

RÉPUBLIQUE TUNISIENNE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Gabès

École Nationale d'Ingénieurs de Gabès

RAPPORT DE PROJET DE FIN D'ÉTUDES

Département de Génie des Communications et Réseaux

Étude et mise en place d'une solution basée sur la blockchain pour la délivrance des attestations de stage

Réalisé par : Ibtissem Aloui

Encadrant académique : [Nom de l'encadrant]

Encadrant professionnel : [Nom de l'encadrant]

Année universitaire : 2024–2025

Dédicace

[Contenu de la dédicace]

Remerciements

[Contenu des remerciements]

Table des matières

Dédicace	i
Remerciements	ii
Liste des Figures	vi
Liste des Tableaux	vii
Liste des Acronymes	viii
Introduction générale	1
1 Étude préliminaire	2
1.1 Introduction	2
1.2 Présentation de l'organisme d'accueil	2
1.3 Problématique	4
1.4 Étude de l'existant	4
1.4.1 Solutions existantes	4
1.4.2 Limites identifiées	5
1.5 Principe et fonctionnement de la technologie blockchain	5
1.5.1 Principe de la blockchain	5
1.5.2 Fonctionnement appliqué à la certification des documents	6
1.6 Solution proposée	6
1.6.1 Description générale de la solution	6
1.6.2 Principe de fonctionnement	6
1.7 Étude comparative des solutions blockchain	8
1.7.1 Présentation d'Ethereum et de l'EVM	8
1.7.2 Langages de développement de smart contracts	8

1.7.3	Testnets compatibles EVM	8
1.7.4	Analyse des outils de développement	9
1.7.5	Justification du choix de Hardhat	10
1.8	Comparaison des frameworks frontend	10
1.8.1	Justification du choix de React.js	11
1.9	Comparaison des frameworks backend	12
1.9.1	Justification du choix de Node.js	13
1.10	Conclusion	13
2	Analyse et conception	14
2.1	Introduction	14
2.2	Analyse des besoins	14
2.2.1	Besoins fonctionnels	14
2.2.2	Besoins non fonctionnels	14
2.3	Architecture globale du système	14
2.4	Modélisation UML	14
2.4.1	Diagramme de cas d'utilisation	14
2.4.2	Diagramme de classes	14
2.4.3	Diagrammes de séquences	14
2.4.4	Diagramme de déploiement	14
2.5	Conclusion	14
3	Réalisation	15
3.1	Introduction	16
3.2	Mise en place de l'environnement de développement	16
3.2.1	Environnement matériel	16
3.2.2	Environnement logiciel	16
3.2.3	Outils et extensions utilisés	16
3.3	Développement du backend	16
3.3.1	API REST pour la gestion des utilisateurs	16
3.3.2	Intégration avec MetaMask	16
3.3.3	Parsing CSV pour l'importation des universités et entreprises	16
3.4	Développement du frontend	16

3.4.1	Interfaces graphiques selon les rôles	16
3.5	Développement des contrats intelligents	16
3.5.1	Contrat Auth pour la gestion des rôles	16
3.5.2	Contrat Attestation pour la génération et validation	16
3.5.3	Événements et logique de validation	16
3.6	Génération et validation des attestations	16
3.6.1	Génération des PDF avec QR Code	16
3.6.2	Hachage et stockage sur IPFS	16
3.6.3	Page de vérification HTML	16
3.7	Tests et validation	16
3.7.1	Tests unitaires des contrats intelligents	16
3.7.2	Tests fonctionnels des interfaces et API	16
3.8	Sécurité et robustesse	16
3.8.1	Audit des contrats intelligents	16
3.8.2	Protection des données et authentification	16
3.9	Déploiement	16
3.9.1	Procédure d'installation et configuration	16
3.9.2	Contraintes techniques de déploiement	16
3.10	Conclusion	16
	Conclusion générale et perspectives	17
	Webographie	18
	Annexes	19
	Résumé	20

Table des figures

1	Logo du Groupe Chimique Tunisien (GCT)	3
2	Organigramme du GCT	3

Liste des tableaux

1	Comparaison des principaux testnets compatibles EVM	9
2	Comparaison des principaux frameworks de développement blockchain	10
3	Comparaison des principaux frameworks frontend	11
4	Comparaison des principaux frameworks backend	12

Liste des Acronymes

- IPFS : InterPlanetary File System
- API : Application Programming Interface
- REST : Representational State Transfer
- CSV : Comma-Separated Values

Introduction générale

Avec la montée en puissance des technologies numériques, les méthodes traditionnelles de gestion des documents officiels sont devenues de moins en moins adaptées aux exigences actuelles d'efficacité, de sécurité et de vérifiabilité. C'est notamment le cas des attestations de stage, souvent traitées manuellement ou de manière centralisée, avec des risques de falsification, de perte ou de délais dans leur validation.

La technologie blockchain se présente comme une solution innovante à ces problématiques, offrant des garanties d'authenticité, de traçabilité et d'intégrité grâce à son caractère décentralisé et immuable. Elle permet d'automatiser et de sécuriser l'ensemble du cycle de vie d'un document sans dépendre d'un tiers de confiance.

C'est dans ce contexte que s'inscrit ce projet de fin d'études. Réalisé au sein de la division informatique du Groupe Chimique Tunisien (GCT) à Gabès, il a pour objectif de concevoir et développer une plateforme web pour la gestion des attestations de stage basée sur la blockchain. Cette solution permet aux entreprises de générer des attestations de manière sécurisée, aux établissements universitaires de les valider, et aux tiers (recruteurs, institutions) de les vérifier à l'aide d'un identifiant unique ou d'un code QR imprimé sur l'attestation. Ce code redirige vers une preuve en ligne hébergée sur le réseau IPFS et liée à une transaction blockchain.

La plateforme repose sur une architecture distribuée utilisant des technologies telles que Node.js, React.js, Solidity, Hardhat, IPFS et MySQL. Elle assure l'automatisation complète du cycle de génération, validation et consultation des attestations. Bien qu'initiée dans un cadre pédagogique, la solution est pensée pour évoluer vers une utilisation concrète en environnement professionnel, répondant aux exigences d'intégrité et de pérennité.

Ce rapport est structuré en trois chapitres :

- Le premier chapitre présente l'organisme d'accueil, analyse l'existant et introduit la solution retenue.
- Le deuxième chapitre traite de l'analyse des besoins et de la conception du système.
- Le troisième chapitre couvre les phases de développement, test, sécurisation et déploiement de la solution.

Nous concluons par une synthèse des résultats obtenus et des perspectives d'évolution du projet.

Chapitre 1

Étude préliminaire

1.1 Introduction

Ce chapitre présente le cadre général du projet ainsi que l'environnement professionnel dans lequel il a été réalisé. Il expose la problématique, les solutions existantes et la technologie retenue.

1.2 Présentation de l'organisme d'accueil

Le Groupe Chimique Tunisien (GCT) 1 est une entreprise publique tunisienne fondée en 1953, spécialisée dans la transformation du phosphate en produits à forte valeur ajoutée tels que l'acide phosphorique et les engrais. Le GCT contribue significativement à l'économie nationale avec environ 4200 employés.

La Tunisie est ainsi le deuxième pays mondial en matière de valorisation du phosphate naturel, avec 85% de la production transformée.

Le projet a été réalisé au sein de la Direction Régionale des Usines de Gabès, plus précisément dans la division informatique, créée en 1981. Elle assure :

- Maintenance informatique ;
- Administration système et réseau ;
- Développement d'applications internes.



FIGURE 1 – Logo du Groupe Chimique Tunisien (GCT)

La structure de l'entreprise se définit comme l'ensemble des relations hiérarchiques et fonctionnelles entre les divers services et le personnel. Le GCT est structuré comme le montre l'organigramme de la figure 2

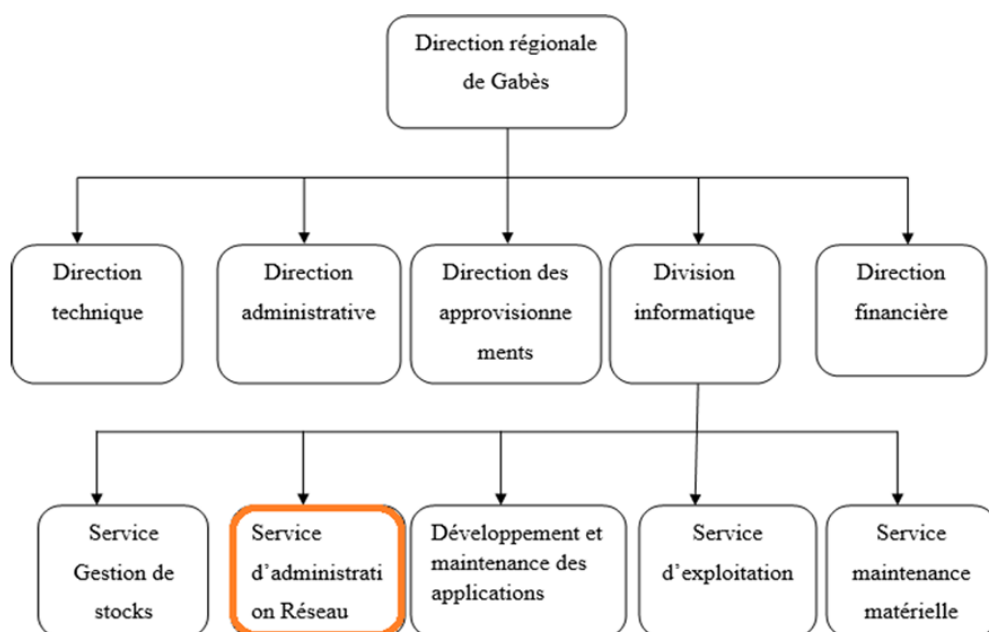


FIGURE 2 – Organigramme du GCT

La division informatique comprend trois services :

- Service bureautique.
- Service administration réseau.
- Centre informatique.

1.3 Problématique

Les attestations de stage sont des documents importants dans le parcours académique et professionnel des étudiants. Remises en format papier ou PDF à la fin du stage, elles peuvent cependant être facilement perdues, modifiées ou falsifiées. Leur vérification par les établissements ou les recruteurs est souvent lente, demande du temps et peut être sujette à des erreurs.

L'absence d'un système centralisé et automatisé rend la gestion de ces documents difficile : manque de traçabilité, lourde charge administrative et risque élevé de fraude. Cela affecte aussi bien les universités que les entreprises.

Face à ces limites, une question se pose : Comment garantir l'authenticité, le suivi et la vérification des attestations de stage tout en réduisant les efforts manuels ?

Dans ce cadre, la technologie blockchain, grâce à sa structure décentralisée, sécurisée et transparente, apparaît comme une solution pertinente pour automatiser la génération, la validation et la consultation des attestations de manière fiable.

1.4 Étude de l'existant

1.4.1 Solutions existantes

En Tunisie, la gestion des attestations de stage repose essentiellement sur des procédures classiques. Les entreprises d'accueil génèrent des documents au format papier ou numérique à l'issue du stage, qui sont ensuite remis aux étudiants pour transmission à leurs établissements. La validation repose souvent sur des échanges par courrier électronique ou sur des remises physiques, impliquant une intervention manuelle de l'administration universitaire.

Certaines institutions universitaires ont mis en place des systèmes numériques internes pour faciliter le suivi des stages, notamment pour l'affectation des encadrants ou la validation académique. Ces plateformes permettent généralement aux étudiants de déposer leurs rapports de stage et, dans certains cas, de télécharger des attestations validées. Cependant, ces systèmes restent isolés, souvent non interopérables avec des acteurs externes, et ne prévoient pas de mécanisme formel permettant une vérification automatisée de l'authenticité des documents émis.

À l'échelle nationale, aucun cadre standardisé n'existe pour garantir la sécurité, la traçabilité

ou l'accès tiers aux attestations. Cela engendre une hétérogénéité dans les pratiques, une charge administrative importante pour les personnels impliqués, et une difficulté pour les tiers à valider l'authenticité des documents reçus.

1.4.2 Limites identifiées

L'analyse des pratiques actuelles fait ressortir plusieurs limites importantes :

- L'absence de vérification automatisée de l'authenticité des attestations ;
- La dépendance à des échanges manuels ou par courriel, générateurs de délais ;
- Le risque de falsification des documents numériques transmis en dehors de tout système sécurisé ;
- L'inexistence d'un accès en ligne aux attestations pour des acteurs tiers autorisés (recruteurs, institutions partenaires) ;
- Le manque de standardisation entre les établissements ;
- L'impossibilité de tracer l'historique des validations de manière infalsifiable.

Ces constats soulignent la nécessité de mettre en œuvre une solution moderne, interopérable et fiable, capable de répondre aux exigences de sécurité, d'intégrité et de simplicité de vérification.

1.5 Principe et fonctionnement de la technologie blockchain

1.5.1 Principe de la blockchain

La blockchain est une technologie de registre distribué permettant de stocker et transmettre des données de manière sécurisée, transparente et sans dépendance à un tiers de confiance. Chaque bloc contient des données horodatées, validées par un consensus, et liées cryptographiquement au bloc précédent, formant une chaîne inaltérable [1].

Une fois qu'une donnée est enregistrée dans la blockchain, elle devient pratiquement immuable : toute tentative de modification nécessiterait de réécrire tous les blocs suivants, ce qui est techniquement irréalisable dans un système bien sécurisé [2]. Cette immuabilité garantit l'intégrité des données et constitue un pilier central de la fiabilité de cette technologie.

1.5.2 Fonctionnement appliqué à la certification des documents

Dans le contexte de la certification de documents, la blockchain est utilisée pour inscrire une empreinte numérique (appelée hachage) d'un document. Cette empreinte, unique et générée à partir du contenu du fichier, est enregistrée dans la blockchain, accompagnée d'un horodatage et d'une signature cryptographique de l'émetteur [3].

Ce procédé permet, sans révéler le contenu du document, de vérifier son authenticité. Toute personne disposant du fichier peut en recalculer le hachage et le comparer à celui inscrit dans la blockchain. Si les deux correspondent, cela prouve que le document est authentique et n'a pas été modifié depuis son enregistrement [4].

Cette approche offre plusieurs avantages :

- *Intégrité* : toute altération modifie l'empreinte et rend la vérification invalide ;
- *Authenticité* : la signature cryptographique permet d'identifier l'émetteur ;
- *Traçabilité* : chaque action est enregistrée de manière permanente ;
- *Accessibilité* : la vérification peut se faire à distance via une interface web ;
- *Confidentialité* : seuls les hachages sont visibles, le contenu reste privé [5].

Cette méthode s'impose donc comme une réponse fiable aux problèmes de fraude, de lenteur de traitement et de vérification des documents numériques.

1.6 Solution proposée

1.6.1 Description générale de la solution

La solution proposée repose sur une plateforme web permettant de gérer tout le cycle de vie des stages : de l'inscription jusqu'à la génération et vérification des attestations. Elle intègre des fonctionnalités adaptées aux différents profils utilisateurs (étudiants, encadrants, responsables, tiers) avec une logique de validation progressive.

1.6.2 Principe de fonctionnement

Le processus de fonctionnement peut être résumé selon les étapes suivantes :

1. **Inscription et connexion avec MetaMask** : l'utilisateur se connecte via son portefeuille MetaMask. Une signature permet d'authentifier son identité sans mot de passe, selon les rôles enregistrés dans un contrat intelligent.
2. **Proposition de sujet de stage** : les étudiants peuvent soumettre un ou plusieurs sujets via leur tableau de bord. Un e-mail est automatiquement envoyé aux deux encadrants (universitaire et professionnel), qui peuvent ensuite accepter ou refuser la proposition directement depuis leur interface respective.
3. **Création du stage** : si les deux encadrants valident le sujet, le stage est officiellement créé et rattaché à l'étudiant dans le système. Le statut de chaque proposition est mis à jour et suivi en temps réel.
4. **Soumission du rapport** : à la fin du stage, l'étudiant soumet son rapport via l'interface. Chaque encadrant reçoit une notification et peut valider ou rejeter le contenu. Des rappels automatiques sont envoyés si aucun retour n'est effectué dans un délai donné.
5. **Déblocage par un tiers** : en cas de blocage (absence de validation), un utilisateur tiers par entité est informé de retard et peut intervenir et valider le rapport, garantissant ainsi la fluidité du processus sans intervention manuelle complexe.
6. **Génération de l'attestation** : une fois le rapport validé, l'attestation est générée au format PDF. Elle contient un QR code pointant vers une page de vérification en ligne.
7. **Certification sur la blockchain** : une empreinte (hash) de l'attestation est calculée et inscrite dans la blockchain via un contrat intelligent. Cela permet une vérification ultérieure de son intégrité.
8. **Vérification publique** : tout tiers (recruteur, institution) peut vérifier l'attestation depuis une page dédiée, en important le PDF ou en scannant le QR code. La correspondance entre l'empreinte et celle enregistrée sur la blockchain permet de valider l'authenticité du document.
9. **Gestion de l'historique et des notifications** : toutes les actions (proposition, validation, refus, génération, déblocage) sont historisées. Des e-mails sont envoyés à chaque étape clé, assurant un suivi transparent et automatisé pour tous les utilisateurs.

1.7 Étude comparative des solutions blockchain

1.7.1 Présentation d'Ethereum et de l'EVM

Ethereum est l'une des blockchains les plus utilisées dans le monde, notamment grâce à sa prise en charge des contrats intelligents (smart contracts) et des applications décentralisées (dApps). Contrairement à Bitcoin, qui est centré sur les transactions monétaires, Ethereum permet le déploiement de programmes autonomes qui s'exécutent automatiquement lorsque certaines conditions sont remplies [?].

EVM (Ethereum Virtual Machine) est une machine virtuelle intégrée à Ethereum. Elle permet d'exécuter des contrats intelligents de manière standardisée, garantissant ainsi la compatibilité entre plusieurs blockchains dites « EVM-compatibles », comme Polygon, BSC, Avalanche, etc. Cela facilite le développement, la migration et l'interopérabilité des applications [?].

1.7.2 Langages de développement de smart contracts

Le langage dominant pour le développement de contrats intelligents sur Ethereum et ses compatibles est **Solidity**, un langage orienté objet spécialement conçu pour la blockchain. Il est largement utilisé grâce à :

- Sa compatibilité directe avec l'EVM [?];
- Une documentation riche [?];
- Un écosystème de développeurs très actif [?].

D'autres blockchains utilisent des langages alternatifs comme *Rust* (Solana) ou *Move* (Aptos), mais ceux-ci nécessitent un apprentissage plus complexe et disposent de moins de ressources pédagogiques [?].

1.7.3 Testnets compatibles EVM

Les **testnets** sont des réseaux de test qui imitent le fonctionnement des blockchains réelles mais sans coût financier. Ils permettent de tester, déployer et vérifier les contrats intelligents avant de les publier sur un mainnet.

Les critères principaux pour le choix d'un testnet sont développés dans le tableau suivant :

- **Performance** : capacité à traiter rapidement les transactions ;
- **Sécurité** : robustesse du réseau face aux attaques et failles ;
- **Interopérabilité** : compatibilité avec d'autres blockchains via l'EVM ;
- **Documentation** : qualité des ressources disponibles ;
- **Coût des transactions** : idéalement gratuit sur un testnet ;
- **Facilité de migration** : possibilité de transférer le code vers un mainnet sans modification majeure.

TABLE 1 – Comparaison des principaux testnets compatibles EVM

Testnet	Compatibilité EVM	Performance	Sécurité	Interopérabilité	Documentation	Migration
BSC Testnet	Oui	Élevée	Bonne	Excellente	Bonne	Facile
Fantom Testnet	Oui	Très élevée	Haute	Bonne	Moyenne	Facile
Avalanche Fuji	Oui	Très élevée	Très haute	Très bonne	Bonne	Facile
Celo Alfajores	Oui	Moyenne	Bonne	Bonne	Bonne	Facile

1.7.4 Analyse des outils de développement

Le développement de contrats intelligents et leur test nécessitent des environnements adaptés. Plusieurs frameworks sont disponibles, notamment :

- **Hardhat** : simulateur local très performant, orienté développeur, avec intégration facile des tests [?];
- **Truffle** : anciennement populaire, mais plus lourd et moins flexible [?];
- **Remix** : IDE en ligne pratique pour les démonstrations ou tests rapides [?];
- **Foundry** : outil moderne performant, mais basé sur Rust et plus complexe à maîtriser [?].

Hardhat se distingue par sa compatibilité avec Ethers.js, sa prise en charge des réseaux EVM, son système de plugins riche, et surtout la possibilité d'exécuter un réseau local sans frais,

sans dépendre d'un faucet de testnet devenu peu fiable [?].

TABLE 2 – Comparaison des principaux frameworks de développement blockchain

Framework	Langage supporté	Réseau local intégré	Facilité de test	Support EVM / Solidity
Hardhat	Solidity	Oui (Hardhat Network)	Très élevée (tests automatisés)	Excellente
Truffle	Solidity	Oui (Ganache)	Moyenne (moins flexible)	Bonne
Remix IDE	Solidity	Non	Très bonne pour test simple	Très bonne
Foundry	Solidity / Rust	Oui	Excellente mais complexe	Excellente (mais plus technique)

1.7.5 Justification du choix de Hardhat

Notre projet utilise **Hardhat** comme environnement principal de développement pour les raisons suivantes :

- Il permet de simuler un réseau local complet sans connexion Internet, évitant les interruptions dues à la saturation des faucets [?];
- Il est totalement gratuit, respectant la contrainte de ne pas utiliser de testnet payant [?];
- Il offre une intégration directe avec les bibliothèques comme Ethers.js, utilisées dans notre backend en Node.js [?];
- Il facilite l'automatisation des tests, améliorant la robustesse de nos contrats intelligents avant déploiement [?].

1.8 Comparaison des frameworks frontend

Le choix d'un framework frontend dépend des besoins spécifiques du projet, tels que la performance, la facilité d'intégration avec d'autres systèmes, et la réactivité. Voici une comparaison

des principaux frameworks frontend utilisés pour le développement d'applications web dans le domaine de la blockchain.

- **React.js** : Très populaire, permettant de créer des interfaces utilisateur dynamiques. Il offre une gestion efficace de l'état et est largement utilisé pour les applications décentralisées (dApps) grâce à son intégration facile avec des bibliothèques comme Web3.js ou Ethers.js.
- **Vue.js** : Un autre framework JavaScript qui est apprécié pour sa simplicité d'utilisation et sa courbe d'apprentissage moins abrupte que React. Il est également bien adapté pour les projets blockchain nécessitant une gestion fluide des interfaces.
- **Angular** : Un framework robuste de Google, idéal pour les applications de grande envergure. Cependant, il peut être plus complexe à maîtriser par rapport à React ou Vue.
- **Svelte** : Un framework plus récent qui compile les composants en JavaScript pur à la compilation, ce qui peut améliorer les performances, en particulier pour des applications décentralisées légères.

TABLE 3 – Comparaison des principaux frameworks frontend

Framework	Courbe d'apprentissage	Performance	Écosystème	Facilité d'intégration avec blockchain
React.js	Moyenne	Élevée	Très riche	Excellente [?]
Vue.js	Faible	Moyenne	Moyen	Bonne [?]
Angular	Élevée	Élevée	Très riche	Bonne [?]
Svelte	Faible	Très élevée	Moyen	Excellente [?]

1.8.1 Justification du choix de React.js

Pour notre projet, **React.js** a été choisi pour plusieurs raisons :

- Large communauté et support actif ;
- Facilité d'intégration avec des bibliothèques JavaScript comme Web3.js et Ethers.js ;
- Performances optimisées grâce à son approche du DOM virtuel ;

- Flexibilité et réutilisabilité des composants, idéales pour des applications décentralisées dynamiques.

1.9 Comparaison des frameworks backend

Le choix du framework backend est crucial pour garantir la robustesse, la sécurité et la capacité à gérer des interactions complexes avec la blockchain. Voici une comparaison des principaux frameworks backend utilisés dans les projets blockchain.

- **Node.js** : Un environnement JavaScript asynchrone idéal pour les applications qui nécessitent une exécution rapide et une gestion de nombreux utilisateurs simultanés. Il s'intègre bien avec les technologies de blockchain via des bibliothèques comme Web3.js ou Ethers.js.
- **Express.js** : Basé sur Node.js, ce framework minimaliste permet de créer rapidement des API RESTful. Il est idéal pour les projets blockchain qui nécessitent une architecture simple et scalable.
- **Django** : Un framework Python robuste qui fournit une structure sécurisée pour le développement d'applications backend. Bien adapté pour les applications qui nécessitent une logique de traitement complexe avant d'interagir avec la blockchain.
- **Spring Boot** : Un framework Java puissant, utilisé principalement dans les grandes entreprises. Il est particulièrement adapté aux applications blockchain nécessitant des microservices robustes et une forte sécurité.

TABLE 4 – Comparaison des principaux frameworks backend

Framework	Langage	Performance	Écosystème	Facilité d'intégration avec blockchain
Node.js	JavaScript	Élevée	Très riche	Excellente [?]
Express.js	JavaScript	Très élevée	Très riche	Excellente [?]
Django	Python	Moyenne	Riche	Bonne [?]
Spring Boot	Java	Élevée	Très riche	Bonne [?]

1.9.1 Justification du choix de Node.js

Pour notre backend, nous avons opté pour **Node.js** en raison de :

- Sa rapidité et sa capacité à gérer des connexions simultanées de manière efficace grâce à son modèle non-bloquant ;
- Une large communauté et un écosystème riche de modules npm pour interagir facilement avec des blockchains via Web3.js ou Ethers.js ;
- Une intégration transparente avec les technologies frontend comme React.js, assurant une architecture homogène et performante.

1.10 Conclusion

Le premier chapitre a permis d'établir le cadre du projet . Après avoir analysé les solutions existantes, il est apparu clairement que les méthodes traditionnelles présentent des limites notables en termes de sécurité et de fiabilité. Ce constat a souligné la nécessité d'adopter des technologies innovantes, telles que la blockchain, pour garantir une gestion plus sécurisée et efficace des attestations.

Chapitre 2

Analyse et conception

2.1 Introduction

2.2 Analyse des besoins

2.2.1 Besoins fonctionnels

2.2.2 Besoins non fonctionnels

2.3 Architecture globale du système

2.4 Modélisation UML

2.4.1 Diagramme de cas d'utilisation

2.4.2 Diagramme de classes

2.4.3 Diagrammes de séquences

2.4.4 Diagramme de déploiement

2.5 Conclusion

Chapitre 3

Réalisation

3.1 Introduction

3.2 Mise en place de l'environnement de développement

3.2.1 Environnement matériel

3.2.2 Environnement logiciel

Node.js

Hardhat

React.js

MySQL

IPFS Desktop pour le stockage décentralisé

ngrok pour l'exposition du backend

3.2.3 Outils et extensions utilisés

MetaMask

Postman

Git et GitHub

Visual Studio Code

3.3 Développement du backend

3.3.1 API REST pour la gestion des utilisateurs

Conclusion générale et perspectives

[Texte de conclusion générale ici...]

Webographie

[1] <https://www.ibm.com/think/topics/blockchain-for-business>

(consulté le 14 février 2025)

[2] <https://www.flur.ee/fluree-blog/why-does-blockchain-immutability-matte>

(consulté le 2 février 2025)

[3] <https://help.verisart.com/en/articles/5050610-how-do-blockchain-certif>

(consulté le 02 mars 2025)

[4] <https://www.alibabacloud.com/blog/how-to-use-blockchain-to-ensure-the-599875>

(consulté le 10 avril 2025)

[5] <https://en.wikipedia.org/wiki/Blockchain>

(consulté le 10 février 2025)

Annexes

- Annexe A : Extraits de code
- Annexe B : Captures d'écran
- Annexe C : Résultats de tests

Résumé

Résumé en français

[À rédiger]

Abstract in English

[À rédiger]

Résumé en arabe

[À rédiger]