


Réseau Littéraire pour Ateliers d'Écriture


 **Projet en groupe – 6 étudiants** (projet créatif et riche en modélisation)

Démarrage

URL : https://github.com/esperluet/esme_webservice_full

Clonez le **squelette Dockerisé** fourni par l'enseignant. Il contient :

- Backend Flask (PostgreSQL, Blueprints, SQLAlchemy)
- Frontend React préconfiguré (Vite ou CRA)
- Fichier `docker-compose.yml` pour le lancement local

 **Le projet doit être construit à partir de ce squelette uniquement.**

Objectif du projet

Créer une plateforme communautaire permettant aux utilisateurs de :

- Publier des extraits de leurs écrits (type atelier d'écriture)
 - Recevoir des commentaires et likes d'autres membres
 - Suivre leur propre activité littéraire dans une interface personnelle
-

Contexte fonctionnel

Les utilisateurs aiment partager leurs écrits : poèmes, récits, scènes inspirées d'un livre...
Cette plateforme leur permet de :

- Publier un texte (avec titre, contenu, genre, livre inspirant facultatif)
- Lire les écrits d'autres personnes et les commenter ou liker
- Voir leur propre historique de publications et de réactions

Exemple d'interaction :

- Mehdi publie un extrait de science-fiction inspiré de Dune
 - Clara commente son texte et clique sur "like"
 - Mehdi retrouve toutes ses publications sur sa page personnelle
-

Fonctionnalités à implémenter

Backend (Flask)

- Création de texte : `title`, `content`, `genre`, `book_id` (optionnel)
- Chaque utilisateur peut publier 1 à 2 textes par semaine (limite)
- Système de like + commentaire sur les textes
- Récupération des textes par popularité ou genre
- Historique d'activité d'un utilisateur (publications, likes, commentaires)

Frontend (React)

- Page "Publier un texte" (formulaire)
- Page "Explorer" avec filtres (genre, popularité)
- Affichage des commentaires et likes d'un texte
- Page "Mon espace" : mes textes, mes likes, mes commentaires

- Boutons d'interaction : like, commenter, supprimer son propre texte

Sécurité attendue

- Authentification JWT obligatoire
 - Seul l'auteur d'un texte peut le modifier ou le supprimer
 - Un utilisateur ne peut publier que dans la limite hebdomadaire fixée (backend)
-

Modèles à utiliser

Fournis

- `User`
- `Book`

À créer

- `TextPost : id, title, content, genre, book_id, user_id, created_at`
 - `Comment : id, text_id, user_id, content, created_at`
 - `Like : id, text_id, user_id`
-

Livrables attendus

1. Repository GitHub (obligatoire)

- Dossier `frontend/`, `backend/`, `docker-compose.yml`, `.env.example`

- README avec :
 - Instructions `docker-compose up`
 - Comptes utilisateur test
 - Routes principales API

2. Rapport PDF (2 à 4 pages)

Structure attendue :

- Membres du groupe + rôles
- Fonctionnalités réalisées avec captures
- Diagramme simplifié des modèles
- Cas d'usage : publication, commentaire, like, limitation
- Problèmes techniques rencontrés et solutions

3. Bonus (facultatifs)

- Affichage des tendances de publication par semaine
- Page "Texte aléatoire à découvrir"
- Indicateur de progression (nb max de textes publiables encore cette semaine)



Ce que ce projet vous fera pratiquer

Compétence

Mise en œuvre



Publication de contenu

Formulaires, validation, stockage de texte



Interaction sociale

Likes, commentaires, filtrage



Limitation d'usage

Règle hebdomadaire imposée backend



Front créatif

Filtres, classements, animations UX

Endpoints recommandés (exemples)

- `POST /api/texts` : publier un texte
- `GET /api/texts?genre=x` : explorer les textes
- `POST /api/texts/<id>/like` : liker un texte
- `POST /api/texts/<id>/comment` : commenter un texte
- `GET /api/users/me/posts` : voir ses propres textes et interactions