Pweb2

- **HelpHub : Gestion de collecte des dons**
- **Année universitaire : 2024-2025**
- **Matière : Programmation Web2**

Ce projet est realisé par :

- Yengui Mohamed Amine
  - Mail : Amineyengui0407@gmail.com
- Timoumi Bassem
  - Mail : timoumibassem126@gmail.com

Introduction :

Objectif général :

Le projet a pour objectif de **concevoir et développer un site web dynamique** nommé **HelpHub**, permettant de **gérer les collectes de dons** pour des associations .

Ce site doit offrir aux associations la possibilité de :

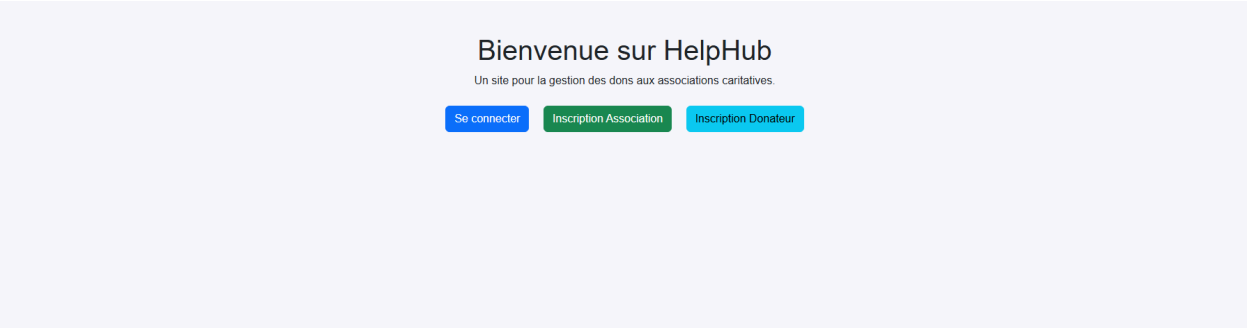
- S'inscrire
- Déposer des demandes de dons pour financer leurs projets d'aide aux personnes nécessiteuses,
- Suivre l'état d'avancement de la collecte.

Il doit aussi permettre aux donateurs de :

- S'inscrire,
- Consulter la liste des projets disponibles,

- Participer en versant un montant de leur choix pour soutenir les projets qui les intéressent.

Description des Pages du Site :



Cette page est le menu principal qui permet de gerer HelpHub

On a utilisé le Bootstrap.

C: > Users > BASS > Desktop > projetweb2 ff v2 > index.html

```html
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>HelpHub - Accueil</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container mt-5 text-center">
    <h1>Bienvenue sur HelpHub</h1>
    <p>Un site pour la gestion des dons aux associations caritatives.</p>
    <a href="login.html" class="btn btn-primary m-2">Se connecter</a>
    <a href="inscription_association.html" class="btn btn-success m-2">Inscription Association</a>
    <a href="inscription_donateur.html" class="btn btn-info m-2">Inscription Donateur</a>
  </div>
</body>
</html>
```

Cette page comporte un formulaire permettant à un nouvel utilisateur de créer un compte en saisissant les informations suivantes :

- **Nom**
- **Prénom**
- **CIN** (numéro d'identité nationale, composé de 8 chiffres)
- **Email**
- **Pseudo** (composé uniquement de lettres)
- **Mot de passe**

Un bouton intitulé **"S'inscrire"** permet de valider les informations et de procéder à l'enregistrement dans la base de données.

C: > Users > BASS > Desktop > projetweb2 ff v2 > traitement_inscription_donateur.php

```php
<?php
try {
    $pdo = new PDO('mysql:host=localhost;dbname=helphub', 'root', '');
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $nom = $_POST['nom'];
    $prenom = $_POST['prenom'];
    $cin = $_POST['cin'];
    $email = $_POST['email'];
    $pseudo = $_POST['pseudo'];
    $password = $_POST['password'];

    $stmt = $pdo->prepare("INSERT INTO donateurs (nom, prenom, email, cin, pseudo, mot_de_passe) VALUES (?, ?, ?, ?, ?, ?)");
    $stmt->execute([$nom, $prenom, $email, $cin, $pseudo, $password]);

    header("Location: login.html");
} catch (PDOException $e) {
    die("Error: " . $e->getMessage());
}
?>
```

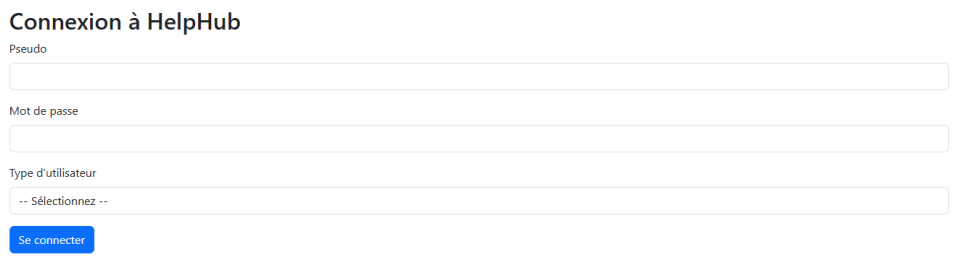
Cette page comporte un formulaire permettant à une nouvelle association de créer un compte en saisissant les informations suivantes :

- **Nom du responsable**
- **Prénom**
- **CIN** (numéro d'identité nationale, composé de 8 chiffres)
- **Email**
- **Nom de l’association**
- **Adresse de l’association**
- **Identifiant fiscal**
- **Logo**
- **Pseudo**
- **Mot de passe**

Un bouton intitulé **"S'inscrire"** permet de valider les informations et de procéder à l'enregistrement dans la base de données.

C: > Users > BASS > Desktop > projetweb2 ff v2 > traitement_inscription_association.php

```php
<?php
try {
    $pdo = new PDO('mysql:host=localhost;dbname=helphub', 'root', '');
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $nom = $_POST['nom'];
    $prenom = $_POST['prenom'];
    $cin = $_POST['cin'];
    $email = $_POST['email'];
    $nom_association = $_POST['nom_association'];
    $adresse = $_POST['adresse'];
    $identifiant_fiscal = $_POST['identifiant_fiscal'];
    $logo = $_POST['logo'];
    $pseudo = $_POST['pseudo'];
    $password = $_POST['password'];

    $stmt = $pdo->prepare("INSERT INTO associations (nom, prenom, email, cin, pseudo, mot_de_passe, nom_association, adresse, identifiant_fiscal, logo) VALUES (?, ?, ?, ?, ?, ?,
    $stmt->execute([$nom, $prenom, $email, $cin, $pseudo, $password, $nom_association, $adresse, $identifiant_fiscal, $logo]);

    header("Location: login.html");
} catch (PDOException $e) {
    die("Error: " . $e->getMessage());
}
?>
```

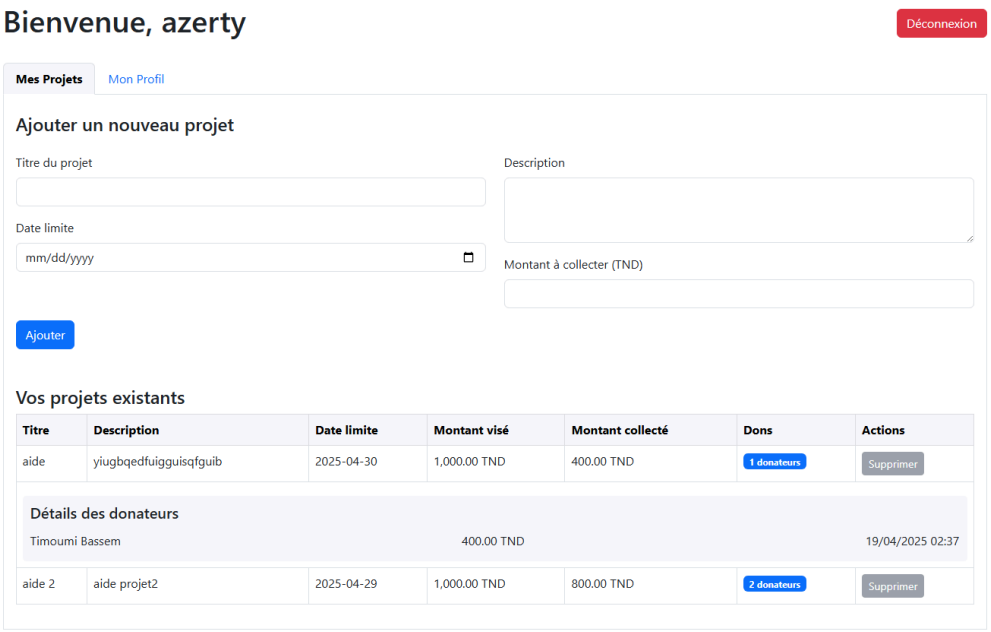


Cette page permet la connection au compte de l'un des deux types d'utilisateur apres saisir le Pseudo et le mot de passe.

On a utilise une session , une **session** permet de conserver des informations (variables) propres à un utilisateur entre plusieurs requêtes HTTP (pages). Contrairement aux cookies, les données de session sont stockées côté serveur, ce qui les rend plus sécurisées et plus volumineuses.

C: > Users > BASS > Desktop > projetweb2 ff v2 > traitement_login.php

```php
<?php
try {
    $pdo = new PDO('mysql:host=localhost;dbname=helphub', 'root', '');
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    session_start();

    $pseudo = $_POST['pseudo'];
    $password = $_POST['password'];
    $type = $_POST['type_utilisateur'];

    if ($type == 'association') {
        $stmt = $pdo->prepare("SELECT * FROM associations WHERE pseudo = ?");
    } else {
        $stmt = $pdo->prepare("SELECT * FROM donateurs WHERE pseudo = ?");
    }

    $stmt->execute([$pseudo]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($user) {
        if ($password === $user['mot_de_passe']) {
            $_SESSION['user'] = $user;
            $_SESSION['type'] = $type;
            header("Location: dashboard_" . $type . ".php");
        } else {
            echo "Mot de passe invalide !";
        }
    } else {
        echo "Utilisateur non trouvé !";
    }
} catch (PDOException $e) {
    die("Error: " . $e->getMessage());
}
?>
```



Après authentification, le Responsable d'Association accède à la page "Mes Projets" composée de :

- **Message de bienvenue** avec bouton de déconnexion.
- **Onglets de navigation** : "Mes Projets" et "Mon Profil".

- **Formulaire d'ajout de projet** :
  - Saisie du titre, description, date limite et montant à collecter.
  - Bouton "Ajouter" pour soumettre le projet.
- **Tableau des projets existants** :
  - Informations affichées : Titre, Description, Date limite, Montant visé, Montant collecté, Nombre de dons, Actions (Supprimer).
- **Détails des donateurs** visibles pour chaque projet :
  - Nom du donateur, montant du don et date.

```php
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['titre'])) {
    $titre = $_POST['titre'];
    $description = $_POST['description'];
    $date_limite = $_POST['date_limite'];
    $montant = floatval($_POST['montant_total_a_collecter']);
    $id_responsable = $user['id'];

    try {
        $stmt = $pdo->prepare("INSERT INTO projet
            (titre, description, date_limite, montant_total_a_collecter, montant_total_collecte, id_responsable_association)
            VALUES (:titre, :description, :date_limite, :montant_total_a_collecter, 0, :id_responsable_association)");

        $stmt->execute([
            ':titre' => $titre,
            ':description' => $description,
            ':date_limite' => $date_limite,
            ':montant_total_a_collecter' => $montant,
            ':id_responsable_association' => $id_responsable
        ]);

        $success_message = "Projet ajouté avec succès!";
    } catch (PDOException $e) {
        $error_message = "Erreur lors de l'ajout du projet: " . $e->getMessage();
    }
}
```

Aucune redirection ne se produit, donc à chaque fois que vous demandez l'ajout d'un nouveau projet, l'affichage des projets se réexécute.

```php
$projects = array();
try {
    $stmt = $pdo->prepare("SELECT * FROM projet WHERE id_responsable_association = :id_responsable_association");
    $stmt->execute([':id_responsable_association' => $user['id']]);
    while ($project = $stmt->fetch(PDO::FETCH_ASSOC)) {
        $donors = array();
        $donor_stmt = $pdo->prepare("SELECT d.*, dp.montant_participation, dp.date_participation
            FROM donateur_projet dp
            JOIN donateurs d ON dp.id_donateur = d.id
            WHERE dp.id_projet = :id_projet");
        $donor_stmt->execute([':id_projet' => $project['id_projet']]);
        while ($donor = $donor_stmt->fetch(PDO::FETCH_ASSOC)) {
            $donors[] = $donor;
        }
        $project['donors'] = $donors;
        $projects[] = $project;
    }
} catch (PDOException $e) {
    $error_message = "Erreur lors de la récupération des projets: " . $e->getMessage();
}
?>
```

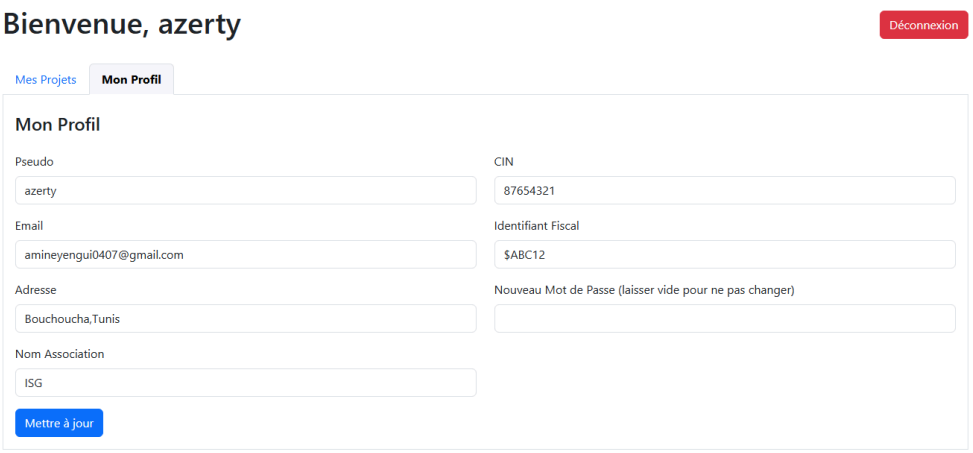
S'il y a des dons pour ce projet, le bouton de suppression est désactivé.

```php
<tbody>
    <?php foreach ($projects as $project): ?>
    <tr>
        <td><?php echo htmlspecialchars($project['titre']); ?></td>
        <td><?php echo htmlspecialchars($project['description']); ?></td>
        <td><?php echo htmlspecialchars($project['date_limite']); ?></td>
        <td><?php echo number_format($project['montant_total_a_collecter'], 2); ?> TND</td>
        <td><?php echo number_format($project['montant_total_collecte'], 2); ?> TND</td>
        <td>
            <span class="badge bg-primary badge-donors"
                onclick="toggleDonors(<?php echo $project['id_projet']; ?>)">
                <?php echo count($project['donors']); ?> donateurs
            </span>
        </td>
        <td>
            <?php if ($project['montant_total_collecte'] == 0): ?>
                <form method="post" style="display:inline;">
                    <input type="hidden" name="delete_project" value="<?php echo $project['id_projet']; ?>">
                    <button type="submit" class="btn btn-danger btn-sm" onclick="return confirm('Êtes-vous sûr de vouloir supprimer ce projet?');">Supprimer</button>
                </form>
            <?php else: ?>
                <button class="btn btn-secondary btn-sm" disabled>Supprimer</button>
            <?php endif; ?>
        </td>
    </tr>
```

```php
    <tr id="donors-<?php echo $project['id_projet']; ?>" style="display:none;">
        <td colspan="7">
            <div class="donor-details">
                <h5>Détails des donateurs</h5>
                <?php if (count($project['donors']) > 0): ?>
                    <?php foreach ($project['donors'] as $donor): ?>
                        <div class="donor-row">
                            <span><?php echo htmlspecialchars($donor['nom'] . ' ' . $donor['prenom']); ?></span>
                            <span><?php echo number_format($donor['montant_participation'], 2); ?> TND</span>
                            <span><?php echo date('d/m/Y H:i', strtotime($donor['date_participation'])); ?></span>
                        </div>
                    <?php endforeach; ?>
                <?php else: ?>
                    <p>Aucun donateur pour ce projet</p>
                <?php endif; ?>
            </div>
        </td>
    </tr>
    <?php endforeach; ?>
</tbody>
```

Avec une fonction de bascule JavaScript, elle rend la section des détails du donateur visible ou non.

```javascript
<script>
    function toggleDonors(projectId) {
        const donorRow = document.getElementById('donors-' + projectId);
        if (donorRow.style.display === 'none') {
            donorRow.style.display = 'table-row';
        } else {
            donorRow.style.display = 'none';
        }
    }
</script>
```



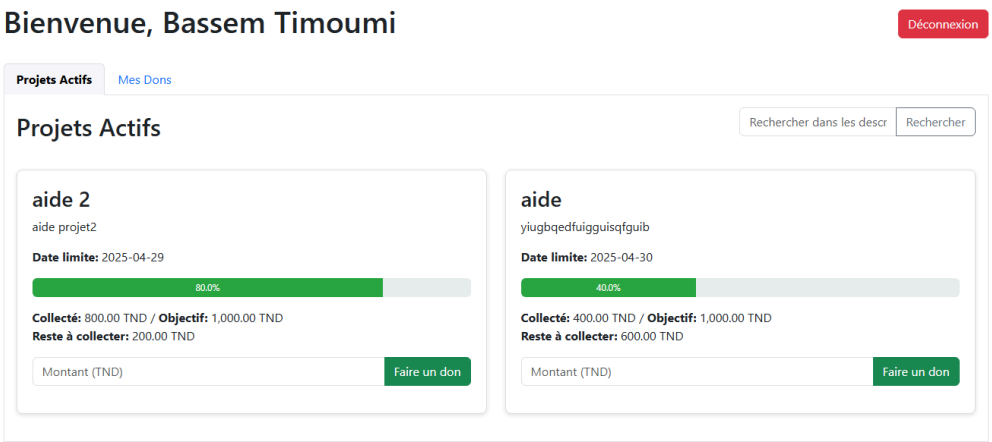
L'onglet de navigation "Mon Profil" :

- **Message de bienvenue** avec bouton de déconnexion.
- **Formulaire d'ajout de profil** :
  - Saisie du Pseudo, CIN, Email et Identifiant Fiscal , Adresse ,Nouveau mot de passe et Nom de l'Association
  - Bouton "Ajouter" pour soumettre le projet.

```php
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['update_profile'])) {
    $pseudo = $_POST['pseudo'];
    $email = $_POST['email'];
    $adresse = $_POST['adresse'];
    $nom_association = $_POST['nom_association'];
    $cin = $_POST['cin'];
    $identifiant_fiscal = $_POST['identifiant_fiscal'];
    $id = $user['id'];

    try {
        $query = "UPDATE associations SET
                  pseudo = :pseudo,
                  email = :email,
                  adresse = :adresse,
                  nom_association = :nom_association,
                  cin = :cin,
                  identifiant_fiscal = :identifiant_fiscal";

        if (!empty($_POST['mot_de_passe'])) {
            $query .= ", mot_de_passe = :mot_de_passe";
        }

        $query .= " WHERE id = :id";

        $stmt = $pdo->prepare($query);

        $params = [
            ':pseudo' => $pseudo,
            ':email' => $email,
            ':adresse' => $adresse,
            ':nom_association' => $nom_association,
            ':cin' => $cin,
            ':identifiant_fiscal' => $identifiant_fiscal,
            ':id' => $id
        ];

        if (!empty($_POST['mot_de_passe'])) {
            $params[':mot_de_passe'] = $_POST['mot_de_passe'];
```

La session est rafraîchie à chaque mise à jour.

```php

        if (!empty($_POST['mot_de_passe'])) {
            $params[':mot_de_passe'] = $_POST['mot_de_passe'];
        }

        $stmt->execute($params);

        // Refresh user data
        $stmt = $pdo->prepare("SELECT * FROM associations WHERE id = :id");
        $stmt->execute[':id' => $id]);
        $_SESSION['user'] = $stmt->fetch(PDO::FETCH_ASSOC);
        $user = $_SESSION['user'];

        $success_message = "Profil mis à jour avec succès!";
    } catch (PDOException $e) {
        $error_message = "Erreur lors de la mise à jour: " . $e->getMessage();
    }
}

```



Cela permet de suivre l'avancement des dons. Une fois le montant total collecté, le projet sera retiré de la page "Projets Actifs" et déplacé vers la page "Historique de Mes Dons".

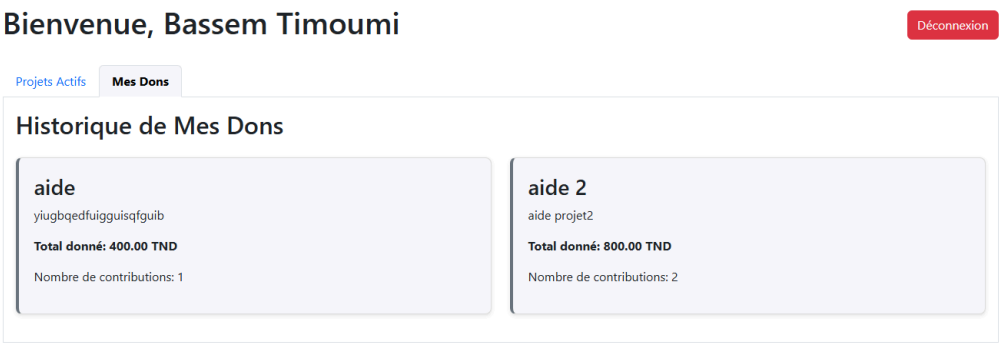
```php

// Build base query for active projects
$base_query = "SELECT * FROM projet WHERE date_limite >= CURDATE() AND montant_total_collecte < montant_total_a_collecter";

// Add search condition if keyword exists
$params = [];
if (!empty($search_keyword)) {
    $base_query .= " AND description LIKE ?";
    $params[] = '%' . $search_keyword . '%';
}

// Complete query with sorting
$query = $base_query . " ORDER BY date_limite ASC";

// Get active projects
$projects = [];
$stmt = $pdo->prepare($query);
$stmt->execute($params);
$projects = $stmt->fetchAll(PDO::FETCH_ASSOC);

// Get donor's donation history
$donation_history = [];
$history_query = "SELECT p.titre, p.description, SUM(dp.montant_participation) as total_don, COUNT(dp.id) as nombre_dons
                  FROM donateur_projet dp
                  JOIN projet p ON dp.id_projet = p.id_projet
                  WHERE dp.id_donateur = ?
                  GROUP BY p.id_projet";
$stmt = $pdo->prepare($history_query);
$stmt->execute([$donateur['id']]);
$donation_history = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>
```

La requête est personnalisée en fonction de la barre de recherche.

```php
<?php if (count($projects) > 0): ?>
<?php foreach ($projects as $project):
    $progress = ($project['montant_total_collecte'] / $project['montant_total_a_collecter']) * 100;
    $remaining = $project['montant_total_a_collecter'] - $project['montant_total_collecte'];
?>
<div class="col-md-6">
    <div class="project-card">
        <h3><?php echo htmlspecialchars($project['titre']); ?></h3>
        <p><?php echo htmlspecialchars($project['description']); ?></p>
        <p><strong>Date limite:</strong> <?php echo htmlspecialchars($project['date_limite']); ?></p>

        <div class="progress">
            <div class="progress-bar" role="progressbar"
                 style="width: <?php echo $progress; ?>%"
                 aria-valuenow="<?php echo $progress; ?>"
                 aria-valuemin="0"
                 aria-valuemax="100">
                <?php echo number_format($progress, 1); ?>%
            </div>
        </div>

        <p>
            <strong>Collecté:</strong> <?php echo number_format($project['montant_total_collecte'], 2); ?> TND /
            <strong>Objectif:</strong> <?php echo number_format($project['montant_total_a_collecter'], 2); ?> TND
            <br>
            <strong>Reste à collecter:</strong> <?php echo number_format($remaining, 2); ?> TND
        </p>

        <form method="post" class="mt-3">
            <input type="hidden" name="project_id" value="<?php echo $project['id_projet']; ?>">
            <div class="input-group mb-3">
                <input type="number" class="form-control" name="amount"
                       min="1" max="<?php echo $remaining; ?>"
                       step="0.01" placeholder="Montant (TND)" required>
                <button class="btn btn-success" type="submit" name="donate">Faire un don</button>
```

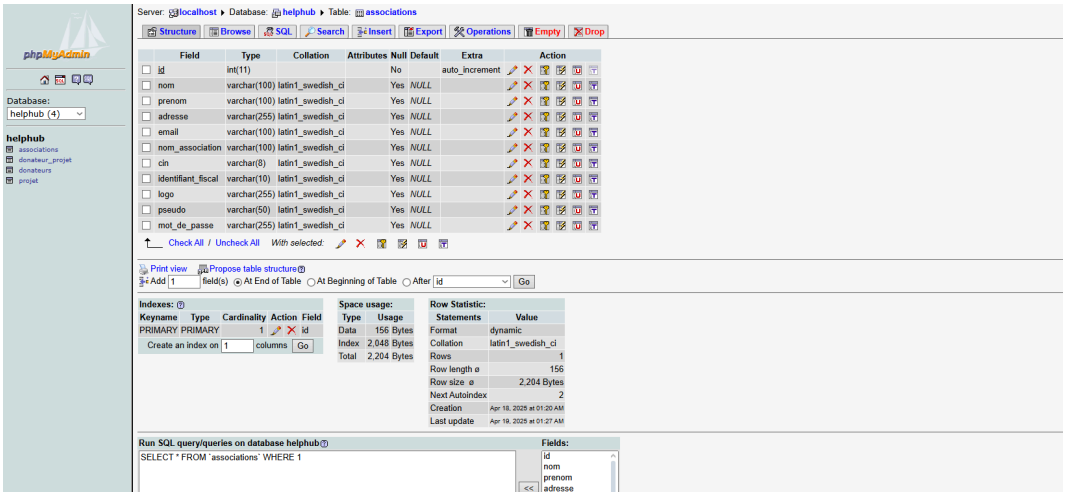
L’affichage des projets est mis à jour en temps réel à chaque don.

```php
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['donate'])) {
    $project_id = (int)$_POST['project_id'];
    $amount = (Float)$_POST['amount'];
    $donor_id = $donateur['id'];

    // Verify project is still active
    $stmt = $pdo->prepare("SELECT * FROM projet WHERE id_projet = ? AND date_limite >= CURDATE() AND montant_total_collecte < montant_total_a_collecter");
    $stmt->execute([$project_id]);
    $project = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($project) {
        $remaining = $project['montant_total_a_collecter'] - $project['montant_total_collecte'];

        // Validate amount
        if ($amount > 0 && $amount <= $remaining) {
            try {
                // Start transaction
                $pdo->beginTransaction();

                // 1. Record the donation
                $insert_query = "INSERT INTO donateur_projet (id_projet, id_donateur, montant_participation, date_participation) VALUES (?, ?, ?, NOW())";
                $stmt1 = $pdo->prepare($insert_query);
                $stmt1->execute([$project_id, $donor_id, $amount]);

                // 2. Update project collected amount
                $update_query = "UPDATE projet SET montant_total_collecte = montant_total_collecte + ? WHERE id_projet = ?";
                $stmt2 = $pdo->prepare($update_query);
                $stmt2->execute([$amount, $project_id]);

                $pdo->commit();
                $success_message = "Merci pour votre don de " . number_format($amount, 2) . " TND!";
            } catch (Exception $e) {
                $pdo->rollBack();
                $error_message = "Erreur lors du traitement de votre don";
            }
        } else {
            $error_message = "Montant invalide. Le montant restant à collecter est " . number_format($remaining, 2) . " TND";
        }
    } else {
```




```php
234            <div class="tab-pane fade" id="history" role="tabpanel">
235                <h2 class="mb-4">Historique de Mes Dons</h2>
236
237                <?php if (count($donation_history) > 0): ?>
238                    <div class="row">
239                        <?php foreach ($donation_history as $donation): ?>
240                        <div class="col-md-6">
241                            <div class="project-card history-card">
242                                <h3><?php echo htmlspecialchars($donation['titre']); ?></h3>
243                                <p><?php echo htmlspecialchars($donation['description']); ?></p>
244                                <p class="fw-bold">Total donné: <?php echo number_format($donation['total_don'], 2); ?> TND</p>
245                                <p>Nombre de contributions: <?php echo $donation['nombre_dons']; ?></p>
246                            </div>
247                        </div>
248                        <?php endforeach; ?>
249                    </div>
250                <?php else: ?>
251                    <div class="alert alert-info">Vous n'avez pas encore effectué de dons</div>
252                <?php endif; ?>
253            </div>
254        </div>
255    </div>
```

Base de Données :



Server: localhost ▸ Database: helphub ▸ Table: donateur_projet

Structure | Browse | SQL | Search | Insert | Export | Operations | Empty | Drop

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	id	int(11)			No		auto_increment	
☐	id_projet	int(11)			No	0		
☐	id_donateur	int(11)			No	0		
☐	montant_participation	float			No	0		
☐	date_participation	timestamp			Yes	CURRENT_TIMESTAMP		

Check All / Uncheck All *With selected:*

Print view | Propose table structure

Add 1 field(s) ◉ At End of Table ○ At Beginning of Table ○ After id Go

Indexes:

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	3		id
id_projet	INDEX	None		id_projet
id_donateur	INDEX	None		id_donateur

Create an index on 1 columns Go

Space usage:

Type	Usage
Data	63 Bytes
Index	4,096 Bytes
Total	4,159 Bytes

Row Statistic:

Statements	Value
Format	fixed
Collation	latin1_swedish_ci
Rows	3
Row length ø	21
Row size ø	1,386 Bytes
Next Autoindex	4
Creation	Apr 18, 2025 at 02:13 AM
Last update	Apr 20, 2025 at 01:43 PM

Run SQL query/queries on database helphub

```
SELECT * FROM `donateur_projet` WHERE 1
```

Fields: id, id_projet, id_donateur, montant_participation, date_participation

Server: localhost ▸ Database: helphub ▸ Table: donateurs

Structure | Browse | SQL | Search | Insert | Export | Operations | Empty | Drop

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	id	int(11)			No		auto_increment	
☐	nom	varchar(100)	latin1_swedish_ci		Yes	*NULL*		
☐	prenom	varchar(100)	latin1_swedish_ci		Yes	*NULL*		
☐	email	varchar(100)	latin1_swedish_ci		Yes	*NULL*		
☐	cin	varchar(8)	latin1_swedish_ci		Yes	*NULL*		
☐	pseudo	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
☐	mot_de_passe	varchar(255)	latin1_swedish_ci		Yes	*NULL*		

Check All / Uncheck All *With selected:*

Print view | Propose table structure

Add 1 field(s) ◉ At End of Table ○ At Beginning of Table ○ After id Go

Indexes:

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	1		id

Create an index on 1 columns Go

Space usage:

Type	Usage
Data	80 Bytes
Index	2,048 Bytes
Total	2,128 Bytes

Row Statistic:

Statements	Value
Format	dynamic
Collation	latin1_swedish_ci
Rows	1
Row length ø	80
Row size ø	2,128 Bytes
Next Autoindex	2
Creation	Apr 09, 2025 at 05:07 PM
Last update	Apr 09, 2025 at 07:17 PM

Run SQL query/queries on database helphub

```
SELECT * FROM `donateurs` WHERE 1
```

Fields: id, nom, prenom, email

Server: localhost ▸ Database: helphub ▸ Table: projet

Structure | Browse | SQL | Search | Insert | Export | Operations | Empty | Drop

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	id_projet	int(11)			No		auto_increment	
☐	titre	varchar(30)	latin1_swedish_ci		No			
☐	description	varchar(200)	latin1_swedish_ci		No			
☐	date_limite	date			No	0000-00-00		
☐	montant_total_a_collecter	double			No	0		
☐	montant_total_collecte	double			Yes	0		
☐	id_responsable_association	int(11)			No	0		

Check All / Uncheck All *With selected:*

Print view | Propose table structure

Add 1 field(s) ◉ At End of Table ○ At Beginning of Table ○ After id_projet Go

Indexes:

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	2		id_projet
id_responsable_association	INDEX	None		id_responsable_association

Create an index on 1 columns Go

Space usage:

Type	Usage
Data	124 Bytes
Index	3,072 Bytes
Total	3,196 Bytes

Row Statistic:

Statements	Value
Format	dynamic
Collation	latin1_swedish_ci
Rows	2
Row length ø	62
Row size ø	1,598 Bytes
Next Autoindex	4
Creation	Apr 18, 2025 at 02:12 AM
Last update	Apr 20, 2025 at 01:43 PM

Run SQL query/queries on database helphub

```
SELECT * FROM `projet` WHERE 1
```

Fields: id_projet, titre, description

1. **donateur_projet** - Table de liaison qui enregistre les dons (montant, date) et relie les donateurs aux projets qu'ils soutiennent.
2. **project** - Stocke les campagnes de financement (titre, description, objectif, montant collecté) avec leur association responsable.
3. **donateur** - Contient les informations personnelles et identifiants de connexion des donateurs.
4. **responsable_association** - Gère les représentants d'associations (coordonnées, infos juridiques, identifiants) et leurs organisations.

Fonctionnalité clé : Une plateforme de crowdfunding caritatif connectant donateurs et porteurs de projets.

Conclusion

Résumé du projet

Dans le cadre de ce mini-projet de Programmation Web2, nous avons conçu et développé une plateforme web dynamique baptisée **HelpHub**, dédiée à la gestion de la collecte de dons pour des associations caritatives.

Le site permet aux **associations** de publier des projets nécessitant des financements, et aux **donateurs** de consulter ces projets et d'y contribuer financièrement.

Grâce à une interface conviviale et une gestion efficace des données (inscription, authentification, ajout et suivi de projets, dons...), HelpHub facilite l'interaction entre les associations et les donateurs tout en assurant une meilleure transparence dans la gestion des dons.

Ce que nous avons appris

Au cours de ce projet, nous avons développé et renforcé plusieurs compétences techniques et organisationnelles essentielles :

- **Développement Frontend** : Maîtrise de la création d'interfaces utilisateurs modernes et intuitives en utilisant les technologies **HTML**, **CSS**, **Bootstrap** et

JavaScript.

- **Développement Backend** : Mise en place du traitement des données à l'aide de **PHP** et gestion efficace de la base de données via **MySQL**.
- **Validation des formulaires** : Implémentation de contrôles de saisie côté client pour assurer la qualité et la cohérence des données collectées.
- **Gestion de projet web** : Organisation structurée du travail en respectant les spécifications fonctionnelles, élaboration d'une architecture claire et validation des fonctionnalités par des tests réguliers.
- **Travail collaboratif** : (en cas de binôme) Coordination des tâches, communication fluide et intégration harmonieuse des différentes contributions pour garantir l'avancement du projet.