

DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ Maghzaoui
Nom d'usage ▶ Mohamed
Prénom ▶ Mohamed
Adresse ▶ 31 Boulevard Vauban Guyancourt

Titre professionnel visé

Concepteur développeur d'applications

MODALITÉ D'ACCÈS :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.
Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.
Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel (DP)** dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.

 <http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Intitulé de l'activité-type n° 1 : Développer une application sécurisée	p.	5
▶ Intitulé de l'exemple n° 1 Développer une authentification sécurisée pour les utilisateurs	p....	5
▶	p....	
▶	p....	
Intitulé de l'activité-type n° 2 : Concevoir et développer une application sécurisée organisée en couches	p.	8
▶ Intitulé de l'exemple n° 1 Création d'un système de gestion des CV	p....	8
▶	p....	
▶	p....	
Intitulé de l'activité-type n° 3 : Préparer le déploiement d'une application sécurisée	p.	11
▶ Intitulé de l'exemple n° 1 Déploiement et mise en production d'une application	p....	11
▶	p....	
▶	p....	
Intitulé de l'activité-type n° 4	p.	
▶ Intitulé de l'exemple n° 1	p....	
▶ Intitulé de l'exemple n° 2	p....	
▶ Intitulé de l'exemple n° 3.....	p....	
Titres, diplômes, CQP, attestations de formation (facultatif)	p.	14
Déclaration sur l'honneur	p.	15
Documents illustrant la pratique professionnelle (facultatif)	p.	16
Annexes (Si le RC le prévoit)	p.	17

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1 Développer une application sécurisée

Exemple n°1 ► Développer une authentification sécurisée pour les utilisateurs

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Objectif : Dans le cadre de ma troisième année à l'École Hexagone, j'ai choisi de réaliser le projet **ReussirSonCv**, une application web répondant à une problématique que j'ai moi-même rencontrée lors de la création de mon CV. Comme beaucoup de candidats, j'ai constaté que les outils classiques (Word, Canva, PowerPoint) ne permettent pas toujours d'obtenir un document clair, professionnel et compatible avec les systèmes de tri automatisés (ATS) utilisés par les recruteurs. En me plaçant à la fois comme développeur et comme utilisateur, j'ai conçu cette solution pour faciliter la création de CV structurés, générer des rendus professionnels, intégrer une analyse de contenu via l'IA Gemini pour optimiser la pertinence des informations, et permettre l'export en PDF. L'une des composantes essentielles du projet est la mise en place d'un système d'authentification sécurisé, garantissant la protection des données et l'accès personnalisé aux fonctionnalités de gestion et d'optimisation des CV.

1. Installer et configurer son environnement de travail en fonction du projet

Pour démarrer le projet **RéussirSonCV**, j'ai configuré un environnement complet permettant de développer, tester et faire communiquer le frontend et le backend.

- **Backend :** installation de Django et Django REST Framework pour gérer les endpoints de l'API, configuration des dépendances pour la gestion des sessions et l'envoi d'emails, mise en place des variables d'environnement pour les connexions à la base de données et à l'API SMTP.
- **Frontend :** installation de React.js, Tailwind CSS et DaisyUI via Vite pour créer une interface moderne et réactive, configuration d'Axios pour gérer les requêtes vers le backend, mise en place de la navigation avec React Router, de la gestion des états pour les formulaires avec Context API, et de l'optimisation des données avec React Query.
- **Sécurité et gestion des sessions :** mise en place de la protection CSRF côté backend, configuration du CORS pour contrôler les origines autorisées, gestion sécurisée des cookies côté frontend, et mise en place d'un système de limitation des requêtes (rate limiting) pour prévenir les abus et renforcer la sécurité des endpoints.
- **Organisation du projet :** structure des dossiers pour séparer clairement composants métier, composants d'interface et services utilitaires, facilitant la maintenance et l'évolution de l'application.

2. Développer des interfaces utilisateur

J'ai conçu et développé les interfaces permettant aux utilisateurs de s'inscrire, se connecter et modifier leur mot de passe.

- **Formulaire d'inscription :** champs pour email, prénom, nom, mot de passe, date de naissance et photo de profil. Validation dynamique pour vérifier les informations saisies et afficher des messages d'erreur

précis.

- **Formulaire de connexion** : saisie des identifiants avec messages dynamiques ; erreur si le compte est inactif ou identifiants incorrects, confirmation visuelle et textuelle en cas de succès.
- **Formulaire de changement de mot de passe** : vérification du mot de passe actuel avant la saisie du nouveau mot de passe, affichage clair des messages de réussite ou d'erreur
- **Activation par email** : interface guidant l'utilisateur pour activer son compte et redirection vers la page de login après activation réussie.
- **Ergonomie et réactivité** : interfaces adaptées aux PC, tablettes et mobiles, avec un design clair et accessible.

3. Développer des composants métier

J'ai mis en place la logique métier côté backend pour gérer l'authentification et la gestion des utilisateurs.

- **Endpoints backend** : création des routes pour l'inscription, la connexion, la déconnexion, le changement de mot de passe et l'activation par email.
- **Validation des données** : vérification de la conformité des identifiants et messages d'erreur adaptés.
- **Gestion des sessions** : maintien de l'état de connexion et contrôle des accès aux pages et fonctionnalités sensibles.
- **Hashage des mots de passe** : sécurisation des données utilisateurs et respect des bonnes pratiques.
- **Activation par email** : génération de liens d'activation uniques et gestion des cas d'erreur (lien expiré ou invalide).

4. Contribuer à la gestion d'un projet informatique

Même en projet individuel, j'ai structuré le travail pour respecter les bonnes pratiques de gestion informatique.

- **Organisation et suivi des tâches** : planification des étapes de développement (installation, création des endpoints, développement des formulaires, validation et ajustements).
- **Versioning et contrôle du code** : utilisation de Git et GitHub pour suivre l'évolution du projet, créer des commits réguliers et documenter les modifications.
- **Tests d'intégration** : vérification que les utilisateurs non connectés ne peuvent pas accéder aux pages protégées et que les messages dynamiques sont correctement affichés.
- **Révisions et ajustements** : amélioration continue de l'interface et des composants métier après tests, pour garantir une expérience utilisateur fluide et cohérente.

2. Précisez les moyens utilisés :

- **Django REST Framework** : gestion des endpoints sécurisés, des sessions et des permissions côté backend

DOSSIER PROFESSIONNEL (DP)

- **React.js** : création des formulaires de connexion, d'inscription et de changement de mot de passe.
- **Tailwind CSS** : stylisation moderne et responsive des interfaces utilisateur.
- **DaisyUI** : composants prêts à l'emploi pour accélérer la conception des interfaces.
- **CSRF et hashage** : protection des données avec hashage des mots de passe et gestion des sessions côté serveur.
- **Axios** : envoi sécurisé des requêtes entre le frontend et le backend.
- **Brevo** : envoi des e-mails d'activation pour la création de compte.
- **Trello** : organisation et suivi des tâches du projet.
- **Git et GitHub** : gestion du code source, versioning et collaboration.

3. Avec qui avez-vous travaillé ?

Projet réalisé individuellement

4. Contexte

Nom de l'entreprise, organisme ou association ► *École Hexagone*

Chantier, atelier, service ► *Développement de l'application web **RéussirSonCV**, permettant la création et la génération de CV personnalisés*

Période d'exercice ► Du : 16/04/2025 au : 20/08/2025

5. Informations complémentaires (facultatif)

- **Code source complet du projet** : <https://github.com/mohamedmaghzaoui/ReussirSonCV>
- **Application en ligne (hébergeur)** : <https://reussirsoncv.netlify.app/>
- **Documentation utilisateur** : github.com/mohamedmaghzaoui/ReussirSonCV/blob/main/docs/Documentation%20utilisateur.pdf
- **Documentation technique** : github.com/mohamedmaghzaoui/ReussirSonCV/blob/main/docs/Documentation%20technique.pdf

Activité-type

2

Concevoir et développer une application sécurisée organisée en couches

Exemple n° 1 ► Création d'un système de gestion des CV

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Objectif : En prenant appui sur le même projet **ReussirSonCv** développé lors de ma troisième année à l'École Hexagone, l'objectif de cette activité était de concevoir un module complet de gestion des CV, fonctionnalité clé de l'application. Cette partie s'inscrit dans la continuité du travail d'authentification déjà mis en place et permet aux utilisateurs, une fois connectés, de créer, organiser et personnaliser leurs CV (sections expériences, formations, compétences, projets, langues et informations personnelles), de les visualiser en temps réel, de les exporter en PDF et d'en analyser la qualité grâce à l'intégration d'une IA (Gemini) pour optimiser leur compatibilité avec les outils de recrutement (ATS).

1. Analyser les besoins et maquetter une application

L'analyse a permis de définir les fonctionnalités essentielles pour gérer les CV : création de plusieurs CV par utilisateur, structuration en sections, prévisualisation en temps réel, personnalisation du thème et export PDF. J'ai réalisé des maquettes des pages principales : tableau de bord listant les CV, éditeur de CV avec formulaires dynamiques, page d'analyse automatique via IA et aperçu final PDF. Ces maquettes ont servi à identifier les composants réutilisables, anticiper les interactions utilisateur et faciliter la communication avec le backend.

2. Définir l'architecture logicielle d'une application

L'architecture a été pensée en couches :

Frontend :

- Utilisation de React.js avec Vite pour une interface rapide et réactive.
- Gestion des routes avec React Router, état global via Context API et synchronisation des données avec React Query.
- Composants réutilisables : Navbar, Footer, PrivateRoute, formulaires dynamiques pour chaque section du CV.
- Gestion des appels API avec Axios configuré pour les cookies et CSRF.

Backend :

- Django + Django REST Framework, structuré en apps pour utilisateurs (user/) et CV (resume/).
- Endpoints REST sécurisés pour CRUD sur CV et sections, protection avec permissions IsAuthenticated.
- Intégration de Gemini AI pour analyser automatiquement le contenu des CV et fournir un rapport détaillé.

4. Concevoir et mettre en place une base de données relationnelle

- Base relationnelle MySQL hébergée sur Aiven, avec haute disponibilité et scalabilité.
- Tables principales : user, cv, profile, experience, education, project, skill, language, personal_info.
- Relations et intégrité :
 - User → CV : 1 à N
 - CV → Profile / PersonalInfo : 1 à 1
 - CV → Expérience / Formation / Projet / Compétence / Langue : 1 à N
- Clés étrangères assurant que chaque sous-élément appartient à un CV existant.
- Normalisation pour éviter la redondance et permettre des opérations CRUD cohérentes via Django ORM.

5. Développer des composants d'accès aux données SQL et NoSQL

- Création de **ViewSet** et **Serializers** pour chaque modèle : CV, expériences, formations, compétences, projets, langues, informations personnelles.
- Mise en place d'**endpoints sécurisés** pour les opérations CRUD :
 - /api/cvs/ pour lister et créer des CV
 - /api/cvs/:id/ pour consulter, modifier ou supprimer un CV
 - /api/experiences/, /api/educations/, /api/projects/, /api/skills/, /api/languages/, /api/personal-infos/ pour gérer les sous-modèles
- Gestion des **relations entre tables** via Django ORM :
 - un utilisateur → plusieurs CV
 - chaque CV → profil + informations personnelles
 - chaque CV → plusieurs expériences, formations, projets, compétences et langues
- Intégration de l'**IA Gemini** via /api/analyse-cv/ :
 - analyse structure, mots-clés, compatibilité ATS, fautes
 - génération d'un **rapport détaillé** consultable depuis le frontend

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Moyens utilisés :

- **Aiven** pour l'hébergement de la base de données.
- **React-to-print** pour l'export PDF.
- **Gemini (IA)** pour l'analyse et l'optimisation des CV.
- **MySQL Workbench** : gestion visuelle et administration de la base de données.
- **Insomnia** : tests et validation des API REST.
- **Figma** : conception des maquettes et prototypage des interfaces.

3. Avec qui avez-vous travaillé ?

Projet réalisé individuellement

4. Contexte

Nom de l'entreprise, organisme ou association ► *École Hexagone*

Chantier, atelier, service ► *Développement de l'application web **RéussirSonCV**, permettant la création et la génération de CV personnalisés*

Période d'exercice ► Du : *16/04/2025* au : *20/08/2025*

5. Informations complémentaires (facultatif)

- **Code source complet du projet** : <https://github.com/mohamedmaghzaoui/ReussirSonCV>
- **Application en ligne (hébergeur)** : <https://reussirsoncv.netlify.app/>
- **Documentation utilisateur** :
github.com/mohamedmaghzaoui/ReussirSonCV/blob/main/docs/Documentation%20utilisateur.pdf
- **Documentation technique** :
github.com/mohamedmaghzaoui/ReussirSonCV/blob/main/docs/Documentation%20technique.pdf

Activité-type 3 Préparer le déploiement d'une application sécurisée

Exemple n° 1 ► Déploiement et mise en production d'une application

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Objectif : En prenant appui sur le projet ReussirSonCv, assurer le déploiement de l'application afin qu'elle soit accessible à tous les utilisateurs de manière fiable et sécurisée. Cela inclut la mise en production du système de gestion et d'analyse des CV, la disponibilité continue des fonctionnalités principales (création, modification, visualisation et export des CV), ainsi que la protection des comptes et de l'intégrité des données, pour offrir une expérience complète et professionnelle à chaque utilisateur

1. Préparer et exécuter les plans de tests d'une application

Backend – Tests d'intégration :

- Vérification complète des fonctionnalités utilisateur : création de compte (register), connexion (login), déconnexion (logout), modification du profil et changement de mot de passe.
- Tests CRUD sur les CV et leurs sous-éléments : création, lecture, mise à jour et suppression des CV, expériences, formations, projets, compétences, langues et sections personnelles.
- Contrôle des permissions pour s'assurer que chaque utilisateur ne peut accéder ou modifier que ses propres CV et sous-éléments.

Frontend – Tests fonctionnels :

- Vérification que les pages protégées ne sont accessibles qu'aux utilisateurs authentifiés via le composant PrivateRoute.
- Validation des interactions principales : navigation fluide entre le Dashboard, l'éditeur de CV et les pages statiques.

2. Préparer et documenter le déploiement d'une application

- **Frontend** : déployé sur Netlify à partir de la branche principale main, avec la commande de build `npm run build` et le répertoire de publication `build`. Configuration des paramètres pour assurer un rendu optimal et une mise à jour automatique lors des nouvelles versions.
- **Backend** : déployé sur Render, entièrement containerisé via Docker pour garantir un environnement stable et identique à la production locale. La build inclut l'installation des dépendances (`pip install -r requirements.txt`) et les migrations de la base de données (`python manage.py migrate`), tandis que le démarrage utilise Gunicorn (`gunicorn reussirsoncv.wsgi:application`) pour un serveur WSGI performant.
- **Sécurité et configuration** : création et gestion des fichiers `.env` pour stocker les variables sensibles (clés secrètes, configurations CSRF et CORS, identifiants de base de données) et assurer la sécurité des échanges entre frontend et backend.

- **Documentation** : rédaction détaillée du processus de build, des commandes Docker et des configurations nécessaires pour reproduire l'environnement de production à tout moment, facilitant la maintenance et le déploiement continu.

3. Contribuer à la mise en production dans une démarche DevOps

- **Pipeline CI via GitHub Actions** : mise en place d'un flux automatisé pour le frontend et le backend afin d'assurer la fiabilité et la cohérence du code avant tout déploiement.
 - **Backend** : installation automatique des dépendances Python, configuration du fichier .env avec les variables sensibles, exécution des migrations de la base de données et lancement des tests d'intégration pour valider les fonctionnalités principales (authentification, CRUD sur les CV et leurs sections).
 - **Frontend** : installation des dépendances Node.js, exécution des tests pour vérifier le rendu et la navigation entre les pages, génération du build de l'application React.
- **Docker et Docker Compose** : configuration pour gérer le développement local et la production.
 - Montage du code en volume pour un développement interactif.
 - Exposition du port 8000 pour le backend afin de permettre les tests et l'accès aux API.
 - Commande par défaut pour appliquer les migrations et démarrer Gunicorn automatiquement, garantissant un environnement identique entre dev et prod.
 - Containerisation complète pour maintenir la cohérence des environnements, simplifier la maintenance et éviter les conflits de dépendances.
- **Surveillance et contrôle post-déploiement** : suivi des logs backend et frontend pour détecter les erreurs, vérification du bon fonctionnement des pages protégées, des fonctionnalités CRUD CV et de l'analyse IA, et validation que les utilisateurs peuvent accéder à toutes les fonctionnalités principales de l'application de manière stable et sécurisée.

2. Précisez les moyens utilisés :

- **Render** : hébergement du backend Django containerisé pour un déploiement stable et scalable.
- **Netlify** : hébergement du frontend React pour un accès rapide et global à l'application.
- **Docker** : containerisation du backend pour garantir un environnement identique entre développement et production.
- **Docker Compose** : orchestration locale et gestion des services pour dev et tests.
- **GitHub Actions** : pipeline CI/CD pour automatiser l'installation des dépendances, l'exécution

DOSSIER PROFESSIONNEL (DP)

des tests, la migration de la base et le build/déploiement du frontend et backend.

- **.env et variables sensibles** : gestion sécurisée des clés, CSRF, CORS et configurations spécifiques pour production.

3. Avec qui avez-vous travaillé ?

Projet réalisé individuellement

4. Contexte

Nom de l'entreprise, organisme ou association ► *École Hexagone*

Chantier, atelier, service ► *Développement de l'application web **RéussirSonCV**, permettant la création et la génération de CV personnalisés*

Période d'exercice ► Du : *16/04/2025* au : *20/08/2025*

5. Informations complémentaires (facultatif)

- **Code source complet du projet** : <https://github.com/mohamedmaghzaoui/ReussirSonCV>
- **Application en ligne (hébergeur)** : <https://reussirsoncv.netlify.app/>
- **Documentation utilisateur** :
github.com/mohamedmaghzaoui/ReussirSonCV/blob/main/docs/Documentation%20utilisateur.pdf
- **Documentation technique** :
github.com/mohamedmaghzaoui/ReussirSonCV/blob/main/docs/Documentation%20technique.pdf

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

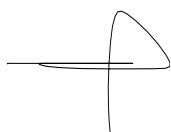
Intitulé	Autorité ou organisme	Date
Bachelor informatique	École Hexagone	2025
Stagiaire Développeur Web	Mona	2024
Stagiaire Développeur Full Stack	Sharing Technologies	2023
Baccalauréat Mathématique	Lycée Abouloubaba	2022

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] Mohamed Maghzaoui,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis
l'auteur(e) des réalisations jointes.

Fait à Guyancourt, le 23/08/2025
pour faire valoir ce que de droit.

Signature :



Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

ANNEXES

Exp1 : Développer une authentification sécurisée pour les utilisateurs

Connexion

Veuillez remplir les informations pour se connecter.

Email ou mot de passe incorrect.

Données personnelles

Email

mohamedmaghzaoui@gmail.com

Mot de passe

Fermer

Se connecter

Vous n'avez pas un compte ? [Inscrire](#)

Inscription

Veuillez remplir les informations pour créer votre compte.

Données personnelles

Prénom

Nom

Email

Mot de passe

Répéter votre mot de passe

jj/mm/aaaa

Ajoutez votre photo de profil

[Cliquez pour téléverser](#)

Fermer


Créer un compte

Formulaire de connexion avec error

Formulaire d'inscription

ReussirSonCV

Jean



Jean Batiste

Prénom

Jean

cancel Savegarder

Nom

Batiste

Email

mohamedmaghzaoui@gmail.com

Date de naissance

18/01/2005

[+ Supprimer ce compte]

Modifier votre mot de passe

Ancien mot de passe

Entrez votre ancien mot de passe

Nouveau mot de passe

Entrez votre nouveau mot de passe

Confirmer le nouveau mot de passe

Confirmez votre nouveau mot de passe

Modifier le mot de passe

Page Profile

ANNEXES

Exp1 : Développer une authentification sécurisée pour les utilisateurs

```
# CORS / CSRF
CORS_ALLOWED_ORIGINS = os.getenv('CORS_ALLOWED_ORIGINS', '').split(',')

CSRF_TRUSTED_ORIGINS = os.getenv('CSRF_TRUSTED_ORIGINS', '').split(',')
CORS_ALLOW_CREDENTIALS = True

SESSION_COOKIE_SAMESITE = "None"
CSRF_COOKIE_SAMESITE = "None"
SESSION_COOKIE_SECURE = True
CSRF_COOKIE_SECURE = True
```

Configuration backend Django de cors et csrf

```
# REST FRAMEWORK
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': [
        'rest_framework.authentication.SessionAuthentication',
    ],
    'DEFAULT_PERMISSION_CLASSES': [
        'rest_framework.permissions.IsAuthenticated',
    ],
    "DEFAULT_THROTTLE_CLASSES": [
        "rest_framework.throttling.UserRateThrottle",
        "rest_framework.throttling.AnonRateThrottle",
    ],
    "DEFAULT_THROTTLE_RATES": {
        "user": "5000/day",
        "anon": "100/hour",
    }
}
```

Configuration sécurisée des sessions et limitation des requêtes

```
import { useEffect } from "react"; 4.1k (gzipped: 1.8k)
import axios from "axios"; 63.1k (gzipped: 23.3k)
const apiUrl = import.meta.env.VITE_API_URL;
export const useCSRF = () => {
  axios.defaults.withCredentials = true;
  useEffect(() => {
    const fetchCSRFToken = async () => {
      try {
        const res = await axios.get(`${apiUrl}/csrf/`, {
          withCredentials: true,
        });

        axios.defaults.headers.common["X-CSRFToken"] = res.data.csrfToken;
      } catch (error) {
        console.error("Erreur lors de la récupération du CSRF token", error);
      }
    };

    fetchCSRFToken();
  }, []);
};
```

Configuration et activation de la protection CSRF côté client

ANNEXES

Exp1 : Développer une authentification sécurisée pour les utilisateurs

```
@api_view(['POST'])
@permission_classes([AllowAny])
def register_view(request):
    serializer = UserSerializer(data=request.data)
    if serializer.is_valid():
        user = serializer.save(is_active=False) # in case problem with serializer set user to not active
        send_verification_email(user, request)

        return Response({'message': 'Utilisateur créé. Vérifie ton email pour l'activer.'}, status=201)
    return Response(serializer.errors, status=400)

#login
@api_view(['POST'])
@permission_classes([AllowAny])
def login_view(request):
    email = request.data.get('email')
    password = request.data.get('password')

    try:
        user = User.objects.get(email=email)
    except User.DoesNotExist:
        return Response({'error': 'invalid'}, status=401)

    if not user.check_password(password):
        return Response({'error': 'invalid'}, status=401)

    if not user.is_active:
        return Response({'error': 'unverified'}, status=401)

    login(request, user)
    return Response({'message': 'Logged in successfully'})
```

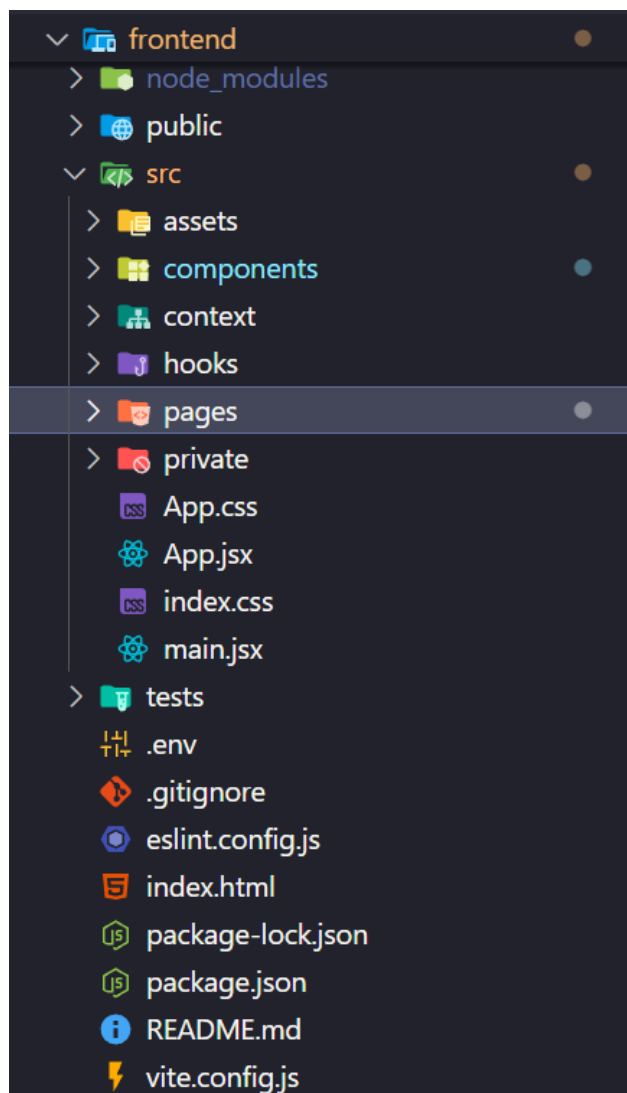
Endpoints d'authentification et d'inscription

```
const handleSubmit = async (e) => {
    e.preventDefault();
    if (!formData.email || !formData.password) {
        setError("Veuillez remplir tous les champs.");
        return;
    }
    setIsSubmitting(true);
    setError("");
    try {
        const response = await axios.post(`${apiUrl}/login/`, formData);
        const csrfRes = await axios.get(`${apiUrl}/csrf/`, {
            withCredentials: true,
        });
        axios.defaults.headers.common["X-CSRFToken"] = csrfRes.data.csrfToken;
        refetch();
        setOpenLogin(false);
    } catch (err) {
        console.error("Login error:", err);
        const errorType = err.response?.data?.error;
        if (errorType === "unverified") {
            setError("Votre compte n'est pas vérifié.");
        } else {
            setError("Email ou mot de passe incorrect.");
        }
    } finally {
        setIsSubmitting(false);
    }
};
```

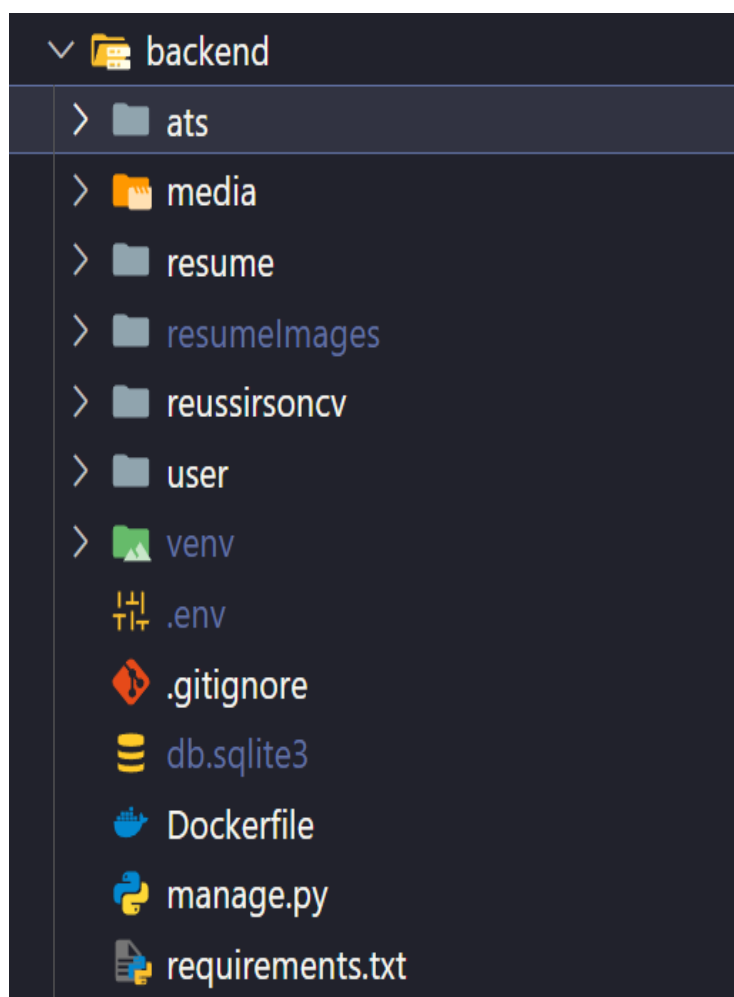
Authentification côté client

ANNEXES

Exp1 : Développer une authentification sécurisée pour les utilisateurs



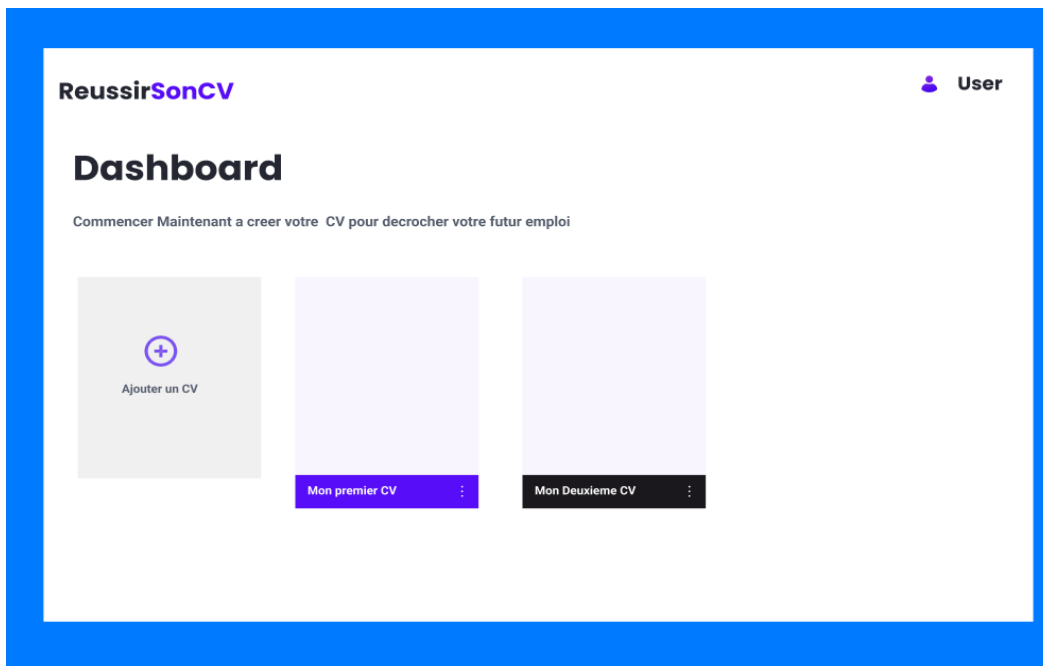
Authentification côté client



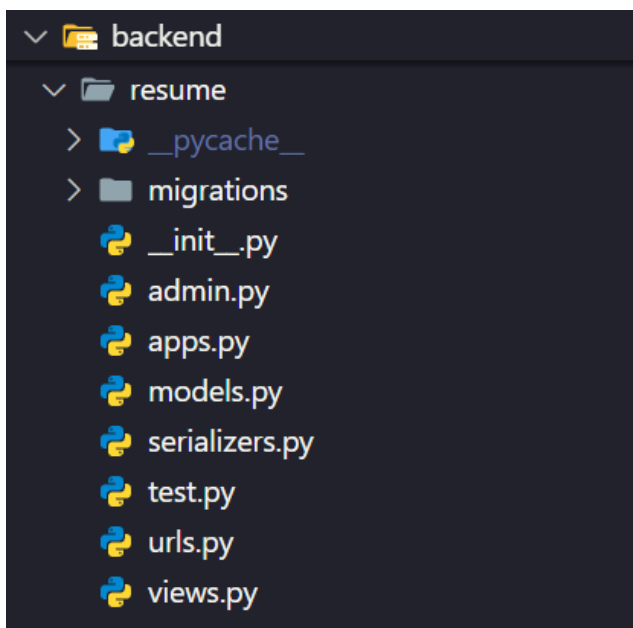
Arborescence backend

ANNEXES

Exp2 : Création d'un système de gestion des CV



Maquettes du tableau de bord des CV



Structure de l'application resume (backend Django)

ANNEXES

Exp2 : Création d'un système de gestion des CV

```
from rest_framework.routers import DefaultRouter
from .views import (
    CVViewSet,
    ExperienceViewSet,
    EducationViewSet,
    ProjectViewSet,
    LanguageViewSet,
    SkillViewSet,
    PersonalInfoViewSet,
    ProfileViewSet
)

router = DefaultRouter()
router.register('cvs', CVViewSet, basename='cv')
router.register('experiences', ExperienceViewSet, basename='experience')
router.register('educations', EducationViewSet, basename='education')
router.register('projects', ProjectViewSet, basename='project')
router.register('languages', LanguageViewSet, basename='language')
router.register('skills', SkillViewSet, basename='skill')
router.register('personal-infos', PersonalInfoViewSet, basename='personalInfo')
router.register('profiles', ProfileViewSet, basename='profile')

urlpatterns = router.urls
```

Architecture et structure des endpoints
API pour la gestion des CV

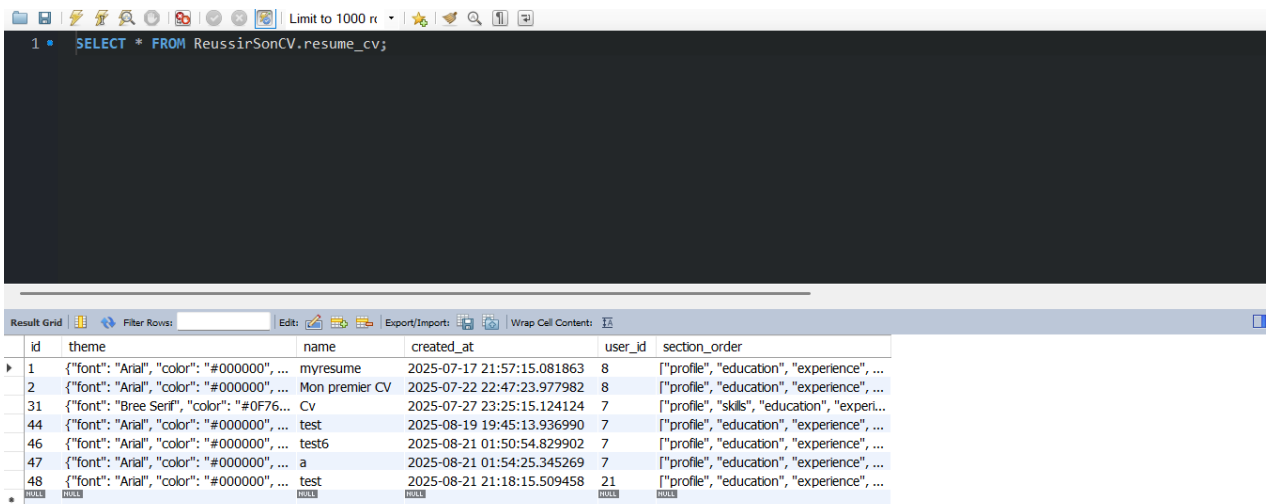
```
function App() {
  useCSRFS();
  return (
    <div className="flex flex-col min-h-screen">
      <Navbar />
      <main className="flex-grow">
        <Routes>
          <Route path="/" element={<Home />} />
          <Route
            path="/dashboard"
            element={<PrivateRoute element={<Dashboard />} />}
          />
          <Route
            path="/dashboard/edit/:id"
            element={<PrivateRoute element={<ResumeEditor />} />}
          />
          <Route
            path="/dashboard/analyse/:id"
            element={<PrivateRoute element={<ResumeAnalyser />} />}
          />
          <Route
            path="/profile"
            element={<PrivateRoute element={<Profile />} />}
          />
          <Route path="/mentions-legales" element={<MentionsLegales />} />
          <Route
            path="/politique-confidentialite"
            element={<PolitiqueConfidentialite />}
          />
          <Route path="/cgu" element={<CGU />} />
          <Route path="/FAQ" element={<FAQ />} />
          <Route path="*" element={<NotFound />} />
        </Routes>
      </main>
      <Footer />
    </div>
  );
}
```

Architecture et organisation du
frontend

DOSSIER PROFESSIONNEL (DP)

ANNEXES

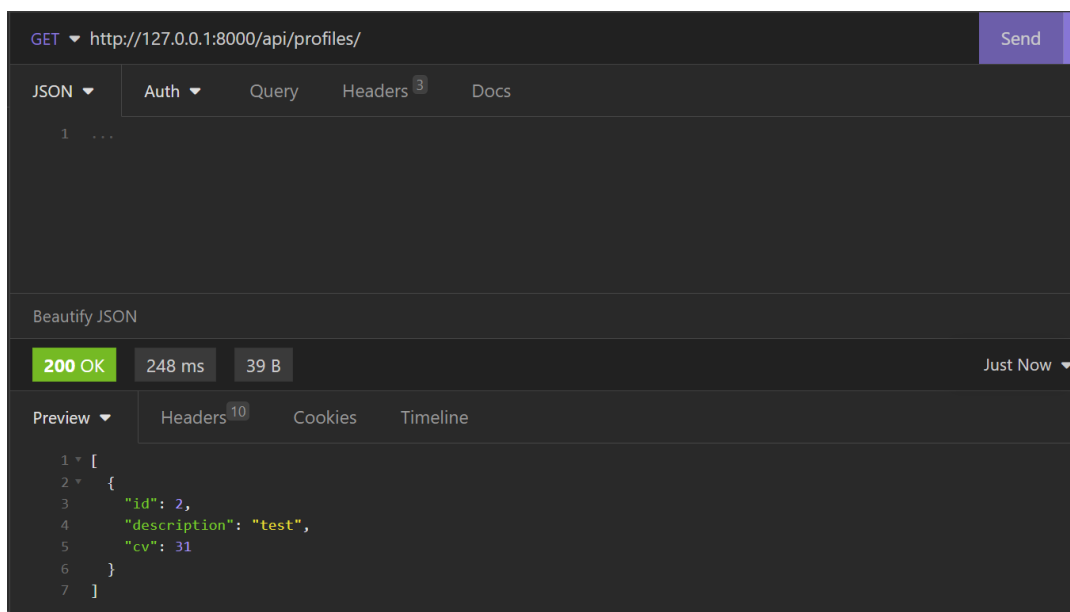
Exp2 : Création d'un système de gestion des CV



```
1 • SELECT * FROM ReussirSonCV.resume_cv;
```

id	theme	name	created_at	user_id	section_order
1	{ "font": "Arial", "color": "#000000", ...	myresume	2025-07-17 21:57:15.081863	8	["profile", "education", "experience", ...
2	{ "font": "Arial", "color": "#000000", ...	Mon premier CV	2025-07-22 22:47:23.977982	8	["profile", "education", "experience", ...
31	{ "font": "Bree Serif", "color": "#0F76...	Cv	2025-07-27 23:25:15.124124	7	["profile", "skills", "education", "exper...
44	{ "font": "Arial", "color": "#000000", ...	test	2025-08-19 19:45:13.936990	7	["profile", "education", "experience", ...
46	{ "font": "Arial", "color": "#000000", ...	test6	2025-08-21 01:50:54.829902	7	["profile", "education", "experience", ...
47	{ "font": "Arial", "color": "#000000", ...	a	2025-08-21 01:54:25.345269	7	["profile", "education", "experience", ...
48	{ "font": "Arial", "color": "#000000", ...	test	2025-08-21 21:18:15.509458	21	["profile", "education", "experience", ...

Affichage de la table principale des CV



GET http://127.0.0.1:8000/api/profiles/ Send

JSON Auth Query Headers 3 Docs

1 ...

Beautify JSON

200 OK 248 ms 39 B Just Now

Preview Headers 10 Cookies Timeline

```
1 [
2 {
3   "id": 2,
4   "description": "test",
5   "cv": 31
6 }
7 ]
```

Exemple d'appel API avec Insomnia

DOSSIER PROFESSIONNEL (DP)

ANNEXES

Exp2 : Création d'un système de gestion des CV

Informations personnelles

Adresse email

Dupont.email@domaine.com

Nom

Dupont

Prénom

Claire

Âge

20

Numéro de téléphone

0612345678


Titre du CV

Chef de projet digital

Accueil

Suivant →

Formulaire CV : informations personnelles



Claire Dupont

Développeur Web Full-Stack

0612345678

Dupont.email@domaine.com

20 ans

LinkedIn

Site web

Portfolio

14 rue Jean-Jacques Rousseau, Montreuil, France

Profil

Développeur web passionné avec 3 ans d'expérience en React et Django, spécialisé dans la création d'interfaces modernes et performantes.

Compétences

HTML

Css

JS

Php

Java

React

Docker

tailwind

python

django

flask

bootstrap

chat gpt

MongoDB

PyTorch

TensorFlow

Redux

Expériences professionnelles

Développeur Front-End

août 2025 - Présent

Agence Web Paris

Paris, France

Développement d'interfaces utilisateur avec React, intégration d'API REST et optimisation des performances.

Stagiaire Développeur Full-Stack

décembre 2024 - août 2025

Tech Solutions

Paris, Ile de france

Participation à la création d'une application de gestion de projets avec Django et React.

Formation

École 92

décembre 2022 - Présent

Bachelor Informatique

Montreuil, France

Projets

Portfolio Personnel

août 2025 - Présent

Site web en Next.js présentant mes projets et expériences.

Application Météo Connectée

août 2025 - Présent

Développement d'une station météo IoT avec ESP32, Django et React pour visualiser les données en temps réel.

Langues

Français : Langue maternelle.

Anglais : Bilingue.

Page 24

DOSSIER PROFESSIONNEL-Version du 11/09/2017

Aperçu d'un CV complété

ANNEXES

Exp3 : Déploiement et mise en production d'une application

```
def test_register_login_logout(self):
    # Register
    response = self.client.post(self.register_url, self.user_data, format='json')
    self.assertEqual(response.status_code, status.HTTP_201_CREATED)

    # Login immediately without activation
    response = self.client.post(self.login_url, self.user_data, format='json')
    self.assertEqual(response.status_code, status.HTTP_200_OK)

    # Logout
    self.client.post(self.logout_url)

def test_user_crud(self):
    # Register
    self.client.post(self.register_url, self.user_data, format='json')
    # Login immediately without activation
    response = self.client.post(self.login_url, self.user_data, format='json')

    # GET user info
    response = self.client.get(self.user_url)
    self.assertEqual(response.status_code, status.HTTP_200_OK)
    self.assertEqual(response.data["email"], self.user_data["email"])

    # Update user info
    update_data = {"birthdate": "2000-01-01"}
    response = self.client.put(self.user_url, update_data, format='json')
    self.assertEqual(response.status_code, status.HTTP_200_OK)
```

Exp de test d'intégration des fonctionnalités utilisateur : inscription, authentification et déconnexion

```
HP@LAPTOP-M610JM08 MINGW64 ~/Desktop/ReussirSonCV (main)
$ cd backend

HP@LAPTOP-M610JM08 MINGW64 ~/Desktop/ReussirSonCV/backend (main)
$ python manage.py test
Found 11 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 11 tests in 32.182s

OK
Destroying test database for alias 'default'...
```

Exécution des tests backend

ANNEXES

Exp3 : Déploiement et mise en production d'une application

```
version: '3.9'

services:
  web:
    build:
      context: ./backend
      dockerfile: Dockerfile
    container_name: reussirsoncv_django
    volumes:
      - ./backend:/app
    ports:
      - "8000:8000"
    env_file:
      - ./backend/.env
    # Override command for local dev
    command: sh -c "python manage.py migrate && gunicorn reussirsoncv.wsgi:application --bind 0.0.0.0:8000"
```

Docker Compose – Backend Django

```
FROM python:3.11-slim

# Install system dependencies
RUN apt-get update && apt-get install -y \
    default-libmysqlclient-dev \
    build-essential \
    gcc \
    libssl-dev \
    pkg-config \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*

WORKDIR /app

# Copy requirements and install
COPY requirements.txt /app/
RUN pip install --upgrade pip \
    && pip install --no-cache-dir -r requirements.txt

# Copy the rest of the code
COPY . /app

# Default command for production (can be overridden in docker-compose for dev)
CMD ["sh", "-c", "python manage.py migrate && gunicorn reussirsoncv.wsgi:application --bind 0.0.0.0:8000 --workers 3"]
```

Fichier Dockerfile – Containerisation du backend Django

ANNEXES

Exp3 : Déploiement et mise en production d'une application

```
frontend:
  name: Frontend React
  runs-on: ubuntu-latest
  defaults:
    run:
      working-directory: frontend
  steps:
    - uses: actions/checkout@v3
    - uses: actions/setup-node@v3
      with:
        node-version: 20

    - name: Install dependencies
      run: npm ci
      working-directory: ./frontend

    - name: Run Tests
      run: npm run test
      working-directory: ./frontend

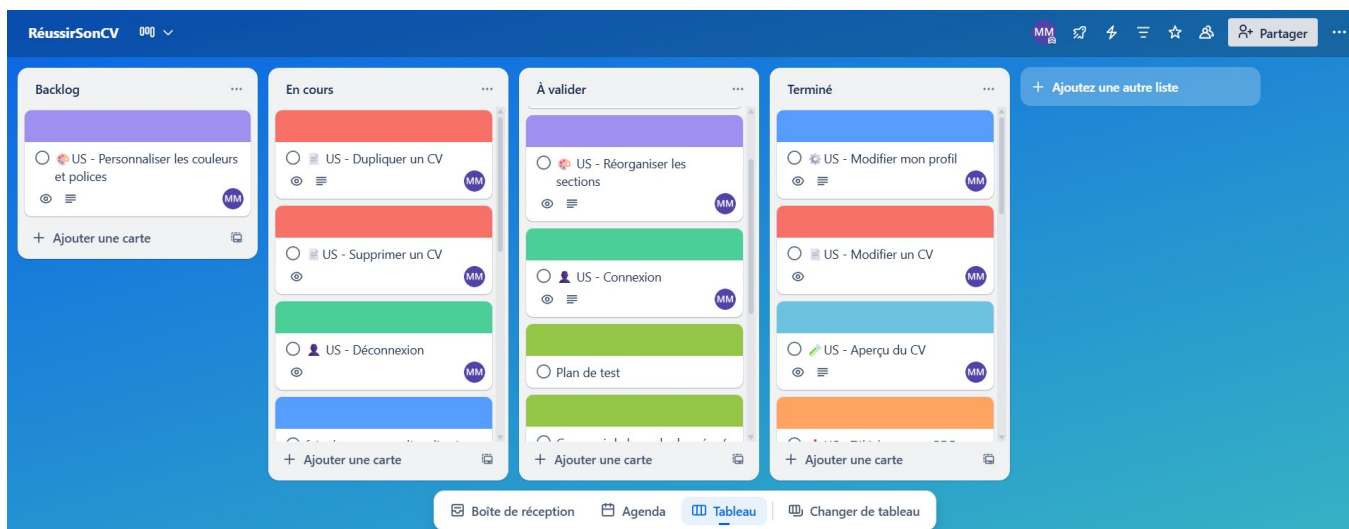
    - name: Build
      run: npm run build
      working-directory: ./frontend
```

Pipeline CI/CD GitHub Actions – Workflow de build et déploiement du frontend React

DOSSIER PROFESSIONNEL (DP)

ANNEXES

Annexes complémentaires



Trello – Organisation et suivi du projet



ReussirSonCV

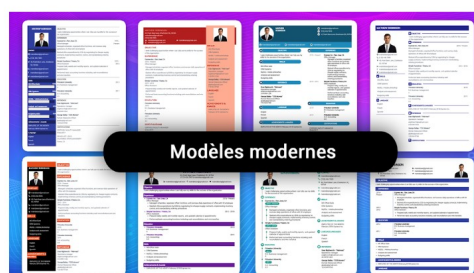
Login

Sign up

Créez votre CV avec ReussirSonCv

Cela prendra seulement quelques minutes.
Modifiez vos informations et confirmez avec un code SMS.

Commencer



Pourquoi Nous

Rapide

Créez votre CV en quelques minutes.

Personnalisé

Adaptez le style à votre profil.

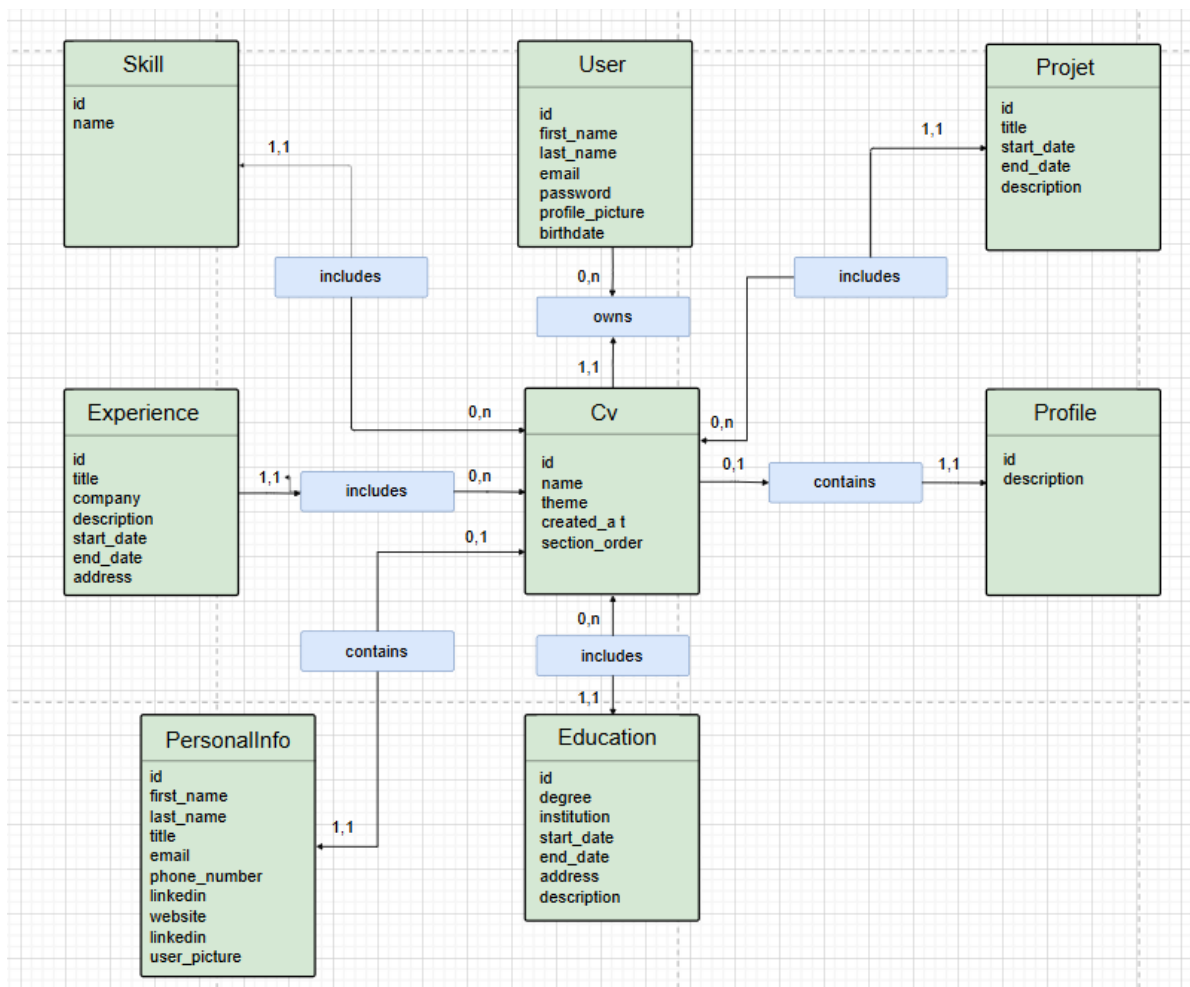
Gratuit

100% gratuit, sans engagement.

Page d'accueil – Vue générale de l'application

ANNEXES

Annexes complémentaires



MCD complet de l'application RéussirSonCV