

République Tunisienne

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Monastir

Institut Supérieur d'Informatique et de Mathématiques de Monastir

RAPPORT DE PROJET

Application de Gestion RH et Documents Administratifs

Réalisé par : Mohamed Kassem Abbassi & Achraf Lamari

Encadré par : Lazhar Hamel

Année universitaire : 2024-2025

Table des matières

1	Introduction	2
2	Analyse des besoins	3
2.1	Identification des acteurs.....	3
2.2	Besoins fonctionnels.....	3
2.3	Cas d'utilisation.....	4
3	Conception	5
3.1	Architecture globale.....	5
3.2	Architecture Backend.....	5
3.3	Architecture Frontend.....	6
3.4	Conception de la base de données.....	6
4	Choix technologiques	7
4.1	Backend.....	7
4.2	Frontend.....	7
4.3	Base de données.....	8
5	Implémentation	9
5.1	Backend (Spring Boot).....	9
5.2	Frontend (Next.js).....	9
5.3	Sécurité.....	10
6	Fonctionnalités implémentées	11
6.1	Gestion des utilisateurs.....	11
6.2	Gestion des congés et absences.....	11
6.3	Gestion des données des employés et enseignants.....	12
6.4	Génération de documents administratifs.....	12
6.5	Statistiques et rapports.....	12
7	Interface utilisateur	13
7.1	Design et ergonomie.....	13
7.2	Expérience utilisateur.....	13
7.3	Présentation des interfaces du projet.....	13
8	Difficultés rencontrées et solutions	14
9	Conclusion et perspectives	15

1. Introduction

Dans le cadre de notre projet, nous avons développé une application web de gestion des ressources humaines et des documents administratifs pour notre institut. Cette application a pour objectif de moderniser et d'optimiser les processus de gestion du personnel administratif et enseignant.

L'application permet la gestion complète des employés, des enseignants, des congés, et la génération automatique de documents administratifs. Elle offre également des fonctionnalités avancées comme la recherche multicritère, la gestion des tâches, ainsi que la production de statistiques et de rapports.

Ce projet répond à un besoin réel de notre institut pour centraliser la gestion des ressources humaines et automatiser la génération de documents administratifs, ce qui permettra d'améliorer l'efficacité et la productivité des services concernés.

2. Analyse des besoins

2.1 Identification des acteurs

Sur la base des spécifications, nous avons identifié les acteurs suivants qui interagissent avec le système :

- **L'Administrateur Système** : Il supervise l'intégralité de la gestion des employés au sein de l'institut. Il dispose d'un accès complet à l'application, ce qui lui permet de gérer efficacement les ressources humaines et de garantir le bon fonctionnement administratif.
- **Personnel RH** : Dispose de droits étendus pour la gestion des employés et des congés. Il peut valider les demandes de congés, modifier les informations des employés et générer des documents administratifs.
- **Employé** : Peut consulter ses informations personnelles, faire des demandes de congés et télécharger des documents administratifs le concernant.
- **Enseignant** : Peut consulter son emploi du temps, ses informations personnelles et faire des demandes de congés.

2.2 Besoins fonctionnels

Les besoins fonctionnels identifiés pour l'application sont les suivants :

1. **Gestion des utilisateurs** : Création, modification, suppression des utilisateurs et gestion des rôles avec différents niveaux d'accès.
2. **Gestion des employés administratifs** : Enregistrement et gestion des informations complètes des employés administratifs.
3. **Gestion des enseignants** : Enregistrement et gestion des informations spécifiques aux enseignants.
4. **Recherche avancée avec filtres** : Possibilité de rechercher des employés et enseignants selon divers critères (grade, département, ancienneté, etc.).
5. **Gestion des congés** : Suivi des demandes de congés, validation, suivi des soldes et historique des congés.
6. **Génération de documents administratifs** : Création automatique de certificats, attestations et autres documents avec pré-remplissage des informations.
7. **Gestion des tâches et responsabilités** : Attribution et suivi des tâches pour le personnel administratif.
8. **Paramètres système** : Configuration des paramètres généraux de l'application.
9. **Statistiques et rapports** : Génération de statistiques sur l'assiduité, les congés et autres indicateurs RH.

2.3 Cas d'utilisation

Conformément aux spécifications fournies, nous avons modélisé les cas d'utilisation suivants :

- **Authentification** : Connexion sécurisée au système.
- **Gestion des utilisateurs** : Création, modification et suppression des comptes utilisateurs.
- **Gestion des employés administratifs** : CRUD des informations des employés.
- **Gestion des enseignants** : CRUD des informations spécifiques aux enseignants.
- **Recherche avec filtres avancés** : Recherche multicritères dans la base de données.
- **Gestion des congés** : Demande, validation et suivi des congés.
- **Génération de documents administratifs** : Création automatique de documents officiels.
- **Configuration des paramètres généraux** : Ajustement des paramètres du système.

3. Conception

3.1 Architecture globale

L'architecture adoptée pour notre application est une architecture **client-serveur**, basée sur une **API REST** pour la communication entre le **frontend** et le **backend**. Cette architecture présente l'avantage de séparer clairement les responsabilités et de faciliter la maintenance et l'évolution de l'application.

Notre système est divisé en trois parties principales :

1. **Le frontend** : Développé avec **Next.js**, il s'agit de l'interface utilisateur avec laquelle interagissent les utilisateurs.
2. **Le backend** : Développé avec **Spring Boot**, il expose les **API REST** qui sont consommées par le **frontend** et implémente la logique métier.
3. **La base de données** : **MySQL** est utilisé pour stocker les données de l'application.

3.2 Architecture Backend

Le backend est structuré selon les principes du Domain-Driven Design (DDD) avec les couches suivantes :

- **Couche Contrôleur** : Gère les requêtes HTTP entrantes et les réponses correspondantes. Elle expose les endpoints REST qui permettent au frontend de communiquer avec le backend.
- **Couche Service** : Contient la logique métier de l'application. Elle orchestre les opérations sur les entités et utilise les repositories pour accéder aux données.
- **Couche Repository** : Responsable de la persistance des données. Elle utilise Spring Data JPA pour interagir avec la base de données MySQL.
- **Couche Modèle** : Définit les entités JPA qui sont mappées sur les tables de la base de données, ainsi que les DTOs (Data Transfer Objects) utilisés pour les communications API.
- **Couche Sécurité** : Implémente l'authentification et l'autorisation en utilisant Spring Security.

3.3 Architecture Frontend

Le frontend est structuré selon une architecture orientée composants, typique des applications React modernes :

- **Pages** : Représentent les différentes vues de l'application. Dans Next.js, chaque fichier dans le dossier pages correspond à une route.
- **Composants** : Éléments réutilisables de l'interface utilisateur. Ils sont organisés par fonctionnalité ou par type.
- **Services** : Encapsulent la logique de communication avec l'API backend.
- **Hooks** : Permettent de gérer l'état et les effets secondaires de manière modulaire.
- **Context** : Gèrent l'état global de l'application, notamment pour l'authentification et les préférences utilisateur.

3.4 Conception de la base de données

La base de données a été conçue pour stocker de manière efficace toutes les informations nécessaires au fonctionnement de l'application. Les principales entités sont :

- **User** : Contient les informations des utilisateurs, incluant leurs rôles et leurs identifiants d'accès.
- **Employé** : Regroupe les données relatives aux employés administratifs et enseignants.
- **Absence** : Enregistre les absences des employés avec les détails associés.
- **Conge** : Gère les demandes de congés, leur durée et leur statut (accepté, refusé, en attente).
- **Journal** : Archive les événements et les activités du système à des fins de traçabilité.
- **Notification** : Stocke les notifications envoyées aux utilisateurs.
- **PDFFile** : Conserve les fichiers PDF générés ou importés dans le système.
- **Quotas** : Définit les quotas ou limites liés aux congés ou absences selon les règles internes.
- **Tache** : Gère les tâches assignées aux employés avec leurs statuts et échéances.

Les relations entre ces entités ont été soigneusement définies pour garantir l'intégrité des données et optimiser les performances des requêtes.

4. Choix technologiques

4.1 Backend

Pour le développement du backend, nous avons choisi les technologies suivantes :

- **Spring Boot** : Framework Java qui facilite la création d'applications autonomes basées sur **Spring**. Il offre une configuration automatique, un serveur embarqué et des fonctionnalités de surveillance.
- **Spring Security** : Module pour gérer l'authentification et l'autorisation. Il fournit une protection contre les attaques courantes comme CSRF, XSS, et session fixation.
- **Spring Data JPA** : Facilite l'accès aux données en réduisant la quantité de code boilerplate nécessaire pour implémenter les repositories.
- **iTextPDF** : Bibliothèque pour la génération dynamique de documents PDF, utilisée pour créer les documents administratifs.
- **Google API Services Gmail** et **Javax Mail API** : Utilisés pour l'envoi automatique d'emails aux utilisateurs pour les notifications.
- **Spring DevTools** : Outils de développement qui améliorent la productivité en permettant le redémarrage automatique de l'application lors des modifications du code.

4.2 Frontend

Pour le frontend, nous avons opté pour les technologies suivantes :

- **Next.js** : Framework **React** qui offre des fonctionnalités comme le rendu côté serveur (SSR), la génération de sites statiques (SSG), et un système de routage intégré.
- **Tailwind CSS** : Framework **CSS** utilitaire qui permet de créer rapidement des interfaces utilisateur personnalisées sans quitter le HTML.
- **Framer Motion** : Bibliothèque d'animations pour **React** qui permet de créer des interfaces utilisateur fluides et interactives.
- **React Hook Form** : Bibliothèque pour la gestion efficace des formulaires dans **React**, avec validation et gestion des erreurs.

4.3 Base de données

Nous avons choisi **MySQL** comme système de gestion de base de données relationnelle pour les raisons suivantes :

- **Fiabilité** : MySQL est reconnu pour sa stabilité et sa fiabilité.
- **Performance** : Il offre de bonnes performances pour les opérations de lecture et d'écriture.
- **Compatibilité** : Il s'intègre parfaitement avec **Spring Data JPA**.
- **Communauté** : Il bénéficie d'une large communauté et d'une documentation abondante.
- **Expérience** : Notre équipe est déjà familière avec cette technologie.

5. Implémentation

5.1 Backend (Spring Boot)

Dans notre implémentation **backend**, nous avons suivi les principes de développement **SOLID** et les bonnes pratiques de Spring Boot :

- **config** : Contient les classes de configuration du projet (sécurité, CORS, etc.), assurant un paramétrage centralisé et flexible.
- **controller** : Gère les endpoints **REST** de l'application, en exposant des interfaces claires pour les différentes fonctionnalités accessibles aux clients.
- **DTO (Data Transfer Objects)** : Facilite l'échange structuré de données entre les couches service et controller, tout en préservant l'encapsulation des entités métiers.
- **entity** : Représente les modèles de données persistés en base. Chaque entité est soigneusement mappée pour refléter la structure de la base de données.
- **filter** : Implémente des filtres (notamment de sécurité ou de journalisation) pour intercepter et traiter les requêtes **HTTP** de manière transversale.
- **repository** : S'appuie sur **Spring Data JPA** pour offrir un accès efficace aux données via des interfaces de persistance, réduisant le code boilerplate.
- **service** : Contient la logique métier principale, organisée selon le principe de responsabilité unique, favorisant la séparation des préoccupations.
- **util** : Regroupe les classes utilitaires et les fonctions communes réutilisables dans l'ensemble du projet.

5.2 Frontend (Next.js)

Pour le frontend, nous avons créé une interface utilisateur moderne et intuitive en utilisant Next.js et Tailwind CSS :

- **Pages Next.js** : Implémentent les différentes vues de l'application.
- **Composants réutilisables** : Créent des éléments d'interface cohérents et mainteneables.
- **Hooks personnalisés** : Encapsulent la logique spécifique à certaines fonctionnalités.
- **Gestion d'état** : Utilisent React Context et SWR pour gérer l'état global et les requêtes API.

5.3 Sécurité

La sécurité a été une préoccupation majeure dans notre implémentation. Nous avons mis en place les mesures suivantes :

- **Authentification JWT** : Utilisation de **JSON Web Tokens** pour l'authentification des utilisateurs.
- **Stockage sécurisé des tokens** : Les **tokens JWT** sont stockés dans des **cookies HTTP-only** pour prévenir les attaques XSS (Cross-Site Scripting).
- **Protection CSRF** : Mise en place de mesures contre les attaques CSRF (Cross-Site Request Forgery).
- **Contrôle d'accès basé sur les rôles** : Implémentation de la gestion des autorisations en fonction des rôles des utilisateurs.
- **Validation des entrées** : Validation rigoureuse de toutes les données d'entrée côté client et côté serveur.
- **Chiffrement des mots de passe** : Utilisation de **BCrypt** pour le hachage sécurisé des mots de passe.
- **Protocole HTTPS** : Configuration du serveur pour n'accepter que les connexions sécurisées.

Une attention particulière a été portée à la sécurisation des sessions utilisateurs en stockant les **tokens JWT** dans des **cookies HTTP-only**, ce qui empêche leur accès par JavaScript et réduit considérablement le risque de vol de session.

6. Fonctionnalités implémentées

6.1 Gestion des utilisateurs

La gestion des utilisateurs comprend les fonctionnalités suivantes :

- **Création de comptes utilisateurs** : Enregistrement de nouveaux utilisateurs avec attribution de rôles.
- **Authentification** : Connexion sécurisée avec email et mot de passe.
- **Gestion des rôles** : Attribution et modification des rôles (administrateur, personnel RH, employé, enseignant).
- **Modification des informations** : Mise à jour des données personnelles des utilisateurs.
- **Réinitialisation de mot de passe** : Processus sécurisé de réinitialisation en cas d'oubli.
- **Journal des activités** : Suivi des actions des utilisateurs pour des raisons de sécurité et d'audit.

6.2 Gestion des congés et absences

Le module de gestion des congés offre les fonctionnalités suivantes :

- **Demande de congés** : Interface permettant aux employés et enseignants de soumettre des demandes de congés.
- **Validation des demandes** : Processus de validation par le personnel RH ou l'administrateur.
- **Suivi des soldes** : Calcul et affichage en temps réel des soldes de congés disponibles.
- **Historique des congés** : Visualisation de l'historique des congés pris et à venir.
- **Conversion des heures supplémentaires** : Système de conversion des heures supplémentaires en jours de congé.
- **Notifications** : Alertes automatiques pour les demandes en attente, les validations et les refus.

6.3 Gestion des données des employés et enseignants

La gestion des données du personnel comprend les fonctionnalités suivantes :

- **Fiches employés et enseignants** : Enregistrement et consultation des informations complètes (personnelles et professionnelles).
- **Recherche avancée** : Système de recherche multicritères (par grade, département, ancienneté, etc.).
- **Filtrage des données** : Possibilité de filtrer les résultats selon différents critères.
- **Modification des informations** : Interface pour la mise à jour des données.
- **Historique des modifications** : Suivi des changements effectués sur les fiches du personnel.

6.4 Génération de documents administratifs

Le module de génération de documents offre les fonctionnalités suivantes :

- **Génération automatique** : Création de documents administratifs (certificats de travail, attestations, etc.) avec pré-remplissage.
- **Personnalisation des modèles** : Possibilité de définir et modifier les modèles de documents.
- **Aperçu avant génération** : Visualisation du document avant sa génération finale.
- **Export en PDF** : Génération de documents au format PDF.
- **Historique des documents** : Conservation de l'historique des documents générés.
- **Notifications** : Alertes automatiques lorsqu'un document est prêt à être téléchargé.

6.5 Statistiques et rapports

Le module de statistiques et rapports comprend les fonctionnalités suivantes :

- **Tableaux de bord** : Affichage des indicateurs clés de performance RH.
- **Statistiques d'assiduité** : Suivi de la présence du personnel.
- **Analyse des congés** : Visualisation des tendances de prise de congés.
- **Rapports périodiques** : Génération automatique de rapports mensuels et annuels.
- **Exportation des données** : Possibilité d'exporter les statistiques au format PDF ou Excel.
- **Graphiques interactifs** : Représentation visuelle des données pour faciliter l'analyse.

7. Interface utilisateur

7.1 Design et ergonomie

L'interface utilisateur a été conçue avec une attention particulière à l'ergonomie et à l'expérience utilisateur :

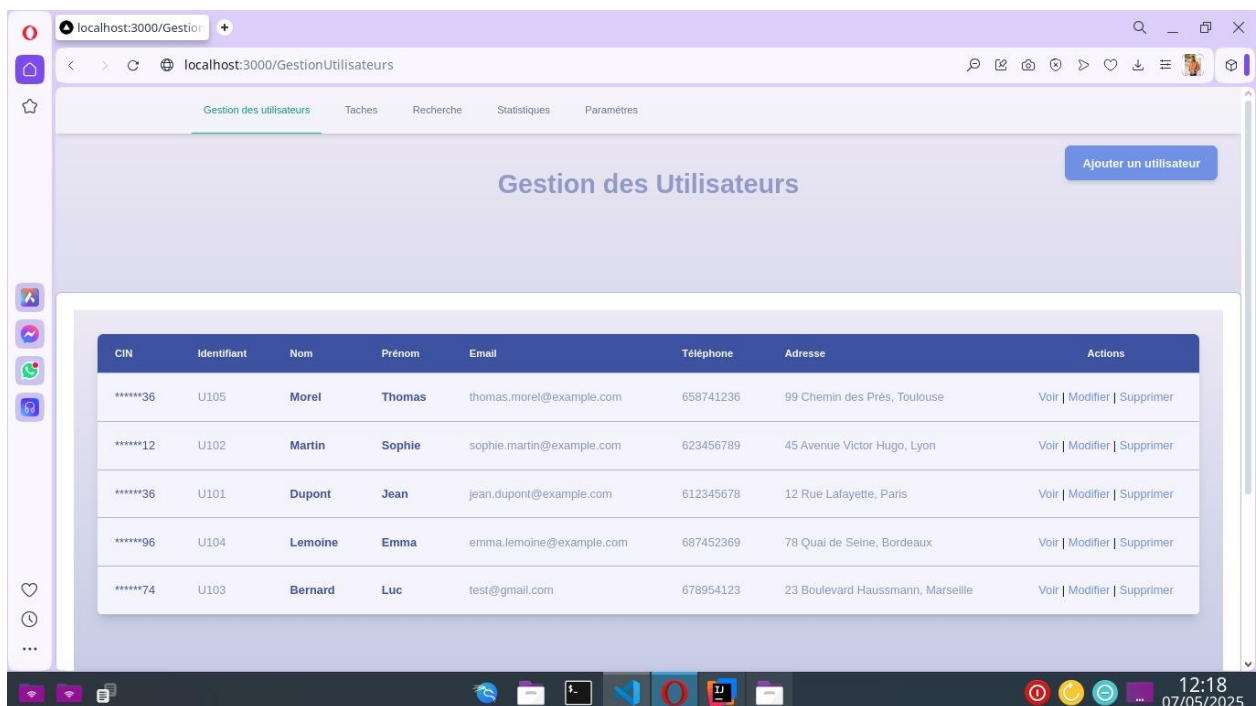
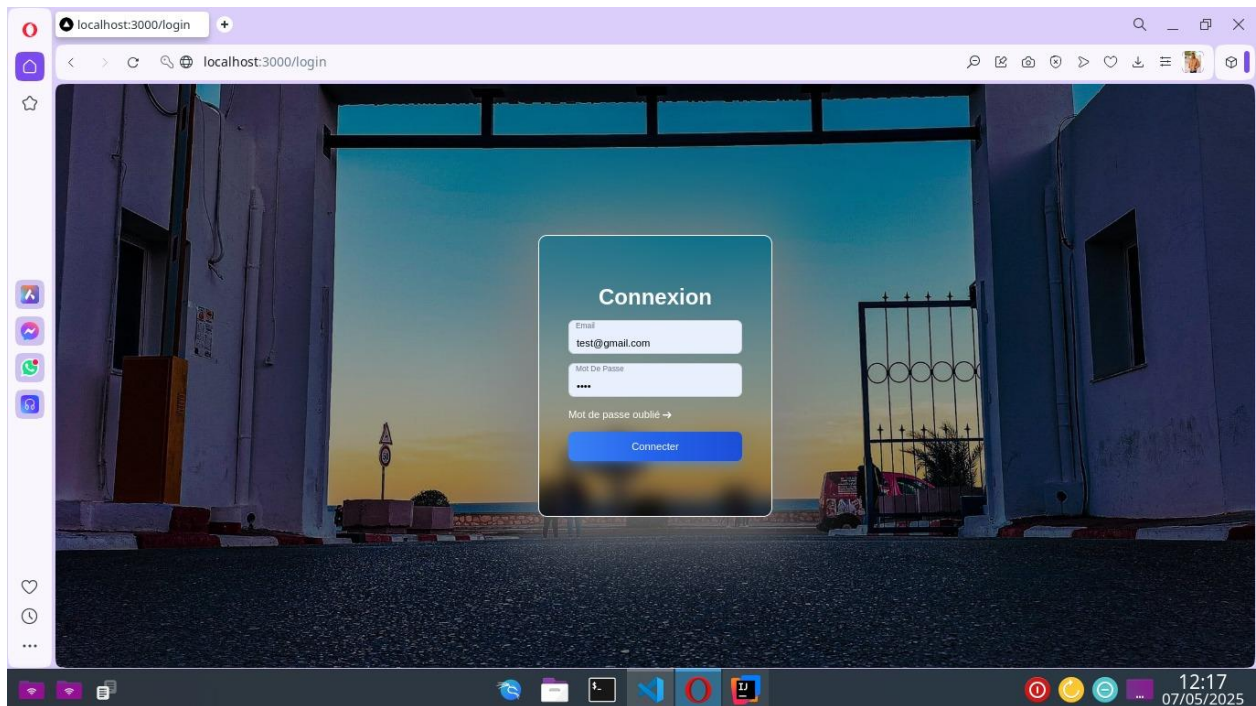
- **Design responsive** : Adaptation automatique à différentes tailles d'écran (desktop, tablette, mobile).
- **Interface moderne** : Utilisation de **Tailwind CSS** pour un design élégant et professionnel.
- **Palette de couleurs cohérente** : Utilisation d'une palette de couleurs harmonieuse et respectant la charte graphique de l'institut.
- **Typographie lisible** : Choix de polices optimisées pour la lisibilité à l'écran.
- **Icônes intuitives** : Utilisation d'icônes significatives pour faciliter la navigation.
- **Cohérence visuelle** : Maintien d'une cohérence dans les éléments d'interface sur l'ensemble de l'application.

7.2 Expérience utilisateur

L'expérience utilisateur a été optimisée pour rendre l'application intuitive et agréable à utiliser :

- **Navigation simplifiée** : Mise en place d'une structure de navigation claire et intuitive.
- **Feedback visuel** : Utilisation d'animations subtiles pour indiquer les actions et les états.
- **Messages d'erreur explicites** : Affichage de messages d'erreur clairs et informatifs.
- **Formulaires optimisés** : Conception de formulaires faciles à remplir avec validation en temps réel.
- **Temps de chargement réduits** : Optimisation des performances pour minimiser les temps d'attente.
- **Accessibilité** : Respect des standards d'accessibilité pour les utilisateurs en situation de handicap.

7.3 Présentation des interfaces du projet



localhost:3000/recherche

Gestion des utilisateurs Tâches Recherche Statistiques Paramètres

Search for items

NUMERO DE CARTE D'IDENTITÉ	IDENTIFIANT (MATRICULE)	NOM	PRENOM	ADRESSE	TELEPHONE	SITUATION	SEXE	DATE DE NAISSANCE	HANDICAPÉ
78452136	U101	Dupont	Jean	12 Rue Lafayette, Paris	612345678	Single	M	1992-07-15	No
45781236	U105	Morel	Thomas	99 Chemin des Prés, Toulouse	658741236	Single	M	1985-09-05	Yes
32569874	U103	Bernard	Luc	23 Boulevard Haussmann, Marseille	678954123	Single	M	1995-11-10	No
87412596	U104	Lemoine	Emma	78 Quai de Seine, Bordeaux	687452369	Divorced	F	1990-02-20	No
65987412	U102	Martin	Sophie	45 Avenue Victor Hugo, Lyon	623456789	Married	F	1988-05-30	No

FILTRE

- Ancienneté
- Departement
- Grade
- Handicapé
- Poste
- Role
- Sexe
- Specialité

Imprimer

12:22 07/05/2025

localhost:3000/GestionTaches

Gestion des utilisateurs Tâches Recherche Statistiques Paramètres

+ Ajouter

À Faire En Cours Terminé

12:21 07/05/2025

localhost:3000/HomeRH

Gestion des utilisateurs Tâches Recherche Statistiques Paramètres

Data Table

CIN	NOM	EMAIL	ROLE	STATUS
45781236	Morel Thomas	thomas.morel@example.com	Employé	Actif
65987412	Martin Sophie	sophie.martin@example.com	Enseignant	Congé
78452136	Dupont Jean	jean.dupont@example.com	Employé	Actif
87412596	Lemoine Emma	emma.lemoine@example.com	Admin	Congé
32569874	Bernard Luc	test@gmail.com	Employé	Congé

Journal Historique

Aucun journal disponible

12:18 07/05/2025

localhost:3000/Parametres

Gestion des utilisateurs Tâches Recherche Statistiques Paramètres

Gestion des Congés

Réinitialiser les soldes

CIN	Identifiant	Nom & Prénom	Heures Supp.	Actions
*****36	U105	Morel Thomas	30 + -	Convertir
*****12	U102	Martin Sophie	8 + -	Convertir
*****36	U101	Dupont Jean	20 + -	Convertir
*****96	U104	Lemoine Emma	12 + -	Convertir
*****74	U103	Bernard Luc	5 + -	Convertir

12:22 07/05/2025

8. Difficultés rencontrées et solutions

Au cours du développement, nous avons rencontré plusieurs défis que nous avons su surmonter :

1. **Gestion des autorisations** : La mise en place d'un système d'autorisation granulaire basé sur les rôles s'est avérée complexe. Nous avons résolu ce problème en utilisant les annotations de **Spring Security** et en définissant des expressions **SpEL** personnalisées.
2. **Performance de la recherche avancée** : Les requêtes de recherche multicritères étaient initialement lentes. Nous avons optimisé les performances en utilisant des requêtes JPA optimisées et en ajoutant des index appropriés dans la base de données.
3. **Génération de PDF complexes** : La création de documents administratifs avec mise en page élaborée a présenté des difficultés. Nous avons utilisé **iTextPDF** avec des templates prédéfinis pour résoudre ce problème.
4. **Gestion des sessions utilisateurs** : La sécurisation des sessions utilisateurs a nécessité une attention particulière. Nous avons implémenté un système de tokens JWT stockés dans des cookies HTTP-only et mis en place une rotation régulière des tokens pour renforcer la sécurité.
5. **Intégration frontend/backend** : La communication entre le frontend et le backend a parfois posé des problèmes de CORS. Nous avons configuré correctement les en-têtes CORS dans Spring Security pour résoudre ces problèmes.

9. Conclusion et perspectives

L'application de gestion RH et documents administratifs développée répond efficacement aux besoins identifiés dans les spécifications. Elle offre une solution moderne, sécurisée et performante pour la gestion du personnel et l'automatisation des tâches administratives.

Les principales réalisations de ce projet sont :

- Une interface utilisateur intuitive et responsive
- Un système d'authentification et d'autorisation robuste
- Une gestion efficace des données des employés et enseignants
- Un système de gestion des congés complet
- Une génération automatisée de documents administratifs
- Des statistiques et rapports pertinents pour la prise de décision

Pour les perspectives d'évolution, nous envisageons les améliorations suivantes :

1. **Module de planning avancé** : Développement d'un module de gestion des plannings plus élaboré, avec gestion des conflits et optimisation automatique.
2. **Application mobile** : Création d'une application mobile pour permettre aux utilisateurs d'accéder aux fonctionnalités essentielles depuis leur smartphone.
3. **Intégration avec d'autres systèmes** : Mise en place d'API pour l'intégration avec d'autres systèmes d'information de l'institut (scolarité, comptabilité, etc.).
4. **Systèmes de notifications avancées** : Implémentation de notifications push et d'alertes personnalisables.
5. **Tableau de bord analytique** : Développement d'un tableau de bord analytique avancé avec prévisions et recommandations.

Ce projet constitue une base solide pour la gestion des ressources humaines de l'institut et pourra être enrichi progressivement pour répondre à des besoins spécifiques émergents.