

Titre du Projet: Développement du Frontend d'un Clone de LinkedIn

Membres :: Mvo'o Meyo Denis Valentin Herve

Liens vers le github : <https://github.com/Denis237HM/Projet-4DWR-denis>

Objectif: Développer un frontend dynamique et réactif pour un clone de LinkedIn, interagissant parfaitement avec un backend conçu en architecture de microservices. Le projet utilise ReactJS, Redux, React Router et Axios, en mettant l'accent sur les pratiques de développement modernes et les principes de conception.

Référentiel Backend: [4AMS LinkedIn Clone Backend](#)

Structure du Projet

- **Page d'Accueil:** Liste des publications avec le nom de l'auteur, le titre et un extrait du contenu. Permet la navigation vers les détails de la publication et le profil de l'utilisateur.
 - **Page de Publication:** Affiche les détails complets de la publication, les informations de l'auteur et les commentaires. Inclut la navigation vers le profil de l'auteur.
 - **Page de Nouvelle Publication:** Permet aux utilisateurs de créer de nouvelles publications.
 - **Page de Profil Utilisateur:** Affiche les informations de l'utilisateur, y compris l'éducation, les compétences, l'expérience et les connexions. Fournit la fonctionnalité de modification du profil.
-

Résolutions Étape par Étape

1. Configuration et Initialisation

- Cloner le référentiel backend et s'assurer qu'il fonctionne avec Docker Compose.
- Initialiser le projet frontend avec Create React App.
- Installer les dépendances nécessaires :

```
bash
Copier le code
npx create-react-app linkedin-clone-frontend
cd linkedin-clone-frontend
npm install redux react-redux react-router-dom axios
tailwindcss ant-design
```

2. Gestion de l'État

- Implémenter Redux pour la gestion de l'état.
- Créer un store Redux et configurer les réducteurs et actions pour gérer les données des utilisateurs, des publications, des profils et des commentaires.
- Configurer React Router pour la navigation de l'application.

3. Routage

- Configurer React Router pour gérer les différentes routes : Accueil, Publication, Nouvelle Publication, Profil Utilisateur.
- S'assurer que chaque page est navigable via des URLs, permettant ainsi le bookmarking.
- 4. **Intégration de l'API**
 - Utiliser Axios pour les requêtes HTTP vers les services backend.
 - Implémenter les opérations CRUD pour les utilisateurs, les publications, les profils et les commentaires.
 - Gérer les erreurs et fournir un retour d'information à l'utilisateur pendant le chargement des données.
- 5. **Conception UI/UX**
 - Utiliser TailwindCSS et Ant Design pour le style et améliorer l'expérience utilisateur.
 - Suivre les principes de conception modernes pour garantir une interface utilisateur cohérente et intuitive.
- 6. **Développement des Composants de Page**
 - **Page d'Accueil :**
 - Récupérer et afficher une liste de publications.
 - Inclure des liens de navigation vers les détails des publications et les profils des utilisateurs.
 - **Page de Publication :**
 - Afficher le contenu complet de la publication, les détails de l'auteur et les commentaires.
 - Permettre la navigation vers le profil de l'auteur.
 - **Page de Nouvelle Publication :**
 - Fournir un formulaire pour créer de nouvelles publications.
 - Gérer la soumission du formulaire et l'envoi des données au backend.
 - **Page de Profil Utilisateur :**
 - Afficher les informations détaillées de l'utilisateur.
 - Inclure la fonctionnalité de modification des détails du profil.

Pour démarrer le projet faire ceci :

```
1. npm install ( pour installer les dépendances )
2. npm start ( pour lancer l'application )
```
