

Projet de Générateur de CV (Travail en Groupe)

Contexte

Après un cours de 35 heures couvrant Node.js, Express, MongoDB et React, vous allez maintenant travailler en groupe de **2 ou 3** pour développer un projet complet. Ce projet vous permettra de démontrer votre compréhension des technologies étudiées et de collaborer efficacement en équipe.

Objectif

Développer une application de **Générateur de CV** où les utilisateurs peuvent s'inscrire, créer et gérer leur CV, le rendre visible ou non, et consulter les CV d'autres utilisateurs. L'application doit également permettre aux utilisateurs authentifiés de laisser des recommandations sur les CV.

Fonctionnalités principales

1. Authentification des utilisateurs

- Les utilisateurs doivent pouvoir créer un compte, se connecter, et accéder aux fonctionnalités liées à la création et gestion de leur CV.
- Implémentez l'authentification via **JWT (JSON Web Token)**.

2. Création et gestion de CV

- Un utilisateur connecté doit pouvoir :
 - Créer et modifier son CV, incluant les champs : nom, prénom, description, expériences pédagogiques (Diplomes, certifications, formations, ...), expériences professionnelles.
 - Choisir de rendre son CV visible ou non pour les autres utilisateurs.

3. Consultation des CV

- Créez une page permettant de consulter un CV **visibles**
- Créez une page listant tous les CV **visibles**. Cette page doit inclure une fonctionnalité de **recherche** par nom et prénom.
- Un utilisateur authentifié pourra voir les détails des CV visibles.

4. Recommandations

- Un utilisateur connecté doit pouvoir laisser une **recommandation** sur le CV d'un autre utilisateur.
- Les recommandations seront visibles pour tous les utilisateurs.

5. Déploiement

- Le projet devra être **déployé** en ligne, à la fois pour le back-end et le front-end

6. Documentation

- Documentez l'API pour que chaque endpoint soit clairement défini.
- Incluez un fichier **README.md** avec les instructions pour lancer le projet en local.
- Si vous utilisez des variables d'environnement, fournissez un fichier `.env.local.example` avec les exemples de configurations nécessaires.

Contraintes techniques

- **Back-end** : Utilisez **Node.js** avec **Express** pour créer une API REST.
 - Gestion des données via **MongoDB**.
 - endpoints API pour gérer les utilisateurs, CV et recommandations.
- **Front-end** : Utilisez **React** pour construire l'interface utilisateur.
 - Gestion des états via **React Hooks**.
 - Interface permettant la création, la modification, et la consultation des CV.
- **Authentification** : Implémentez un système de connexion sécurisé avec **JWT**.
- **Déploiement** : Assurez-vous que le front-end et le back-end sont déployés sur des plateformes accessibles.

Organisation du projet

- Travaillez en groupe de **2 à 3** personnes.
- Utilisez un **système de gestion de versions (Git)** et créez un **repository GitHub/Gitlab** partagé pour faciliter la collaboration.
- Répartissez les tâches équitablement entre les membres de l'équipe. Assurez-vous que chacun contribue au développement du projet.

Critères d'évaluation

1. **Fonctionnalité** : Le projet inclut toutes les fonctionnalités demandées.
2. **Collaboration** : Chaque membre du groupe a contribué de manière équilibrée.
3. **Qualité du code** : Le code est propre, lisible et bien organisé.
4. **Sécurité** : L'authentification et les autorisations sont correctement implémentées.
5. **Expérience utilisateur** : L'application est intuitive et agréable à utiliser.
6. **Déploiement** : Le projet est accessible via un lien déployé.
7. **Documentation** : L'API et le projet sont bien documentés.

Détails de soumission

- **Déployez** l'application front-end et back-end et fournissez les liens.
- Soumettez les liens des repositories **GitHub/GitLab** (back-end et front-end) en précisant votre groupe à l'adresse :
levy.zire@intervenants.efrei.net
- Chaque groupe devra inclure les noms des membres dans le **README**.

Date limite

- **Vendredi 18 octobre à 00h00.**

Bon courage à tous et bonne collaboration !