

Rapport de projet

Application de Gestion des Entretiens Forum



Sara Ferraa & Hajar Afdel

2024/2025

DotNet – Classe de 4IIR G42

1. Introduction

Dans le cadre de notre formation en développement logiciel, nous avons été amenés à réaliser un projet concret mettant en pratique nos compétences en **développement web full-stack**. Le projet, intitulé "**Application de Gestion des Entretiens Forum**", vise à répondre à un besoin réel identifié dans l'organisation des forums de recrutement, où la gestion des candidatures et des entretiens reste souvent manuelle et fastidieuse.

Ce rapport présente le processus complet de conception, développement et déploiement de cette application. Il détaille les différentes phases du projet, les choix technologiques, les défis rencontrés, ainsi que les solutions mises en œuvre pour y répondre. Ce projet nous a permis d'appliquer nos connaissances en **.NET 8**, en gestion de **bases de données relationnelles (SQL Server)**, ainsi qu'en développement d'interfaces utilisateur avec le modèle **MVC et Razor Pages**.

Équipe de Développement :

Le projet a été réalisé par une équipe de **deux personnes**, à savoir :

- **Sara Ferraa** : Développement backend, gestion de la base de données, intégration frontend, tests.
- **Hajar Afdel** : Développement frontend, conception de la base de données, implémentation des fonctionnalités de sécurité, tests.

Les tâches ont été réparties de manière équilibrée, chaque membre contribuant à la fois au **backend**, au **frontend**, à la **gestion de la base de données** et aux **tests**. Cette collaboration a favorisé une compréhension globale du projet par chaque membre de l'équipe, tout en assurant la cohérence et l'intégration des différentes parties de l'application.

Durée du Projet :

Le développement de l'application s'est déroulé sur une période de **un mois**. Le projet a été structuré en plusieurs phases :

1. **Analyse des besoins** et rédaction du cahier des charges.

2. **Conception technique** de l'architecture et des bases de données.
 3. **Développement** des fonctionnalités principales (authentification, dépôt de CV, évaluation des candidats).
 4. **Tests** et validation de l'application.
 5. **Déploiement** et rédaction de la documentation.
-

Objectifs du Projet :

L'objectif principal de ce projet était de développer une application web permettant de :

- **Simplifier la gestion des candidatures** pour les forums de recrutement.
- **Offrir une solution en temps réel** pour l'organisation des entretiens.
- **Faciliter l'évaluation et le suivi des candidats** par les recruteurs.
- **Centraliser les données** dans une base de données sécurisée, accessible par un administrateur.

À travers ce projet, nous avons également cherché à renforcer nos compétences techniques en **programmation orientée objet**, en **gestion de bases de données relationnelles**, et en **sécurisation des applications web**.

Ce rapport détaillera l'ensemble des étapes suivies, des **choix techniques** réalisés, des **problématiques rencontrées**, ainsi que des **solutions** apportées pour mener à bien ce projet.

2. Contexte et Objectifs

Contexte du Projet :

Les **forums de recrutement** sont des événements clés où des candidats rencontrent des recruteurs de différentes entreprises pour postuler à des offres d'emploi ou de stage. Cependant, la gestion de ces forums repose souvent sur des processus manuels : les candidats apportent leurs CV en version papier, les recruteurs prennent des notes sur des feuilles volantes, et l'organisation des entretiens se fait de manière informelle,

entraînant des risques de perte d'informations, des erreurs de suivi, et une inefficacité globale.

Dans ce contexte, il est apparu essentiel de développer une solution numérique capable de **centraliser et automatiser la gestion des candidatures** et des entretiens. Le projet "**Application de Gestion des Entretiens Forum**" a été conçu pour répondre à ces défis en offrant une plateforme web qui simplifie ces processus pour tous les acteurs impliqués : candidats, recruteurs, et administrateurs.

L'idée est de proposer une **interface intuitive et sécurisée** qui permette :

- Aux **candidats** de soumettre facilement leur CV en ligne et de choisir les entreprises ciblées.
- Aux **recruteurs** de consulter rapidement les profils, de planifier les entretiens et d'évaluer les candidats directement via l'application.
- Aux **administrateurs** de gérer l'ensemble des forums, des entreprises participantes, et des recruteurs, tout en accédant à des statistiques globales sur le déroulement des forums.

Problématique :

La problématique principale à laquelle ce projet répond est la **complexité de la gestion des candidatures et des entretiens dans les forums de recrutement**. Actuellement, ces processus sont souvent :

- **Lents** en raison de la saisie manuelle des informations.
- **Sujet à des erreurs** de suivi des candidatures et des évaluations.
- **Non centralisés**, ce qui complique la supervision globale pour les organisateurs des forums.

Notre application vise à pallier ces lacunes en offrant une solution numérique efficace et fiable.

Objectifs Généraux :

L'objectif principal de ce projet est de **développer une application web complète**

permettant de **numériser et automatiser** l'ensemble du processus de gestion des entretiens dans les forums de recrutement. L'application doit être :

- **Intuitive** : Facile à utiliser pour les candidats et les recruteurs, même sans compétences techniques avancées.
 - **Sécurisée** : Assurer la confidentialité des données des candidats et le contrôle d'accès pour les recruteurs et administrateurs.
 - **Efficace** : Permettre un traitement rapide des candidatures, des évaluations, et des statistiques.
-

Objectifs Spécifiques :

Pour répondre à ces objectifs généraux, l'application doit intégrer les fonctionnalités suivantes :

1. Pour les Candidats :

- Dépôt de CV en ligne via un formulaire interactif, sans nécessité de créer un compte.
- Sélection des entreprises auprès desquelles ils souhaitent postuler à partir d'une liste prédéfinie.
- Confirmation de la soumission de leur candidature via une notification.

2. Pour les Recruteurs :

- Recherche et consultation des CV grâce à des filtres (nom, compétences, entreprises ciblées).
- Évaluation des candidats via un système de notation (étoiles ou points) et des commentaires personnalisés.
- Accès à un tableau de bord pour visualiser l'historique des entretiens et les statistiques d'évaluation.

3. Pour les Administrateurs :

- Gestion des forums, des entreprises participantes et des comptes recruteurs.
- Supervision de la base de données pour garantir l'intégrité et la sécurité des informations.

- Accès à un tableau de bord global avec des statistiques sur les candidatures, les entretiens réalisés, et les performances des entreprises.
-

Bénéfices Attendus :

Le développement de cette application apporte des **bénéfices significatifs** aux différents utilisateurs :

- **Gains de temps** pour les recruteurs grâce à une gestion numérique des CV et des entretiens.
 - **Suivi efficace** des candidatures, réduisant les risques de perte d'informations.
 - **Meilleure organisation** des forums pour les administrateurs, avec des données centralisées et accessibles en temps réel.
 - **Expérience utilisateur améliorée** pour les candidats, avec un processus de dépôt simplifié et rapide.
-

Ce projet constitue non seulement une **solution innovante** pour les forums de recrutement, mais également une **opportunité d'apprentissage** pour notre équipe, en mettant en pratique des compétences en développement web, gestion de base de données, et sécurité des applications.

3. Cahier des Charges

1. Contexte et Objectifs du Projet

Dans le cadre de forums de recrutement, les candidats souhaitent déposer leur CV auprès d'entreprises ciblées, et les recruteurs doivent pouvoir consulter les profils, organiser des entretiens en temps réel, et évaluer les candidats directement. Ce projet vise à développer une application web permettant de :

1. Simplifier la gestion des candidatures pour les forums.
 2. Offrir une solution en temps réel pour l'organisation des entretiens.
 3. Permettre une évaluation et un suivi efficaces des candidats.
 4. Centraliser les données dans une base gérée par un administrateur.
-

2. Description des Acteurs et Rôles

1. Candidats :

- Déposent leur CV via un formulaire interactif.
- Sélectionnent les entreprises auprès desquelles ils souhaitent postuler.
- Ne créent pas de compte.

2. Recruteurs :

- Recherchent et consultent les CV déposés.
- Évaluent les candidats avec un système de notation (fiche d'évaluation) et de commentaires.

3. Administrateur :

- Gère les forums, entreprises, et recruteurs.
 - Supervise et manipule la base de données.
 - Accède à un tableau de bord pour visualiser les statistiques globales.
-

3. Fonctionnalités Détaillées

3.1. Côté Candidat :

● Dépôt de CV :

- Formulaire interactif pour remplir les informations de base :
 - Nom, prénom, email, compétences principales.
 - Téléchargement d'un CV au format PDF (taille maximale à définir, ex. 5 Mo).
- Possibilité de sélectionner les entreprises auxquelles ils postulent parmi une liste préchargée.

● Confirmation de soumission :

- Une fois le CV soumis, une notification confirme la réussite du dépôt.
-

3.2. Côté Recruteur :

- **Recherche et consultation des candidats :**
 - Barre de recherche avec filtres par :
 - Nom/prénom.
 - Compétences spécifiques.
 - Entreprise ciblée par le candidat.
 - Accès direct aux CV des candidats sous forme de liste ou de carte.
 - **Évaluation des candidats :**
 - Fiche d'évaluation pour chaque candidat comprenant :
 - Un champ de **notation** (sur 5 étoiles ou sur 10).
 - Un champ de **commentaire libre** pour les impressions ou remarques.
 - Historique des évaluations disponible pour consultation.
-

3.3. Côté Administrateur :

- **Gestion des forums :**
 - Création, modification et suppression de forums.
 - Association des entreprises à un forum.
- **Gestion des entreprises :**
 - Ajout et mise à jour des informations sur les entreprises participantes (nom, secteur, etc.).
- **Gestion des recruteurs :**
 - Création de comptes pour les recruteurs, incluant :
 - Nom du recruteur.
 - Entreprise associée.
 - Accès sécurisé via identifiants.
- **Tableau de bord et statistiques :**
 - Nombre de candidats inscrits par forum.
 - Nombre d'entretiens réalisés.

- Performance des entreprises (ex. nombre de candidatures reçues).
-

4. Interface Utilisateur (UI/UX)

1. Candidats :

- Formulaire clair et simple avec indication des étapes complètes.
- Sélection facile des entreprises via une liste déroulante ou des cases à cocher.

2. Recruteurs :

- Tableau de bord avec :
 - Liste des candidats (recherchable et filtrable).
 - Accès direct aux fiches des candidats.
- Section pour les évaluations.

3. Administrateurs :

- Interface avancée avec un tableau de bord global.
- Accès à toutes les tables pour ajout/modification/suppression.

4. Technologies Utilisées

Le choix des technologies est une étape cruciale dans le développement d'une application web. Pour garantir la **fiabilité**, la **performance** et la **sécurité** de notre application de gestion des entretiens pour les forums de recrutement, nous avons sélectionné des technologies modernes et robustes, adaptées aux besoins spécifiques du projet.

4.1. Environnement de Développement

- **Framework : .NET 8**

Le choix de .NET 8 s'est imposé naturellement pour plusieurs raisons :

- **Performance et fiabilité** : .NET est reconnu pour ses performances élevées, sa robustesse, et sa capacité à gérer des applications complexes.
- **Sécurité** : Le framework offre des fonctionnalités de sécurité intégrées (authentification, gestion des rôles, protection contre les attaques courantes comme l'injection SQL).
- **Support du modèle MVC (Model-View-Controller)** : Ce modèle a permis de séparer clairement la logique métier, les données, et l'interface utilisateur, facilitant ainsi le développement et la maintenance de l'application.

- **Langage de Programmation : C#**

Le langage C# a été utilisé pour développer l'ensemble de la logique métier. C'est un langage orienté objet puissant, bien intégré à l'écosystème .NET, offrant des fonctionnalités avancées pour le développement web.

4.2. Gestion de la Base de Données

- **Système de Gestion de Base de Données (SGBD) : SQL Server**

Nous avons choisi **SQL Server** pour gérer les données de l'application en raison de :

- **Sa robustesse** dans la gestion des bases de données relationnelles.
- **Son intégration facile** avec le framework .NET via **Entity Framework** pour l'ORM (Object-Relational Mapping), simplifiant ainsi les opérations de création, lecture, mise à jour et suppression (CRUD).
- **Sécurité et performance** dans la gestion des transactions et des requêtes complexes.

- **Outils utilisés pour la gestion de la base de données :**

- **SQL Server Management Studio (SSMS)** : Pour la création, la gestion et l'optimisation des tables et des requêtes.

- **Entity Framework Core** : Pour faciliter l'interaction entre le code C# et la base de données SQL grâce à l'approche ORM.
-

4.3. Technologies Front-End

- **Modèle : MVC avec Razor Pages**

Le modèle **MVC (Model-View-Controller)** a été utilisé en combinaison avec **Razor Pages** pour le développement des interfaces utilisateur :

- **Model (Modèle)** : Représente les données de l'application et la logique métier.
 - **View (Vue)** : Composée de pages Razor, elle est responsable de l'affichage des données à l'utilisateur. Razor offre une syntaxe fluide pour combiner le HTML avec le C#.
 - **Controller (Contrôleur)** : Gère les interactions entre le modèle et la vue, traitant les requêtes des utilisateurs et renvoyant les réponses appropriées.
 - **HTML, CSS et JavaScript**
 - **HTML5** : Pour la structure des pages web.
 - **CSS3** : Pour le style et la mise en page des différentes interfaces (formulaires de dépôt de CV, tableau de bord des recruteurs, etc.).
 - **JavaScript** : Pour rendre l'application plus interactive (par exemple, validation de formulaires côté client, notifications dynamiques après soumission).
-

4.4. Sécurité et Gestion des Utilisateurs

- **Authentification et Autorisation :**

- **Identity Framework de .NET** : Utilisé pour la gestion de l'authentification des utilisateurs et des rôles (Administrateur, Recruteur).
- **Role-Based Access Control (RBAC)** : Mise en place de contrôles d'accès basés sur les rôles pour restreindre l'accès aux fonctionnalités en fonction

du profil de l'utilisateur.

- **Protection des Données :**
 - **Chiffrement des mots de passe :** Les mots de passe des recruteurs et des administrateurs sont stockés de manière sécurisée grâce aux mécanismes de hachage intégrés dans .NET.
 - **Validation des entrées utilisateur :** Prévention des injections SQL et des attaques XSS grâce à la validation et l'échappement des données entrées par les utilisateurs.
-

4.5. Outils de Développement et de Collaboration

- **Environnement de Développement Intégré (IDE) : Visual Studio 2022**

Visual Studio a été utilisé pour coder, déboguer et tester l'application. Il offre des outils puissants pour le développement en .NET, notamment l'intégration avec SQL Server, des outils de débogage avancés, et la gestion des packages via **NuGet**.
 - **Gestion de Version :**
 - **Git :** Utilisé pour le contrôle de version, permettant un suivi précis des modifications apportées au code.
 - **Plateforme de dépôt :** [GitHub/GitLab/Autre] pour héberger le code source, faciliter la collaboration et gérer les versions.
-

4.6. Méthodes de Test

- **Tests Unitaires :**

Nous avons utilisé les outils de tests intégrés à .NET pour écrire des **tests unitaires**, permettant de valider le bon fonctionnement des différentes parties de la logique métier.
- **Tests Manuels :**

Des **tests fonctionnels manuels** ont été réalisés pour s'assurer que les différentes interfaces utilisateur (dépôt de CV, tableau de bord, évaluation des candidats)

fonctionnent correctement.

- **Outils de Debugging :**
 - **Debugging avec Visual Studio :** Utilisation des points d'arrêt pour identifier les bugs dans la logique du code.
 - **Logging :** Mise en place de logs pour suivre le comportement de l'application, notamment pour déboguer les problèmes liés à l'authentification et à la gestion des rôles.
-

4.7. Déploiement

- **Serveur de Test Local :**

L'application a été déployée sur un serveur local pour les phases de test et de validation.
 - **Outils de Déploiement :**
 - **IIS (Internet Information Services) :** Utilisé pour héberger l'application web en environnement local.
 - **Scripts de migration SQL :** Pour automatiser le déploiement des structures de la base de données.
-

Conclusion sur le Choix des Technologies :

Le choix de ces technologies nous a permis de développer une **application web robuste, sécurisée et performante**, répondant aux besoins spécifiques des forums de recrutement. L'intégration harmonieuse entre **.NET 8, SQL Server**, et le modèle **MVC avec Razor Pages** a facilité la conception et la mise en œuvre des fonctionnalités tout en assurant une bonne évolutivité pour d'éventuelles améliorations futures.

5. Développement et Répartition des Tâches

5.1. Méthodologie de Développement

Pour mener à bien ce projet, nous avons adopté une **approche itérative** inspirée de la méthodologie **Agile**. Cette approche nous a permis de diviser le projet en plusieurs cycles de développement (ou sprints), chacun comprenant des phases de **planification, développement, test et révision**. Cette méthode a favorisé une **flexibilité** dans l'ajout ou la modification de fonctionnalités en cours de projet et a permis une **meilleure gestion des imprévus**.

Chaque sprint était organisé autour d'un ensemble de fonctionnalités à développer et à tester. À la fin de chaque cycle, nous procédions à une **revue des tâches réalisées** et à un ajustement des priorités pour le sprint suivant.

5.2. Phases du Développement

Le projet a été structuré en plusieurs phases clés, de l'analyse des besoins au déploiement final :

1. **Analyse des besoins et rédaction du cahier des charges :**
Cette phase a consisté à identifier les besoins des utilisateurs (candidats, recruteurs, administrateurs) et à définir les fonctionnalités essentielles de l'application.
2. **Conception technique et modélisation :**
 - Élaboration des **diagrammes UML** (cas d'utilisation, diagrammes de classes, séquence) pour modéliser le fonctionnement de l'application.
 - Conception de la **base de données relationnelle** sous SQL Server.
3. **Développement du backend :**
 - Mise en place de l'architecture **MVC** avec .NET 8.
 - Implémentation de la logique métier (authentification, gestion des rôles, dépôt de CV, évaluation des candidats).
4. **Développement du frontend :**

- Création des **interfaces utilisateur** avec Razor Pages pour les différentes parties prenantes (candidats, recruteurs, administrateurs).
 - Intégration des fonctionnalités interactives avec HTML, CSS et JavaScript.
5. **Tests et validation :**
- Réalisation de **tests unitaires** pour la logique métier et de **tests fonctionnels manuels** pour les interfaces utilisateur.
 - Identification et correction des bugs.
6. **Déploiement et finalisation :**
- Déploiement de l'application sur un environnement de test local.
 - Rédaction de la **documentation technique** et du rapport final.
-

5.3. Répartition des Tâches

Le projet a été réalisé par une équipe de **deux membres**, chacun ayant participé activement à toutes les étapes du développement. La répartition des tâches a été équilibrée afin de garantir une bonne compréhension du projet dans sa globalité par chaque membre.

Membre 1 : Sara Ferraa

- **Backend :**
 - Mise en place de l'architecture MVC.
 - Implémentation de la logique métier pour l'authentification et la gestion des rôles.
 - Développement de la fonctionnalité de dépôt de CV et gestion des fichiers.
 - **Base de données :**
 - Conception du schéma de la base de données (tables, relations, clés primaires/étrangères).
 - Rédaction des requêtes SQL pour la gestion des candidatures et des évaluations.
 - **Tests :**
 - Réalisation de tests unitaires sur les fonctionnalités backend.
 - Débogage des problèmes liés à l'authentification.
-

Membre 2 : Hajar Afdel

- **Frontend :**
 - Développement des interfaces utilisateur avec Razor Pages.
 - Création du tableau de bord pour les recruteurs et des formulaires pour le dépôt de CV.
 - Intégration de fonctionnalités interactives avec JavaScript (validation des formulaires, notifications).
 - **Sécurité et gestion des accès :**
 - Implémentation des contrôles d'accès basés sur les rôles (RBAC).
 - Sécurisation des données utilisateurs (chiffrement des mots de passe, validation des entrées).
 - **Tests :**
 - Réalisation de tests fonctionnels manuels sur les interfaces utilisateur.
 - Correction des incohérences dans l'interface et les flux de navigation.
-

5.4. Collaboration et Outils Utilisés

La collaboration entre les deux membres de l'équipe a été facilitée par l'utilisation d'outils de gestion de projet et de versionnage :

- **Gestion de version : Git**
 - Suivi des modifications du code source.
 - Fusion des différentes branches de développement pour garantir l'intégration continue.
 - Plateforme utilisée : **GitHub**.
 - **Outils de communication :**
 - **Whatsapp et Discord** pour la coordination quotidienne et les discussions techniques.
 - **Réunions hebdomadaires** pour faire le point sur l'avancement du projet et ajuster les priorités.
 - **Suivi des tâches :**
 - Utilisation de **Trello** pour organiser les tâches à réaliser, suivre les progrès et identifier les points bloquants.
-

5.5. Gestion des Délais et Adaptation

Malgré une durée de développement limitée à **un mois**, nous avons réussi à respecter le calendrier prévu grâce à une planification rigoureuse et une bonne communication au sein de l'équipe. Cependant, certains imprévus ont nécessité des ajustements :

- **Problèmes rencontrés :**
 - Gestion des redirections après authentification (problèmes de permissions pour les recruteurs).
 - Complexité dans la mise en place du système de dépôt de CV avec gestion des fichiers PDF.
 - **Solutions apportées :**
 - Mise en place de sessions de **débogage intensif** avec des logs détaillés pour identifier les problèmes.
 - Ajustement des rôles et des permissions pour assurer la cohérence des accès.
-

Conclusion sur la Répartition des Tâches :

La collaboration entre les deux membres de l'équipe a été efficace, permettant de **couvrir l'ensemble des aspects du projet** (backend, frontend, base de données, tests). La **polyvalence** des membres a permis une meilleure compréhension des différentes couches de l'application et une **intégration fluide** des fonctionnalités. La répartition équilibrée des tâches a contribué à la réussite du projet, tout en offrant à chaque membre l'opportunité de renforcer ses compétences dans des domaines variés du développement web.

6. Fonctionnalités Implémentées

L'application "**Gestion des Entretiens Forum**" a été conçue pour répondre aux besoins spécifiques des **candidats**, des **recruteurs**, et des **administrateurs** lors des forums de recrutement. Toutes les fonctionnalités principales prévues dans le cahier des charges

ont été développées, permettant une **gestion fluide des candidatures**, une **évaluation structurée des candidats**, et une **supervision efficace** des forums.

6.1. Fonctionnalités pour les Candidats

L'expérience des candidats a été pensée pour être **simple, rapide et intuitive**. Les candidats peuvent interagir avec l'application sans avoir à créer de compte, ce qui facilite l'accès et réduit les barrières à l'utilisation.

1. Dépôt de CV en Ligne :

- Formulaire interactif pour remplir les informations personnelles :
 - **Nom, prénom, email, et compétences principales.**
- **Téléchargement de CV** au format PDF, avec une **limite de taille de 5 Mo**.
- Validation automatique des champs pour éviter les erreurs de saisie (vérification de l'email, format du fichier PDF, etc.).

2. Sélection des Entreprises Ciblées :

- Liste déroulante ou cases à cocher permettant aux candidats de **sélectionner les entreprises** auxquelles ils souhaitent postuler.
- La liste des entreprises est préchargée par l'administrateur, garantissant des informations à jour.

3. Confirmation de Soumission :

- Une **notification de confirmation** apparaît à la soumission du formulaire pour informer le candidat que son CV a bien été enregistré.
- Envoi automatique d'un **email de confirmation** (si cette fonctionnalité est activée dans la version finale).

6.2. Fonctionnalités pour les Recruteurs

Les recruteurs disposent d'un **tableau de bord personnalisé** leur permettant de gérer

efficacement les candidatures, d'effectuer des recherches précises et d'évaluer les candidats de manière structurée.

1. Authentification et Accès Sécurisé :

- Les recruteurs accèdent à leur espace via une **authentification sécurisée** avec identifiants fournis par l'administrateur.
- La gestion des accès est basée sur un **système de rôles**, garantissant que seuls les recruteurs autorisés peuvent accéder aux candidatures.

2. Recherche et Consultation des Candidats :

- **Barre de recherche avancée** avec des filtres :
 - **Nom/Prénom** du candidat.
 - **Compétences spécifiques** recherchées par le recruteur.
 - **Entreprise ciblée** par le candidat.
- Visualisation des candidatures sous forme de **liste** ou de **cartes**, avec un accès direct aux CV en PDF.

3. Évaluation des Candidats :

- **Fiche d'évaluation** intégrée pour chaque candidat :
 - **Système de notation** (par exemple, sur **5 étoiles** ou **10 points**).
 - **Champ de commentaire** libre pour que les recruteurs puissent noter leurs impressions et remarques spécifiques.
- Les recruteurs peuvent **consulter l'historique des évaluations**, ce qui permet un suivi des candidats sur plusieurs entretiens.

4. Tableau de Bord Statistique :

- Visualisation des **statistiques sur les évaluations** réalisées, incluant :
 - Le nombre de candidatures consultées.
 - Le nombre d'évaluations complétées.
 - Les moyennes des notes attribuées aux candidats.

6.3. Fonctionnalités pour les Administrateurs

L'administrateur dispose d'un **tableau de bord global** pour gérer l'ensemble des forums,

des entreprises participantes, des recruteurs, et des statistiques de performance.

1. Gestion des Forums de Recrutement :

- Création, modification et suppression de forums via une interface dédiée.
- Association des **entreprises à chaque forum**, facilitant l'organisation des événements.

2. Gestion des Entreprises Participantes :

- Ajout et mise à jour des informations sur les entreprises :
 - **Nom, secteur d'activité, description**, etc.
- Possibilité de **désactiver** ou de **supprimer** une entreprise du forum si nécessaire.

3. Gestion des Recruteurs :

- Création et gestion des comptes recruteurs :
 - Enregistrement du **nom du recruteur**, de **l'entreprise associée** et des **informations de connexion**.
- Attribution des rôles et gestion des **droits d'accès** pour garantir la sécurité des informations.

4. Tableau de Bord Administratif :

- Visualisation des **statistiques globales** sur les forums :
 - **Nombre de candidats inscrits** par forum.
 - **Nombre d'entretiens réalisés** et d'évaluations effectuées.
 - **Performance des entreprises** (ex : nombre de candidatures reçues par entreprise).

5. Supervision de la Base de Données :

- Accès à l'ensemble des **tables de la base de données** pour effectuer des opérations de :
 - **Ajout, modification** ou **suppression** de données.
 - Supervision des informations pour assurer leur cohérence et leur sécurité.

6.4. Fonctionnalités Additionnelles Développées

En plus des fonctionnalités prévues dans le cahier des charges, nous avons intégré des **améliorations** pour enrichir l'expérience des utilisateurs, notamment pour les recruteurs.

1. Statistiques d'Évaluation Avancées pour les Recruteurs :

- Ajout de graphiques et de tableaux de bord interactifs permettant aux recruteurs de visualiser :
 - La **répartition des notes** attribuées aux candidats.
 - Les **compétences les plus recherchées** parmi les candidats évalués.
 - Le **suivi des entretiens** réalisés par recruteur.

2. Notifications Dynamique :

- Implémentation de **notifications en temps réel** lors de la soumission des évaluations ou des modifications dans les profils des candidats.

3. Amélioration de l'Interface Utilisateur :

- Interface plus fluide et interactive avec des **animations légères** pour améliorer l'expérience utilisateur.
- Optimisation de la navigation avec des **filtres intelligents** et des **barres de recherche dynamiques**.

Conclusion sur les Fonctionnalités Implémentées :

Toutes les **fonctionnalités principales** décrites dans le cahier des charges ont été **développées et testées avec succès**. L'ajout de fonctionnalités supplémentaires, comme les **statistiques d'évaluation avancées** pour les recruteurs, a permis de rendre l'application plus complète et plus performante. L'application est ainsi **prête à être utilisée dans un contexte de forum de recrutement**, avec des possibilités d'évolution pour intégrer des fonctionnalités encore plus avancées à l'avenir.

7. Fonctionnalités Manquantes et Améliorations

Malgré le respect du calendrier de développement et l'implémentation de la majorité des fonctionnalités prévues dans le cahier des charges, certaines fonctionnalités n'ont pas pu être finalisées en raison de **contraintes de temps**, de **complexités techniques** ou de **priorisation des tâches**. Par ailleurs, au cours du projet, des **améliorations** ont été ajoutées pour optimiser l'expérience utilisateur et enrichir les fonctionnalités existantes.

7.1. Fonctionnalités Manquantes

1. Fonctionnalités de Reporting Avancées pour les Administrateurs :

Bien que les statistiques de base (nombre de candidats inscrits, nombre d'entretiens réalisés) aient été implémentées, certaines fonctionnalités de **reporting avancé** prévues dans le cahier des charges n'ont pas pu être finalisées. Il s'agit notamment de :

- **Rapports détaillés sur la performance des entreprises** : Prévoir des graphiques détaillant les performances de chaque entreprise en termes de nombre de candidatures reçues, taux de conversion (candidatures -> entretiens -> embauches), etc.
- **Exportation des données** : Génération de rapports téléchargeables (formats PDF, Excel) pour les administrateurs souhaitant analyser les performances hors ligne.
- **Filtres temporels pour les statistiques** : Possibilité d'afficher les statistiques sur des périodes spécifiques (par jour, semaine, mois).

2. Système de Notification Automatique par Email :

Le cahier des charges prévoyait l'envoi automatique de notifications par email pour :

- **Confirmation de dépôt de CV** pour les candidats.

- **Notifications aux recruteurs** lorsqu'un candidat postule à une entreprise spécifique.
 - **Rappels automatiques** pour les entretiens à venir.
Cette fonctionnalité n'a pas été finalisée en raison des défis techniques liés à la configuration des serveurs SMTP et de la gestion des permissions d'envoi d'emails.
3. **Planification Automatique des Entretiens :**
Bien que les recruteurs puissent consulter les CV et évaluer les candidats, la **fonctionnalité de planification automatique des entretiens en temps réel** n'a pas été intégrée. Cette fonctionnalité devait permettre :
- La **prise de rendez-vous** automatique entre candidats et recruteurs via l'application.
 - L'intégration d'un **calendrier interactif** pour gérer la disponibilité des recruteurs et des candidats.
4. **Multi-langues (Internationalisation) :**
L'application est actuellement disponible uniquement en **français**. La **traduction de l'interface** en d'autres langues (comme l'anglais) pour une utilisation dans des forums internationaux était envisagée, mais n'a pas été mise en œuvre.

7.2. Raisons des Fonctionnalités Manquantes

1. **Contraintes de Temps :**
Le projet devait être complété dans un délai de **un mois**, ce qui a limité le temps disponible pour des fonctionnalités secondaires ou avancées. La priorité a été donnée aux **fonctionnalités essentielles** pour garantir une application fonctionnelle et stable.
2. **Complexité Technique :**
Certaines fonctionnalités, comme le **reporting avancé** et la **planification automatique des entretiens**, nécessitent des intégrations complexes (ex. : graphiques interactifs, gestion de calendriers) qui auraient exigé plus de temps de développement et de tests.
3. **Priorisation des Tâches :**

Nous avons fait le choix de prioriser la **stabilité** et la **sécurité** de l'application avant d'intégrer des fonctionnalités supplémentaires. L'accent a été mis sur des fonctionnalités robustes comme l'authentification sécurisée, le dépôt de CV, et l'évaluation des candidats.

7.3. Améliorations Apportées au Cours du Projet

En parallèle du développement des fonctionnalités principales, nous avons intégré plusieurs **améliorations** qui n'étaient pas initialement prévues dans le cahier des charges. Ces ajouts ont permis d'améliorer l'expérience utilisateur et d'optimiser certaines fonctionnalités.

1. Tableau de Bord Amélioré pour les Recruteurs :

- Ajout de **statistiques d'évaluation détaillées** pour les recruteurs :
 - Moyenne des notes attribuées aux candidats.
 - Répartition des évaluations par compétence.
 - Graphiques interactifs pour visualiser les performances des candidats.
- Ces améliorations offrent aux recruteurs une **vue d'ensemble plus précise** de leurs entretiens et des profils évalués.

2. Amélioration de l'Interface Utilisateur (UI/UX) :

- **Optimisation des formulaires** de dépôt de CV pour les rendre plus intuitifs et accessibles.
- **Affichage dynamique** des notifications lors de la soumission des candidatures et des évaluations, pour un retour immédiat à l'utilisateur.
- Mise en place d'un **design responsive** pour que l'application soit accessible sur différents appareils (ordinateurs, tablettes, smartphones).

3. Sécurisation Renforcée des Données :

- **Chiffrement avancé des mots de passe** et des informations sensibles des recruteurs et administrateurs.
- Mise en place de **contrôles d'accès plus stricts** pour garantir la confidentialité des données des candidats.

- **Validation approfondie des fichiers** lors du dépôt de CV (vérification du format PDF, taille maximale, et protection contre les fichiers malveillants).
4. **Optimisation des Requêtes SQL :**
- Amélioration des **performances des requêtes** pour le filtrage et la recherche des candidats.
 - Mise en place d'**indexation** sur les champs fréquemment consultés (nom, compétences, entreprises ciblées).
-

7.4. Fonctionnalités à Envisager pour des Évolutions Futures

Bien que l'application soit pleinement fonctionnelle, plusieurs axes d'amélioration peuvent être envisagés pour les versions futures :

1. **Intégration d'un Système de Messagerie Interne :**
Permettre aux recruteurs et aux candidats de communiquer directement via l'application pour organiser les entretiens.
2. **Application Mobile Dédiée :**
Développer une version mobile native (Android/iOS) pour faciliter l'accès aux fonctionnalités depuis un smartphone.
3. **Automatisation des Processus Administratifs :**
 - Intégration de **rapports automatiques** envoyés aux administrateurs à intervalles réguliers.
 - Mise en place de **rappels automatisés** pour les recruteurs en fonction des échéances des entretiens.
4. **Personnalisation Avancée des Interfaces :**
Permettre aux recruteurs et aux administrateurs de **personnaliser leur tableau de bord** en fonction de leurs besoins spécifiques (par exemple, choisir les statistiques affichées, les filtres par défaut, etc.).
5. **Intégration avec des Outils Tiers :**
Connecter l'application à des outils externes comme **Google Calendar** ou

Microsoft Outlook pour synchroniser les entretiens et notifications.

Conclusion sur les Fonctionnalités Manquantes et les Améliorations :

Le projet a atteint ses **objectifs principaux**, avec une application fonctionnelle qui répond aux besoins des **candidats**, **recruteurs** et **administrateurs**. Bien que certaines fonctionnalités avancées n'aient pas été finalisées, les **améliorations apportées** ont permis d'enrichir l'application et d'améliorer l'expérience utilisateur. Ces fonctionnalités manquantes pourront être intégrées dans des **évolutions futures** pour rendre l'application encore plus complète et performante.

8. Défis Rencontrés et Solutions

Tout au long du développement de l'application de gestion des entretiens, plusieurs défis ont été rencontrés, nécessitant des solutions techniques et organisationnelles adaptées pour garantir la réussite du projet.

1. Problèmes d'intégration entre le backend et le frontend :

L'un des défis majeurs a été l'intégration du backend (.NET) avec le frontend, notamment en ce qui concerne la gestion des formulaires et le téléchargement des CV. Les difficultés sont survenues lors de la synchronisation des données entre la base de données et l'interface utilisateur, ce qui a entraîné des erreurs lors de l'affichage des

informations soumises par les candidats. Pour résoudre ce problème, nous avons mis en place des API RESTful efficaces pour assurer une communication fluide entre les différentes couches de l'application, en veillant à une validation rigoureuse des données côté serveur avant de les transmettre au frontend.

2. Gestion de la performance et de la scalabilité :

Un autre défi a été de garantir la performance de l'application face à un grand nombre de candidats et de recruteurs simultanés. L'application risquait de devenir lente en cas d'augmentation du nombre d'utilisateurs. Pour pallier ce problème, nous avons optimisé les requêtes SQL et mis en place des mécanismes de mise en cache pour accélérer les opérations fréquentes. De plus, une attention particulière a été portée à la gestion des fichiers téléchargés (CV) en utilisant des solutions de stockage cloud pour alléger la charge sur le serveur principal.

3. Conception de l'interface utilisateur (UI/UX) :

L'ergonomie et la simplicité de l'interface ont constitué un autre défi, notamment pour la création d'un formulaire interactif pour les candidats. Les utilisateurs devaient pouvoir naviguer facilement sans confusion. Après avoir effectué des tests utilisateurs, nous avons ajusté l'interface en simplifiant les étapes de soumission du CV et en offrant des retours visuels clairs sur chaque étape du processus, notamment en cas d'erreur.

4. Gestion des utilisateurs et sécurité :

Assurer la sécurité des données personnelles des candidats a été une priorité. Nous avons mis en place des mécanismes d'authentification sécurisés pour les recruteurs, utilisant des techniques comme l'authentification basée sur des tokens (JWT) pour garantir la confidentialité des informations. Pour prévenir les failles de sécurité, des vérifications de validité ont été appliquées aux données entrées par les utilisateurs, notamment lors du téléchargement des CV pour éviter tout fichier malveillant.

9. Tests et Validation

Les tests et la validation de l'application ont été des étapes cruciales pour garantir son bon fonctionnement et son efficacité avant sa mise en production. Un processus rigoureux a été mis en place pour tester les différentes fonctionnalités et assurer une expérience utilisateur optimale.

1. Tests unitaires :

Dès le début du développement, des tests unitaires ont été réalisés pour chaque composant individuel de l'application. Ces tests ont permis de valider le bon fonctionnement des différentes méthodes et logiques métier, notamment pour les processus de soumission de CV, la recherche des candidats par les recruteurs, ainsi que les systèmes de notation et d'évaluation. L'utilisation de frameworks comme MSTest et NUnit a facilité l'automatisation de ces tests, assurant ainsi une couverture maximale du code.

2. Tests d'intégration :

Une fois les tests unitaires effectués, des tests d'intégration ont été réalisés pour vérifier la bonne interaction entre les différents modules de l'application. Les échanges entre le frontend, les API backend et la base de données ont été minutieusement testés pour s'assurer qu'aucune donnée n'était corrompue et que les actions des utilisateurs (candidats, recruteurs, administrateurs) étaient correctement enregistrées dans la base. Des outils comme Postman ont été utilisés pour tester les API et garantir leur stabilité.

3. Tests de performance :

Afin de garantir que l'application reste performante même en cas d'afflux de trafic important, des tests de charge et de performance ont été réalisés. Ces tests ont permis de simuler l'enregistrement simultané de plusieurs candidats et de plusieurs recruteurs, afin d'évaluer la capacité du système à supporter de nombreux utilisateurs.

simultanément sans perte de performance. Des outils comme JMeter ont été utilisés pour effectuer ces tests, et des ajustements ont été faits pour optimiser les requêtes SQL et améliorer la gestion des fichiers.

4. Tests fonctionnels et d'acceptation utilisateur :

Des tests fonctionnels ont été réalisés pour vérifier que toutes les fonctionnalités de l'application répondaient bien aux exigences spécifiées dans le cahier des charges. Ces tests ont porté sur l'ergonomie de l'interface utilisateur, la soumission et le téléchargement des CV, la recherche des candidats, ainsi que la gestion des évaluations par les recruteurs. Parallèlement, des tests d'acceptation utilisateur ont été effectués en collaboration avec des utilisateurs réels (candidats et recruteurs) pour valider l'expérience utilisateur et s'assurer que l'application répondait à leurs attentes.

5. Tests de sécurité :

La sécurité étant un enjeu majeur pour cette application, des tests ont été menés pour identifier d'éventuelles vulnérabilités. Des tests de pénétration ont été réalisés pour vérifier la résistance de l'application face aux attaques courantes, telles que les injections SQL et les attaques par script intersite (XSS). Des mesures de sécurisation des données personnelles ont été mises en place, notamment via le chiffrement des informations sensibles et l'utilisation d'authentifications sécurisées pour les recruteurs.

Au terme de ces étapes de tests et de validation, l'application a pu être lancée en production, avec la certitude qu'elle répondait aux exigences fonctionnelles et techniques du cahier des charges tout en offrant une expérience utilisateur de qualité.

10. Déploiement

Explication du processus de déploiement de l'application et des éventuels obstacles rencontrés.

11. Retour d'Expérience

Le projet de développement de l'application de gestion des entretiens a été une expérience riche et formatrice pour l'ensemble de l'équipe. En analysant les points forts et les axes d'amélioration, plusieurs enseignements précieux ont été tirés, qui pourront être appliqués à de futurs projets.

1. Points forts :

Le principal point fort de ce projet a été la collaboration étroite entre les membres de l'équipe, ce qui a permis de surmonter efficacement les défis techniques et organisationnels. La mise en place d'une méthodologie agile, avec des réunions régulières, a facilité l'organisation des tâches et la répartition des responsabilités, garantissant ainsi un suivi constant de l'avancement du projet. Cette approche a également permis de détecter rapidement les problèmes et de les résoudre dans les délais. D'un point de vue technique, l'intégration des différentes fonctionnalités de l'application (soumission de CV, recherche des candidats, évaluation des profils, gestion des forums) a été une réussite. L'utilisation d'API RESTful et de bonnes pratiques en matière de gestion de base de données a permis de garantir une communication fluide entre le frontend et le backend, ainsi qu'une performance stable de l'application. Les tests rigoureux effectués à chaque étape du projet ont également été un point fort, permettant de valider que chaque fonctionnalité répondait aux exigences et offrait une expérience utilisateur de qualité.

2. Axes d'amélioration :

Cependant, plusieurs axes d'amélioration ont émergé tout au long du projet. Tout d'abord, le temps alloué à la phase de planification initiale était limité, ce qui a occasionné quelques ajustements en cours de développement. Il aurait été bénéfique de passer davantage de temps sur la définition des spécifications techniques et fonctionnelles avant de commencer le développement afin d'éviter des modifications importantes en cours de route. De plus, bien que des tests de performance aient été réalisés, certaines problématiques liées à la gestion de fichiers (tels que des CV volumineux) ont nécessité des ajustements tardifs pour optimiser le stockage et la gestion des ressources serveur. L'optimisation des requêtes SQL, bien qu'efficace, aurait pu être envisagée plus tôt dans le processus de développement afin d'éviter les

ralentissements observés lors des tests de charge. En termes de l'interface utilisateur, bien que celle-ci ait été jugée fonctionnelle, des retours ont suggéré qu'une simplification supplémentaire aurait permis de rendre l'expérience encore plus fluide, notamment pour les candidats qui déposent leur CV. La mise en place de conseils plus clairs sur chaque étape du processus aurait également amélioré l'accessibilité pour les utilisateurs moins familiers avec les outils numériques.

3. Enseignements tirés :

L'un des principaux enseignements tirés de ce projet est l'importance d'une communication constante au sein de l'équipe, en particulier dans un projet de développement complexe où de nombreuses variables sont à prendre en compte. Une gestion de projet agile, avec des révisions régulières et des ajustements au fur et à mesure, est essentielle pour s'adapter aux imprévus et aux nouvelles exigences. Un autre enseignement majeur réside dans l'importance des tests. Bien que des tests unitaires et fonctionnels aient été réalisés, il aurait été judicieux de commencer plus tôt la phase de tests d'acceptation utilisateur pour recueillir des retours précoces et ajuster l'interface en fonction des besoins réels des utilisateurs. Enfin, ce projet a permis de renforcer l'importance de la sécurité des données. L'intégration de mécanismes de sécurité robustes dès le début a évité des risques potentiels et a assuré la confidentialité des informations personnelles des utilisateurs.

En conclusion, bien que des défis aient été rencontrés, ce projet a permis à l'équipe de se renforcer sur plusieurs aspects techniques et organisationnels. Les expériences vécues seront essentielles pour aborder plus efficacement des projets similaires à l'avenir.

Partie Technique

12. Captures d'Écran de l'Application

- Interface de dépôt de CV pour les candidats.
 - Tableau de bord des recruteurs (recherche et évaluation des candidats).
 - Interface administrateur (gestion des forums, entreprises, et statistiques).
-

13. Diagrammes UML

Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation présente les différentes interactions entre les acteurs (candidats, recruteurs, et administrateur) et le système de gestion des entretiens. Il permet de visualiser les fonctionnalités principales offertes par l'application ainsi que les rôles et responsabilités de chaque utilisateur. Ce diagramme facilite la compréhension des besoins fonctionnels du système en mettant en évidence les actions possibles pour chaque type d'utilisateur.

Les cas d'utilisation illustrés incluent le dépôt de CV par les candidats, la recherche et l'évaluation des profils par les recruteurs, ainsi que la gestion des forums, des entreprises et des utilisateurs par l'administrateur.

Diagramme de classes :

Le diagramme de classes représente la structure statique de l'application en modélisant

les différentes classes, leurs attributs, méthodes, ainsi que les relations entre elles. Ce diagramme est essentiel pour comprendre l'architecture du système et la manière dont les différentes entités interagissent entre elles.

Dans le cadre de ce projet, le diagramme de classes met en évidence les entités clés telles que **Candidat**, **Recruteur**, **Administrateur**, **Entreprise**, et **Forum**. Il illustre également les associations, héritages et dépendances entre ces classes, facilitant ainsi la conception et le développement de l'application.

Ce schéma aide non seulement à structurer le code de manière efficace, mais aussi à anticiper les évolutions futures en identifiant les relations et les responsabilités de chaque composant du système.

Diagramme de séquence :

Le diagramme de classes représente la structure statique de l'application en modélisant les différentes classes, leurs attributs, méthodes, ainsi que les relations entre elles. Ce diagramme est essentiel pour comprendre l'architecture du système et la manière dont les différentes entités interagissent entre elles.

Dans le cadre de ce projet, le diagramme de classes met en évidence les entités clés telles que **Candidat**, **Recruteur**, **Administrateur**, **Entreprise**, et **Forum**. Il illustre également les associations, héritages et dépendances entre ces classes, facilitant ainsi la conception et le développement de l'application.

Ce schéma aide non seulement à structurer le code de manière efficace, mais aussi à anticiper les évolutions futures en identifiant les relations et les responsabilités de chaque composant du système.

14. Schéma de la Base de Données

15. Extraits de Code Significatifs

16. Conclusion

Le projet de développement de l'application de gestion des entretiens a permis d'atteindre les objectifs fixés tout en offrant des solutions pratiques aux candidats, recruteurs et administrateurs. En facilitant la soumission de CV, la gestion des candidatures, et l'organisation des entretiens en temps réel, l'application répond efficacement aux besoins d'un forum de recrutement. Grâce à une interface intuitive, des fonctionnalités adaptées aux utilisateurs, et une gestion des données sécurisée, l'application représente un outil précieux pour moderniser et optimiser les processus de recrutement.

Tout au long du projet, l'équipe a fait face à des défis techniques et organisationnels qui ont été résolus grâce à une approche collaborative et une gestion agile. Les tests rigoureux et la validation de chaque fonctionnalité ont permis de garantir la stabilité et la performance de l'application, offrant ainsi une expérience utilisateur optimale. En outre, les retours d'expérience ont permis d'identifier des axes d'amélioration et d'enrichir l'application avec des fonctionnalités supplémentaires, tout en optimisant l'ergonomie et la sécurité.

Les enseignements tirés de ce projet, notamment sur la gestion de projet, les tests, et la sécurité des données, seront précieux pour de futurs développements. Ce projet a permis à l'équipe de renforcer ses compétences techniques et organisationnelles, et de mieux comprendre les besoins réels des utilisateurs finaux.

En conclusion, bien que des défis aient été rencontrés, le projet a été un véritable succès. Il a permis de réaliser une application fonctionnelle, sécurisée et évolutive, prête à être utilisée dans des contextes de forums de recrutement à grande échelle. L'application peut

maintenant être déployée avec confiance, en sachant qu'elle répond aux exigences des utilisateurs tout en étant capable de s'adapter aux évolutions futures.

17. Annexe

1. Introduction	1
2. Contexte et Objectifs	2
Contexte du Projet :	2
Problématique :	3
Objectifs Généraux :	3
Objectifs Spécifiques :	4
Bénéfices Attendus :	5
3. Cahier des Charges	5
1. Contexte et Objectifs du Projet	5
2. Description des Acteurs et Rôles	6
3. Fonctionnalités Détaillées	6
3.1. Côté Candidat :	6
3.2. Côté Recruteur :	7
3.3. Côté Administrateur :	7
4. Interface Utilisateur (UI/UX)	8
4. Technologies Utilisées	8
4.1. Environnement de Développement	9
4.2. Gestion de la Base de Données	9
4.3. Technologies Front-End	10
4.4. Sécurité et Gestion des Utilisateurs	10
4.5. Outils de Développement et de Collaboration	11
4.6. Méthodes de Test	11
4.7. Déploiement	12
Conclusion sur le Choix des Technologies :	12
5. Développement et Répartition des Tâches	13
5.1. Méthodologie de Développement	13
5.2. Phases du Développement	13
5.3. Répartition des Tâches	14
5.4. Collaboration et Outils Utilisés	15

5.5. Gestion des Délais et Adaptation	16
Conclusion sur la Répartition des Tâches :	16
6. Fonctionnalités Implémentées	16
6.1. Fonctionnalités pour les Candidats	17
6.2. Fonctionnalités pour les Recruteurs	17
6.3. Fonctionnalités pour les Administrateurs	18
6.4. Fonctionnalités Additionnelles Développées	20
Conclusion sur les Fonctionnalités Implémentées :	20
7. Fonctionnalités Manquantes et Améliorations	21
7.1. Fonctionnalités Manquantes	21
7.2. Raisons des Fonctionnalités Manquantes	22
7.3. Améliorations Apportées au Cours du Projet	23
7.4. Fonctionnalités à Envisager pour des Évolutions Futures	24
Conclusion sur les Fonctionnalités Manquantes et les Améliorations :	25
8. Défis Rencontrés et Solutions	25
1. Problèmes d'intégration entre le backend et le frontend :	25
2. Gestion de la performance et de la scalabilité :	26
3. Conception de l'interface utilisateur (UI/UX) :	26
4. Gestion des utilisateurs et sécurité :	26
9. Tests et Validation	27
1. Tests unitaires :	27
2. Tests d'intégration :	27
3. Tests de performance :	27
4. Tests fonctionnels et d'acceptation utilisateur :	28
5. Tests de sécurité :	28
10. Déploiement	28
11. Retour d'Expérience	29
1. Points forts :	29
2. Axes d'amélioration :	29
3. Enseignements tirés :	30
12. Captures d'Écran de l'Application	31
13. Diagrammes UML	31
14. Schéma de la Base de Données	31
15. Extraits de Code Significatifs	32
16. Conclusion	32
17. Annexe	32

