

## **Projet Tutoré Django: Développer un réseau social**

**Membres du groupe:**

- Boris Hermann NANA
- Wendyida Alphonse Tanguy DJIGMA

**Génie logiciel L2 / Université virtuelle du Burkina Faso/ P22**

## **Documentation du Projet de réseau social**

### **1. Introduction**

Ce projet est un réseau social développé en Java. Il permet aux utilisateurs de créer un compte, se connecter, publier du contenu, recevoir des likes sur leurs publications et échanger des messages avec d'autres utilisateurs.

### **2. Objectifs**

- Développer une application simple de réseau social.
- Permettre aux utilisateurs de s'inscrire, se connecter et interagir avec d'autres.
- Créer une base de données pour gérer les utilisateurs, publications et conversations.
- Fournir une interface utilisateur intuitive et facile à utiliser.

### **3. Fonctionnalités Principales**

#### **3.1. Gestion des Utilisateurs**

- Inscription : Les utilisateurs peuvent créer un compte avec un nom d'utilisateur, un mot de passe et d'autres informations personnelles.
- Connexion : Les utilisateurs peuvent se connecter en utilisant leurs identifiants.
- Déconnexion : Les utilisateurs peuvent se déconnecter de leur compte.
- Gestion du Profil : Les utilisateurs peuvent mettre à jour leurs informations personnelles (nom, photo de profil, etc.).

#### **3.2. Publications**

- Créer une Publication : Les utilisateurs peuvent écrire du contenu et le publier sur leur profil.
- Afficher les Publications : Les publications des utilisateurs sont visibles sur le fil d'actualité.
- Recevoir des Likes : Les autres utilisateurs peuvent aimer les publications.

#### **3.3. Interactions**

- Commenter : Les utilisateurs peuvent commenter les publications.
- Réagir avec des Likes : Les utilisateurs peuvent liker les publications des autres.

#### **3.4. Messagerie**

- Envoyer des Messages : Les utilisateurs peuvent échanger des messages textuels avec d'autres utilisateurs.
- Boîte de Réception : Interface pour voir les messages reçus et envoyés.
- Notifications : Recevoir des notifications lors de la réception de nouveaux messages.

## 4. Technologies Utilisées

### 4.1. Langage

- Java

### 4.2. Frameworks et Bibliothèques

- Spring Boot : Pour simplifier le développement de l'application et gérer la logique de backend.
- Spring Data JPA : Pour l'accès et la gestion de la base de données.
- Thymeleaf : Pour le rendu des templates côté serveur.
- MySQL : Pour la gestion de la base de données relationnelle.

### 4.3. Gestion des Dépendances

- Maven : Pour gérer les dépendances et les configurations du projet.

## 5. Architecture du Système

L'application suit une architecture classique MVC (Modèle-Vue-Contrôleur), permettant de séparer la logique de traitement des données, la logique de présentation et les données elles-mêmes.

### 5.1. Diagramme UML Simplifié

#### 1. Modèle Utilisateur

- User : Contient des informations sur les utilisateurs (ID, nom, email, mot de passe).

#### 2. Modèle Publication

- Post : Contient le contenu des publications, le nombre de likes, la date de création.

#### 3. Modèle Message

- Message : Contient l'expéditeur, le destinataire et le contenu des messages.

## 6. Configuration de la Base de Données

La base de données stocke toutes les informations nécessaires pour faire fonctionner l'application, y compris les utilisateurs, les publications et les messages.

### **Schéma de Table**

- Users Table
  - id (Primary Key)
  - username
  - password
  - email
  - profile\_picture
- Posts Table
  - id (Primary Key)
  - content
  - likes\_count
  - created\_at
  - user\_id (Foreign Key)
- Messages Table
  - id (Primary Key)
  - sender\_id (Foreign Key)
  - receiver\_id (Foreign Key)
  - content
  - sent\_at

## **8. Conclusion**

Ce projet de réseau social simple fournit une base solide pour implémenter les fonctionnalités essentielles d'un réseau social. Il peut être étendu pour inclure des fonctionnalités supplémentaires comme les groupes, la gestion des amis, les notifications en temps réel et bien plus encore.