



## Déploiement d'une Application de Gestion des Notes des étudiants

Réalisé par :

ILHAM OUAKRIM

&

KONE MODIBO

Sous l'encadrement :

Mr Maizate

ANNEE ACADEMIQUE

2023-2024

## **Remerciements**

Au terme de ce projet, on tient à exprimer notre profonde gratitude envers deux acteurs cruciaux de cette réussite : le professeur de cloud computing Mr Maizate et l'administration. Le dévouement et le soutien inestimable de chacun ont été les piliers fondamentaux qui ont permis d'atteindre nos objectifs.

Tout d'abord, on souhaite adresser nos sincères remerciements à Mr Maizate pour son engagement exceptionnel. Sa guidance éclairée, son expertise et son dévouement ont été des éléments déterminants dans la réalisation de ce projet. Nous avons grandement bénéficié de sa vision, de sa bonne volonté de partager ses connaissances et de sa passion pour l'éducation.

Un merci particulier également à l'administration pour son soutien constant. Les efforts déployés par l'administration ont été cruciaux durant tout le semestre. Votre engagement envers l'excellence a été une source d'inspiration pour nous tous.

## **Contexte**

Ce projet intitulé **Déploiement d'une Application de Gestion des Notes des étudiants** a été entrepris dans le cadre du cours de cloud computing , supervisé par le professeur Mr Maizate.

Cette initiative a pour objectif de mettre en pratique les concepts enseignés durant le semestre.

Ce projet vise à concevoir, développer et déployer une application de gestion des notes.

L'application permettra aux utilisateurs de créer, consulter, modifier et supprimer des notes de manière sécurisée et efficace en suivant cette architecture :

- Un serveur d'application accessible par une adresse ip publique et une adresse IP privée.
- Un serveur de base de données accessible avec une adresse IP privée uniquement.

**NB :**

**Le prof a également souligné qu'on pouvait ajouter de nouvelles fonctionnalités à l'application tant au niveau développement qu'au niveau déploiement.**

## Résumé

Notre application de gestion des notes intitulé “**myschool**”

**p**ermet non seulement aux utilisateurs de créer, consulter, modifier et supprimer des notes de manière sécurisée et efficace comme demandé mais aussi elle comporte beaucoup d’autres fonctionnalités telles que :

- Développement :

La gestion des profs, la gestion des étudiants, des modules, des matières, des méthodes de paiements, etc.

- Déploiement :

La containerisation de l’appli avec **Docker** , une appli pour le frontend et une pour le backend, la communication avec les APIs, l’intégration et le versionning avec **Git** et **GitHub**, etc.

L’application est vraiment riche en contenu, que nous allons expliquer dans les sections suivantes.

.

# Table des matières

Remerciements .....	2
Contexte .....	3
Résumé .....	4
I. Introduction .....	6
II. Mise en place de l'application .....	6
A. Etude de l'existant.....	6
B. Analyse de l'Existant.....	6
C. Critique de l'Existant .....	7
D. Solutions Proposées.....	7
III. Analyse des besoins.....	7
A. Besoins Fonctionnels.....	7
B. Besoins Non Fonctionnels .....	8
C. Acteurs .....	8
D. Public Visé .....	8
IV. Conception.....	9
A. Modèle conceptuel de données (MCD) .....	9
B. Modèle physique de données (MPD) .....	10
V. Réalisation .....	10
A. Description des fonctionnalités de l'application.....	10
B. Choix de la technologie Django.....	11
C. Choix de la base de données SQLite.....	12
D. Présentation et description de l'interface utilisateur.....	14
E. Difficultés rencontrées .....	16
F. Conclusion .....	17
G. Annexes .....	17

# **I. Introduction**

Dans le domaine de l'éducation, la gestion des notes des étudiants joue un rôle crucial dans le suivi de leur progression académique. Aujourd'hui, alors que les technologies de l'information se développent rapidement, les applications de gestion des notes offrent des solutions numériques pour simplifier et améliorer ce processus vital. Dans le cadre de notre rapport, nous entreprenons une analyse approfondie de l'existant concernant les applications de gestion des notes, dans le but de mieux comprendre leurs fonctionnalités, leurs avantages et leurs limites.

Cette étude vise à éclairer les forces et les faiblesses des applications de gestion des notes actuellement disponibles sur le marché, en mettant l'accent sur les aspects liés au déploiement. En examinant de près ces systèmes, nous serons en mesure d'évaluer leur efficacité, leur convivialité et leur adaptabilité aux besoins changeants des enseignants et des étudiants. De plus, nous chercherons à déterminer les opportunités d'amélioration et à proposer des solutions innovantes pour optimiser l'expérience utilisateur et la performance globale de ces applications.

## **II. Mise en place de l'application**

### **A. Etude de l'existant**

Dans le cadre de notre rapport, une analyse approfondie de l'existant a été entreprise, se concentrant sur les applications de gestion des notes. Cette étude visait à examiner en détail les forces et les faiblesses des applications de gestion des notes actuellement disponibles sur le marché.

### **B. Analyse de l'Existant**

L'analyse de l'existant a permis d'identifier plusieurs points forts des applications de gestion des notes. Ces applications offrent généralement une multitude de fonctionnalités, permettant aux enseignants de saisir, de suivre et de partager les résultats académiques des étudiants. Elles proposent également des interfaces intuitives, des options de personnalisation fournissant ainsi une expérience conviviale et efficace.

Cependant, malgré ces avantages, certaines lacunes ont été identifiées. Certains systèmes peuvent souffrir de complexités dans la saisie des données, entraînant des retards et des erreurs potentielles. De plus, la nécessité d'une meilleure intégration de fonctionnalités de rétroaction automatique pour les étudiants a été soulignée. Enfin, la performance du système pendant les périodes de charge intensive, comme la clôture des semestres, peut être un défi à résoudre.

### **C. Critique de l'Existant**

Notre analyse critique souligne la nécessité d'offrir une expérience utilisateur plus fluide, efficace et interactive dans le domaine des applications de gestion des notes.

### **D. Solutions Proposées**

Dans ce contexte, des solutions sont envisagées pour répondre aux défis identifiés. Il est impératif de simplifier la saisie des notes, d'intégrer des fonctionnalités de rétroaction automatique pour les étudiants et d'optimiser la performance du système lors de périodes de charge élevée. Ces ajustements visent à améliorer l'efficacité globale des applications de gestion des notes, offrant ainsi une meilleure expérience tant pour les enseignants que pour les étudiants.

## **III. Analyse des besoins**

### **A. Besoins Fonctionnels**

- **Gestion des étudiants :** Les utilisateurs doivent pouvoir créer des profils personnels avec des informations telles que le nom, la photo de profil, la biographie, etc.
- **Gestion des profs:** Les utilisateurs doivent être en mesure de publier du contenu sur leur profil et dans les fils d'actualités.
- **Gestions des modules :** Les utilisateurs doivent pouvoir interagir avec les publications en commentant, en aimant et en partageant.
- **Gestion des matières :** Les utilisateurs devraient pouvoir envoyer et accepter des demandes d'amitié, gérer leur liste d'amis, et voir les activités de leurs amis.

## **B. Besoins Non Fonctionnels**

- **Sécurité des Données** : Assurer la sécurité des données des utilisateurs, y compris le chiffrement des données sensibles.
- **Containérisation** :
- **Intégration et le versionning**:
- **Évolutivité** : Concevoir l'application pour qu'elle puisse gérer une croissance éventuelle du nombre d'utilisateurs.
- **Performance** : Assurer des temps de chargement rapides et une expérience utilisateur fluide.
- **Disponibilité** : Garantir une disponibilité élevée de la plateforme, minimisant les temps d'arrêt.

## **C. Acteurs**

- **Etudiants** :
- **Profs**
- 
- 

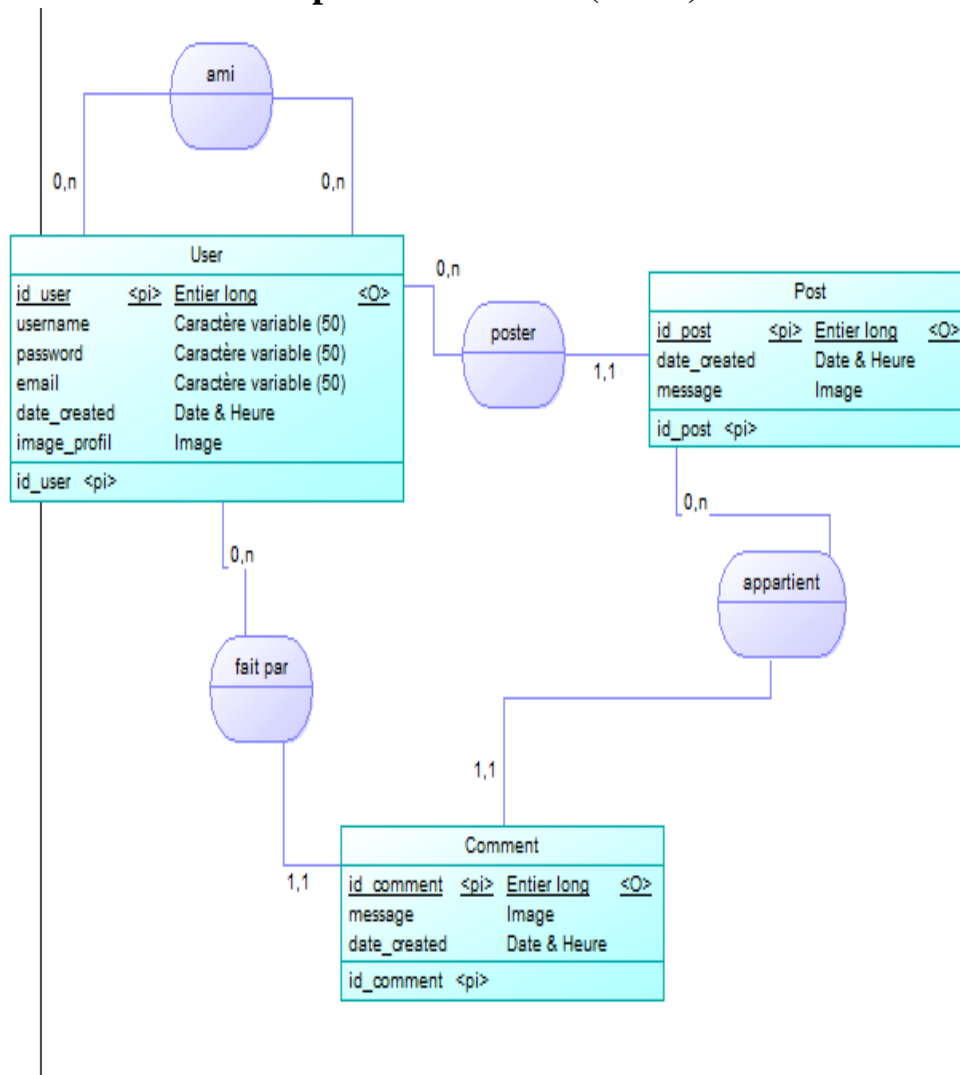
## **D. Public Visé**

Le public visé englobe un large éventail de parties prenantes, toutes impliquées de près ou de loin dans la gestion des notes des étudiants et intéressées par l'efficacité et l'optimisation de ces processus à l'aide de solutions technologiques.

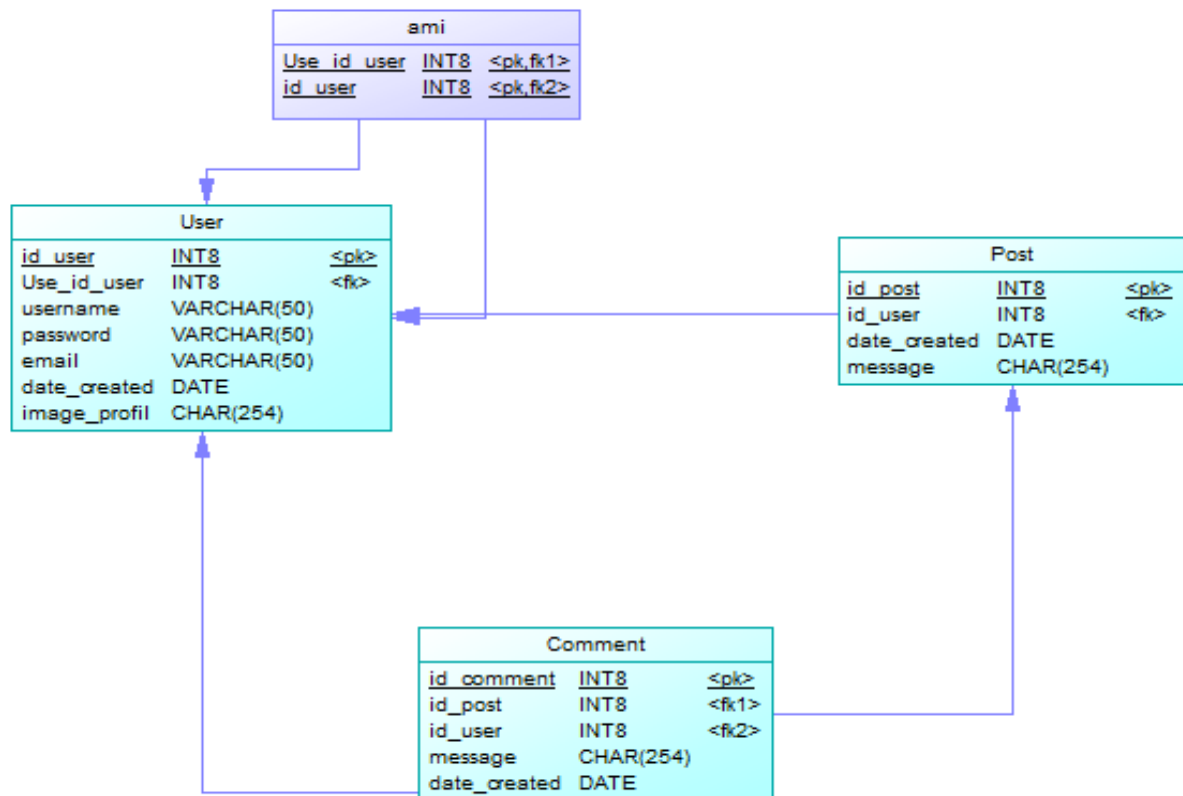


## IV. Conception

### A. Modèle conceptuel de données (MCD)



## B. Modèle physique de données (MPD)



## V. Réalisation

### A. Description des fonctionnalités de l'application

- **Création de Profil Utilisateur :** Les utilisateurs peuvent créer un profil personnel en fournissant des informations telles que leur nom, une photo de profil, une biographie, etc.

- **Fil d'Actualités :** Les utilisateurs ont accès à un fil d'actualités chronologique qui affiche les publications de leurs amis.

**Publication de Contenu :** Les utilisateurs peuvent publier du contenu.

- **Interaction Sociale :** Les utilisateurs peuvent interagir avec les publications en aimant, en commentant et en partageant.

## B. Choix de la technologie Django

**Django** est un Framework de développement web open source, écrit en Python, qui facilite la création d'applications web puissantes et évolutives. Il a été conçu pour aider les développeurs à construire des applications web rapidement et en favorisant une architecture de développement propre et modulaire. Voici une description de Django et de ses principales caractéristiques :

- Productivité élevée : Django offre une productivité élevée grâce à son ensemble complet de fonctionnalités et de bibliothèques intégrées. Il comprend des outils pour la gestion de bases de données, l'authentification utilisateur, la gestion des sessions, la création de formulaires, la gestion des fichiers médias, etc. Cela permet aux développeurs de se concentrer sur la logique métier de leur application plutôt que de réinventer la roue.
- Architecture MVT : Django suit l'architecture MVT (Modèle-Vue-Template) qui est similaire au modèle MVC (Modèle-Vue-Contrôleur). Cette architecture encourage la séparation claire des préoccupations, ce qui rend le code plus propre et maintenable.
  - ✓ Modèle : Représente les données et la logique métier de l'application.
  - ✓ Vue : Gère la logique de présentation et interagit avec le modèle.
  - ✓ Template : Gère le rendu HTML et la présentation des données.
- ORM (Object-Relational Mapping) : Django possède un ORM puissant qui permet aux développeurs de travailler avec la base de données en utilisant des objets Python au lieu de SQL brut. Cela simplifie considérablement la gestion des bases de données et permet de travailler avec différents systèmes de gestion de bases de données.
- Administration automatique : Django génère automatiquement une interface d'administration pour les modèles de base de données que vous définissez. Cela facilite la gestion des données de l'application sans avoir à créer une interface d'administration personnalisée.
- Sécurité intégrée : Django met en œuvre de nombreuses mesures de sécurité par défaut, telles que la protection contre les attaques CSRF, la gestion des sessions sécurisées, l'évasion de modèles automatiques pour prévenir les attaques XSS, etc. Il encourage également le cryptage des mots de passe et la sécurisation des médias téléchargés.

- **Authentification et autorisation** : Django propose un système d'authentification et d'autorisation robuste, permettant de gérer les utilisateurs, les groupes d'utilisateurs et les autorisations d'accès aux différentes parties de l'application.
- **Extensibilité** : Django est extensible grâce à son système de middleware, de gestionnaires de signaux, de plugins et à une vaste communauté de développeurs qui contribuent à des packages tiers.
- **Documentation complète** : Django dispose d'une documentation complète et de qualité qui facilite l'apprentissage et la résolution de problèmes.
- **Large communauté** : Django est pris en charge par une grande communauté de développeurs et dispose de nombreuses ressources en ligne, forums, didacticiels, et packages tiers disponibles pour résoudre presque tous les problèmes.
- **Développement rapide** : Grâce à toutes les fonctionnalités intégrées, Django permet de développer rapidement des applications web de haute qualité.

### C.     **Choix de la base de données SQLite**

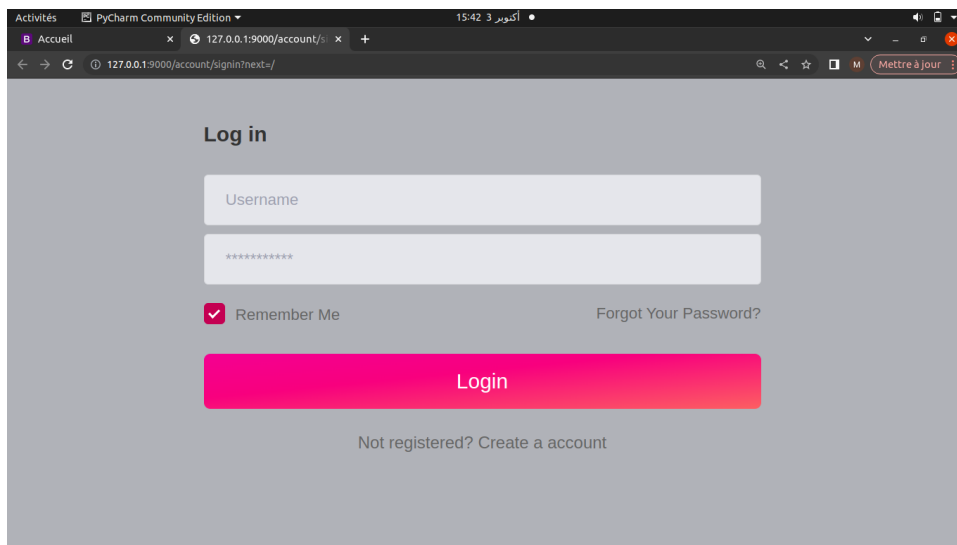
**SQLite** est un système de gestion de base de données relationnelles (SGBDR) open source, léger, autonome et sans serveur. Conçu pour être simple, rapide et efficace, SQLite est largement utilisé dans une variété d'applications, allant des logiciels embarqués aux applications de bureau, en passant par les applications mobiles et les navigateurs web. Voici une description des principales caractéristiques de SQLite :

- **Léger et autonome** : SQLite est une bibliothèque C légère et autonome qui n'a pas besoin d'un serveur de base de données distinct pour fonctionner. Cela signifie qu'il est facile à intégrer dans des applications et ne nécessite pas de configuration complexe.
- **Pas de processus de serveur** : Contrairement à d'autres systèmes de gestion de base de données tels que MySQL ou PostgreSQL, SQLite n'a pas de processus de serveur distinct. Les opérations de base de données sont gérées directement par la bibliothèque SQLite dans le même processus que l'application.
- **Stockage local** : SQLite stocke les données localement dans un fichier unique, ce qui le rend adapté au stockage de petites à moyennes quantités de données. Il est souvent utilisé pour stocker des données d'application locales, telles que les préférences utilisateur, les journaux d'application et les caches.
- **Zéro configuration** : Vous n'avez pas besoin de configurer un serveur ou de gérer des autorisations complexes pour utiliser SQLite. Il suffit de créer un fichier de base de données SQLite et de commencer à y insérer des données.

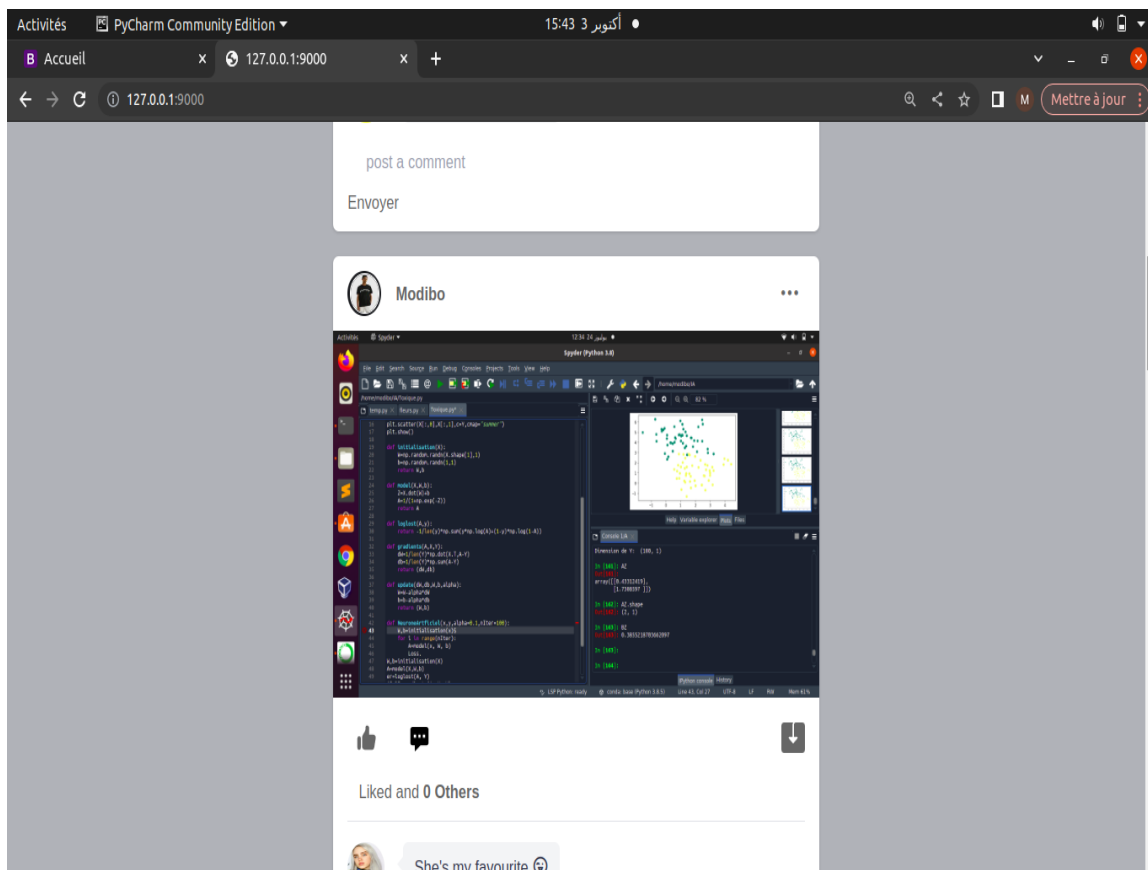
- **Transactions ACID** : SQLite prend en charge les transactions ACID (Atomicité, Cohérence, Isolation, Durabilité), ce qui garantit la fiabilité et la cohérence des données, même en cas de panne du système.
- **Langage SQL complet** : SQLite prend en charge un sous-ensemble complet du langage SQL, ce qui permet d'effectuer des opérations de base de données complexes telles que la création de tables, l'insertion, la mise à jour, la suppression et la sélection de données.
- **Haute performance** : Malgré sa légèreté, SQLite est connu pour ses performances élevées, en particulier dans les cas d'accès concurrents et de lecture intensive.
- **Portabilité** : Les fichiers de base de données SQLite sont généralement compatibles avec différentes plates-formes, ce qui facilite la portabilité des données entre les systèmes.
- **Large adoption** : SQLite est largement adopté et intégré dans de nombreuses applications et bibliothèques logicielles, ce qui en fait un choix populaire pour le stockage de données locales.
- **Licence open source** : SQLite est distribué sous une licence open source permissive (domaine public), ce qui signifie qu'il peut être utilisé gratuitement dans des projets commerciaux et open source sans restriction de licence.

## D. Présentation et description de l'interface utilisateur

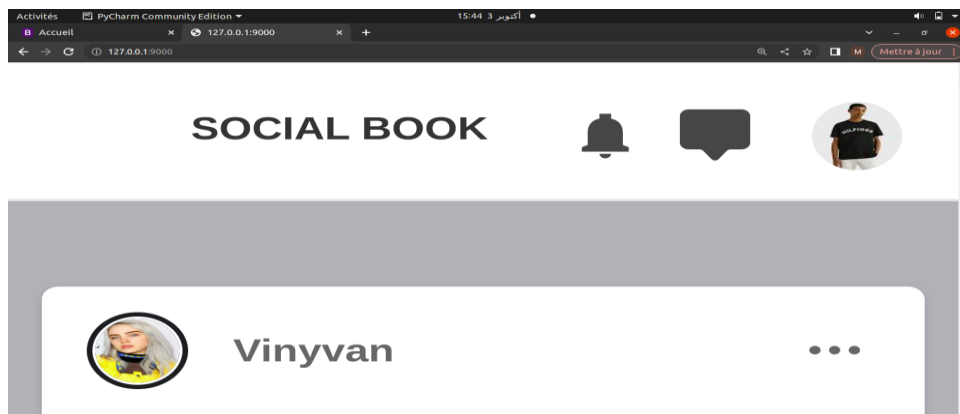
- Page de connexion : Elle permet à un utilisateur de se connecter



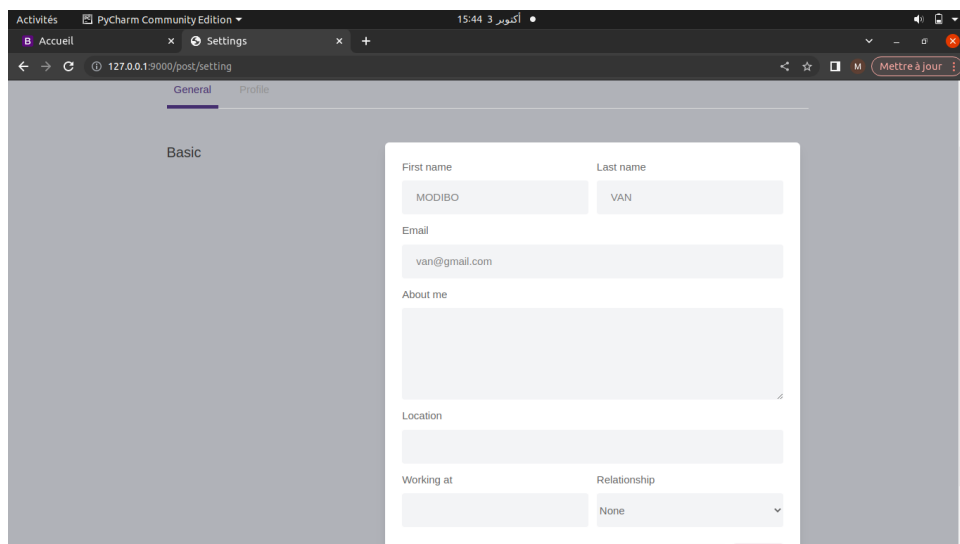
- Page d'accueil : Contient l'ensemble des posts et commentaires



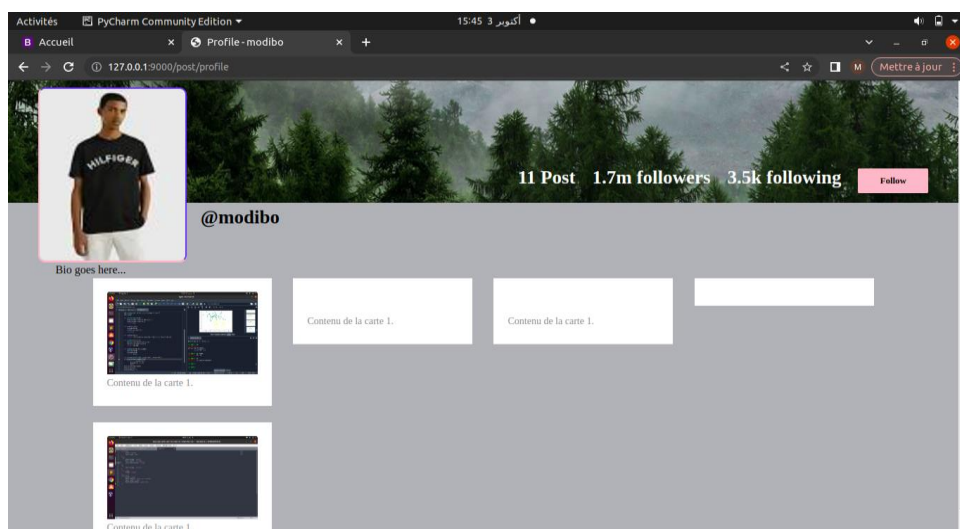
- Section de navigation : Permet de naviguer sur le réseau.



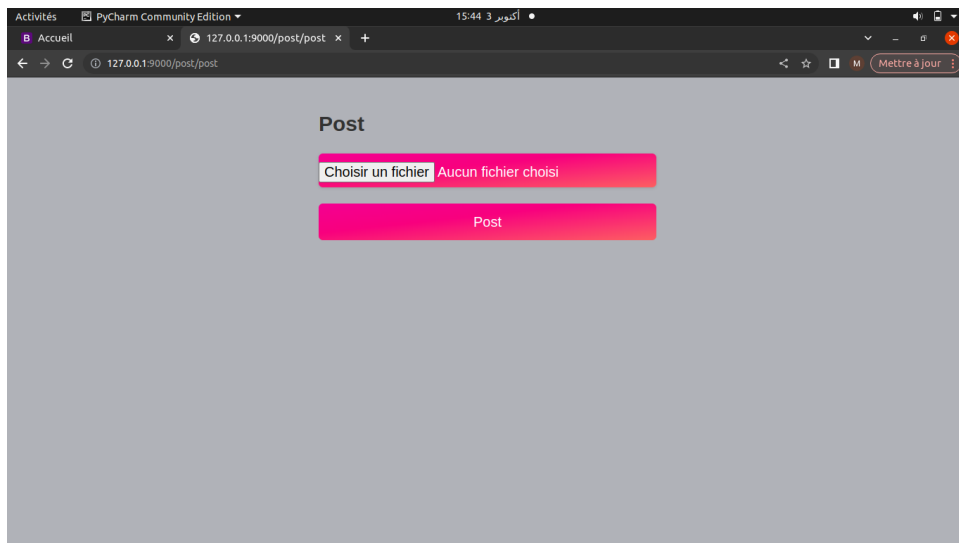
- Settings : Pour modifier les paramètres de l'utilisateur.



- Page de profil : Contient les posts et les informations de l'utilisateur.



- Page de post : Permet à l'utilisateur de faire un post.



## E. Difficultés rencontrées

Lors de la réalisation de ce projet, la difficulté majeure a été la dockerisation (utilisation des conteneurs de docker) du projet ; finalement le problème a été résolu après plusieurs recherches.



## **F. Conclusion**

La réalisation de ce projet m'a permis d'approfondir mes connaissances en développement web avec Django, de déployer des applications en utilisant les conteneurs de Docker, de versionner le projet avec git et de collaborer sur github.

J'ai appris beaucoup d'autres concepts que ça soit dans la conception d'une application que dans sa réalisation.

Ce stage a été une expérience vraiment enrichissante.

## **G. Annexes**

<https://docs.djangoproject.com>

[GitHub: Let's build from here · GitHub](#)

[Docker: Accelerated Container Application Development](#)