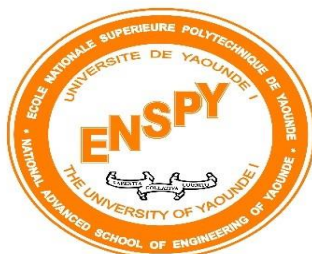


UNIVERSITÉ DE YAOUNDÉ I
ÉCOLE NATIONALE SUPÉRIEURE
POLYTECHNIQUE DE YAOUNDÉ

DÉPARTEMENT DES GÉNIES
ÉLECTRIQUE ET DES
TÉLÉCOMMUNICATIONS



THE UNIVERSITY OF YAOUNDE I
NATIONAL ADVANCED SCHOOL
OF ENGINEERING OF YAOUNDE

DEPARTMENT OF ELECTRICAL
AND TELECOMMUNICATIONS
ENGINEERING

UE : PROGRAMMATION PYTHON

EC 3522

THÈME :

**CONCEPTION ET REALISATION D'UN
LOGICIEL DE GESTION DES TONTINES**

GROUPE 2

MEMBRES :

NOMS	PRENOMS	MATRICULE	POURCENTAGE
ADER EBEN	YOHAN CHRISTOPHER	22P468	16.0%
DJOMEKOUM LAAGNEY	ANGE JOYCE	23P441	16.0%
MEFIRE	JOSAPHAT	25P856	16.0%
UMARU	NAFISSATU LADI UMU SALAMATU	23P185	20.0%
VOUNDI TSOUNGUI	ARMEL	23P377	16.0%
WAFO WOUAFO	JOSEPH WILLIAM	22P211	16.0%

RESPONSABLE : M. MBIETIEU AMOS
Année académique : 2025/2026

Table des matières

1. INTRODUCTION	1
2. OBJECTIFS DU PROJET.....	1
3. PLANNING DU TRAVAIL	2
4. CHOIX DU PROJET	3
5. FONCTIONNEMENT DE L'APPLICATION.....	5
6. PISTES D'AMELIORATIONS	6
7. IMPLEMENTATION DES PISTES D'AMELIORATIONS	7
8. DIFFICULTÉS RENCONTRÉES ET SOLUTIONS APPORTÉES	12
9. DIAGRAMME DE GANTT	12
10. CONCLUSION	13

RAPPORT D'ACTIVITÉ ET D'ANALYSE DE **PROJET PYTHON**

1. INTRODUCTION

Le présent rapport a pour objectif de synthétiser les activités menées par notre groupe dans le cadre du cours de python, axé sur l'analyse et la sélection d'un projet open-source en langage Python. Ce travail s'inscrit dans une démarche d'apprentissage pratique visant à confronter les concepts théoriques aux réalités du développement de projets concrets.

L'activité principale a consisté à identifier, cloner et comparer plusieurs projets Python répondant à une thématique commune, celle de la **gestion d'une application de tontine**. Après une analyse comparative rigoureuse basée sur des critères techniques et fonctionnels prédéfinis, un projet a été sélectionné pour une étude approfondie.

Ce document est structuré conformément au plan de travail établi et détaille successivement les objectifs du projet, le planning de travail, le processus de choix, l'analyse fonctionnelle et architecturale du projet retenu, et enfin, les pistes d'amélioration envisagées pour le futur.

2. OBJECTIFS DU PROJET

La réalisation de ce travail répond à plusieurs d'objectifs spécifiques, au-delà des exigences du cours, notre groupe s'est fixé des objectifs d'apprentissage spécifiques pour enrichir notre expérience collective :

- **Approfondir la connaissance des frameworks Python** : Comprendre les différences pratiques de framework comme, Django dans le contexte d'une application de tontine.
- **Développer l'esprit critique technique** : Être capable de justifier le choix d'une technologie ou d'une architecture par rapport à une autre en se basant sur des critères objectifs.
- **Renforcer la collaboration** : Améliorer notre organisation interne et la répartition des tâches pour une efficacité maximale, en simulant un environnement de travail professionnel.
- **Identifier des pistes d'innovation** : Ne pas se contenter de l'existant, mais proposer des améliorations concrètes et des solutions d'implémentation pour le projet choisi.

Ces objectifs ont guidé l'ensemble de nos activités, de la recherche initiale à l'analyse détaillée du projet retenu.

3. PLANNING DU TRAVAIL

La gestion rigoureuse du temps et des ressources a été un facteur clé de succès pour ce projet. Une approche structurée, s'appuyant sur les principes de la gestion de projet, a permis de séquencer les activités et d'assurer un suivi précis de l'avancement.

3.1. LES GRANDES PHASES DU PROJET

Le projet a été décomposé en six grandes phases logiques, allant de la préparation initiale à l'évaluation finale :

- 1 **Préparation et Recherche** : Création de l'environnement de travail (GitHub) et recherche initiale des projets potentiels.
- 2 **Analyse et Sélection** : Clonage des projets, analyse comparative et sélection du projet cible.
- 3 **Étude Approfondie** : Analyse détaillée de l'architecture, de l'application et des modules du projet choisi.
- 4 **Identification des Améliorations** : Détection des pistes d'amélioration fonctionnelles et/ou techniques.
- 5 **Implémentation** : Réalisation concrète des améliorations identifiées (phase future).
- 6 **Évaluation** : Rédaction du rapport final et évaluation de l'atteinte des objectifs.

3.2. ORGANISATION DU TRAVAIL ET ATTRIBUTION DES TÂCHES

Le groupe est composé de six membres, et l'organisation du travail a reposé sur une répartition des responsabilités en fonction des compétences et des phases du projet. Cette approche a permis de garantir que chaque section du rapport bénéficie d'une expertise dédiée.

Membre du Groupe	Rôle Principal	Tâches Attribuées
1. UMARU NAFISSATU LADI UMU SALAMATU	Gestion de Projet et Coordination (chef de groupe)	Faire marcher le projet 3. Travailler sur la piste d'amélioration 2.
2. DJOMEKOUM LAAGNEY ANGE JOYCE	Étude du fonctionnement du projet 2	Comprendre comment fonctionne le projet 2, réalisation de l'analyse comparative. Travailler sur la piste d'amélioration 1.
3. MEFIRE JOSAPHAT	Étude du fonctionnement du projet 4	Explication du principe plus rédaction du rapport. Travailler sur la

		piste d'amélioration 2.
4. VOUNDI TSOUNGUI ARMEL	Étude du fonctionnement du projet 1	Faire marcher le projet 1. Travailler sur la piste d'amélioration 1.
5. WAFO WOUAFO JOSEPH WILLIAM	Analyse du projet	Faire marcher le projet du groupe 5. Travailler sur la piste d'amélioration 3.
6. ADER EBEN YOHAN CHRISTOPHER	Lecture plus analyse	Proposition d'implémentation des améliorations relecture et vérification technique du rapport. Travailler sur la piste d'amélioration 3.

Tableau 1 : récapitulatif des tâches et répartition aux membres du groupe

4. CHOIX DU PROJET

Le choix du projet à analyser en profondeur a été l'étape la plus critique de la phase initiale. Il a été guidé par une démarche méthodologique visant à garantir que le projet retenu offre la meilleure opportunité d'apprentissage et de compréhension des enjeux du développement d'une application de gestion de tontine.

4.1. TABLEAU COMPARATIF

Pour commencer, nous avons pensé à utiliser comme critère le type de base de données utilisée et les fonctionnalités de l'application. Mais nous avons constaté que pour tous les groupes, la base de données est la même ainsi que les fonctionnalités.

La base de données utilisée est MySQL.

Les applications sont exploitables à tout instant et de façon pratique. Les fonctionnalités basiques des tontines réelles ont bien été mises en place (ajouter des membres dans la tontine, faire des dons, souscrire à une tontine, faire un prêt ou effectuer un remboursement...). Par contre, étant donné que les projets ne sont pas complètement terminés, les fonctionnalités ne sont pas toutes opérationnelles dans certains projets.

Notre tableau comparatif (**Tableau 2**) est le suivant :

Critère	1. Interface	2. Flexibilité	1. Modularité	4. Critère de sécurité
Groupe 1	L'interface est structurée et belle (mélange de couleur ordonné et cohérent, image de fond parlante) ; pourtant il ne peut être modifié à convenance.	Les données ne sont pas récupérables : Puisqu' il n' y a pas une façon de quelconque le sauvegarder le travail fait Le projet 1 est donc peu flexible	L' application étant divisé en parties ou modules, mais certains ne sont pas indépendants. Elle a donc une faible modularité.	L' application présente un niveau de sécurité faible suivant les critères : La gestion des accès : les permissions demandées ne sont pas toutes justifiées (justificatif pour les dons mais pas pour les prêts) ; Mise à jour : il n'y a aucune mention de sécurité ; Paramètres de l' application : disponibles et
Groupe 2	L' interface est structurée et belle (mélange de couleur ordonné et cohérent). Visuel très attirant.	Les données ne sont pas récupérables : Puisqu' il n' y a pas une façon de quelconque le sauvegarder le travail fait Le projet 2 est donc	L' application étant divisé en parties, mais il est difficile de savoir si elles sont dépendantes ou non car l' application n' est pas utilisable. Elle a donc une	L' application présente un niveau de sécurité moyen suivant les critères : La gestion des accès : l'accès à l'application est sécurisé par un mot de passe unique à l'administrateur principal appelé « <i>superuser</i> ».
Groupe 3	L' interface est structurée et mais le jeu de couleurs n' est pas uniforme dans toutes les pages de l' application ce qui rend l' utilisation un peu gênant pour les	Les données sont récupérables car il y a des sauvegardes qui sont fait à chaque utilisation donc l' application du projet 3 est assez flexible	L' application étant divisé en parties, mais la plupart sont indépendants. Elle a donc une bonne modularité	L' application présente un niveau de sécurité moyen suivant les critères : La gestion des accès : l'accès à l'application est sécurisé par un mot de passe unique aux membres ou aux administrateurs. Mise à jour : seuls les administrateurs peuvent faire des modifications de grande envergure. Statistiques indisponibles.
Groupe 5	L' interface est structurée et facile d' utilisation.	Facilité de compréhension du projet L' application du projet 5 est assez flexible.	L' application étant divisé en parties, mais certains ne sont pas indépendants. Elle a donc une faible	L' application présente un niveau de sécurité moyen suivant les critères : Il a le chiffrement des données. Une forte authentification pour accéder aux modules et paramètres de l' application

4.2. CONCLUSION SUR LE CHOIX

Après une étude comparative des quatre projets, étude résumée dans le tableau ci-dessus, notre choix de projet pour le travail demandé s'est porté sur le projet 3 intitulé « **GROUPE-3_PROJET-** ».

Bien que ce projet ne soit pas fonctionnel à 100%, qu'il ait une interface utilisateur pas très attirante et un niveau de sécurité moyen nous pensons que sa flexibilité, son niveau de modularité et les fonctionnalités opérationnelles sont des atouts exploitables pour une bonne compréhension du projet et donc des améliorations plus efficaces.

5. FONCTIONNEMENT DE L'APPLICATION

L'application s'appelle « **Hels Association** » et elle est conçue pour deux types d'utilisateur cibles qui sont : l'administrateur qui est le responsable principal qui accède à toutes les fonctionnalités et les membres qui peuvent consulter leurs informations et faire des transactions. Cette application est constituée d'une interface utilisateur et sept modules d'utilisation selon les rapports et le guide d'utilisation de l'application.

Nous allons essayer de décrire brièvement le fonctionnement de cette application comme suit :

5.1. L'interface utilisateur

Ça revient aux icônes, listes, boutons, menus qui aide à la compréhension et à l'utilisation de l'application dans son ensemble sans forcément une aide extérieure. Nous avons une interface dédiée aux administrateurs et une interface dédiée aux membres partir de laquelle ils peuvent avoir accès à toutes les fonctionnalités de l'application relatives à aux selon la conception de l'application.

5.2. Les modules de l'application

L'application est subdivisée en sept grands modules qui sont nécessaires pour le bon fonctionnement de l'application. Ces modules cherchent à couvrir les fonctionnalités possibles d'une « tontine » réelle et sont accessibles à tout utilisateur de l'application selon leur poste ou statut. Ces modules sont :

1. *Le module « Membres » :*

- ✓ Il y a la possibilité pour tous (membre comme utilisateur) d'accéder à la **Liste des membres** pour connaître ceux qui sont inscrit et actifs dans l'association.
- ✓ Il y a la possibilité d'ajouter un membre (**Ajouter un membre**), modifier les informations concernant un membre (**Modifier un membre**) et de supprimer un membre de la base de données de l'application (**Supprimer un membre**) mais ceux-ci sont exclusivement réservés aux administrateurs de l'application.

2. *Le module « Tontines » :*

- ✓ Pour chaque utilisateur, il y a la possibilité de voir la **Liste des tontines** disponibles dans l'application.
- ✓ Souscrire à une tontine (cliquer sur le bouton **Souscrire** dans la **Liste des tontines** disponibles) : ici tous les utilisateurs sont concernés, souscrire à une tontine veut dire

que l'utilisateur se crée un compte dans lequel il peut participer dans toutes les activités de la tontine choisie (prêts, dons, remboursements, ...).

- ✓ On peut aussi ajouter un autre type de tontine avec le montant qui lui est approprié (**Ajouter une tontine**) et aussi ajouter un membre de la tontine dans le registre des membres souscrit à une tontine particulière (**Ajouter un membre dans une tontine**) mais ceux-ci exclusivement réservé aux administrateurs.

3. Le module « Prêts » :

- ✓ Demander un prêt : tout utilisateur à droit de faire un ou plusieurs prêts à un ou plusieurs tontines.
- ✓ Gestion des prêts : l'administrateur à accès aux demandes de prêts pour soit les **approuver** ou les **refuser**.
- ✓ Pour les membres le statut de leur demande de prêt est soit **Approuvé**, **En attente** ou **Refusé**.

4. Le module « Remboursements » :

- ✓ Effectuer un remboursement : Les remboursements se font en fonctions des prêts. Il existe aussi l'opportunité de rembourser des prêts en plusieurs tranches et quand ceci est fait le reste à rembourser s'affiche juste au-dessus.

5. Le module « Dons » :

- ✓ Effectuer un don : Ce module est ouvert à tous les utilisateurs de l'application. Il y a la possibilité d'effectuer des dons en nature (en faisant une description du don) ou en espèces (en spécifiant le montant).

6. Le module « Tableau de bord » :

- ✓ Dans cette partie, on peut voir le récapitulatif des tontines, prêt, remboursement et dons effectué dans par un membre dans l'application.

7. Le module « Statistique » :

- ✓ Selon le guide d'utilisation, dans cette partie il doit y avoir deux diagrammes circulaires, le premier représentant les différents membres en fonctions de leur tranche d'âge et le deuxième représentant le pourcentage d'adhésion aux différentes tontines.
- ✓ Problème survenu : cette partie de l'application reste indisponible quel que soit le compte ou le statut de l'utilisateur

6. PISTES D'AMELIORATIONS

L'application de tontine n'étant pas fini ou complète nous avons pu relever quelques points de modifications à apporter dans l'application pour qu'elle puisse satisfaire au maximum les attentes d'une tontine réelle. Les trois (3) pistes d'amélioration sont subdivisées par modules comme suit :

1. Le module « Membres » :

- En inscrivant, chacun rempli pour dire s'il est un membre actif ou non. Dans la **Liste des membres** de l'administrateur nous proposons d'ajouter une colonne pour indiquer le **statut** actif ou non des membres pour l'aider à prendre les décisions comme l'approbation des prêts.

2. Les modules « Prêts » et « Remboursements » :

- En faisant un prêt, il n'est pas spécifié à quel tontine le membre souhaite faire un prêt ni le délai de remboursement. Aussi n'est pas spécifier la tontine dans laquelle on souhaite faire un remboursement de prêt si le membre est inscrit dans plusieurs tontines de « **Hels Association** ». Nous pensons ajouter ces deux options dans l'application.

3. Le module « Statistique » :

- Nous pensons à faire deux diagrammes circulaires, le premier représentant les différents membres en fonctions de leurs tranches d'âge et le deuxième représentant le pourcentage d'adhésion aux différentes tontines ouvertes à tous car cette fonctionnalité reste inaccessible.

7. IMPLEMENTATION DES PISTES D'AMELIORATIONS

Les pistes d'amélioration citées plus haut ont bel et bien été implémentées dans l'application. Le processus de modification était de modifier les fichiers python *models*, *forms*, *views* et *urls* dans le dossier « mon_application » ainsi que les fichiers html (pages webs) concernées. Ainsi nous avons ci-après les captures d'écran de l'application de gestion des tontines avant et après les modifications dans les modules concernés ;

1. Module « Membre »

a. Avant (pour l'administrateur et les membres) :

Accueil Membres ▼ Tontines ▼ Prêts ▼ Remboursement ▼ Dons ▼ Tableau de bord Statistique						
Liste des Membres						
Nom	Prénom	Actions				
MAATSING	Armanda	Modifier	Supprimer			
EMMANUELLE	Cindy	Modifier	Supprimer			
SOBOTH	Shekinah	Modifier	Supprimer			
MEWALI	Tatiana	Modifier	Supprimer			
TENDJIANG	Isnel	Modifier	Supprimer			
DONGMO	Romarth	Modifier	Supprimer			
Akam	patrick	Modifier	Supprimer			

- b. Après :
- Pour l'administrateur

Hels Association Accueil Membres Tontines Prêts Remboursement Dons Tableau de bord **Statistiques**

Liste des Membres

Nom	Prénom	Rôle	Statut	Actions
Ader	Christopher	Admin	Actif	Modifier Supprimer
Ange	Joyce	Membre	Actif	Modifier Supprimer
Dauda	Adamu	Membre	Actif	Modifier Supprimer
Mefire	Josaphat	Membre	Inactif	Modifier Supprimer
Umaru	Nafi	Membre	Inactif	Modifier Supprimer
Voundi	Armel	Membre	Inactif	Modifier Supprimer

Ajouter un membre

- Pour les membres

127.0.0.1:8000/liste_membres Summarise ☆ ... Chat

Hels Association Accueil Membres Tontines Prêts Remboursement Dons Tableau de bord **Statistiques**

Liste des Membres

Nom	Prénom
Ader	Christopher
Ange	Joyce
Dauda	Adamu
Mefire	Josaphat
Umaru	Nafi
Voundi	Armel

© 2025 Hels Association - Tous droits réservés

2. Modules « Prêts » et « Remboursement »

- a. Avant
- Prêt :

Soumettre une demande de prêt

Montant du prêt

Observations

Choisir un garant -- Sélectionner un membre -- ▼

CNI du garant (image)

Parcourir...
Aucun fichier sélectionné.

Soumettre

- Remboursement :

Accueil Membres Tontines Prêts Remboursement Dons Tableau de bord Statistique

Remboursement partiel effectué. Montant restant : 17000000.00 F.

Montant du prêt : 20000000.00 F

Montant restant à rembourser : 17000000.00 F

Montant à rembourser :

Date de remboursement :

Rembourser

b. Après :

- Prêt :

Soumettre une demande de prêt

Montant du prêt

Observations

RAS

Choisir un garant Voundi Armel ▼

CNI du garant (image)

Parcourir...
667081.jpg

Tontine concernée presence (1000.00 F) ▼

Soumettre

- Remboursement

[Accueil](#) [Membres ▾](#) [Tontines ▾](#) [Prêts ▾](#) [Remboursement ▾](#) [Dons ▾](#) [Tableau de bord](#) [Statistique](#)

Choisir une tontine

-- Toutes les tontines -- ▾

Filtrer

Veuillez sélectionner une tontine pour voir les prêts.

[Accueil](#) [Membres ▾](#) [Tontines ▾](#) [Prêts ▾](#) [Remboursement ▾](#) [Dons ▾](#) [Tableau de bord](#) [Statistique](#)

Choisir une tontine

presence (1000.00 F) ▾

Filtrer

Montant du prêt : 27000.00 F

Tontine : presence

Montant restant à rembourser : 27000.00 F

Montant à rembourser :

10000

Date de remboursement :

11 / 12 / 2025

Rembourser

[Accueil](#) [Membres ▾](#) [Tontines ▾](#) [Prêts ▾](#) [Remboursement ▾](#) [Dons ▾](#) [Tableau de bord](#) [Statistique](#)

Choisir une tontine

-- Toutes les tontines -- ▾

Filtrer

Remboursement de 10000 F enregistré.

Veuillez sélectionner une tontine pour voir les prêts.

3. Module « Statistiques »

a. Avant :

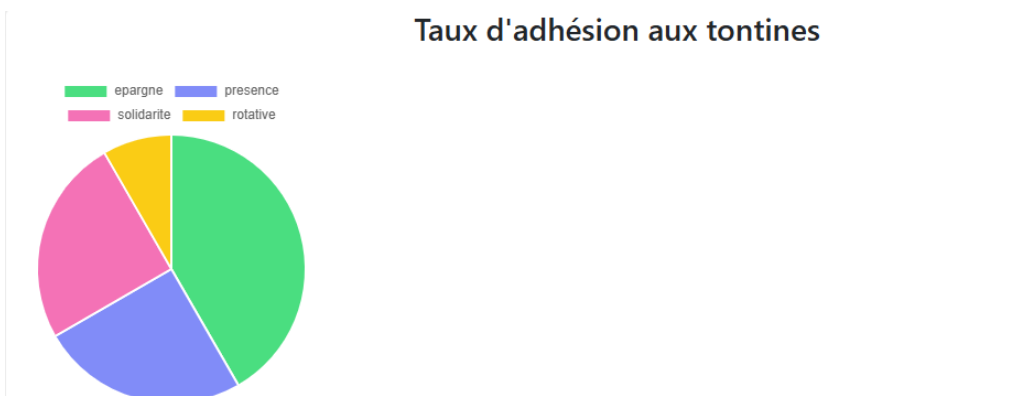
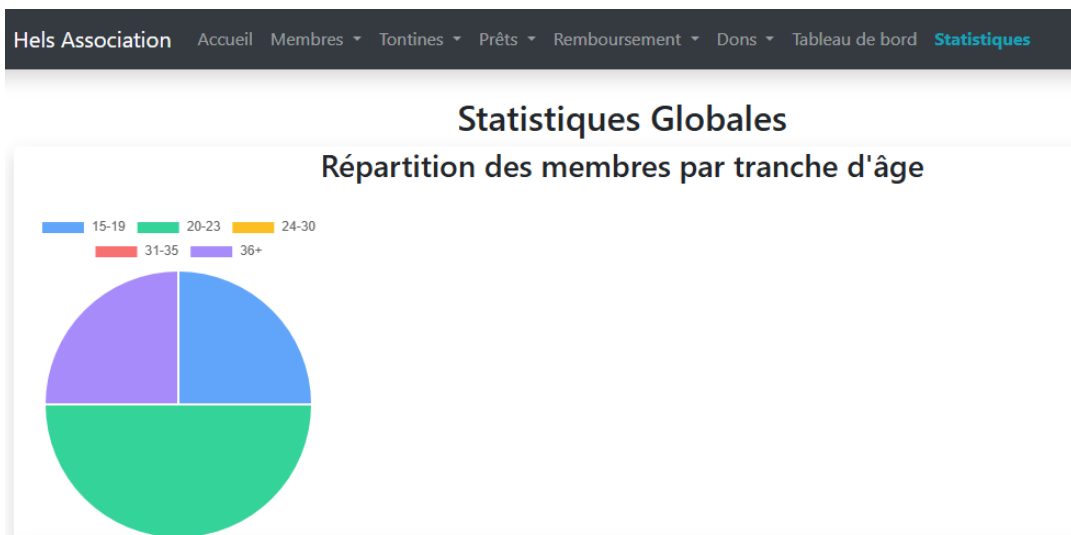
Page not found (404)

Request Method: GET
Request URL: http://127.0.0.1:8000/membres/statistiques_view

Using the URLconf defined in mon_association.urls, Django tried these URL patterns, in this order:

```
1. [name='connexion']
2. inscription [name='inscription']
3. home [name='home']
4. home2 [name='home2']
5. admin_dashboard [name='admin_dashboard']
6. member_dashboard [name='member_dashboard']
7. liste_membres [name='liste_membres']
8. liste_membres2 [name='liste_membres2']
9. ajouter_membre [name='ajouter_membre']
10. ajouter_membre/ [name='ajouter_membre']
11. modifier_membre/<int:idmembre>/ [name='modifier_membre']
12. supprimer_membre/<int:idmembre>/ [name='supprimer_membre']
13. liste_tontine [name='liste_tontine']
14. liste_tontine2 [name='liste_tontine2']
15. ajouter_tontine [name='ajouter_tontine']
16. ajouter_membre_tontine [name='ajouter_membre_tontine']
17. modifier_tontine/<int:idTontines>/ [name='modifier_tontine']
18. supprimer_tontine/<int:idTontines>/ [name='supprimer_tontine']
19. souscrire_tontine/<int:idTontines>/ [name='souscrire_tontine']
20. voir_membres_tontines/<int:idTontines>/ [name='voir_membres_tontines']
21. tableau_de_bord/ [name='tableau_de_bord']
22. tableau_de_bord2/ [name='tableau_de_bord2']
23. membres/ [name='liste_membres']
24. demande_pret/ [name='demande_pret']
25. gestion_demandes/ [name='gestion_demandes']
26. valider_demande/<int:idpret>/ [name='valider_demande']
27. refuser_demande/<int:idpret>/ [name='refuser_demande']
28. liste_prets/ [name='liste_prets']
29. effectuer_remboursement/ [name='effectuer_remboursement']
30. effectuer_remboursement2/ [name='effectuer_remboursement2']
31. effectuer_don [name='effectuer_don']
32. liste_dons_connecte [name='liste_dons_connecte']
33. liste_tous_les_dons [name='liste_tous_les_dons']
```

b. Après :

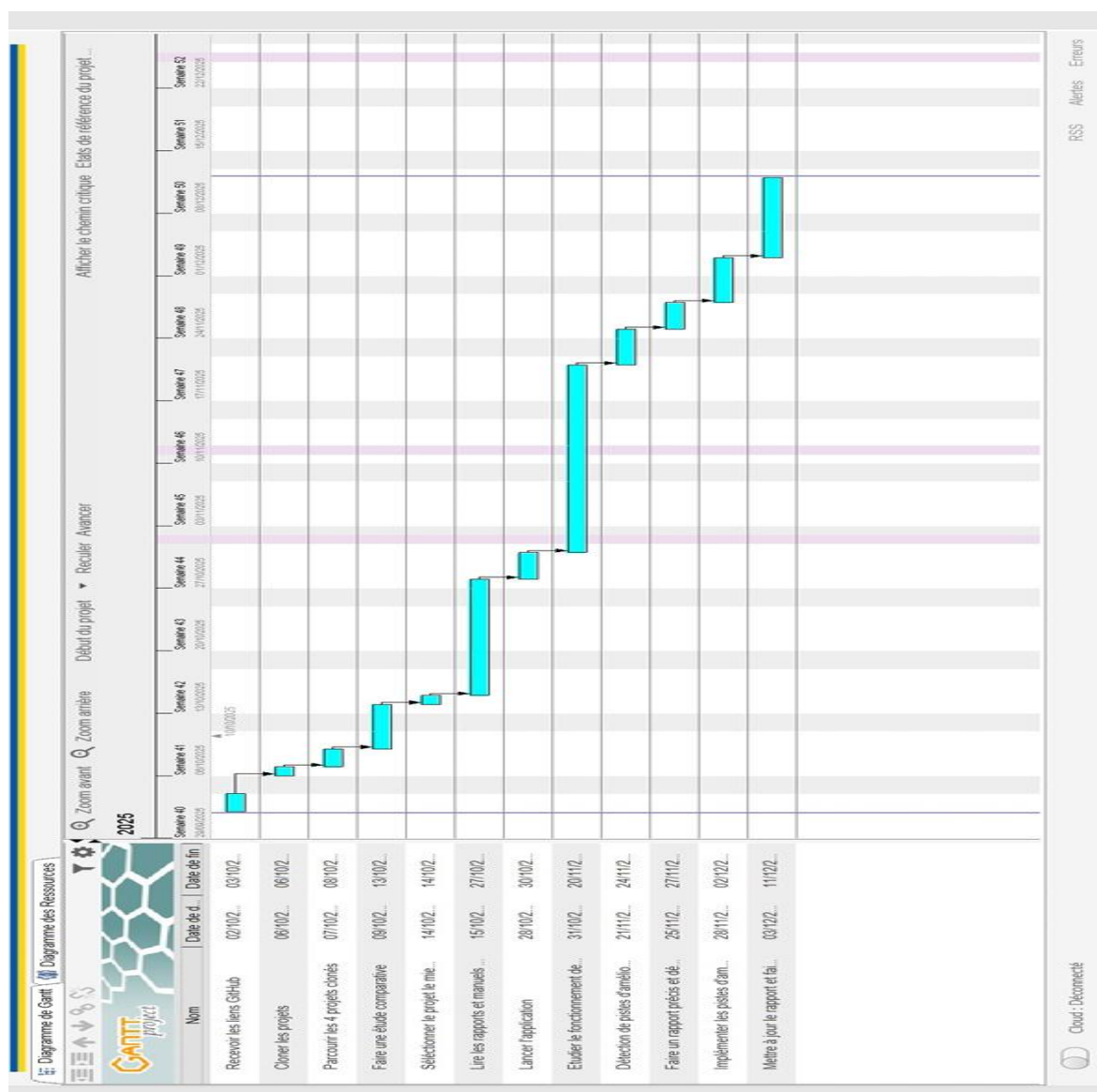


8. DIFFICULTÉS RENCONTRÉES ET SOLUTIONS APPORTÉES

- Certaines machines n'ont pas pu installer l'environnement MySQL. Pour résoudre ce problème nous avons utilisé l'application XAMPP Control Panel qui nous a permis de manipuler les bases de données afin de lancer l'application.
- La procédure d'installation de l'environnement donnée par nos aînés dans leur guide d'utilisation était incomplète du fait de l'absence des instructions de déclaration et activation de l'environnement virtuel nécessaire pour héberger l'application. Pour résoudre ce problème, nous avons effectué des recherches pour trouver ces instructions manquantes qui devaient être exécutées avant les instructions pour la configuration et le lancement de l'application. Ces instructions sont les suivantes :
 - `python -m venv venv`
 - `venv\Scripts\activate`
- La non maîtrise du langage html et la programmation orientée objet ont été un problème majeur dans l'implémentation de nos améliorations. Pour résoudre ce problème, nous avons fait des recherches sur internet et nous avons fait usage de l'intelligence artificielle.

9. DIAGRAMME DE GANTT

Le **Diagramme de Gantt** a servi d'outil central pour la planification, la visualisation des dépendances et le suivi de l'exécution des tâches. Il a permis de définir des étapes intermédiaires claires et mesurables. Le tableau ci-dessous présente les principales tâches, leur durée planifiée et leur état d'avancement au moment de la rédaction de ce rapport.



10. CONCLUSION

Ce rapport marque l'achèvement de notre projet. Il était question pour nous de lancer une application de gestion de tontine, l'analyser et y apporter des améliorations. Pour ce faire nous sommes passés par le clonage des projets logés sur la plateforme GitHub, le choix du projet le plus avancé après une étude comparative des projets, une analyse approfondie de l'application puis la détection et l'implémentations de pistes d'améliorations. Ce travail nous a permis de mettre en pratique les méthodologies de gestions de projet à travers l'élaboration du diagramme de Gantt, ainsi que de renforcer notre capacite à travailler en équipe.