d'accès (`getId`, `getDateFacture`, etc.) : Ces méthodes, souvent appelées "getters", sont des fonctions qui permettent d'accéder aux valeurs des propriétés privées de la classe. Par exemple, la méthode `getId` renvoie la valeur de la propriété `id`.

SETTERS

```php
public function setId($id): void {
        $this->id = $id;
    }

    public function setDateFacture($dateFacture): void {
        $this->dateFacture = $dateFacture;
    }

    public function setIdCong($idCong): void {
        $this->idCong = $idCong;
    }

    public function setIdHotel($idHotel): void {
        $this->idHotel = $idHotel;
    }

    public function setIdAct($idAct): void {
        $this->idAct = $idAct;
    }

    public function setIdSession($idSession): void {
        $this->idSession = $idSession;
    }
```

Méthodes de modification (`setId`, `setDateFacture`, etc.) : Les méthodes de modification, ou "setters", sont utilisées pour définir les valeurs des propriétés privées de la classe. Par exemple, la méthode `setId` permet de définir la valeur de la propriété `id`.

```php
public function getFacture() {
        $sql = "SELECT * FROM facture f
        JOIN congressistes c ON c.id = f.idCong
        JOIN PAIEMENT p ON p.idCong = c.id";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_OBJ);
        return $result;
    }
```

Cette fonction permet d’avoir les informations de la facture avec 2 jointures.

```php
public function payFacture($idCong) {
        $sql = "UPDATE paiement SET paye = 1 WHERE idCong = ?";
        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(1, $idCong);
        $stmt->execute();
     }
```

Vérifier si le congressiste a déjà payé ou pas.

```php
public function addFacture() {
        $sql = "INSERT INTO facture (dateFacture, idCong, idHotel, idAct,
idSession) VALUES (?, ?, ?, ?, ?)";
        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(1, $this->dateFacture);
        $stmt->bindValue(2, $this->idCong);
        $stmt->bindValue(3, $this->idHotel);
        $stmt->bindValue(4, $this->idAct);
        $stmt->bindValue(5, $this->idSession);
        $stmt->execute();
    }
```

Ajouter une facture lors de la création avec 5 colonne et paramètres.

```php
public function existCongFacturePaye($idCong) {
        $sql = "SELECT *
         FROM congressistes c
            JOIN paiement p ON p.idCong = c.id
         WHERE idCong = ? AND paye = 1";
        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(1, $idCong);
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_OBJ);
        return $result;
    }
    public function existCongFactureNonPaye($idCong) {
        $sql = "SELECT *
         FROM congressistes c
            JOIN paiement p ON p.idCong = c.id
         WHERE idCong = ? AND paye = 0";
        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(1, $idCong);
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_OBJ);
        return $result;
    }
```

Les deus fonctions vérifient si le congressiste choisi (passer en paramètre $idCong) à déjà payé ou pas.

```php
public function congressisteHasInvoice($idCong) {
        $sql = "SELECT COUNT(*) FROM facture WHERE idCong = ?";
        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(1, $idCong);
        $stmt->execute();
        $count = $stmt->fetchColumn();

        // Retourner true si le count est supérieur à 0 (le congressiste a
déjà une facture), sinon false
        return $count > 0;
    }
    public function congressisteNonHasInvoice($idCong) {
        $sql = "SELECT COUNT(*) FROM facture WHERE idCong = ?";
        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(1, $idCong);
        $stmt->execute();
        $count = $stmt->fetchColumn();

        // Retourner true si le count est supérieur à 0 (le congressiste a
déjà une facture), sinon false
        return $count == 0;
    }
```

Les deux fonctions vérifient si le congressiste à déjà une facture ou pas encore. COUNT(*) c'est pour compter id de ce congressiste choisi et on retourne la variable $count sil est plus de ou égale 0.

```php
public function getFactureInfoById($id) {
        $sql = "SELECT *
                FROM facture f
                JOIN congressistes c ON c.id = f.idCong
                JOIN sessions s ON s.id = f.idSession
                JOIN activites a ON a.id = f.idAct
                JOIN hotels h ON h.id = f.idHotel
                WHERE c.id = ? ";
        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(1, $id);
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_OBJ);
        return $result;
    }
```

Avoir toutes les informations de $id(congressiste) choisi nécessaires en faisant 4 jointures.

```php
public function getFacturePaye() {
        $sql = "SELECT *
        FROM paiement p
        JOIN congressistes c ON c.id = p.idCong
        JOIN facture f ON f.idCong = c.id
        WHERE paye = 1";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_OBJ);
        return $result;
    }
    public function getFactureNonPaye() {
        $sql = "SELECT * FROM paiement p
        JOIN congressistes c ON c.id = p.idCong
        JOIN facture f ON f.idCong = c.id
        WHERE paye = 0";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_OBJ);
        return $result;
    }
```

Les deux fonctions trient que les factures payées ou pas payées.

```php
    public function getMontantHotelById($congressisteId) {
        $sql = "SELECT prix as montant_hotel
                FROM facture f
                JOIN hotels h ON h.id = f.idHotel
                WHERE f.idCong = ?";

        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(1, $congressisteId);
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_OBJ);

        return $result;
    }

    public function getMontantSessionById($congressisteId) {
        $sql = "SELECT prix as montant_session
                FROM facture f
                JOIN sessions s ON s.id = f.idSession
                WHERE f.idCong = ?";

        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(1, $congressisteId);
```

```
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_OBJ);


        return $result;

    }

    public function getMontantActiviteById($congressisteId) {
        $sql = "SELECT prix as montant_activite
                FROM facture f
                JOIN activites a ON a.id = f.idAct
                WHERE f.idCong = ?";

        $stmt = $this->pdo->prepare($sql);
        $stmt->bindValue(1, $congressisteId);
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_OBJ);


        return $result;

    }
```

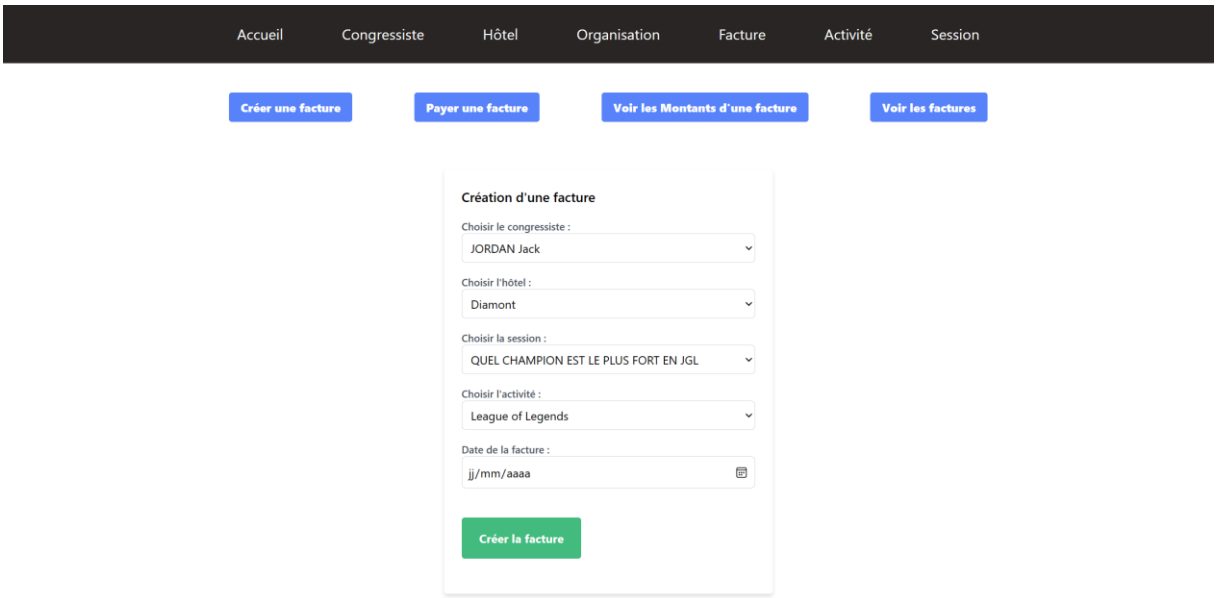
Les 3 fonctions permettent d'afficher les montants de l'hôtel, l'activité et la session par id choisi de congressite.

```
public function getSession () {
        $sql = "SELECT * FROM sessions";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_OBJ);

        return $result;
    }
    public function getActivite () {
        $sql = "SELECT * FROM activites";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_OBJ);

        return $result;
    }
```

Les deux fonctions permet de récupérer les informations trouvées sur la table de session et activité.

La vue de la facture

Vue principale :



Affichage des 4 boutons :qui contrôlent la vue des différentes 4 parties


```
<form method="POST">
                <button class="bg-blue-500 hover:bg-blue-700 text-white font-
bold py-2 px-4 rounded mt-10"
                    type="submit" name="action" value="creerFacture">
                    <b>Créer une facture</b>
                </button>
                <button class="bg-blue-500 hover:bg-blue-700 text-white font-
bold py-2 px-4 rounded ml-20 mt-10"
                    type="submit" name="action" value="payerFacture">
                    <b>Payer une facture</b>
                </button>
                <button class="bg-blue-500 hover:bg-blue-700 text-white font-
bold py-2 px-4 rounded ml-20 mt-10"
                    type="submit" name="action" value="voirMontantFacture">
                    <b>Voir les Montants d'une facture</b>
                </button>
```

```
                <button class="bg-blue-500 hover:bg-blue-700 text-white font-
bold py-2 px-4 rounded ml-20 mt-10"
                    type="submit" name="action" value="tousFactures">
                    <b>Voir les factures</b>
                </button>
            </form>
```

Faire une balise HTML de <form> pour mettre 4 boutons principales qui gèrent la vue de facture :

- **CREER UNE FACTURE**
- **PAYER UNE FACTURE**
- **VOIR LES INFORMATIONS D'UNE FACTURE**
- **VOIR LES FACTURES**

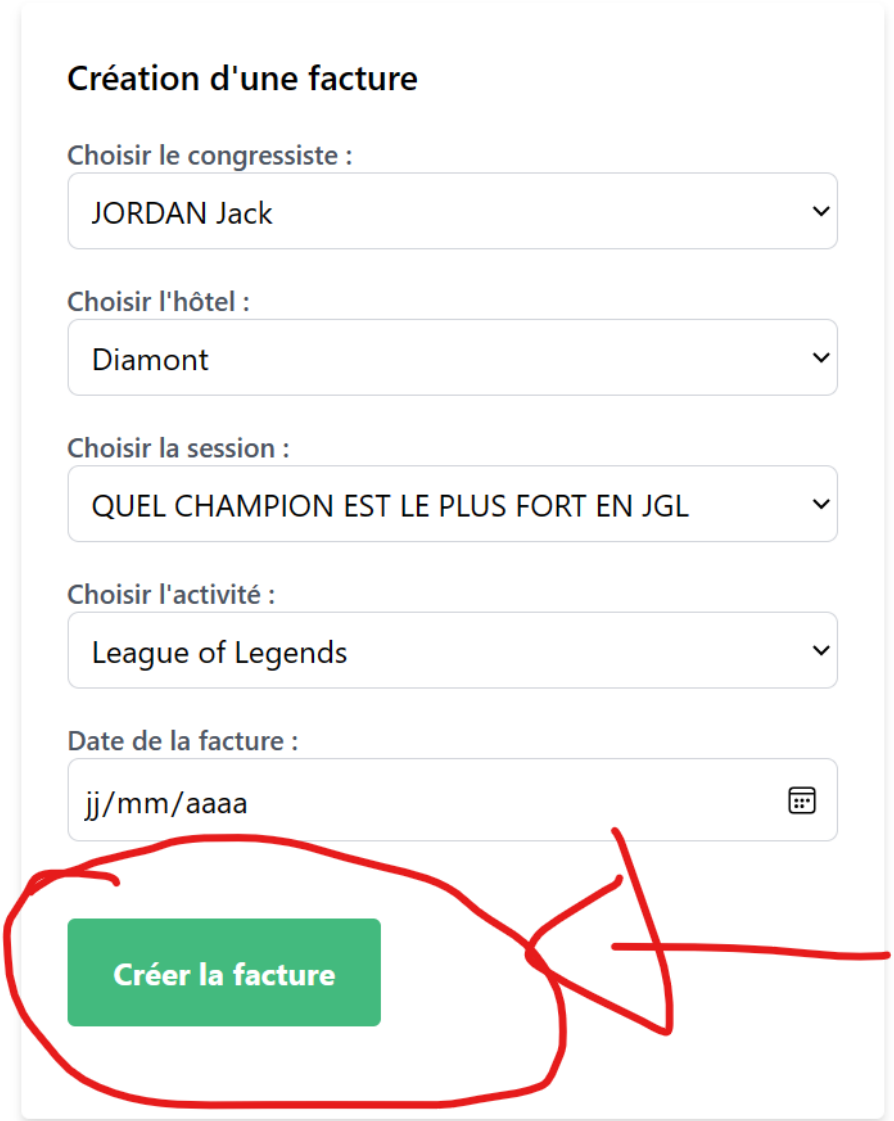
```
if (isset($_POST['action'])) {
    $action = $_POST['action'];

    // Handle different actions
    switch ($action) {

        case 'creerFacture':
            // Handle the action to create facture
            afficherFormulaireCreationFacture();
            break;
        case 'payerFacture':
            // Handle the action to pay facture
            afficherFormulairePayerFacture();

            break;
        case 'voirMontantFacture':
            // Handle the action to view facture information
            ?>
            <div class="mb-8">

                <form id="FactureForme" action="./?action=facture"
method="POST" class="max-w-md mx-auto bg-white p-6 rounded shadow-md">
                    <h2 class="text-lg font-semibold mb-4">Informations de la
facture</h2>

                    <div class="mb-4">
                        <label for="congressiste" class="block text-sm font-
medium text-gray-600">Choisir le congressiste :</label>
```

```
                        <select name="congressistee" id="congressisteSelect"
class="congressiste-select block w-full p-2 border border-gray-300 rounded-
md">
                            <?php
                            $congret = new Congressiste();
                            $congressistes = $congret->getCong();

                            foreach ($congressistes as $unCongressiste) {
                                echo "<option value='" . $unCongressiste->id .
"' >" . $unCongressiste->nom . " " . $unCongressiste->prenom . "</option>";
                            }
                            ?>
                        </select>
                    </div>

                    <div class="mb-6">
                        <button class="bg-blue-500 hover:bg-blue-700 text-
white font-bold py-2 px-4 rounded" type="submit" name="valider">
                            Valider
                        </button>
                    </div>
                </form>
            </div>

            <?php



            break;


        case 'tousFactures':
            // Handle the action to view all factures
            $facture = new Facture();
            afficherFormulaireTriFactures();
            afficherFactures($facture->getFacture(), "Tous les factures:");


            break;



    }
}
```

Créer une condition if pour afficher les différents cas de la boucle switch quand on clique de l'un de ces boutons avec (isset($_POST['action'])).

Pour le cas de creerFacture, payerFacture et tousFactures , il suffit d'appeler les fonctions que j'ai crée à la fin de page vueFacture.php pour avoir un code plus visible et facile à lire.

Créer une facture :



```php
// CREER UNE FACTURE
            if (isset($_POST['creerFacture'])) {
                $idCongressiste = $_POST['congressistes'];
                $facture = new Facture();
                // Vérifier si le congressiste a déjà payé et a une facture
                $facturePaye = $facture-
>existCongFacturePaye($idCongressiste);
                $factureExist = $facture-
>congressisteHasInvoice($idCongressiste);
                $factureNonExist = $facture-
>congressisteNonHasInvoice($idCongressiste);
                $factureNonPaye = $facture-
>existCongFactureNonPaye($idCongressiste);

                if ($facturePaye && $factureExist) {
                    // Le congressiste a déjà payé et a une facture
                    echo "<div class='max-w-md mx-auto bg-yellow-100 border-l-
4 border-yellow-500 text-yellow-700 p-4 rounded shadow-md'>";
                    echo "    <div class='flex items-center'>";
                    echo "        <div class='mr-2 text-3xl'>&#9888;</div>";
// Symbole d'avertissement
                    echo "        <h2 class='text-xl font-semibold mb-4'>Le
congressiste a déjà payé et a une facture existante.</h2>";
                    echo "    </div>";
                    echo "</div>";

                }
                if($facturePaye && $factureNonExist){
                    // Le congressiste a déjà payé mais n'a pas de facture, on
peut créer une nouvelle facture
                    $facture = new Facture();
                    $facture->setDateFacture($_POST['dateFacture']);
                    $facture->setIdCong($_POST['congressistes']);
                    $facture->setIdHotel($_POST['hotel']);
                    $facture->setIdAct($_POST['activite']);
                    $facture->setIdSession($_POST['session']);
                    $facturee=$facture->addFacture();


                        // La facture a été créée avec succès
                        echo "<div class='max-w-md mx-auto bg-green-100
border-l-4 border-green-500 text-green-700 p-4 rounded shadow-md'>";
                        echo "    <div class='flex items-center'>";
                        echo "        <div class='mr-2 text-
3xl'>&#10004;</div>"; // Symbole de check mark
                        echo "        <h2 class='text-xl font-semibold mb-
4'>La facture a été créée avec succès</h2>";
                        echo "    </div>";
                        echo "</div>";
```

```
                }
                if($factureNonPaye && $factureNonExist) {
                    $facture->setDateFacture($_POST['dateFacture']);
                    $facture->setIdCong($_POST['congressistes']);
                    $facture->setIdHotel($_POST['hotel']);
                    $facture->setIdAct($_POST['activite']);
                    $facture->setIdSession($_POST['session']);
                    $factureee=$facture->addFacture();
                    //le congressiste n'a pas payé, donc il faut payer d'abord
                    echo "<div class='max-w-md mx-auto bg-green-100 border-l-4
border-green-500 text-green-700 p-4 rounded shadow-md'>";
                    echo "    <div class='flex items-center'>";
                    echo "        <div class='mr-2 text-3xl'>&#10004;</div>";
// Symbole de check mark
                    echo "        <h2 class='text-xl font-semibold mb-4'>La
facture a été crée avec succées, maintenant il faut passer au paiement!</h2>";
                    echo "    </div>";
                    echo "</div>";
                }
                if($factureNonPaye && $factureExist){
                    //le congressiste n'a pas payé, donc il faut payer d'abord
                    echo "<div class='max-w-md mx-auto bg-yellow-100 border-l-
4 border-yellow-500 text-yellow-700 p-4 rounded shadow-md'>";
                    echo "    <div class='flex items-center'>";
                    echo "        <div class='mr-2 text-3xl'>&#9888;</div>";
// Symbole d'avertissement
                    echo "        <h2 class='text-xl font-semibold mb-4'>Le
congressiste a déjà une facture existante, mais il n'a pas encore payé!</h2>";
                    echo "    </div>";
                    echo "</div>";
                }

            }
```

Une condition si on clique sur le bouton de créer une facture, ca va afficher des messages de succès de paiement si la fonction addFacture() a bien été exécuté et des autres conditions pour contrôler toutes les situations possibles. Ex : le congressiste a déjà une facture…….

Récupération des données du formulaire : $idCongressiste = $_POST['congressistes'];

Instanciation de la classe Facture et Vérifications : $facture = new Facture();

Vérification si le congressiste a déjà payé et a une facture existante :

$facturePaye = $facture->existCongFacturePaye($idCongressiste);

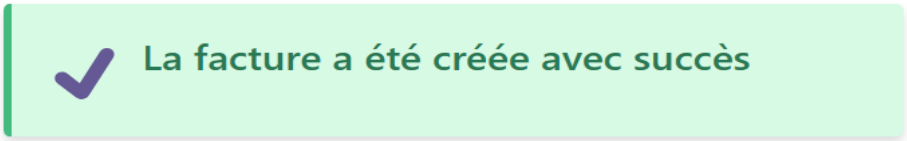
$factureExist = $facture->congressisteHasInvoice($idCongressiste);

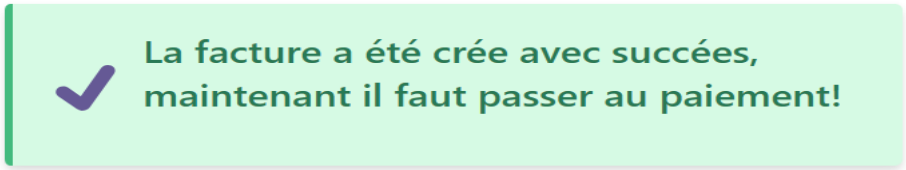
if ($facturePaye && $factureExist) {

// Le congressiste a déjà payé et a une facture existante

// Affichage d'un message d'avertissement

} et ainsi de suite ….






```
$facture->setDateFacture($_POST['dateFacture']);
                    $facture->setIdCong($_POST['congressistes']);
                    $facture->setIdHotel($_POST['hotel']);
                    $facture->setIdAct($_POST['activite']);
                    $facture->setIdSession($_POST['session']);
                    $facturee=$facture->addFacture();
```

Insérer les informations de facture sur la base de données si la condition est juste

	id ▲ 1	dateFacture	idCong	idHotel	idAct	idSession
Éditer Copier Supprimer	1	2024-01-03	1	3	2	3
Éditer Copier Supprimer	2	2024-01-09	2	2	2	2
Éditer Copier Supprimer	3	2024-01-18	3	4	2	3

```
if(isset($_POST['afficherPrix'])){
                $facture = new Facture();
                $selectedCongressisteId = $_POST['congressistes'];
                $factureExist = $facture-
>congressisteHasInvoice($selectedCongressisteId);
                $factureNonExist = $facture-
>congressisteNonHasInvoice($selectedCongressisteId);
                // Vérifier si le congressiste a déjà une facture
                if ($factureExist) {
                    // Récupérer les montants
                    $montantHotel = $facture-
>getMontantHotelById($selectedCongressisteId);
                    $montantSession = $facture-
>getMontantSessionById($selectedCongressisteId);
                    $montantActivite = $facture-
>getMontantActiviteById($selectedCongressisteId);
                    ?>

                    <div class="max-w-md mx-auto bg-white p-8 rounded shadow-
md" id="voir">
                        <h2 class="text-xl font-semibold mb-4">Les
informations de la facture</h2>

                        <div class="mb-4">
                            <label for="montantHotel" class="block text-sm
font-medium text-gray-600">Montant pour chaque hôtel :</label>
                            <p class="mt-1 p-2 border border-gray-300 rounded-
md w-full">
                                <?php echo $montantHotel->montant_hotel; ?>
                            </p>
                        </div>

                        <div class="mb-4">
                            <label for="montantSession" class="block text-sm
font-medium text-gray-600">Montant pour chaque session :</label>
                            <p class="mt-1 p-2 border border-gray-300 rounded-
md w-full">
                                <?php echo $montantSession->montant_session;
?>
                            </p>
                        </div>

                        <div class="mb-4">
                            <label for="montantActivite" class="block text-sm
font-medium text-gray-600">Montant pour chaque activité :</label>
                            <p class="mt-1 p-2 border border-gray-300 rounded-
md w-full">
```

```
                                <?php echo $montantActivite->montant_activite;
?>
                            </p>
                        </div>

                        <div class="mb-4">
                            <label for="montantTotal" class="block text-sm
font-medium text-gray-600">Montant total :</label>
                            <p class="mt-1 p-2 border border-gray-300 rounded-
md w-full">
                                <?php echo $montantHotel->montant_hotel +
$montantSession->montant_session + $montantActivite->montant_activite; ?>
                            </p>
                        </div>
                    </div><br><br>

                <?php
                }
                if($factureNonExist) {
                    // Le congressiste doit d'abord créer une facture
                    echo "<div class='max-w-md mx-auto bg-red-100 border-l-4
border-red-500 text-red-700 p-4 rounded shadow-md'>";
                    echo "    <div class='flex items-center'>";
                    echo "        <div class='mr-2 text-3xl'>&#10007;</div>";
// Symbole de croix
                    echo "        <h2 class='text-xl font-semibold mb-4'>Vous
devez d'abord créer une facture.</h2>";
                    echo "    </div>";
                    echo "</div>";
                }
            }
```

La condition si on clique sur Afficher les prix, on fait appel à la classe facture avec $facture= new Facture() pour récupérer les fonctions qu'on veut utiliser et pour afficher les montants.

Le code vérifie si le congressiste a déjà créé une facture. Si c'est le cas, il affiche les informations de facturation (montants pour chaque hôtel, session, activité, et le montant total). Si le congressiste n'a pas encore créé de facture, un message d'avertissement est affiché pour l'inciter à créer une facture d'abord.

Payer une facture

⚠ Le congressiste a déjà payé et a une facture existante.

⚠ Vous n'avez pas encore créer la facture pour payer

✓ La facture a été payée avec succès

```
if (isset($_POST['payerFacture'])) {
                $facture = new Facture();
                $selectedCongressisteId = $_POST['congressistes'];
                $facturePaye = $facture-
>existCongFacturePaye($selectedCongressisteId);
                $factureNonPaye = $facture-
>existCongFactureNonPaye($selectedCongressisteId);
                $factureNonExist = $facture-
>congressisteNonHasInvoice($selectedCongressisteId);
                $factureExist = $facture-
>congressisteHasInvoice($selectedCongressisteId);
                // Vérifiez si la facture n'est pas payée
                if ($factureNonPaye && $factureExist) {

                        $processPaye = $facture-
>payFacture($selectedCongressisteId);

                        // Vérifiez si le paiement a réussi
```

```
                            echo "<div class='max-w-md mx-auto bg-green-100
border-l-4 border-green-500 text-green-700 p-4 rounded shadow-md'>";
                            echo "    <div class='flex items-center'>";
                            echo "        <div class='mr-2 text-
3xl'>&#10004;</div>"; // Symbole de check mark
                            echo "        <h2 class='text-xl font-semibold mb-
4'>La facture a été payée avec succès</h2>";
                            echo "    </div>";
                            echo "</div>";


                }
                if($factureNonPaye && $factureNonExist){


                            // Vérifiez si le paiement a réussi

                            echo "<div class='max-w-md mx-auto bg-yellow-100
border-l-4 border-yellow-500 text-yellow-700 p-4 rounded shadow-md'>";
                            echo "    <div class='flex items-center'>";
                            echo "        <div class='mr-2 text-
3xl'>&#9888;</div>"; // Symbole d'avertissement
                            echo "        <h2 class='text-xl font-semibold mb-
4'>Vous n'avez pas encore créer la facture pour payer </h2>";
                            echo "    </div>";
                            echo "</div>";


                        }
                if($facturePaye && $factureNonExist){
                    //afficher un message que le congressiste à deja payé et
de creer une facture aprés
                    echo "<div class='max-w-md mx-auto bg-yellow-100 border-l-
4 border-yellow-500 text-yellow-700 p-4 rounded shadow-md'>";
                    echo "    <div class='flex items-center'>";
                    echo "        <div class='mr-2 text-3xl'>&#9888;</div>";
// Symbole d'avertissement
                    echo "        <h2 class='text-xl font-semibold mb-4'>Le
congressiste a déjà payé mais il n'a pas encore creer une facture.</h2>";
                    echo "    </div>";
                    echo "</div>";
                }
                if($facturePaye && $factureExist) {
                    echo "<div class='max-w-md mx-auto bg-yellow-100 border-l-
4 border-yellow-500 text-yellow-700 p-4 rounded shadow-md'>";
                    echo "    <div class='flex items-center'>";
                    echo "        <div class='mr-2 text-3xl'>&#9888;</div>";
// Symbole d'avertissement
```

```
                    echo "        <h2 class='text-xl font-semibold mb-4'>Le
congressiste a déjà payé et a une facture existante.</h2>";
                    echo "    </div>";
                }
            }
```

La condition si on clique sur le bouton de payer La facture, on fait un update de colonne paye avec la fonction payFacture() qui met 1 à la place de 0, mais sauf si le congressiste n'a pas encore payé sinon , on affiche que le congressiste a déjà payé.

```
function afficherFormulaireTriFactures()
{
    echo '<div class="mt-8 ml-10">';
    echo '    <h2 class="text-2xl font-semibold mb-4">Trier les factures
:</h2>';
    echo '    <form id="triForm" action="./?action=facture" method="POST">';
    echo '        <div class="space-x-4">';
    echo '            <button type="submit" name="triTous" class="btn-tri1">';
    echo '                Tous Les Factures';
    echo '            </button>';
    echo '            <button type="submit" name="triPaye" class="btn-tri bg-
green-300 hover:bg-green-400">';
    echo '                Factures Payées';
    echo '            </button>';
    echo '            <button type="submit" name="triNonPaye" class="btn-tri
bg-red-400 hover:bg-red-500">';
    echo '                Factures Non Payées';
    echo '            </button>';
    echo '        </div>';
    echo '    </form>';
    echo '</div>';
    echo '<br><br>';
}
```

Fonction d'Affichage du Formulaire de Tri des Factures :

La fonction `afficherFormulaireTriFactures` génère et affiche un formulaire permettant de trier les factures. Voici une explication détaillée du code :

La fonction commence par afficher un conteneur `<div>` avec une classe `mt-8 ml-10` pour définir la marge en haut et à gauche du formulaire.

Ensuite, un titre `<h2>` est affiché avec les classes `text-2xl font-semibold mb-4`, définissant la taille du texte, la graisse de la police et la marge inférieure.

À l'intérieur du formulaire (`<form>`), l'attribut `action` est défini sur `./?action=facture`, ce qui indique que le formulaire sera soumis à l'action "facture" dans le script PHP.

À l'intérieur du formulaire, un ensemble de boutons de tri est affiché. Chaque bouton est un `<button>` avec un attribut `type="submit"` pour permettre la soumission du formulaire lorsqu'il est cliqué. Les boutons portent les noms `triTous`, `triPaye`, et `triNonPaye` respectivement.

Chaque bouton a des classes CSS spécifiques (`btn-tri1`, `btn-tri`, `bg-green-300`, `hover:bg-green-400`, etc.) qui définissent le style visuel du bouton, y compris les couleurs de fond, les effets de survol, etc.

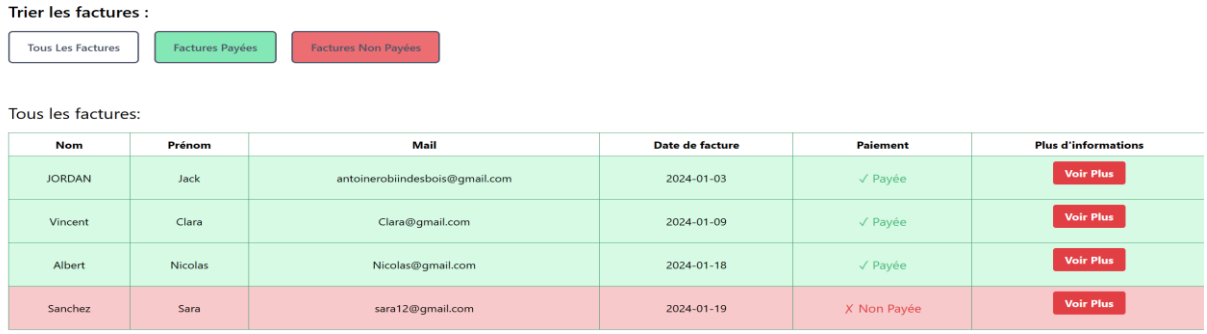
Les boutons sont regroupés dans une `<div>` avec la classe `space-x-4` pour gérer l'espacement horizontal entre les boutons.

Enfin, la fonction ajoute une balise de fermeture `</div>` pour fermer le conteneur principal du formulaire, et `<br>` est utilisé pour ajouter quelques sauts de ligne après le formulaire.

En résumé, cette fonction génère un formulaire de tri avec trois boutons correspondant aux options "Tous Les Factures", "Factures Payées", et "Factures Non Payées". Les boutons permettent de filtrer les factures en fonction de leur état de paiement.

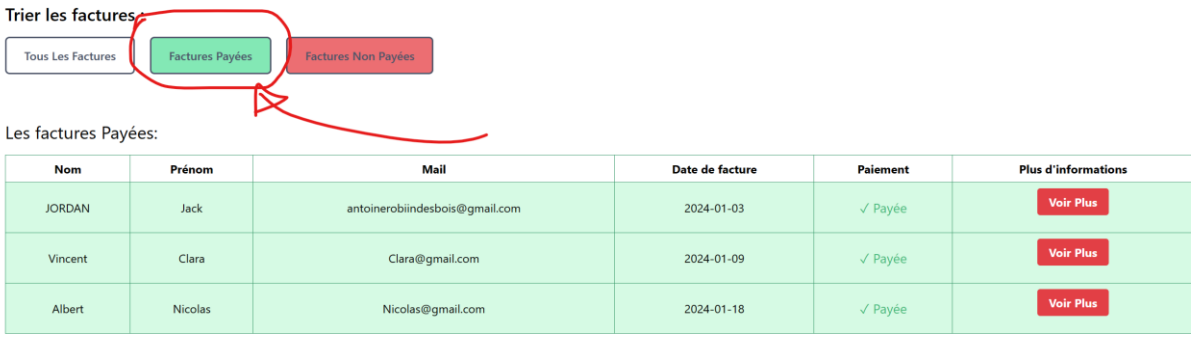
Trier les factures

```
elseif (isset($_POST['triTous'])) {
                afficherFormulaireTriFactures();
                afficherFactures($facture->getFacture(), "Tous les
factures:");
            }
```



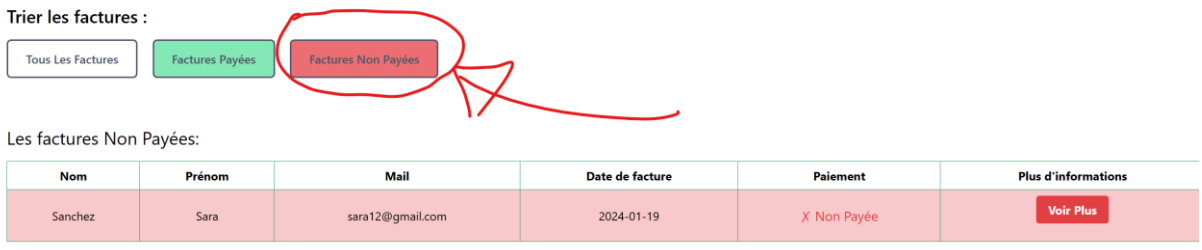
La fonction afficherFactures() est appelée avec le tableau de toutes les factures ($facture->getFacture()) et un titre "Tous les factures:".

```php
if (isset($_POST['triPaye'])) {
                afficherFormulaireTriFactures();
                afficherFactures($facture->getFacturePaye(), "Les factures
Payées:");
            }
```



La fonction afficherFactures() est appelée avec le tableau de factures payées ($facture->getFacturePaye()) et un titre "Les factures Payées:".

```php
elseif (isset($_POST['triNonPaye'])) {
                afficherFormulaireTriFactures();
                afficherFactures($facture->getFactureNonPaye(), "Les factures
Non Payées:");
            }
```



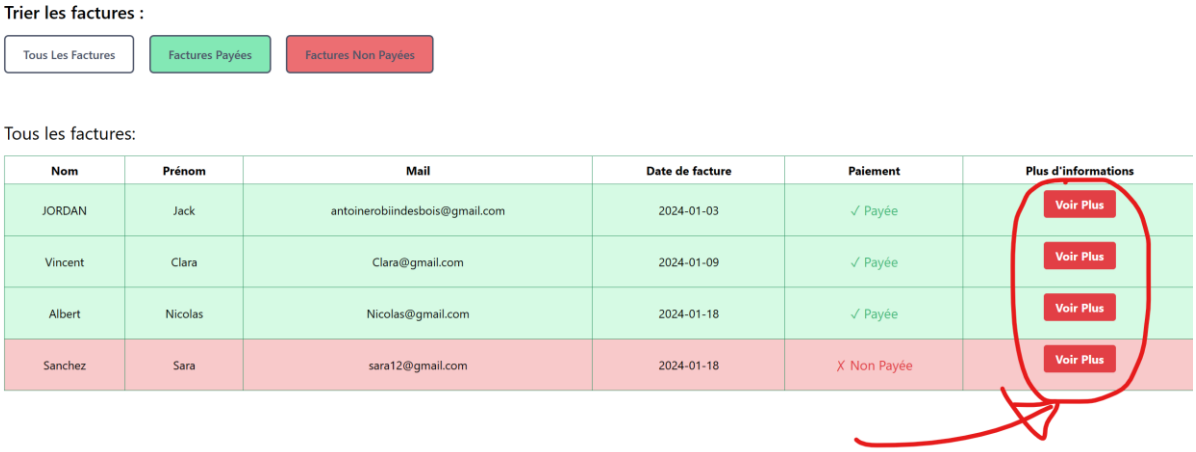
La fonction afficherFactures() est appelée avec le tableau de factures non payées ($facture->getFactureNonPaye()) et un titre "Les factures Non Payées:".

Voir les prix

Si on clique sur le bouton « **Afficher les prix** » , on affiche les montants de ce congressiste choisi :

- if(isset($_POST[‘afficherPrix’])
- echo $montantHotel->montant_hotel
- echo $montantSession->montant_session
- echo $montantActivite->montant_activite

Voir plus



Lorsqu'on clique sur le bouton voir plus, il affiche des informations supplémentaires concernant cette facture choisi en récupérant id de congressiste associé à cette facture :

- If (isset($_POST['infoPlus'']))
- $facture= new Facture() ; **//récupérer les fonctions de cette classe.**
- $selectedId=$_POST['listeCong'] ; **//récupérer id de congressiste choisi associé à cette facture.**

```
- $factureInfo = $facture->getFactureInfoById($selectedId);
- $montantActivite = $facture->getMontantActiviteById($selectedId);
- $montantHotel = $facture->getMontantHotelById($selectedId);
- $montantSession = $facture->getMontantSessionById($selectedId);
- $facturePaye = $facture->existCongFacturePaye($selectedId);
- $factureNonPaye = $facture->existCongFactureNonPaye($selectedId);
```

- Déclarer plusieurs variables pour rappeler les fonctions nécessaires à utiliser de la classe Facture.
- Afficher les informations :

```
- echo $factureInfo->nom;
- echo $factureInfo->prenom;
```

```
-   echo $factureInfo->adresse;
-   cho $factureInfo->ville;
-   echo $factureInfo->mail;
-   echo $montantHotel->montant_hotel;
-   echo $montantActivite->montant_activite;
-   echo $montantSession->montant_session;
-   echo $montantHotel->montant_hotel + $montantSession-
    >montant_session + $montantActivite->montant_activite;
```

- La statut de la facture sil est payé ou pas :

```
-   <?php if($facturePaye){
-                                   echo "Payé";
-                               }
-                               if($factureNonPaye){
-                                   echo "Non Payé";
-                               } ?>
```