

Doc technique

Liste des membres:

ETU003214, ETU3172, ETU003098

Base de données :

```
CREATE TABLE pret_fond (  
    id_fond INT AUTO_INCREMENT PRIMARY KEY,  
    motif VARCHAR(50),  
    montant DECIMAL(15,2) NOT NULL  
);  
  
CREATE TABLE pret_client (  
    id_client INT AUTO_INCREMENT PRIMARY KEY,  
    nom VARCHAR(50) NOT NULL,  
    email VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE pret_type_pret (  
    id_type_pret INT AUTO_INCREMENT PRIMARY KEY,  
    nom VARCHAR(50) NOT NULL,  
    taux DECIMAL(15,2) NOT NULL,  
    duree INT,  
    assurance DECIMAL(5,2) DEFAULT 0  
);  
  
CREATE TABLE pret_pret (  
    id_pret INT AUTO_INCREMENT PRIMARY KEY,  
    montant DECIMAL(15,2) NOT NULL,  
    id_client INT NOT NULL,  
    id_type_pret INT NOT NULL,  
    date_pret DATE NOT NULL,  
    statuts ENUM('en attente', 'valide', 'refuse') NOT NULL DEFAULT 'en attente',  
    FOREIGN KEY (id_client) REFERENCES pret_client(id_client),  
    FOREIGN KEY (id_type_pret) REFERENCES pret_type_pret(id_type_pret)  
);  
  
CREATE TABLE pret_remboursement (  
    id_remboursement INT AUTO_INCREMENT PRIMARY KEY,  
    id_pret INT NOT NULL,
```

```

montant_total DECIMAL(15,2) NOT NULL,
interet DECIMAL(15,2) NOT NULL,
capital DECIMAL(15,2) NOT NULL,
capital_restant DECIMAL(15,2) NOT NULL,
mois INT NOT NULL,
annee INT NOT NULL,
assurance DECIMAL(5,2) DEFAULT 0,
FOREIGN KEY (id_pret) REFERENCES pret_pret(id_pret)
);

CREATE TABLE pret_simulation (
    id_simulation INT AUTO_INCREMENT PRIMARY KEY,
    id_pret INT,
    montant DECIMAL(15,2) NOT NULL,
    taux DECIMAL(15,2) NOT NULL,
    duree INT NOT NULL,
    assurance DECIMAL(5,2) DEFAULT 0,
    date_simulation DATETIME DEFAULT CURRENT_TIMESTAMP,
    details TEXT,
    FOREIGN KEY (id_pret) REFERENCES pret_pret(id_pret)
);

```

Fonctionnalite : Ajouter des fonds à l'établissement financier

Utilisateurs : Administrateur

Input :montant

Scenario nominal :

.L'administrateur se connecte et accède à la page `admin_accueil`.

.Un lien ou bouton "Ajouter des fonds" est affiché.

.Un formulaire s'affiche pour saisir le montant

.L'administrateur valide le formulaire, et les fonds sont ajoutés.

Scenario alternatif

.Redirection vers la page `admin_accueil` avec un message d'erreur précis (ex. "Montant invalide").

Règles de gestion :

.Le montant doit être un nombre positif (DECIMAL > 0).

.Mettre à jour le champ `fonds_total` dans la table `etablissement_financier`.

.Enregistrer la transaction dans la table `transaction_fonds`

- **Résultat :**

.Si l'ajout est possible :

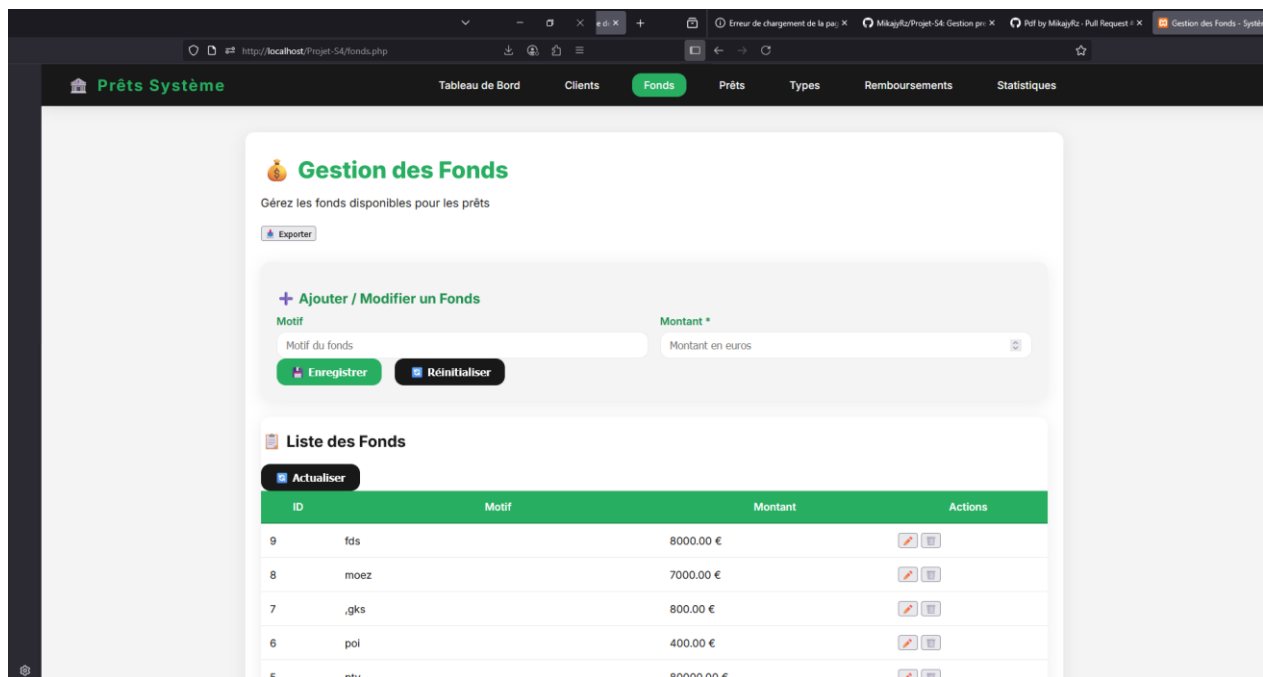
.Les fonds sont ajoutés à `fonds_total`.

.Une entrée est créée dans `transaction_fonds`.

.Si l'ajout échoue :

.Aucun changement dans la base de données.

.Affichage d'un message d'erreur précis.



Fonctionnalité : Créer un type de prêt

Utilisateur : Administrateur

Inputs : Nom du prêt, Taux d'intérêt, Durée maximale (en mois), Montant maximal

Scénario nominal :

- .L'administrateur se connecte et accède à la page `admin_accueil`.
- .Un lien ou bouton "Créer un type de prêt" est affiché.
- .Un formulaire s'affiche pour saisir les détails du type de prêt.
- .L'administrateur valide le formulaire, et le type de prêt est créé.

Scénario alternatif :

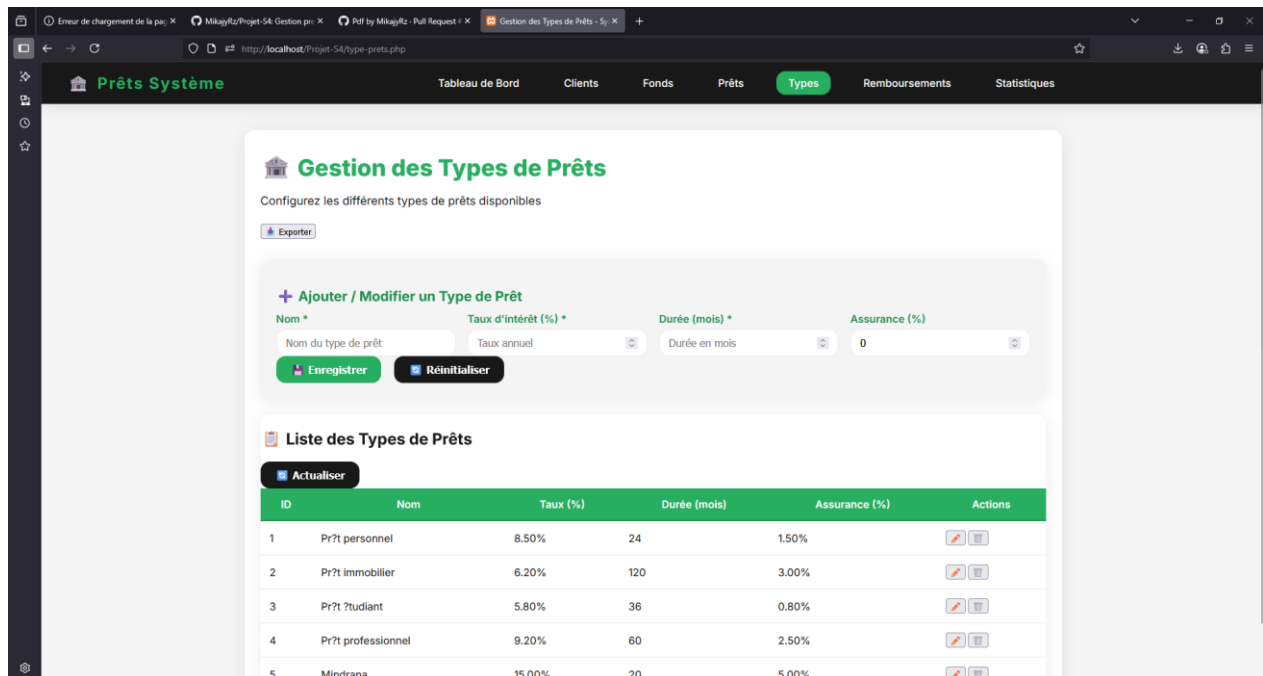
- .Redirection vers la page `admin_accueil` avec un message d'erreur précis (ex. "Taux d'intérêt invalide").

Règles de gestion :

- .Le nom du prêt doit être unique et non vide.
- .Le taux d'intérêt doit être un pourcentage valide (ex. entre 0 et 100).
- .La durée maximale doit être un entier positif (en mois).
- .Le montant maximal doit être un nombre positif.

.Résultat :

- .Si la création est possible :
 - .Une entrée est ajoutée dans la table `type_pret`.
- .Si la création échoue :
 - .Aucun changement dans la base de données.
 - .Affichage d'un message d'erreur précis.



Fonctionnalité : Demander un prêt

Utilisateur : Admin

Inputs : id_client, id_type_pret, Montant, Durée (en mois), date debut

Scénario nominal :

Un lien ou bouton "Faire un prêt" est affiché.

Un formulaire s'affiche pour sélectionner le client, un type de prêt, saisir le montant, et la durée et date debut

Le client valide le formulaire, et la demande de prêt est enregistrée.

Scénario alternatif :

Affichage d'un message d'erreur précis sur la même page

Règles de gestion :

L'id_client doit exister dans la table client.

L'id_type_pret doit exister dans la table type_pret.

Le montant demandé ne doit pas dépasser `montant_max` du type de prêt.

La durée demandée ne doit pas dépasser `duree_max_mois` du type de prêt.

Les fonds disponibles (`fonds_total`) dans `etablissement_financier` doivent être suffisants.

Vérifier si le client n'a pas de prêts en attente ou refusés

Résultat :

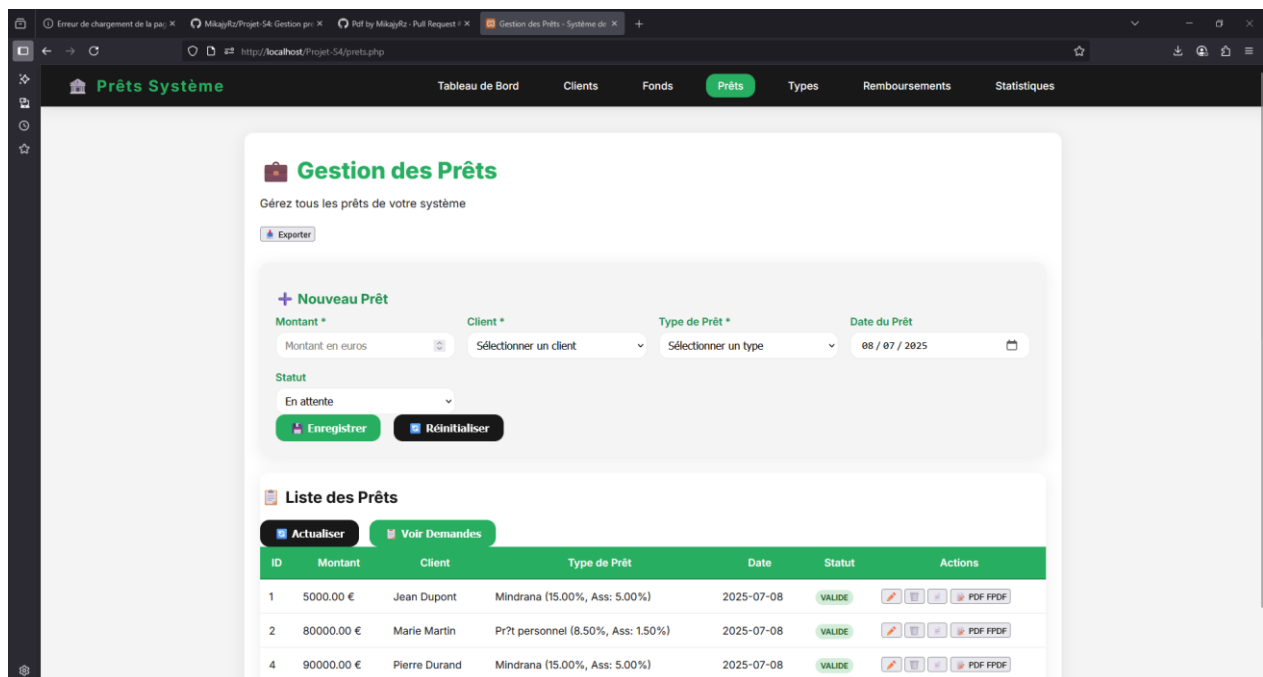
Si la demande est possible :

Une entrée est ajoutée dans la table `pret` avec `statut = 'en attente'`.

Si la demande échoue :

Aucun changement dans la base de données.

Affichage d'un message d'erreur précis.



Fonctionnalité : Valider ou refuser un prêt

Utilisateur : Administrateur

Inputs : `id_pret`, Statut (valide ou refuse)

Scénario nominal :

L'administrateur se connecte et accède à la page `admin_accueil`.

Un lien ou bouton "Gérer les prêts" affiche la liste des prêts en attente.

L'administrateur sélectionne un prêt et choisit d'approuver ou de refuser via un bouton ou formulaire.

La validation met à jour le statut du prêt.

Scénario alternatif :

Affichage d'un message d'erreur précis sur la même page (ex. "Prêt déjà traité").

Règles de gestion :

L'`id_pret` doit exister dans la table `pret`.

Le prêt doit être en statut `en attente`.

Si approuvé :

Vérifier que `fonds_total` dans `etablissement_financier` est suffisant.

Déduire le montant du prêt de `fonds_total`.

Enregistrer une transaction dans `transaction_fonds` avec
`type_transaction = 'retrait'`.

Si refusé :

- Mettre à jour le statut à `refusé` sans modifier les fonds.

Résultat :

Si approuvé :

Le statut du prêt devient `approuvé`.

Les fonds sont déduits, et une transaction est enregistrée.

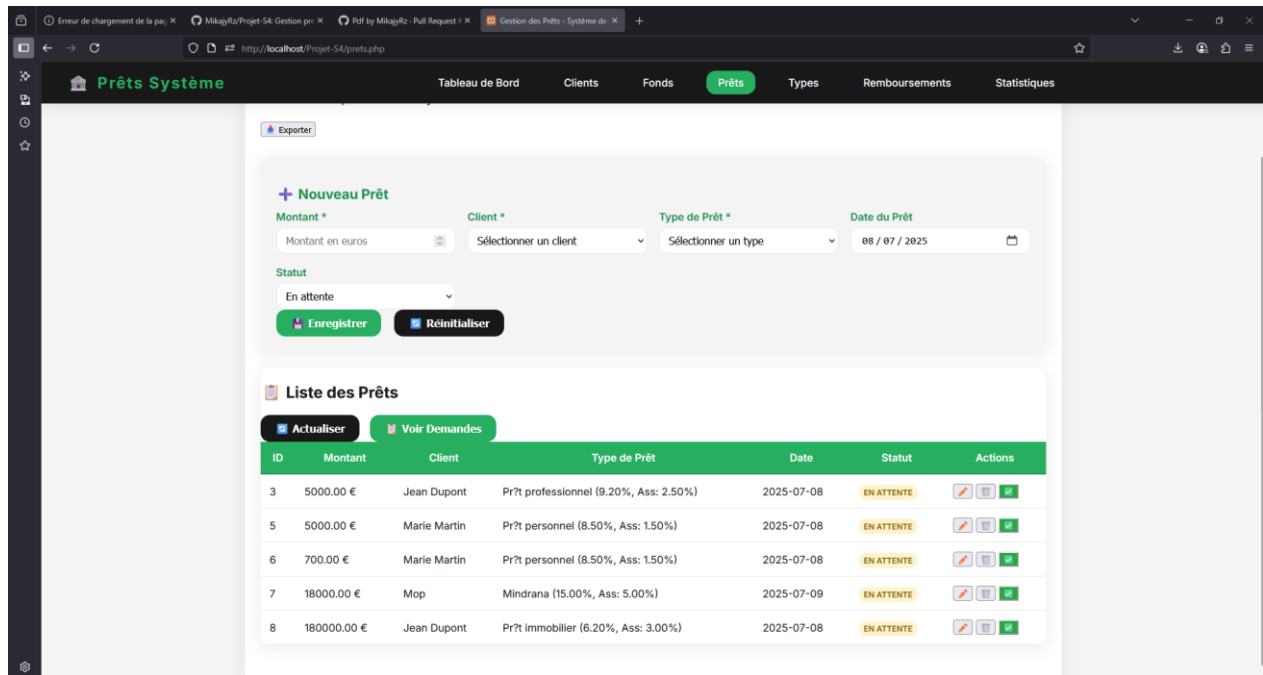
Si refusé :

Le statut du prêt devient `refusé`.

Si impossible :

Aucun changement dans la base de données.

Affichage d'un message d'erreur précis.



Fonctionnalité : Afficher un tableau des intérêts gagnés par mois pour l'établissement financier

Utilisateur : Administrateur

Inputs : Filtre pour mois et année de début, mois et année de fin

Scénario nominal :

L'administrateur se connecte et accède à la page *admin_accueil*.

Un lien ou bouton "Remboursements" affiche un formulaire pour sélectionner le mois/année de début et de fin.

Après validation, un tableau affiche les intérêts gagnés par mois dans la période sélectionnée.

Scénario alternatif :

Si la période est invalide (ex. date de fin antérieure à la date de début), afficher un message d'erreur précis sur la même page (ex. "Période invalide").

Règles de gestion :

Les intérêts sont calculés à partir des prêts approuvés dans la table *pret*, en utilisant le taux d'intérêt du *type_pret* et la formule de l'annuité constante.

Pour chaque mois, sommer les intérêts payés via les remboursements (basés sur la table *remboursement* ou calculés via la formule d'annuité).

Les filtres doivent être valides (mois entre 1 et 12, année positive, période cohérente).

Résultat :

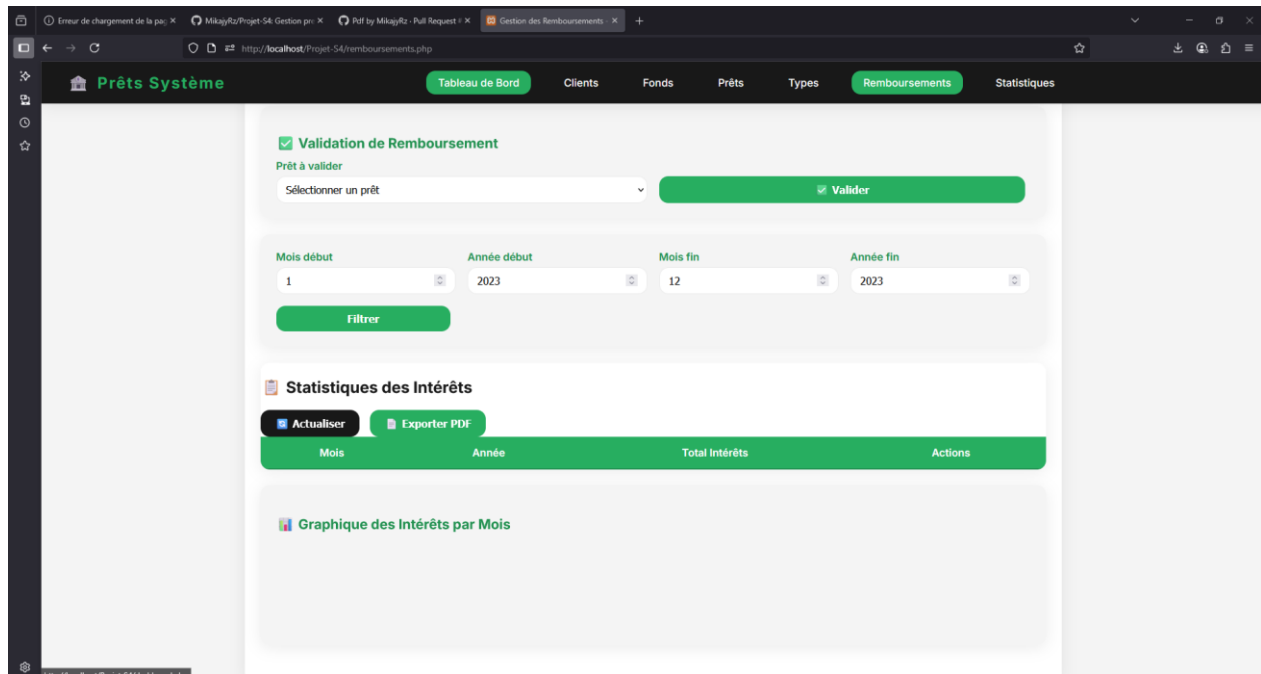
Si la requête est valide :

Un tableau est affiché avec les colonnes : Mois/Année, Intérêts gagnés (en unité monétaire).

Si la requête échoue :

Aucun tableau n'est affiché.

Affichage d'un message d'erreur précis.



Fonctionnalité : Créer un graphique à partir du tableau des intérêts gagnés

Utilisateur : Administrateur

Inputs : Données du tableau généré par la Fonctionnalité 8

Scénario nominal :

Après génération du tableau des intérêts gagnés, un graphique (type : ligne) est affiché sous le tableau pour visualiser les intérêts gagnés par mois.

Scénario alternatif :

Si aucune donnée n'est disponible (ex. période sans prêts), afficher un message indiquant l'absence de données.

Règles de gestion :

Utiliser Chart.js pour générer un graphique de type *line*.

Les données sont extraites du tableau des intérêts gagnés.

Les axes sont : X (Mois/Année), Y (Intérêts gagnés).

Utiliser des couleurs distinctes adaptées aux thèmes clair et sombre (ex. bleu pour la ligne, fond blanc/gris).

Résultat :

Si des données sont disponibles :

Un graphique est affiché sous le tableau.

Si aucune donnée :

Affichage d'un message indiquant l'absence de données.

Fonctionnalité : Système de simulation de prêt

Utilisateur : Client, Administrateur

Inputs : id_type_pret, Montant, Durée (en mois), Taux d'assurance (facultatif)

Scénario nominal :

L'utilisateur accède à une page de simulation via un lien "Simuler un prêt".

Un formulaire permet de sélectionner un type de prêt, saisir le montant, la durée, et le taux d'assurance.

Après validation, un tableau affiche les mensualités, les intérêts, et le coût total.

L'utilisateur peut sauvegarder la simulation.

Scénario alternatif :

Si les inputs sont invalides (ex. montant dépasse *montant_max*), afficher un message d'erreur précis.

Règles de gestion :

Les simulations sont temporaires sauf si sauvegardées (stockées dans une table *simulation_pret*).

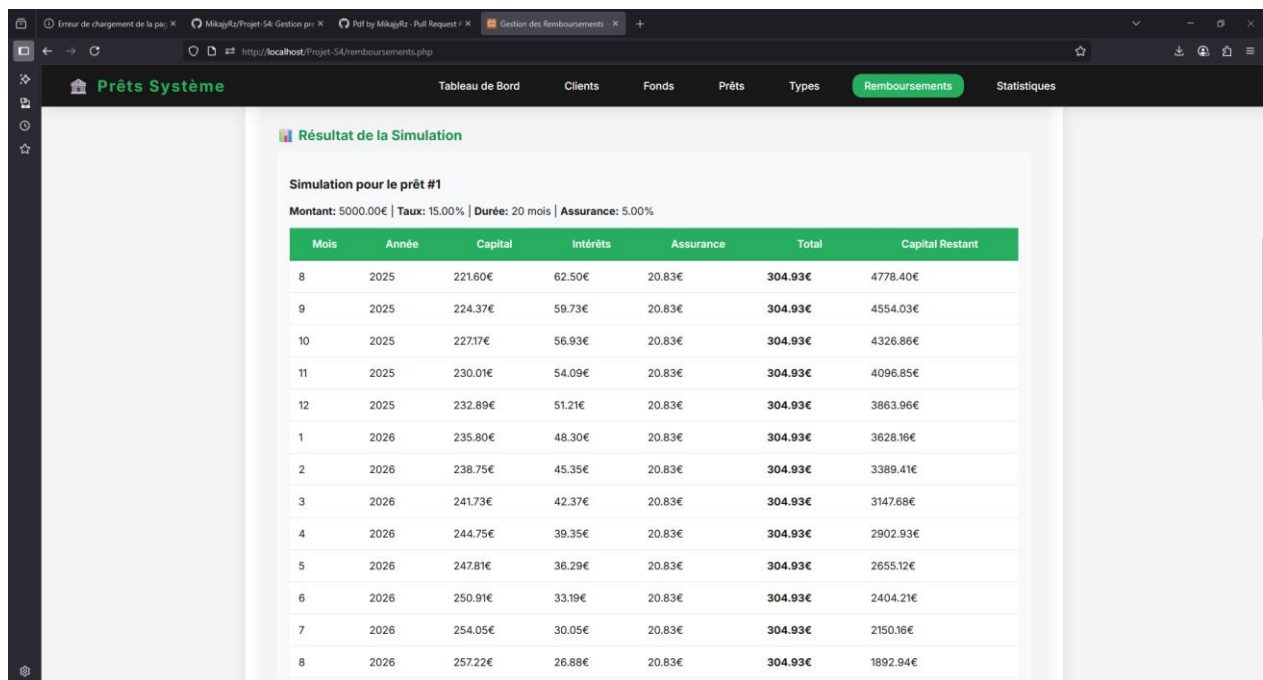
Un prêt simulé doit être validé par un administrateur pour devenir une demande (Fonctionnalité 4).

Vérifier les contraintes du *type_pret* (montant_max, duree_max_mois).

Résultat :

Un tableau affiche les détails de la simulation.

La simulation peut être sauvegardée ou convertie en demande.



The screenshot shows a web application interface with a sidebar on the left containing icons for home, settings, and other functions. The main content area is titled "Prêts Système" and has a navigation bar with tabs: "Tableau de Bord", "Clients", "Fonds", "Prêts", "Types", "Remboursements" (highlighted in green), and "Statistiques". The main content displays a "Résultat de la Simulation" for "Simulation pour le prêt #1". The simulation parameters are: Montant: 5000.00€, Taux: 15.00%, Durée: 20 mois, Assurance: 5.00%. Below this is a table with 7 columns: Mois, Année, Capital, Intérêts, Assurance, Total, and Capital Restant. The table shows 20 rows of data, with the first 12 rows for the year 2025 and the next 8 rows for the year 2026. The "Total" column is consistently 304.93€ for all rows.

Mois	Année	Capital	Intérêts	Assurance	Total	Capital Restant
8	2025	221.60€	62.50€	20.83€	304.93€	4778.40€
9	2025	224.37€	59.73€	20.83€	304.93€	4554.03€
10	2025	227.17€	56.93€	20.83€	304.93€	4326.86€
11	2025	230.01€	54.09€	20.83€	304.93€	4096.85€
12	2025	232.89€	51.21€	20.83€	304.93€	3863.96€
1	2026	235.80€	48.30€	20.83€	304.93€	3628.16€
2	2026	238.75€	45.35€	20.83€	304.93€	3389.41€
3	2026	241.73€	42.37€	20.83€	304.93€	3147.68€
4	2026	244.75€	39.35€	20.83€	304.93€	2902.93€
5	2026	247.81€	36.29€	20.83€	304.93€	2655.12€
6	2026	250.91€	33.19€	20.83€	304.93€	2404.21€
7	2026	254.05€	30.05€	20.83€	304.93€	2150.16€
8	2026	257.22€	26.88€	20.83€	304.93€	1892.94€

Fonctionnalité : Générer un PDF pour un prêt

Utilisateur : Administrateur, Client

Inputs : id_pret

Scénario nominal :

L'utilisateur sélectionne un prêt approuvé via un bouton "Générer PDF".

Un PDF est généré avec les détails du prêt (montant, durée, taux, mensualités, assurance, etc.).

Le PDF est téléchargeable.

Scénario alternatif :

Si le prêt n'existe pas ou n'est pas approuvé, afficher un message d'erreur précis.

Règles de gestion :

Utiliser une bibliothèque PHP (ex. Dompdf) pour générer le PDF.

Inclure : Nom du client, ID du prêt, Type de prêt, Montant, Durée, Taux d'intérêt, Assurance, Mensualités, Tableau d'amortissement.

Le PDF doit être formaté clairement (ex. logo de l'EF, tableau lisible).

Résultat :

Un PDF est généré et téléchargeable.

