Mohamed Khider University of Biskra
Faculty of Exact Sciences
Department of Computer Science

MASTER THESIS

Speciality : Bachelor - Management Information Systems for Local Administration (Professional) - Ref : P1792/2026

TITLE

A Blockchain-Based Platform for Managing Grant Applications for Young Entrepreneurs

Prepared and defended by

ABDELHAY REGHIOUA

Mohamed zakaria BENBRAIKA

Hadjer LAHOUEL

Defense date : 09/06/2026

DEFENSE COMMITTEE

Full Name	Grade	Role
Mohamed SAHRAOUI	MCA	**President**
Sadek Laanani	MAA	**Supervisor**
Hamza BELAICHE	MAA	**Examiner**
amina djoudi	MCB	**Incubator Representative**

Academic Year : 2025/2026

Acknowledgement

First and foremost, we thank God for granting us the strength and patience to successfully complete this project.

We express our deepest gratitude to our thesis supervisors, **Dr. Sadok LANANI** and **Prof. Laid KAHLOUL**, who believed in us and supported us unconditionally. Their availability, insightful guidance, and objectivity have been invaluable to us. We can never thank them enough for their attentiveness in all circumstances, despite their numerous scientific responsibilities.

We would like to express our gratitude to all the staff of the university's **PFE/STARTUP** initiative for their vision and effort. They gave us the opportunity to take on new challenges in our academic lives and to push beyond our limits as students by providing us with a small yet meaningful entrepreneurial experience.

Finally, we wish to express our most heartfelt gratitude to our families, friends, and colleagues who have supported us throughout our journey. Their encouragement, advice, and unwavering support have been invaluable in helping us reach our goals.

We would also like to thank all those who played a role in our journey but have not been specifically mentioned. Your support has not gone unnoticed, and we are sincerely grateful for everything you have done to help us reach this point.

Abstract

Youth entrepreneurship has become a strategic priority in Algeria as the country seeks to diversify its economy beyond hydrocarbons and to absorb growing youth unemployment. Despite significant public investment in support mechanisms NESDA, CNAC, ANGEM, and the national startup ecosystem persistent barriers related to bureaucracy, financing gaps, and inadequate digital infrastructure continue to limit the impact of these policies.

This thesis investigates how blockchain technology can strengthen the trust and traceability of entrepreneurial support processes, with a particular focus on digital document certification. The study presents **GRANTA**, a blockchain-based document verification platform built on Hyperledger Fabric 2.5, Next.js 15, TypeScript, SQLite, and IPFS. GRANTA supports a complete workflow covering candidate application submission, multi-level review, certificate registration, NFT minting, and tamper-proof public verification.

The system architecture relies on a hybrid trust model: the web layer ensures usability, a relational database ensures operational consistency, IPFS provides content-addressed decentralised file storage, and Hyperledger Fabric anchors certificate hashes immutably on a permissioned ledger using Raft consensus. Chaincode transactions including provide end-to-end auditability.

Performance evaluation using Hyperledger Caliper demonstrates that the platform sustains a practical throughput of up to 228 TPS under volume-controlled conditions with zero transaction failures, confirming its suitability for institutional certificate registration workloads.

The results show that blockchain-based certification can meaningfully complement existing entrepreneurial support infrastructure in Algeria by reducing fraud risk, accelerating verification, and providing a transparent audit trail for institutional actors.

Keywords: *Youth entrepreneurship, Algeria, blockchain, Hyperledger Fabric, digital certification, IPFS, smart contracts, GRANTA platform, document verification, NFT.*

Résumé

L'entrepreneuriat des jeunes est devenu une priorité stratégique en Algérie dans le cadre de la diversification économique et de la réduction du chômage des jeunes. Malgré des investissements publics importants notamment au travers de l'ANADE, de la CNAC, de l'ANGEM et de l'écosystème national des startups des obstacles persistants liés à la bureaucratie, aux déficits de financement et à une numérisation insuffisante continuent de limiter l'impact de ces politiques.

Ce mémoire examine comment la technologie blockchain peut renforcer la confiance et la traçabilité des processus d'accompagnement entrepreneurial, en se concentrant sur la certification numérique de documents. L'étude présente **GRANTA**, une plateforme

de vérification de documents basée sur la blockchain, construite avec Hyperledger Fabric 2.5, Next.js 15, TypeScript, SQLite et IPFS. GRANTA prend en charge un workflow complet comprenant la soumission des dossiers de candidature, la révision multi-niveaux, l'enregistrement des certificats, la création de NFT et la vérification publique infalsifiable.

L'architecture du système repose sur un modèle de confiance hybride : la couche web assure l'ergonomie, une base de données relationnelle garantit la cohérence opérationnelle, IPFS offre un stockage décentralisé adressé par contenu, et Hyperledger Fabric ancre les empreintes cryptographiques des certificats de manière immuable sur un registre distribué autorisé utilisant le consensus Raft. Les transactions chaincode notamment assurent une auditabilité de bout en bout.

L'évaluation des performances à l'aide de Hyperledger Caliper montre que la plateforme maintient un débit pratique allant jusqu'à 228 TPS dans des conditions de charge volumique, avec zéro transaction échouée, confirmant son aptitude aux charges de travail institutionnelles d'enregistrement de certificats.

Les résultats démontrent que la certification basée sur la blockchain peut utilement compléter les infrastructures existantes de soutien à l'entrepreneuriat en Algérie, en réduisant les risques de fraude, en accélérant la vérification et en fournissant une piste d'audit transparente pour les acteurs institutionnels.

Mots-clés : *Entrepreneuriat des jeunes, Algérie, blockchain, Hyperledger Fabric, certification numérique, IPFS, contrats intelligents, plateforme GRANTA, vérification de documents, NFT.*

ملخص

باتت ريادة أعمال الشباب أولويةً استراتيجيةً في الجزائر في إطار مساعي التنويع الاقتصادي والحدّ من البطالة في صفوف الشباب. وعلى الرغم من الاستثمارات العامة الضخمة — لا سيما عبر الوكالة الوطنية لتطوير ودعم المقاولاتية (أناد)، والصندوق الوطني للتأمين على البطالة ،(CNAC) والوكالة الوطنية لتسيير القرض المصغر ،(ANGEM) ومنظومة الشركات الناشئة الوطنية — تُعيق عقبات هيكلية مستمرة تتعلق بالبيروقراطية وفجوات التمويل وضعف الرقمنة جدوى هذه السياسات.

تبحث هذه الأطروحة في كيفية توظيف تقنية سلسلة الكتل لتعزيز الثقة وإمكانية التتبع في مسارات دعم المقاولاتية، مع التركيز على توثيق الوثائق الرقمية. تُقدّم الدراسة منصة **GRANTA**، وهي منصة للتحقق من الوثائق مبنية على سلسلة الكتل باستخدام Hyperledger Fabric 5.2 وNext.js 15 وTypeScript وSQLite وIPFS. تدعم المنصة سير عمل متكاملاً يشمل تقديم ملفات الترشح، والمراجعة متعددة المستويات، وتسجيل الشهادات، وإصدار الرموز غير القابلة للاستبدال ،(NFT) والتحقق العلني المقاوم للتزوير.

يرتكز النظام على نموذج ثقة هجين: تضمن طبقة الويب سهولة الاستخدام، وتكفل قاعدة البيانات العلائقية الاتساق التشغيلي، ويوفر IPFS تخزيناً لامركزياً مُعنوَناً بالمحتوى، فيما يُثبّت Fabric Hyperledger البصمات التشفيرية للشهادات بصورة غير قابلة للتغيير على سجل موزع مصرّح به يستخدم خوارزمية توافق .Raft وتوفر معاملات السلسلة البرمجية (Chaincode) — بما فيها قابلية تدقيق كاملة من طرف إلى طرف.

أثبت تقييم الأداء باستخدام Caliper Hyperledger أن المنصة تحافظ على إنتاجية عملية تصل إلى 228 معاملة في الثانية في ظروف التحكم بالحجم مع انعدام فشل المعاملات، مما يؤكد ملاءمتها لأعباء عمل المؤسسات في تسجيل الشهادات.

تُبيّن النتائج أن التوثيق القائم على سلسلة الكتل يُكمّل البنية التحتية القائمة لدعم المقاولاتية في الجزائر بشكل فعّال، من خلال الحدّ من مخاطر التزوير وتسريع عمليات التحقق وتوفير مسار تدقيق شفاف للجهات المؤسسية.

الكلمات المفتاحية: ريادة أعمال الشباب، الجزائر، سلسلة الكتل، Hyperledger ،Fabric الشهادات الرقمية، ،IPFS العقود الذكية، منصة ،GRANTA التحقق من الوثائق، .NFT

Contents

General introduction **1**

1 General Study and State of the Art **4**
1.1 Chapter Introduction 4
1.2 Youth Entrepreneurship 4
1.2.1 Definition of Entrepreneurship 4
1.2.2 Economic and Social Importance 4
1.2.3 The Role of Entrepreneurship in Job Creation 5
1.2.4 Specifics of Youth Entrepreneurship 5
1.3 Youth Entrepreneurship in Algeria 5
1.3.1 Socio-economic Situation of Algeria 5
1.3.2 Youth Unemployment Figures 6
1.3.3 Public Policies to Support Entrepreneurship 6
1.3.4 Obstacles Encountered by Young Entrepreneurs 6
1.3.5 Importance of Digital Transformation 6
1.4 Support Mechanisms in Algeria 6
1.4.1 NESDA (ex-ANSEJ) 6
1.4.2 CNAC 7
1.4.3 ANGEM 8
1.4.4 Ministry of Knowledge Economy, Startups, and Micro-Enterprises . 8
1.4.5 Self-Employed Entrepreneur (Law 22-23) 8
1.5 The Role of Insurance in Entrepreneurial Projects 9
1.5.1 Definition and Objectives of Insurance 9
1.5.2 The Insurance Sector in Algeria 9
1.5.3 Types of Insurance for Businesses 9
1.5.4 The Importance of Insurance in Securing Investments 9
1.5.5 Algerian Insurance Companies 9
1.6 Algerian Legal and Regulatory Framework 9
1.7 Digitalization of Public Administration 10
1.7.1 E-Government in Algeria 10
1.7.2 Advantages of Digitalization 10
1.7.3 Limitations of Manual Procedures 10
1.7.4 Contribution of Digital Platforms 11

1.8 State of the Art . . . 11
1.9 Chapter Conclusion . . . 11

2 Blockchain Technology and Its Application in Secure Digital Certification 12
2.1 Introduction . . . 12
2.2 Fundamentals of Blockchain Technology . . . 12
2.2.1 Definition of Blockchain . . . 12
2.2.2 Historical Evolution of Blockchain . . . 12
2.2.3 Core Principles . . . 13
2.3 Blockchain Architecture . . . 14
2.3.1 Blocks and Transactions . . . 14
2.3.2 Cryptographic Hash Functions . . . 14
2.3.3 Digital Signatures and Public-Key Cryptography . . . 14
2.3.4 Distributed Ledger Structure . . . 15
2.4 Consensus Mechanisms . . . 15
2.4.1 Proof of Work (PoW) . . . 15
2.4.2 Proof of Stake (PoS) . . . 15
2.4.3 Practical Byzantine Fault Tolerance (PBFT) . . . 15
2.4.4 Raft Consensus Algorithm . . . 15
2.5 Types of Blockchain Networks . . . 17
2.5.1 Public Blockchain . . . 17
2.5.2 Private Blockchain . . . 17
2.5.3 Consortium Blockchain . . . 17
2.6 Smart Contracts . . . 18
2.6.1 Concept and Functionality . . . 18
2.6.2 Smart Contracts in Enterprise Systems . . . 18
2.7 Enterprise Blockchain Frameworks . . . 19
2.7.1 Hyperledger Fabric Architecture . . . 19
2.7.2 Permissioned Blockchain Model . . . 19
2.8 Blockchain and Decentralized Storage . . . 20
2.8.1 IPFS Overview . . . 20
2.8.2 On-Chain vs Off-Chain Storage . . . 20
2.9 Application of Blockchain in Digital Certification Systems . . . 21
2.9.1 Traditional Certificate Verification Challenges . . . 21
2.9.2 Blockchain-Based Certificate Anchoring . . . 21
2.10 Advantages and Limitations of Blockchain Technology . . . 23
2.11 Conclusion . . . 23

3 System Design and Conception 24
3.1 Introduction . . . 24
3.2 Global Functional View . . . 24
3.2.1 System actors . . . 24
3.2.2 Use case diagram . . . 25

3.3 Static Design View . . . 26
3.3.1 Class diagram . . . 26
3.4 Cross-Workflow Architecture . . . 28
3.4.1 Overview . . . 28
3.4.2 Architectural layers . . . 29
3.4.3 Workflow integration . . . 30
3.4.4 Key design patterns . . . 31
3.4.5 Key guarantees . . . 31
3.5 WF01 — Candidate Registration and Grant Application . . . 31
3.5.1 Overview . . . 31
3.6 WF02 — Supervisor Multi-Level Review . . . 33
3.6.1 Overview . . . 33
3.7 WF03 — Final Supervisor: Decision and Invitation . . . 34
3.7.1 Overview . . . 34
3.8 WF04 — Cert Supervisor: Add CDE Certificate and NFT Mint . . . 36
3.8.1 Overview . . . 36
3.9 WF05 — Public Dual Verification . . . 37
3.9.1 Overview . . . 37
3.10 WF06 — CDE Training Registration (4-step Form) . . . 38
3.10.1 Overview . . . 38
3.11 Hyperledger Fabric Deployment Architecture . . . 40
3.11.1 Complete Fabric network diagram . . . 40
3.11.2 Certificate registration workflow over Fabric . . . 41
3.11.3 NFT Minting Workflow . . . 42
3.12 On-chain Transactions and Their Meaning . . . 43
3.12.1 Explanation of the most important transactions . . . 45
3.13 Conclusion . . . 45

4 Implementation **47**
4.1 Introduction . . . 47
4.2 Development Environment . . . 47
4.2.1 Technologies and Versions . . . 47
4.2.2 Project Structure . . . 48
4.2.3 Environment Configuration . . . 49
4.3 Database Implementation . . . 49
4.3.1 Schema Definition . . . 50
4.3.2 Database Client . . . 50
4.4 Blockchain Implementation . . . 51
4.4.1 Hyperledger Fabric Network Setup . . . 51
4.4.2 Chaincode Implementation . . . 51
4.4.3 Fabric Gateway Client . . . 54
4.5 IPFS Integration and Document Hashing . . . 55
4.5.1 SHA-256 Hash Generation . . . 55

4.5.2 IPFS Upload via Pinata 55
4.6 API Routes Implementation 56
4.6.1 Authentication — POST /api/auth/login 56
4.6.2 Blockchain Anchoring — POST /api/anchor 56
4.6.3 Document Verification — GET /api/verify 57
4.6.4 API Routes Summary 58
4.7 User Interface Implementation 58
4.7.1 Page Routing Structure 58
4.7.2 Landing Page and Authentication 59
4.7.3 Candidate Register for CDE training certificate 59
4.7.4 Register as a candidate 62
4.7.5 Candidate Dashboard 64
4.7.6 Supervisor Dashboard 64
4.7.7 Final Supervisor Dashboard 65
4.7.8 Certificate Supervisor Interface 66
4.7.9 Admin Dashboard 68
4.7.10 Public Verification Page 72
4.8 Hyperledger Fabric and Caliper Setup 73
4.8.1 Fabric network configuration 73
4.8.2 Orderer parameters used in the evaluation write-up 73
4.9 Performance Evaluation with Hyperledger Caliper 74
4.9.1 Benchmark methodology 74
4.9.2 Rate-controlled benchmark results 74
4.9.3 Volume-controlled benchmark results 76
4.9.4 Unified comparison chart 77
4.9.5 Discussion of benchmark results 77
4.10 Conclusion 78

General Conclusion **79**

List of Figures

1.1 The National Entrepreneurship Support and Development Agency - NESDA 7
1.2 The Caisse Nationale d'Assurance Chômage - CNAC 7
1.3 The Agence Nationale de Gestion du Micro-crédit - ANGEM 8

2.1 Evolution of blockchain technology from Blockchain 1.0 (cryptocurrencies) to Blockchain 2.0 (smart contracts) and Blockchain 3.0 (decentralized applications across industries) 13
2.2 Blockchain architecture: hash chain of blocks, each containing a Merkle tree of transactions [32] 14
2.3 Performance comparison of blockchain consensus algorithms (PoW,Raft, PBFT). 16
2.4 The three main types of blockchain networks: (a) public (permissionless), (b) consortium (semi-permissioned), and (c) private (permissioned) 17
2.5 Workflow of smart contract development, deployment, and execution on a blockchain network 18
2.6 Hyperledger Fabric architecture showing the separation of ordering, endorsement, and validation phases, along with membership services and chaincode execution 19
2.7 Off-chain storage methods with IPFS integration: files are stored off-chain and only their cryptographic hash identifiers are anchored on-chain, optimizing scalability while preserving integrity 21
2.8 End-to-end blockchain-based digital certificate verification process: issuance by the institution, hash anchoring on-chain, and tamper-proof verification by any third party 22
2.9 Summary of the main advantages and limitations of blockchain technology in enterprise and certification contexts 23

3.1 Global use case diagram of the GRANTA platform. 26
3.2 Class diagram of the GRANTA platform. 27
3.3 GRANTA — Cross-Workflow Architecture (Big Picture): end-to-end view of all workflows and system layers. 29
3.4 WF01 — Candidate Registration and Grant Application sequence. 32
3.5 WF02 — Supervisor Multi-Level Review sequence. 33

3.6 WF03 — Final Supervisor: Decision and Invitation sequence. 35
3.7 WF04 — Cert Supervisor: Add CDE Certificate and NFT Mint sequence. 36
3.8 WF05 — Public Dual Verification (SHA-256 path and NFT path). 37
3.9 WF06 — CDE Training Registration (4-step form) and approval phase. . . 38
3.10 Complete Fabric network diagram GRANTA. 40
3.11 End-to-end certificate registration and verification workflow. Steps 1–3 are executed off-chain (hashing and IPFS upload). Steps 4–8 correspond to the Fabric execute-order-validate transaction flow. Step 9 illustrates a verifier independently checking the authenticity of a certificate through a read-only chaincode query. 41
3.12 NFT minting and verification workflow. Steps 1–4 cover the minting path: after the certificate is registered on the ledger, the client invokes mintCertificateNFT; the chaincode enforces the double-mint guard and writes the token atomically. Steps 5–6 propagate the token identifier back to the certificate record via linkNFT. Steps 7–8 illustrate a verifier retrieving the full OpenSea-compatible metadata and validity status through a single read-only peer query. 43

4.1 GRANTA platform landing page 59
4.2 Candidate registration — Step 1: personal information. The candidate selects their wilaya (region) and universtiy and Decision 1275 (yes or no). All fields are validated client-side before the form advances to the next step. 60
4.3 Candidate registration — Step 2: certificate and project details. The candidate selects the certificate type (university or institute), enters the certificate number, and writes the project description. This information is stored in the `applications` table upon submission. 60
4.4 Candidate registration — Step 3: project file upload. The candidate uploads their project PDF. The system calls `POST /api/upload`, which sends the file to Pinata (IPFS), generates a SHA-256 fingerprint using the Web Crypto API, and stores both the IPFS CID and the hash in the `documents` table with status `PENDING`. 61
4.5 Candidate registration — Step 4: review and submit. The candidate reviews all entered information before final submission. Clicking *Submit* calls `POST /api/applications`, creates the application record in the database with status `pending`, and dispatches a notification to the assigned supervisor. . 62
4.6 Certificate number validation 63
4.7 Candidate register 63
4.8 Candidate dashboard 64
4.9 Supervisor dashboard — application review interface. 65
4.10 Final supervisor dashboard 66
4.11 Certificate supervisor — certificate list view (Step 1). 67
4.12 Certificate supervisor — certificate detail view (Step 2). 67
4.13 Certificate supervisor — validation confirmation (Step 3). 68

4.14 Admin dashboard — main overview panel. The main admin panel displays platform-wide statistics: total number of registered users, submitted applications, anchored documents, and active certificates. Navigation links provide direct access to each sub-module. The sidebar shows role-based menu items exclusive to the administrator. 69
4.15 Admin dashboard — document management panel (`/admin/GRANTA`). This panel lists all documents stored in the platform with their SHA-256 hash, IPFS CID, Cloudflare R2 backup URL, blockchain status (`PENDING` / `ANCHORED` / `FAILED`), and transaction ID. The admin can filter by status, search by hash, and trigger manual re-anchoring for failed documents. 69
4.16 Admin dashboard — Hyperledger Fabric live logs panel. This view streams real-time Docker container logs from the Fabric peer and orderer nodes directly into the browser via WebSocket. The admin can observe block commit events, chaincode invocation results, and endorsement policy validations as they happen, enabling immediate diagnosis of any blockchain transaction issue. 70
4.17 Admin dashboard — live blockchain activity monitor (`/admin/live`). The live activity panel displays a real-time event feed of all blockchain operations: document uploads, anchoring transactions, and verification queries. Each event card shows the event type, document filename, SHA-256 hash (truncated), IPFS CID, and timestamp. Events are pushed to the browser via the WebSocket notification module (`websocket-notify.ts`). 70
4.18 Admin dashboard — certificate NFT/CDE management panel. This panel provides the administrator with an overview of all training certificates registered by Certificate Supervisors. It displays the certificate holder, issuing CDE centre, course name, validity flag, and wilaya. The admin can search, filter, and manually override certificate validity status when required. . . . 71
4.19 Admin dashboard — document verification tool (`/admin/verify`). The admin verification panel allows the administrator to query any document by its SHA-256 hash, combining the database record with the live Hyperledger Fabric ledger state. The result shows whether the document is anchored on-chain, the blockchain timestamp, the issuer X.509 identity, and the IPFS CID — the same data returned by the public `/api/verify` endpoint. . . . 71
4.20 Admin dashboard — Hyperledger Caliper benchmark monitor (`/admin/live2`). This panel streams live results from the Caliper benchmark tool as it executes the six configured rounds (R1–R6). For each round it displays the target TPS, the number of successful and failed transactions, the average latency, and the effective throughput. The R2 normal-load result of 45.5 TPS with zero failures is highlighted, confirming production-grade performance. 72
4.21 Rate-controlled throughput for `registerCertificate`. 75
4.22 Rate-controlled latency for `registerCertificate`. 75
4.23 Volume-controlled throughput for `registerCertificate`. 76
4.24 Volume-controlled latency for `registerCertificate`. 77

4.25 Unified comparison of throughput across all benchmark rounds. 77

List of Tables

3.1 Main actors and responsibilities in the platform 25
3.2 Interpretation of the main classes in the diagram 28
3.3 Main blockchain transactions used in the platform 44

4.1 Technology stack and versions 48
4.2 Complete API routes reference 58
4.3 Benchmark environment summary 73
4.4 Orderer parameters considered in the benchmark discussion 74
4.5 Rate-controlled results for `registerCertificate` (1000 transactions per round) 74
4.6 Volume-controlled results for `registerCertificate` 76

Listings

4.1 Project directory structure 48
4.2 Environment variables (.env) 49
4.3 Core database schema (src/db/schema.ts) 50
4.4 Drizzle database client (src/db/index.ts) 50
4.5 Network startup and chaincode deployment 51
4.6 grantacore chaincode (grantacore/index.js) 51
4.7 Fabric Gateway client (src/lib/blockchain.ts) 54
4.8 SHA-256 hash utility (src/lib/hash.ts) 55
4.9 IPFS upload via Pinata (src/lib/ipfs.ts) 55
4.10 Login endpoint (src/app/api/auth/login/route.ts) 56
4.11 Anchoring endpoint (src/app/api/anchor/route.ts) 56
4.12 Verification endpoint (src/app/api/verify/route.ts) 57
4.13 App Router page structure 58
4.14 Public document verification flow (simplified) 72

General introduction

Youth unemployment and economic diversification have become two of the most urgent challenges facing Algeria in the post-hydrocarbon era. The Algerian state has responded with a broad array of institutional and financial support mechanisms including the National Entrepreneurship Support and Development Agency (NESDA), the Caisse Nationale d'Assurance Chômage (CNAC), the Agence Nationale de Gestion du Micro-Crédit (ANGEM), and a dedicated Ministry of Knowledge Economy, Startups, and Micro-Enterprises aimed at stimulating the creation of new enterprises by young people. Legal reforms since 2019 have further reinforced this direction, introducing startup labeling, university incubators, digital administrative portals, and fiscal incentives designed to accelerate business creation.

Despite this significant institutional effort, the entrepreneurial ecosystem continues to face structural weaknesses. Bureaucratic complexity, limited access to finance, skills deficits, and an uneven digitalization of public administration all slow the practical impact of public policies. Young entrepreneurs frequently encounter opaque procedures, manual workflows prone to error, and a near-total absence of verifiable digital records linking support applications to the institutions that process them. In this environment, document fraud, loss of institutional memory, and the inability to verify credentials across organizations are not exceptional incidents but recurring operational risks.

These challenges point to a deeper infrastructure problem: the absence of a trusted, decentralized, and tamper-evident layer for document registration and verification. It is precisely in this gap that blockchain technology offers a compelling and increasingly mature solution.

Problem Statement

The core problem this thesis addresses can be stated as follows: *How can a permissioned blockchain platform be designed, implemented, and evaluated to provide secure, transparent, and auditable digital certificate management within the context of Algerian youth entrepreneurship support?*

Several sub-questions derive from this central problem:

- What are the key technical and institutional constraints of the current document verification landscape for young entrepreneurs in Algeria?

- Which blockchain architecture public, private, or consortium and which consensus mechanism best fit the requirements of an institutional certification platform?
- How can off-chain storage (IPFS), on-chain certificate anchoring (Hyperledger Fabric), and a modern web application be integrated into a cohesive and usable system?
- What is the performance profile of such a system under realistic transaction loads, and does it meet the reliability requirements of institutional workloads?

Objectives

This thesis pursues four primary objectives.

The first is to establish a rigorous state of the art covering youth entrepreneurship in Algeria, the supporting legal and institutional framework, and the fundamentals of blockchain technology as applied to digital certification.

The second is to design a complete system architecture for a blockchain-based certificate platform, specifying actors, data models, interaction flows, and infrastructure components in a manner that is reproducible and coherent.

The third is to implement this architecture as a fully functional platform called **GRANTA** integrating Hyperledger Fabric 2.5, Next.js 15, TypeScript, SQLite, and IPFS, with chaincode supporting the full certificate lifecycle from submission to verification.

The fourth is to evaluate the platform under controlled experimental conditions using Hyperledger Caliper, producing quantitative evidence of throughput and latency behavior that supports conclusions about the practical deployability of the system.

Methodology

The methodology followed in this work is structured in four stages aligned with the four thesis chapters.

The first stage involves a literature review and state-of-the-art analysis. Academic sources, policy documents, legal texts, and technical reports are reviewed to characterize the socio-economic context of youth entrepreneurship in Algeria, identify existing support mechanisms and their limitations, and survey the relevant blockchain literature with particular attention to permissioned systems, consensus algorithms, and digital certification use cases.

The second stage is theoretical and architectural. Blockchain fundamentals including cryptographic hash functions, distributed ledger design, consensus mechanisms (PoW, PoS, PBFT, Raft), smart contracts, and enterprise frameworks such as Hyperledger Fabric are analyzed in relation to the requirements identified in the previous stage.

The third stage is system engineering. Using UML modeling (use case, class, and sequence diagrams) and native architectural diagrams, a complete conception of the GRANTA platform is developed. This covers the static data structure, actor responsibilities, dynamic

transaction flows, Hyperledger Fabric network topology, and the semantics of each chaincode operation.

The fourth stage is implementation and experimental evaluation. The platform is built, deployed, and tested. A Caliper benchmark campaign is designed and executed to measure the certificate registration transaction under both rate-controlled and volume-controlled conditions, producing quantitative performance data.

Thesis Structure

This thesis is organized into four chapters.

Chapter 1 General Study and State of the Art presents the socio-economic context of youth entrepreneurship in Algeria. It reviews the major support agencies and mechanisms, examines legal and regulatory frameworks, discusses the role of digitalization and e-government, and identifies the persistent barriers that motivate the need for an improved digital certification infrastructure.

Chapter 2 Blockchain Technology and Its Application in Secure Digital Certification provides a comprehensive technical foundation. It covers blockchain architecture and core principles, cryptographic building blocks, consensus mechanisms (with particular attention to Raft as used in Hyperledger Fabric), smart contracts, enterprise blockchain frameworks, decentralized storage with IPFS, and the state of the art in blockchain-based digital certification systems.

Chapter 3 System Design and Conception translates the theoretical foundation into a concrete system design. It presents the global use case diagram, the class diagram, sequence diagrams for application submission and certificate registration, a full Hyperledger Fabric deployment diagram, and a detailed explanation of the chaincode transactions that govern the GRANTA platform.

Chapter 4 Implementation describes the technical realization of the platform. It details the technology stack, project structure, database schema, Hyperledger Fabric network and chaincode implementation, IPFS integration, Next.js API routes, and role-based user interfaces. It concludes with a Hyperledger Caliper benchmark evaluation that provides quantitative evidence of the platform's performance under realistic load conditions.

A **General Conclusion** synthesizes the findings, discusses limitations, and proposes directions for future work.

The contributions of this thesis lie at the intersection of applied blockchain engineering and the study of institutional support for youth entrepreneurship in Algeria. By designing, implementing, and evaluating the GRANTA platform, this work provides both a concrete technical artifact and an empirical argument for the role that decentralized trust infrastructure can play in modernizing public entrepreneurship support systems.

Chapter 1

General Study and State of the Art

1.1 Chapter Introduction

Youth entrepreneurship is increasingly viewed as a strategic lever for economic diversification, innovation, and employment generation in Algeria, particularly as the country seeks to reduce dependence on hydrocarbons. Public authorities have multiplied legal, financial, and institutional measures to promote startups and small enterprises led by young people. Despite these efforts, structural constraints such as bureaucratic rigidity, limited financing, skills gaps, and uneven digitalization continue to weigh on the entrepreneurial ecosystem.

1.2 Youth Entrepreneurship

1.2.1 Definition of Entrepreneurship

Entrepreneurship can be defined as the process of identifying opportunities, mobilizing resources, and creating value through the establishment or development of a new business venture. It involves innovation, risk-taking, and strategic decision-making in order to generate economic and social impact. According to Organisation for Economic Co-operation and Development (OECD), entrepreneurship refers to "the phenomenon associated with entrepreneurial activity, which is the enterprising human action in pursuit of the generation of value, through the creation or expansion of economic activity, by identifying and exploiting new products, processes, or markets." [1].

1.2.2 Economic and Social Importance

Entrepreneurship holds significant economic and social importance for both developing and developed economies. Economically, it contributes to job creation, innovation, competitiveness, and sustainable growth by introducing new products, services, and business models. It also enhances employment opportunities and supports economic dynamism at

both local and national levels. Socially, entrepreneurial activities improve societal welfare by reducing unemployment, fostering inclusion, and enabling individuals to actively participate in economic life. Research shows that entrepreneurship has a strong positive impact on economic and social development, contributing to improved standards of living and inclusive growth outcomes. [2].

1.2.3 The Role of Entrepreneurship in Job Creation

Entrepreneurship plays a fundamental role in job creation and employment growth across economies. New businesses established by entrepreneurs inherently generate direct employment opportunities as they require a workforce to operate and expand. In addition to direct jobs, entrepreneurial ventures stimulate indirect employment through supply chains and related service industries. Empirical research shows that entrepreneurial activity can significantly influence employment trends, contributing to reduced unemployment rates and labor market dynamism. This impact is particularly evident in developing economies, where entrepreneurship helps absorb unemployed individuals and supports economic resilience through innovative ventures and small business growth. [15].

1.2.4 Specifics of Youth Entrepreneurship

Youth entrepreneurship refers to the initiatives and activities undertaken by young individuals, typically aged between 15 and 35, to create and develop new business ventures. Unlike general entrepreneurship, it is characterized by limited financial resources, lower levels of professional experience, and a higher propensity for innovation and risk-taking. Young entrepreneurs often face unique challenges, such as difficulty accessing financing, limited networks, and regulatory barriers. At the same time, youth entrepreneurship contributes significantly to economic growth, job creation, and social development by harnessing the creativity, energy, and technological skills of young people. Governments and institutions frequently design tailored support programs, including grants, training, and mentorship, to stimulate youth entrepreneurial activity [35].

1.3 Youth Entrepreneurship in Algeria

1.3.1 Socio-economic Situation of Algeria

Algeria's economy largely depends on the hydrocarbon sector, which represents a major source of national income. However, the country faces several socio-economic challenges, including youth unemployment and the need for economic diversification. In this context, promoting entrepreneurship and supporting startups have become important strategies to stimulate growth and create job opportunities. [10].

1.3.2 Youth Unemployment Figures

Youth unemployment remains a significant challenge in Algeria. In 2024, the youth unemployment rate (ages 15–24) was estimated at about 29.7 , meaning nearly one out of three young people in the labor force was unemployed. This rate is significantly higher than the overall national unemployment rate [9].

1.3.3 Public Policies to Support Entrepreneurship

The Algerian government has implemented several public policies to promote entrepreneurship and support young project holders. These policies include financial support programs, startup incubators in universities, and specialized agencies such as the National Agency for Entrepreneurship Support and Development (NESDA). In addition, initiatives like the Algerian Startup Fund provide financial investment and mentoring to innovative startups. These measures aim to encourage business creation, stimulate innovation, and reduce youth unemployment [44].

1.3.4 Obstacles Encountered by Young Entrepreneurs

Young entrepreneurs in Algeria face several challenges that hinder the creation and development of new businesses. The main obstacles include limited access to financing, complex administrative procedures, and a lack of entrepreneurial experience and professional networks. These difficulties can slow down the creation of startups and reduce the chances of success for young project holders [37]

1.3.5 Importance of Digital Transformation

Digital transformation plays an important role in developing entrepreneurship in Algeria. It allows startups and small businesses to improve efficiency, reduce operational costs, and access wider markets through online platforms and digital services. The increasing use of the internet and mobile technologies also facilitates e-commerce, digital marketing, and innovation in new business models. Moreover, digital transformation supports economic diversification and encourages the creation of technology-based startups. With the growth of internet users and government initiatives to promote the digital economy, digital transformation is becoming a key factor in strengthening the entrepreneurial ecosystem in Algeria. [3]

1.4 Support Mechanisms in Algeria

1.4.1 NESDA (ex-ANSEJ)

The National Entrepreneurship Support and Development Agency is a public governmental institution operating under the supervision of the Ministry of Knowledge Economy,

Start-ups and Micro-Enterprises. Its main objective is to support young entrepreneurs in creating and developing micro-enterprises in the production of goods and services in order to promote wealth creation and sustainable employment.

The agency provides several services, including entrepreneurial training, project accompaniment from the idea stage to business creation, and financial support through different funding mechanisms in collaboration with public banks. NESDA also contributes to spreading entrepreneurial culture among young people through Entrepreneurship Development Centers located in universities and training institutions across Algeria. [4]



Figure 1.1: The National Entrepreneurship Support and Development Agency - NESDA

1.4.2 CNAC

The Caisse Nationale d'Assurance Chômage (CNAC) is a public institution in Algeria that supports unemployed people in creating their own businesses. It provides financial assistance, training, and guidance to encourage entrepreneurship and reduce unemployment. Through these support programs, CNAC contributes to the development of micro-enterprises and promotes entrepreneurial initiatives in the country. [5]



Figure 1.2: The Caisse Nationale d'Assurance Chômage - CNAC

1.4.3 ANGEM

The Agence Nationale de Gestion du Micro-crédit (ANGEM) is a public institution in Algeria established in 2004 to support entrepreneurship through microcredit programs. It helps people with low or unstable income start small businesses by providing micro-loans, training, and technical assistance. The main objective of ANGEM is to promote self-employment, reduce unemployment, and encourage the creation of micro-enterprises across different economic sectors. [6]



Figure 1.3: The Agence Nationale de Gestion du Micro-crédit - ANGEM

1.4.4 Ministry of Knowledge Economy, Startups, and Micro-Enterprises

A dedicated ministry was established to signal Algeria's shift towards a knowledge-based, innovation-driven economy [1]. It oversees:

- The startup and innovative project labeling system.
- The national startup fund and fiscal incentives.
- A network of incubators and accelerators.

This institutional move underpins the rise of labeled startups, particularly after 2019, and supports sectors such as tourism and ICT [1].

1.4.5 Self-Employed Entrepreneur (Law 22-23)

Recent legislation (including provisions like Law 22-23 and Decree 20-254) has introduced a more flexible status for individual or small entrepreneurs and simplified procedures for emerging enterprises [9, 44, 7]. The legal framework now allows:

- Simplified joint-stock companies under the startup label [44].
- Public employees and students to create enterprises while maintaining their status under certain conditions [44, 11, 7].
- Electronic procedures for labeling and registration via national platforms [25, 7].

1.5 The Role of Insurance in Entrepreneurial Projects

1.5.1 Definition and Objectives of Insurance

Insurance is a mechanism for risk transfer and risk pooling, designed to protect individuals and enterprises against financial consequences of uncertain events, thereby stabilizing investment and economic activity [2, 37]. In entrepreneurship, insurance reduces the vulnerability of young firms to shocks and encourages risk-taking.

1.5.2 The Insurance Sector in Algeria

Government support and business climate studies highlight that financial and institutional environments, including insurance, remain underdeveloped relative to entrepreneurial needs [2, 9, 37, 39].

1.5.3 Types of Insurance for Businesses

Typical products relevant for SMEs and startups include property insurance, liability insurance, workers' compensation, and specific coverage related to professional activity, transport, or technology risks. Studies on perceived risks in SMEs highlight the importance of mechanisms—public support, guarantees, and by extension insurance—to mitigate these risks [37, 39, 40].

1.5.4 The Importance of Insurance in Securing Investments

Perceived risk significantly affects sustainable entrepreneurship in Algerian SMEs; government support partly mediates this relationship by reducing uncertainty and making startups more resilient [37]. In this context, appropriate insurance products can complement state programs by securing assets, contracts, and employees, thus protecting both entrepreneurs and financiers and encouraging long-term investment [37, 39, 40].

1.5.5 Algerian Insurance Companies

The literature suggests a need for better integration between financial institutions (banks, funds, insurers) and entrepreneurial support policies to ensure comprehensive risk-management solutions for emerging firms [2, 35, 37].

1.6 Algerian Legal and Regulatory Framework

Algeria has adopted an extensive arsenal of legal texts to foster emerging institutions and startups [1, 9, 44, 7]. Key features include:

- Executive Decree No. 20-254, which defines conditions for granting labels “emerging enterprise”, “innovative project”, and “business incubator” [7].
- Simplified incorporation rules for startup joint-stock companies and digital procedures via electronic portals [44, 7].
- Amendments to labor and public service laws enabling employees and students to engage in private enterprise under specific conditions [44, 11].
- Integration of tax incentives and startup support measures into finance laws to encourage innovation and investment [1, 9, 18].

These reforms reflect a shift from a rent-based model towards a knowledge-based development paradigm, though gaps between legal design and practical implementation remain significant [35].

1.7 Digitalization of Public Administration

1.7.1 E-Government in Algeria

The move towards e-government accompanies Algeria’s digital transformation, with electronic platforms supporting startup registration, project labeling, and interaction with public bodies [1, 12, 25, 7]. These efforts aim to reduce bureaucratic burdens, enhance transparency, and facilitate access to support measures.

1.7.2 Advantages of Digitalization

Digitalization of public administration and support systems offers several advantages for young entrepreneurs: reduced administrative delays, easier access to information and services, and better integration with digital business tools [1, 10, 12]. For universities, digital platforms enable online project submission, monitoring, and coordination among stakeholders [10, 25, 11].

1.7.3 Limitations of Manual Procedures

Traditional manual procedures have long been criticized for their complexity, opacity, and slowness, which discourage entrepreneurial initiatives and undermine incentives [9, 39]. Entrepreneurs—especially returnees and innovative startups—report significant difficulties related to rigid administrative frameworks and the cost of compliance [39]. Digitalization seeks to address these limitations, but implementation remains uneven.

1.7.4 Contribution of Digital Platforms

Digital platforms created under startup policy and Ministerial Decision 1275 support:

- Registration and evaluation of student startup projects (over 9,000 projects nationally) [10, 25, 11].
- Labeling of startups, innovative projects, and incubators [1, 25, 7].
- Online support, training, and networking for startups, particularly in tourism and technology sectors [10, 12].

However, studies in tourism and regional digital transformation show strong geographical disparities and a lack of robust e-governance structures, especially in high plateau and eastern regions [12].

1.8 State of the Art

Recent literature converges on several key points:

- Centrality of youth entrepreneurship to Algeria's development and diversification strategy [1].
- Rapid evolution of legal and institutional frameworks to support startups, especially since 2019 [1].
- Expansion of university-based entrepreneurship initiatives and the integration of innovation, digital skills, and project-based learning [10].
- Recognition of multi-level barriers: financial, bureaucratic, socio-cultural, skills-related, and digital [35].
- Limited but growing attention to sustainable entrepreneurship, territorial development, and sector-specific digital challenges [37].

There remain important research gaps regarding: long-term performance of supported youth enterprises; effectiveness of insurance and advanced financial products; rural vs. urban disparities; and impact evaluation of incubators and accelerators [37].

1.9 Chapter Conclusion

Youth entrepreneurship in Algeria is framed by an evolving legal and institutional architecture, significant public investment in support mechanisms, and a gradual digital transformation of administration and universities. While progress is evident particularly in the emergence of startups and student enterprises persistent barriers and implementation gaps continue to limit the full potential of young entrepreneurs.

Chapter 2

Blockchain Technology and Its Application in Secure Digital Certification

2.1 Introduction

Blockchain has become a key technology for creating secure, tamper-proof, and transparent digital systems across different sectors, including finance, supply chains, healthcare, and identity management[17, 21]. Its decentralized design gets rid of single points of failure. Also, cryptographic methods and agreement mechanisms make sure data is complete and that there is trust between parties that don't fully trust each other [14, 50, 36]. These qualities position blockchain as a top choice for secure digital certification, where being real, not being able to deny something, and being hard to fake are very important [34, 16].

2.2 Fundamentals of Blockchain Technology

2.2.1 Definition of Blockchain

A blockchain serves as a shared, append-only record where data is organized into blocks, linked using cryptography, and copied across many nodes within a peer-to-peer network [14, 50, 47]. Each block includes a group of approved transactions, a timestamp, and a hash pointer that links back to the previous block, creating an unchangeable chain [14, 36, 42].

2.2.2 Historical Evolution of Blockchain

Bitcoin introduced the first working blockchain in 2008, it was a solution for digital cash that didn't rely on a central authority [14, 27]. This stage, often called Blockchain 1.0, was mainly about cryptocurrencies. Blockchain 2.0 expanded its abilities with smart contracts

that could be programmed on platforms like Ethereum, This allowed for general decentralized applications [50, 32]. Blockchain 3.0 and later versions extend blockchain to areas like IoT, supply chains, e-governance, and digital identity and certification [34, 16, 48, 43].

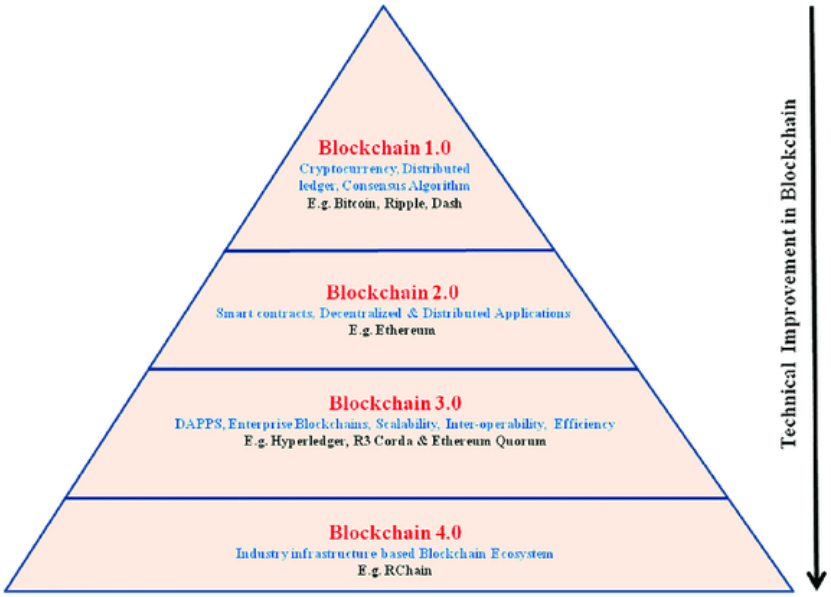


Figure 2.1: Evolution of blockchain technology from Blockchain 1.0 (cryptocurrencies) to Blockchain 2.0 (smart contracts) and Blockchain 3.0 (decentralized applications across industries)

2.2.3 Core Principles

Decentralization: Instead of a central authority, control and data are spread across many nodes, which makes the system stronger and reduces the risk of single-point failures [14, 36, 42].
Immutability: Once blocks are confirmed, changing past records becomes very difficult from a computational or organizational perspective, offering strong evidence against tampering [17, 14, 27].
Transparency and Auditability: In public or consortium chains, participants can see and check transactions, which improves accountability and the ability to trace activity [34, 21, 42].
Security: Blockchain security depends on cryptographic hash functions, digital signatures, and consensus algorithms that can handle faults. Together, they protect data's integrity, authenticity, and availability [17, 36, 33, 53].

2.3 Blockchain Architecture

2.3.1 Blocks and Transactions

A transaction is a change in state (like transferring value or issuing a certificate). It is sent out to the network and collected into a block [14, 50]. A block usually has a header (block hash, previous block hash, Merkle root, timestamp, nonce) and a body that lists the transactions [14, 47, 52]. The Merkle tree structure makes it possible to efficiently prove that transactions are included and verify them [50, 42].

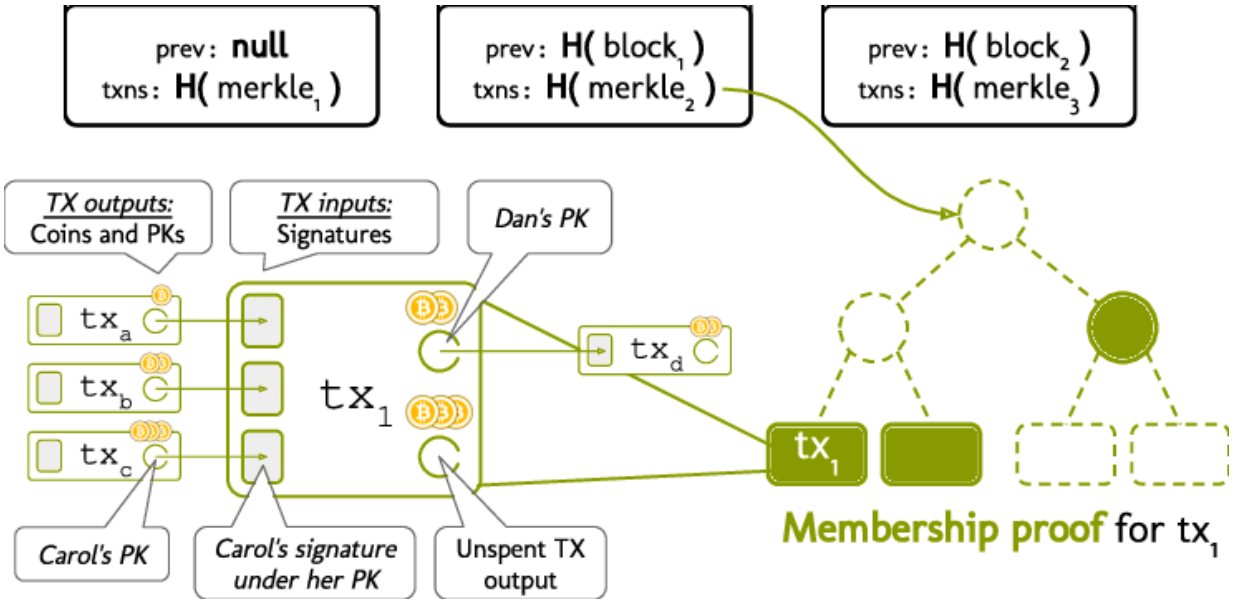


Figure 2.2: Blockchain architecture: hash chain of blocks, each containing a Merkle tree of transactions [32]

2.3.2 Cryptographic Hash Functions

Cryptographic hash functions (like SHA-256, Keccak) convert inputs of any length into outputs of a fixed length. They are designed to resist attacks that try to find the original input (preimage-resistant), find two different inputs that produce the same output (collision-resistant), and find a second input that produces the same output as a known input (second-preimage-resistant) [17, 14]. Hashes connect blocks and are used to calculate Merkle roots for sets of transactions, Any change to the block's contents changes the hash, which reveals tampering [17, 50, 33].

2.3.3 Digital Signatures and Public-Key Cryptography

Blockchain users are identified by public keys, Transactions are authorized using digital signatures created with corresponding private keys [14, 36]. Public-key cryptography makes

sure that only the real owners can make transactions, and anyone can check the signatures, which provides authentication and makes sure that transactions can't be denied later [21, 33, 57].

2.3.4 Distributed Ledger Structure

The blockchain ledger is copied across many nodes that keep synchronized copies using a consensus protocol [14, 36, 42]. Each node independently checks transactions and blocks, which increases resistance to failures and attacks. Ledgers can be permissionless (open to anyone) or permissioned (restricted participation), depending on how they are set up [16, 32, 8].

2.4 Consensus Mechanisms

Consensus algorithms help nodes agree on the valid state of the ledger even when there are failures and attackers [27, 53, 29].

2.4.1 Proof of Work (PoW)

PoW, which Bitcoin uses, requires miners to solve a difficult computational problem (finding a nonce that makes a block hash meet a certain difficulty level) [14, 27]. It is strong and has been studied well, but it uses a lot of energy and has limited throughput [36, 53, 29].

2.4.2 Proof of Stake (PoS)

PoS, chooses block proposers and validators based on how much cryptocurrency they have "staked" instead of computational power [32, 53]. This greatly reduces energy use and can increase throughput. However, it introduces new ways to attack the system (like nothing-at-stake, long-range attacks) and requires careful protocol design [53, 29].

2.4.3 Practical Byzantine Fault Tolerance (PBFT)

PBFT-style protocols allow a known set of nodes to reach an agreement, tolerating up to one-third of Byzantine faults through multiple rounds of voting [21, 27]. They have low latency and high throughput, making them good for permissioned and enterprise blockchains. However, their communication overhead increases with the number of nodes [34, 8, 20].

2.4.4 Raft Consensus Algorithm

Raft is a leader-based, crash-fault-tolerant (CFT) consensus algorithm originally designed for distributed systems and now widely adopted in private/consortium blockchains such

as Hyperledger Fabric[22, 54, 49]. It aims to provide the same safety as Paxos but with a simpler and more understandable structure[22].

Basic operation:

- Nodes can be in three states: Leader, Follower, Candidate [56].
- If Followers do not receive periodic heartbeat messages, they become Candidates and start a leader election, requesting votes from other nodes.
- Once a Candidate receives a majority of votes, it becomes Leader and starts log replication: client commands (transactions) are appended to the Leader's log and then replicated to Followers; an entry is committed when stored by a majority [46, 22, 56].

Suitability for blockchain:

- Raft offers high throughput and low latency for permissioned blockchains where nodes are authenticated and primarily need to handle crash faults rather than Byzantine faults [22, 54].
- It has been used and extended in Hyperledger Fabric and other consortium chains to improve performance via better leader election and log replication [19, 45, 54, 55].

Limitations and enhancements:

- Standard Raft does not tolerate Byzantine (malicious) nodes, only crashes [22, 23].
- Many works propose improved Raft variants:
- Performance-oriented (AdRaft, NK-Raft, CBR, P-Raft, KRaft, Praft) that optimize leader election, load balancing, and message complexity [19, 45, 38, 55, 49, 30].
- Security-enhanced (hhRaft, VSSB-Raft, PB-Raft, DLCA_R_P, PRAFT, X-RAFT)that add supervision nodes, zero-trust models, reputation, or combine Raft with PBFT/BFT signatures to gain Byzantine fault tolerance while keeping Rafts efficiency [46, 28, 56, 51, 31, 41].

Consensus Algorithm	Raft	Pbft	PoW
Environmental Trust	High	Relatively high	Low
Energy Consumption	Low (communication)	High (communication)	Very high (communication+mining)
Transaction Throughput	High	Medium	Low
Communication Complexity	O(n)	O(n2)	O(n)
Maximum number of fault-tolerant nodes	2f+1≤n	3f+1≤n	2f+1≤n

Figure 2.3: Performance comparison of blockchain consensus algorithms (PoW,Raft, PBFT).

2.5 Types of Blockchain Networks

2.5.1 Public Blockchain

Public blockchains (like Bitcoin, Ethereum) are permissionless, so anyone can join, read, and often write transactions [14, 36]. They offer high transparency and resist censorship, but they may have problems with scalability and privacy [34, 42].

2.5.2 Private Blockchain

Private blockchains limit participation to a single organization or a closely controlled group, An administrator controls access [16, 32]. They usually use efficient consensus mechanisms (like PBFT, Raft) and offer higher throughput and privacy but with less decentralization [16, 8].

2.5.3 Consortium Blockchain

Consortium (or federated) blockchains are managed by a group of organizations that jointly handle validation and governance [21, 42]. They balance decentralization and performance and are widely considered suitable for applications between institutions, like supply chains and shared certification infrastructures [34, 21, 16].

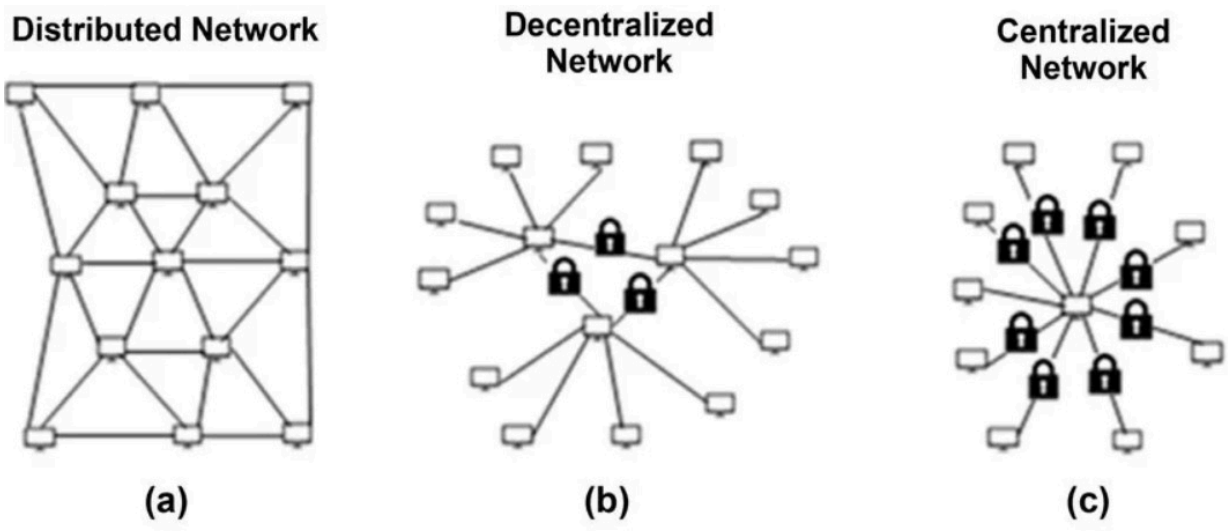


Figure 2.4: The three main types of blockchain networks: (a) public (permissionless), (b) consortium (semi-permissioned), and (c) private (permissioned)

2.6 Smart Contracts

2.6.1 Concept and Functionality

Smart contracts are programs that run automatically on a blockchain, They enforce agreed rules when certain conditions are met [50, 32, 26]. They can change the state of the blockchain, check inputs, and trigger events without needing an intermediary, which ensures execution is consistent and transparent [50, 32].

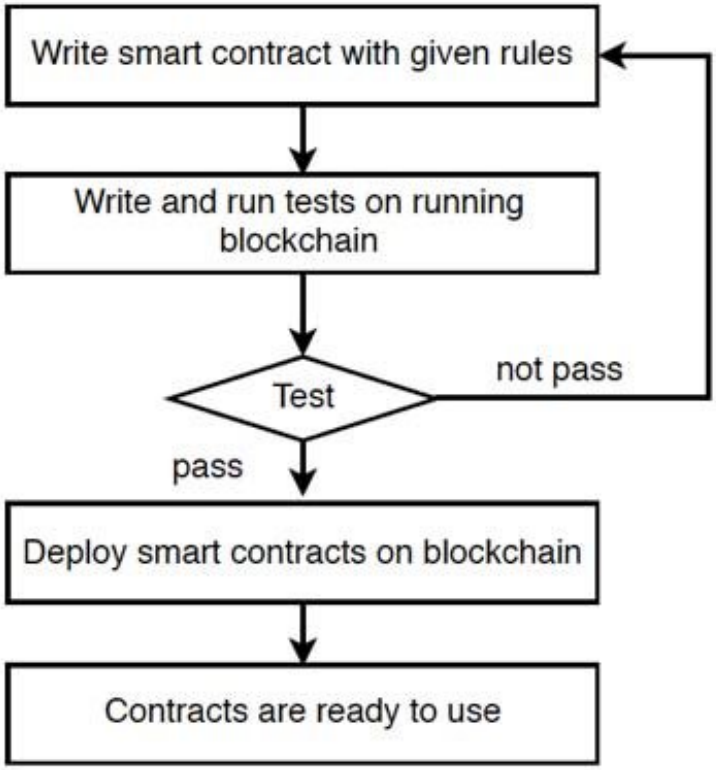


Figure 2.5: Workflow of smart contract development, deployment, and execution on a blockchain network

2.6.2 Smart Contracts in Enterprise Systems

In enterprise settings, smart contracts encode business logic like access control policies, workflows, and settlement rules [50, 8, 26]. Platforms like Ethereum and Hyperledger Fabric support complex contract development, which allows them to integrate with existing systems and meet regulatory requirements while maintaining auditability and trust [32, 8, 24].

2.7 Enterprise Blockchain Frameworks

2.7.1 Hyperledger Fabric Architecture

Hyperledger Fabric is a modular, permissioned blockchain framework that supports pluggable consensus, membership services, and chaincode (smart contracts) [16, 8]. Its architecture separates the ordering, endorsement, and validation stages, which enables high throughput, confidentiality through channels, and detailed access control suitable for enterprises [8, 24].

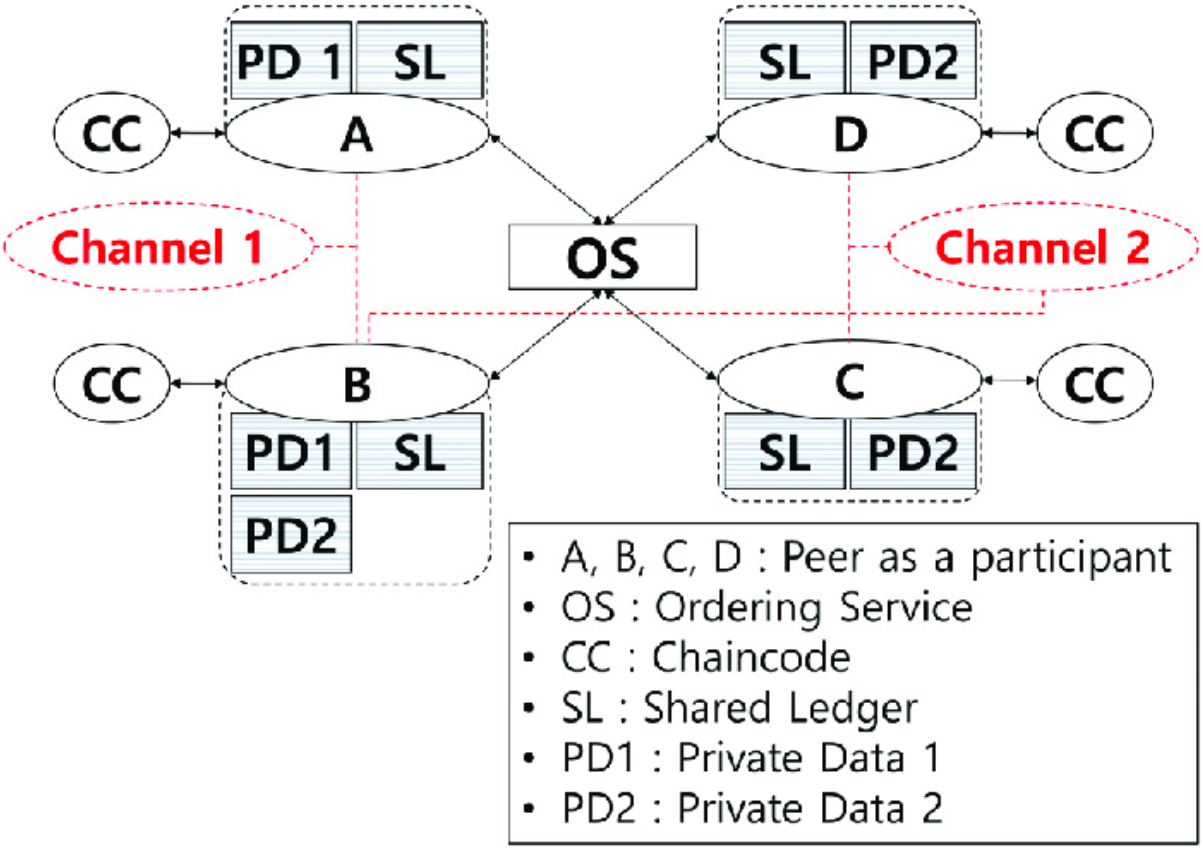


Figure 2.6: Hyperledger Fabric architecture showing the separation of ordering, endorsement, and validation phases, along with membership services and chaincode execution

2.7.2 Permissioned Blockchain Model

Permissioned blockchains restrict who can read, submit, and validate transactions. They are typically supported by a public-key infrastructure (PKI) and a membership service provider [16, 8]. This model is preferred when identity, compliance, and governance are important, such as in financial consortia and inter-university certification systems [21, 16, 24].

2.8 Blockchain and Decentralized Storage

2.8.1 IPFS Overview

The InterPlanetary File System (IPFS) is a content-addressable, peer-to-peer storage network where files are identified by their cryptographic hashes [17, 13]. It enables decentralized, verifiable storage, and references (hashes) can be placed on a blockchain to prove integrity and existence [17, 13].

2.8.2 On-Chain vs Off-Chain Storage

On-chain storage keeps data directly on the blockchain, which maximizes immutability and availability but leads to high costs and limited scalability [17, 34].

Off-chain storage keeps large amounts of data in external systems (like databases, IPFS), while the blockchain only stores hashes or pointers, which improves scalability and privacy [17, 13].

Hybrid approaches carefully combine both, storing critical metadata on-chain and large or sensitive content off-chain [17, 34, 13].

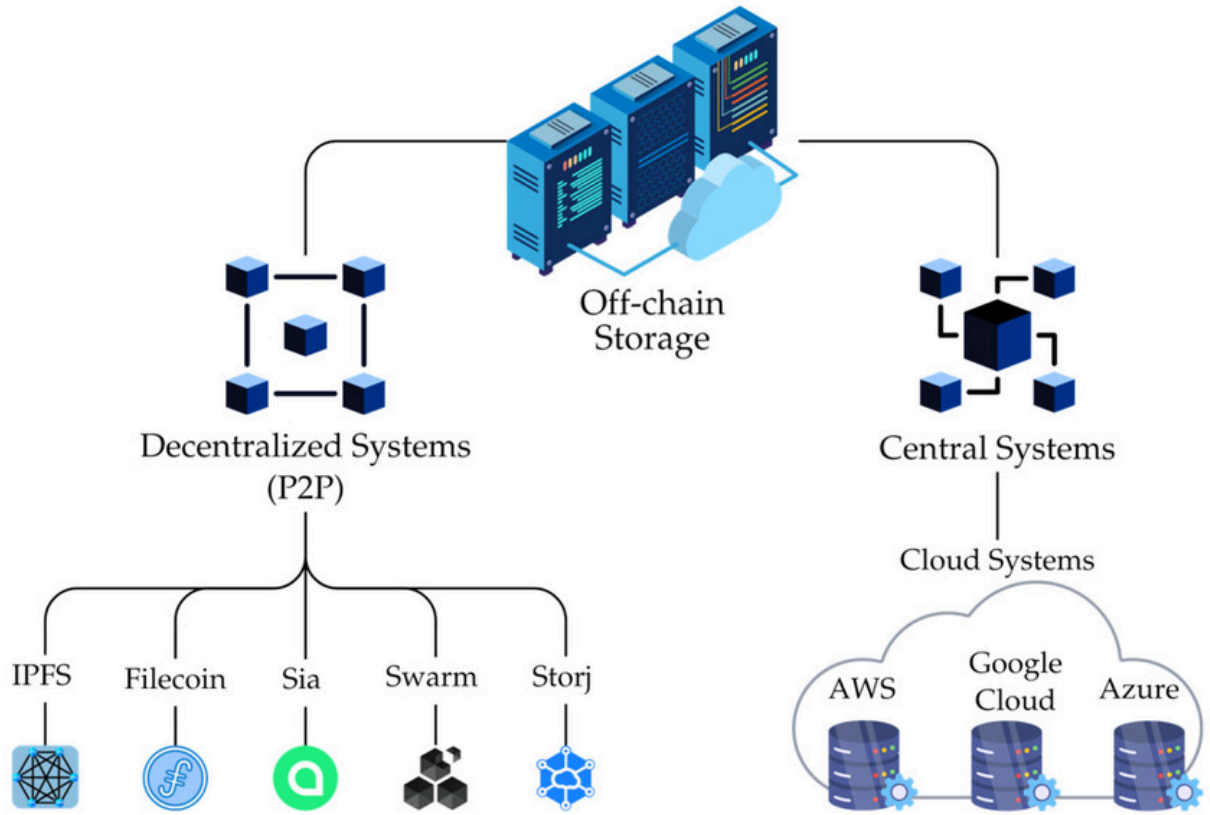


Figure 2.7: Off-chain storage methods with IPFS integration: files are stored off-chain and only their cryptographic hash identifiers are anchored on-chain, optimizing scalability while preserving integrity

2.9 Application of Blockchain in Digital Certification Systems

2.9.1 Traditional Certificate Verification Challenges

Traditional certificate systems (academic, professional, or governmental) often depend on central databases or paper documents, which are at risk of being forged, lost, and experiencing single-point failures [16, 48]. Verification is often done manually, takes a lot of time, and depends on the institutions that issued them, which leads to inefficiencies and chances for fraud [48, 43].

2.9.2 Blockchain-Based Certificate Anchoring

In blockchain-based certification, the key details of a certificate or a hash of the entire document are stored on the ledger, while the full document may be stored off-chain [34,

16]. To verify a certificate, the hash is recalculated and compared to the record on the blockchain, which ensures integrity and authenticity without needing to contact the issuer [48, 43]. Smart contracts can automatically handle issuance, revocation, and expiry, which enables credential systems that can work together and are tamper-evident across different institutions [50, 32, 26].

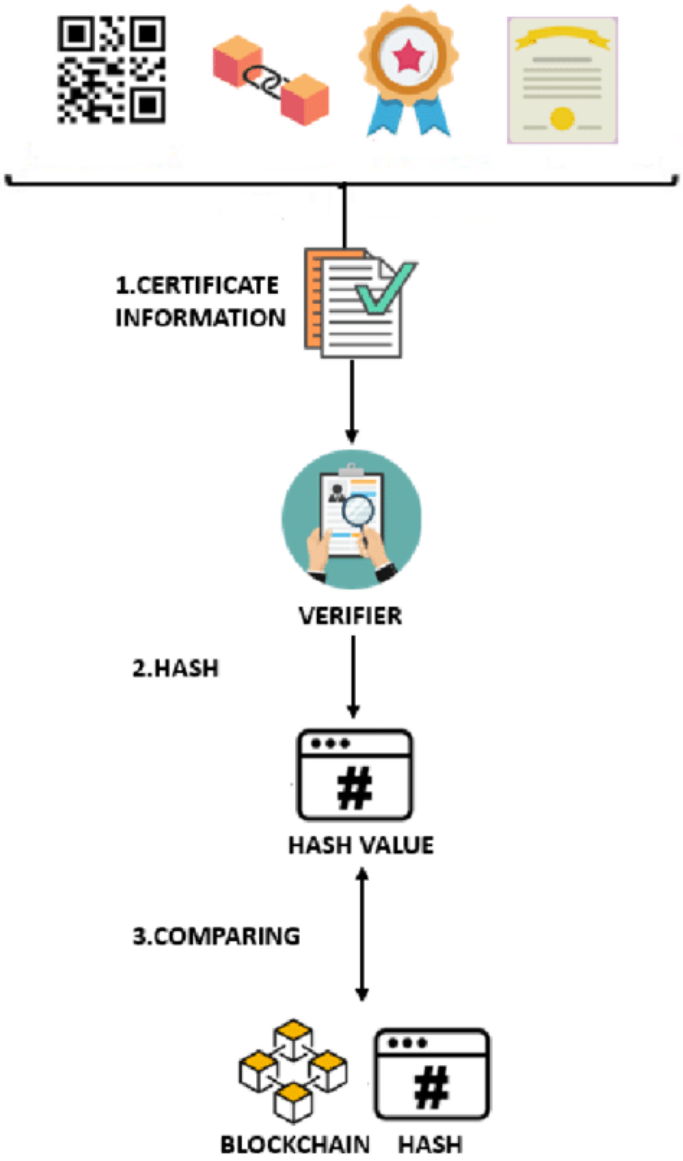


Figure 2.8: End-to-end blockchain-based digital certificate verification process: issuance by the institution, hash anchoring on-chain, and tamper-proof verification by any third party

2.10 Advantages and Limitations of Blockchain Technology

Blockchain provides records that are resistant to tampering, trust that is decentralized, better auditability, and automation that can be programmed using smart contracts. Together, these improve security, transparency, and efficiency in digital certification [17, 34, 14, 50, 16]. However, it has significant limitations: constraints in scalability and latency, especially on public chains; high energy consumption in PoW systems, complex governance and interoperability and privacy/regulatory challenges when dealing with personal data [34, 36, 27, 48, 57, 29]. Choosing appropriate architectures (public vs permissioned), consensus algorithms, and storage models is essential to balance these trade-offs in real-world deployments [17, 21, 8].

Advantages of Blockchain Technology

1.	*Data integrity and Immutability:* Participants can reduce fraud while strengthening regulatory compliance. Once a record has been stored in the ledger, it can only be deleted after a consensus.	(Swan 2015; Fanning & Centers 2016).
2.	*Security:* All transactions will be digitally time-stamped with a cryptographic hash code, a unique 64-digit alpha-numeric signature is recorded corresponding to every single transaction.	Swan (2015).
3.	*High availability and Accessibility:* Due to decentralized networks, Blockchain Technology data would be complete, timely and accurate.	(Bahga & Madisetti, 2016; Bahga & Madisetti 2014).
4.	*Reliability:* Blockchain Technology it is not regulated by a single control center and there's no single point of failure.	(Glaser & Bezzenberger 2015; Tapscott & Tapscott 2016).

Figure 2.9: Summary of the main advantages and limitations of blockchain technology in enterprise and certification contexts

2.11 Conclusion

Blockchain provides a robust foundation for secure digital certification by combining decentralization, cryptographic security, and programmable smart contracts. Understanding its fundamentals architecture, consensus mechanisms, network types, and storage strategies enables the design of scalable and trustworthy certification systems. Nonetheless, careful consideration of performance, privacy, governance, and legal compliance is required to fully realize its benefits in practice.

Chapter 3

System Design and Conception

3.1 Introduction

This chapter redesigns the GRANTA platform from a system engineering perspective and aligns the manuscript with the final certificate-oriented workflow. The objective is to explain, in a coherent way, how the web interface, database, IPFS storage, and Hyperledger Fabric cooperate to support certificate application submission, certificate registration, NFT minting, notification management, and public verification. Because some exported UML diagrams still contain legacy labels from the development phase, the narrative below interprets them according to the final certificate registration and verification process.

The chapter is organised around several complementary views of the system. First, the global use case diagram identifies the actors and the services exposed to them. Second, the class diagram explains the static data structure. Third, a dedicated cross-workflow architectural view presents the global integration of the six business workflows (WF01–WF06) that compose the platform, followed by a detailed presentation of each workflow as an independent dynamic view. Fourth, a complete Hyperledger Fabric deployment diagram used in the platform. Finally, the chaincode transactions are listed and explained in order to clarify the behaviour of each on-chain operation.

The decomposition into seven workflow sections (one big-picture cross-workflow view and six per-workflow views) is motivated by the fact that the GRANTA platform is not a monolithic application but a coordinated set of independent services. Each workflow has its own actors, API routes, data tables, and blockchain interactions. Treating them as autonomous units makes the system easier to understand, maintain, and extend.

3.2 Global Functional View

3.2.1 System actors

The platform involves six main actor categories: the *Candidate*, the *Supervisor*, the *Final Supervisor*, the *Certificate Supervisor*, the *Administrator*, and the *Public/Employer verifier*.

The candidate is responsible for submitting an application and following its status. The supervisor reviews applications and issues intermediate decisions. The final supervisor validates approved applications and triggers anchoring on the blockchain. The certificate supervisor manages certificate registration and issuance operations. The administrator supervises the platform, users, logs, and benchmark execution. Finally, an external verifier can confirm document authenticity without needing a privileged account.

Table 3.1: Main actors and responsibilities in the platform

Actor	Responsibilities
Candidate	Register an account, upload project files, submit an application, track status, receive notifications, and verify personal certificates.
Supervisor	Review assigned applications, accept or reject them, add comments, and send interview invitations.
Final Supervisor	Make the final decision, validate approved records, and trigger blockchain anchoring for accepted documents or certificates.
Certificate Supervisor	Register certificates, validate eligibility, issue certificates, and initiate NFT minting when required.
Administrator	Manage users, inspect all documents, monitor blockchain logs, and supervise benchmark execution.
Public / Employer	Verify a certificate or document by hash, certificate number, or NFT identifier.

3.2.2 Use case diagram

Figure 3.1 presents the global use case diagram of the GRANTA platform. It synthesises the complete functional perimeter of the solution and visually separates the responsibilities attached to each actor. The left part of the diagram groups the operational users of the platform, whereas the right part highlights administration, verification, and blockchain-related services.

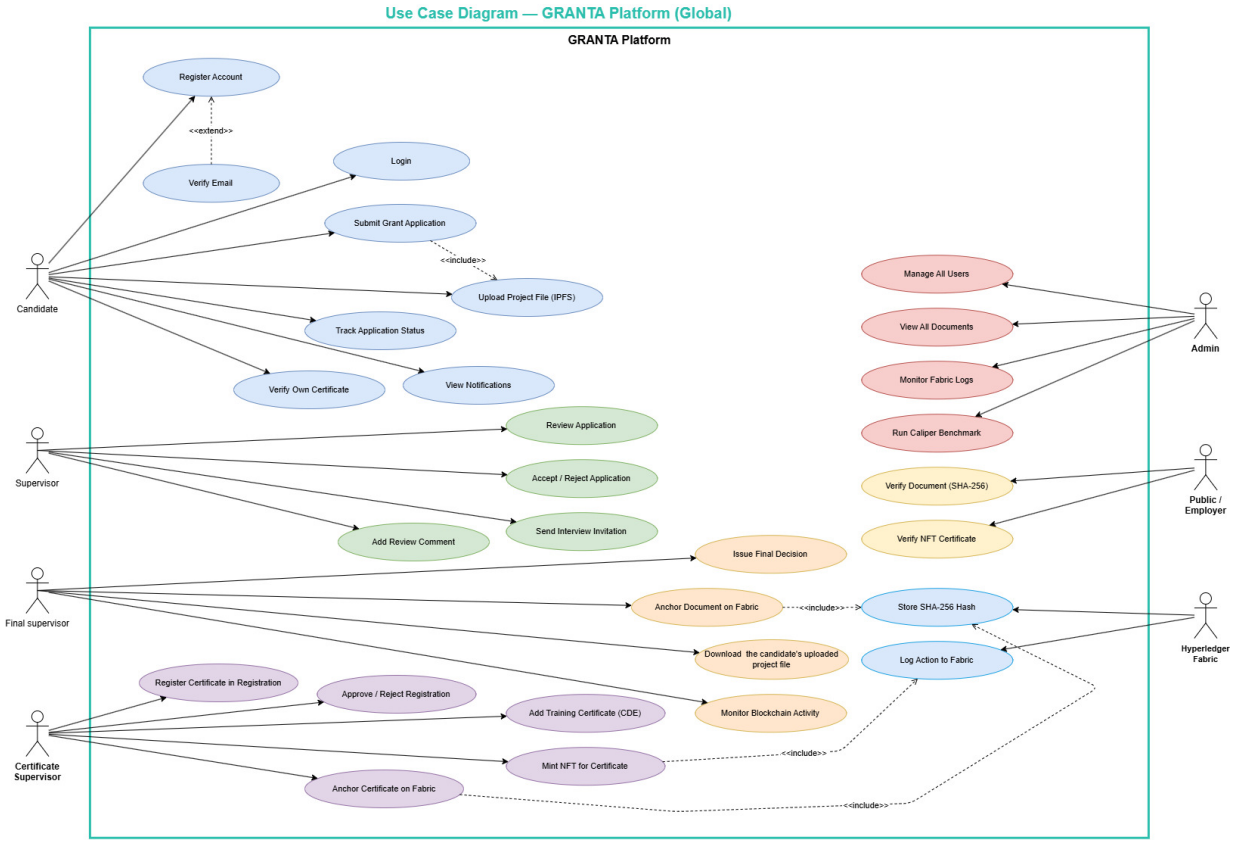


Figure 3.1: Global use case diagram of the GRANTA platform.

The diagram shows that the platform is not limited to a simple upload-and-store process. Instead, it supports a full business workflow: account creation, email verification, application submission, review, interview invitation, final decision, certificate registration, certificate anchoring, NFT minting, notification delivery, and external verification. The diagram also reveals that blockchain integration is not isolated; it is embedded in several use cases such as *Anchor Document on Fabric*, *Store SHA-256 hash*, *Mint NFT for Certificate*, and *Log Action to Fabric*. This confirms that the blockchain acts as a trust infrastructure rather than a separate subsystem.

3.3 Static Design View

3.3.1 Class diagram

The class diagram in Figure 3.2 models the static structure of the platform. It identifies the persistent entities manipulated by the application and shows how they are connected. The most important classes are `User`, `Application`, `Review`, `Document`, `Certificate`, `CertificateRegistration`, `Notification`, and `FabricLog`.

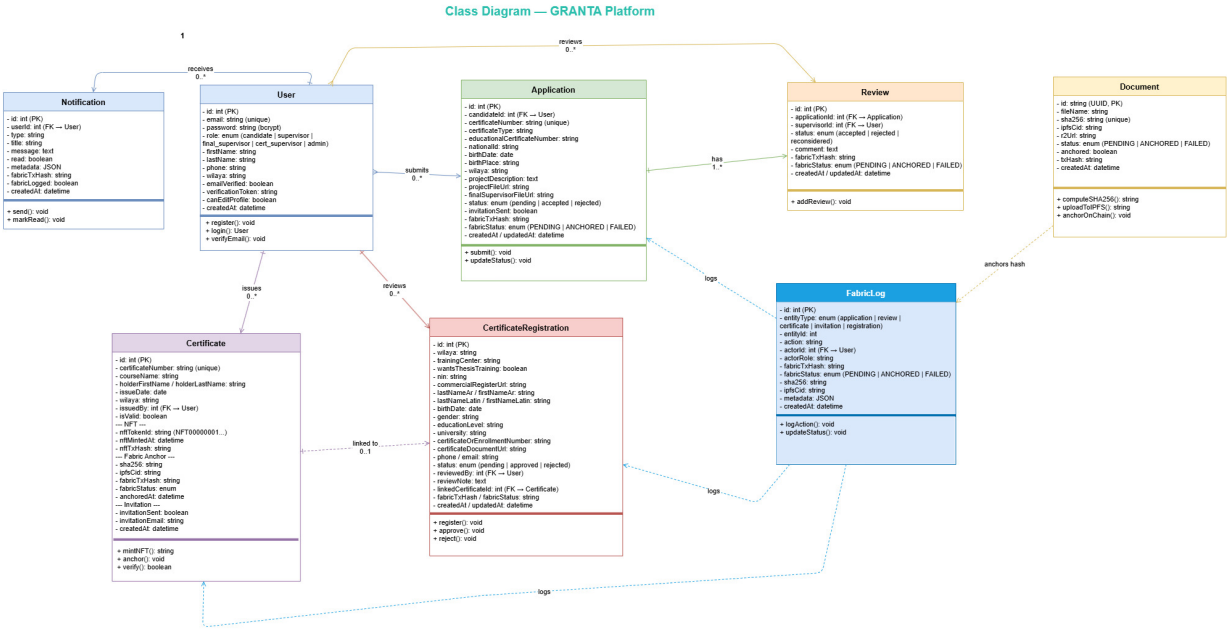


Figure 3.2: Class diagram of the GRANTA platform.

The `User` class is the root identity object of the platform. It stores authentication data, role information, and personal identity fields. The `Application` class represents a candidate request and contains business attributes such as certificate type, educational level, birth date, wilaya, and project description. The `Review` class records supervisor decisions and comments associated with an application. The `Document` class stores the fingerprint and IPFS location of the uploaded file. The `Certificate` class keeps the final issued certificate data, while `CertificateRegistration` models the upstream registration request that precedes approval or rejection. Finally, `FabricLog` and `Notification` capture traceability and asynchronous user communication.

Table 3.2: Interpretation of the main classes in the diagram

Class	Design role
User	Central identity object controlling authentication, authorisation, and role-based access.
Application	Captures the candidate submission lifecycle from initial request to final decision.
Review	Stores supervisor decisions, comments, and status changes during evaluation.
Document	Keeps the document hash, IPFS CID, anchoring status, and transaction traceability fields.
Certificate	Represents the final approved and verifiable certificate issued by the platform.
CertificateRegistrat	Tracks the administrative request that leads to certificate approval or rejection.
FabricLog	Records blockchain actions, statuses, actors, and timestamps for auditability.
Notification	Delivers system events to the concerned users.

The relationships represented in the class diagram are equally important. A user can submit multiple applications, produce multiple reviews depending on the role, and receive multiple notifications. An application can have many reviews during reconsideration cycles. A document can be linked to an application and, once validated, anchored on the ledger. A certificate registration can produce exactly one accepted certificate, which may later be linked to an NFT record for advanced traceability and public verification.

3.4 Cross-Workflow Architecture

3.4.1 Overview

The GRANTA system is designed as a multi-layered architecture that integrates application logic, data persistence, decentralized storage, and blockchain-based verification. The system is organized around six core workflows (WF01–WF06), which together define the full lifecycle of candidate registration, application processing, certificate issuance, and public verification.

Each workflow operates independently at the application level, while sharing common services such as validation, storage, and blockchain anchoring. This design enables scalability, modularity, and strong guarantees of data integrity. Figure 3.3 presents the end-to-end view of all workflows and the system layers that support them.

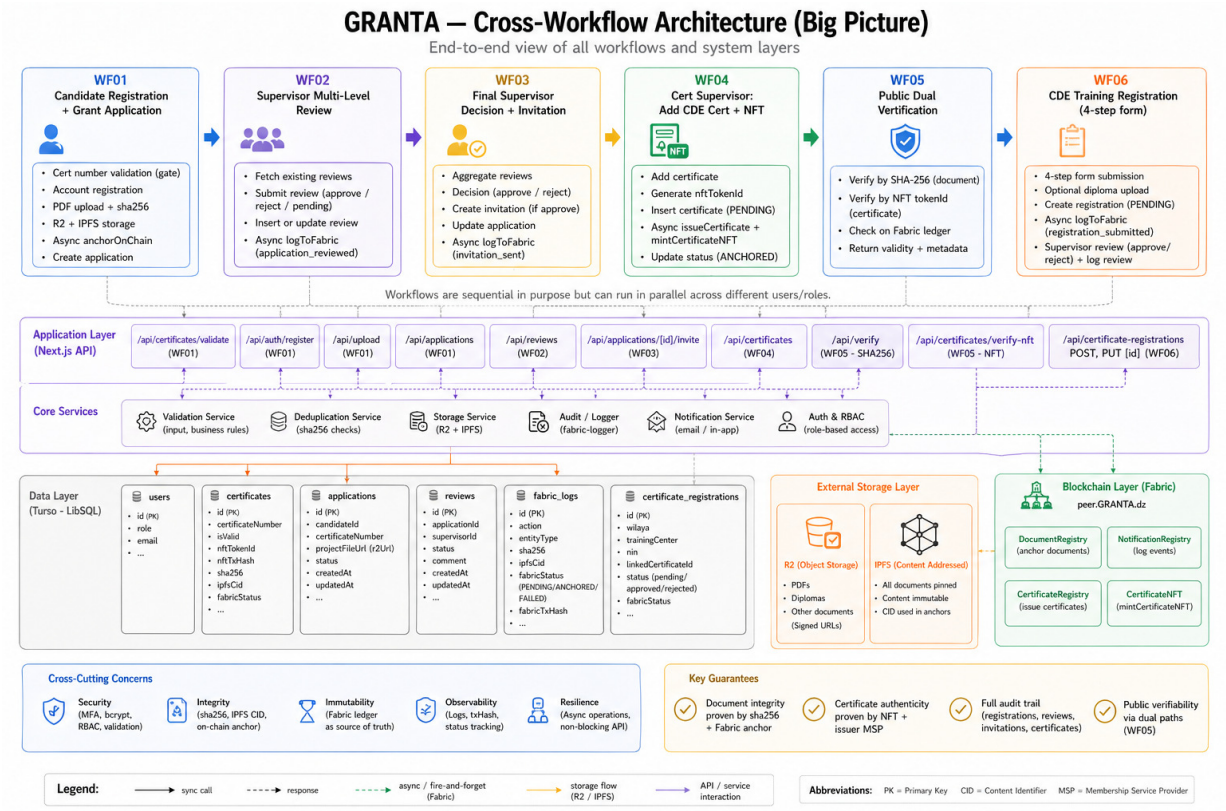


Figure 3.3: GRANTA — Cross-Workflow Architecture (Big Picture): end-to-end view of all workflows and system layers.

3.4.2 Architectural layers

Application Layer (Next.js API)

The application layer exposes RESTful endpoints that orchestrate all workflows. Each workflow maps to a set of API routes:

- **WF01:** `/api/certificates/validate`, `/api/auth/register`, `/api/upload`, `/api/applications`
- **WF02:** `/api/reviews`
- **WF03:** `/api/applications/[id]/invite`
- **WF04:** `/api/certificates`
- **WF05:** `/api/verify`, `/api/certificates/verify-nft`
- **WF06:** `/api/certificate-registrations`

This layer is responsible for input validation, business rule enforcement, request orchestration, and triggering asynchronous blockchain operations.

Core Services Layer

The system relies on several reusable services:

- **Validation Service:** ensures correctness of user input and business constraints.
- **Deduplication Service:** uses SHA-256 hashing to prevent duplicate document storage.
- **Storage Service:** manages uploads to R2 and IPFS.
- **Audit / Logger Service:** records events in `fabric_logs`.
- **Notification Service:** handles invitation and review notifications.
- **Authentication & RBAC:** enforces role-based access control.

Data Layer (Turso / LibSQL)

The relational database acts as the primary operational datastore. Core tables include `users`, `certificates`, `applications`, `reviews`, `certificate_registrations`, and `fabric_logs`. This layer ensures fast query performance, transactional consistency, and referential integrity.

External Storage Layer (R2 + IPFS)

Documents are stored using a hybrid model: **R2 (Object Storage)** stores the actual files (PDFs), while **IPFS** provides content-addressable storage via CID. Each document is associated with a SHA-256 hash (content fingerprint), an IPFS CID (immutable reference), and an R2 URL (accessible file).

Blockchain Layer (Hyperledger Fabric)

The blockchain layer provides immutability and auditability through several chaincodes: **DocumentRegistry** (anchors document hashes), **CertificateRegistry** (issues certificates), **CertificateNFT** (manages NFT-like tokens), and **NotificationRegistry** (logs system events). All blockchain interactions are executed asynchronously via a `fabric-logger` mechanism, ensuring non-blocking API responses.

3.4.3 Workflow integration

The six workflows interact as follows:

- **WF01** initializes the system by validating candidates and storing documents.
- **WF02** enables parallel supervisor reviews.

- **WF03** aggregates reviews and produces final decisions.
- **WF04** issues certificates and mints NFT identifiers.
- **WF05** exposes public verification mechanisms.
- **WF06** manages training registration and approval.

Although these workflows are logically sequential, they are executed independently and can run concurrently across different actors.

3.4.4 Key design patterns

Eventual Consistency. Blockchain operations are performed asynchronously. The database is updated immediately, while blockchain confirmation occurs later.

Idempotency. Duplicate checks are enforced at multiple levels: certificate uniqueness, review uniqueness per supervisor, and invitation uniqueness.

Content-Addressable Storage. All documents are identified by their SHA-256 hash, enabling deduplication, integrity verification, and efficient storage reuse.

Event Logging. All critical actions are recorded in `fabric_logs`, providing audit trails, debugging capability, and blockchain synchronization tracking.

3.4.5 Key guarantees

The architecture ensures **document integrity** (verified via SHA-256 and blockchain anchoring), **certificate authenticity** (ensured through NFT and issuer identity), **auditability** (all actions logged and anchored on-chain), and **public verifiability** (dual verification via hash or NFT).

3.5 WF01 — Candidate Registration and Grant Application

3.5.1 Overview

WF01 covers the full onboarding path of a candidate: from the initial certificate-number gate, through account creation, to project PDF upload and final application submission. It is the entry point of the GRANTA platform and the only workflow that produces both a new `user` record and a new `application` record. Figure 3.4 illustrates the full sequence of operations.

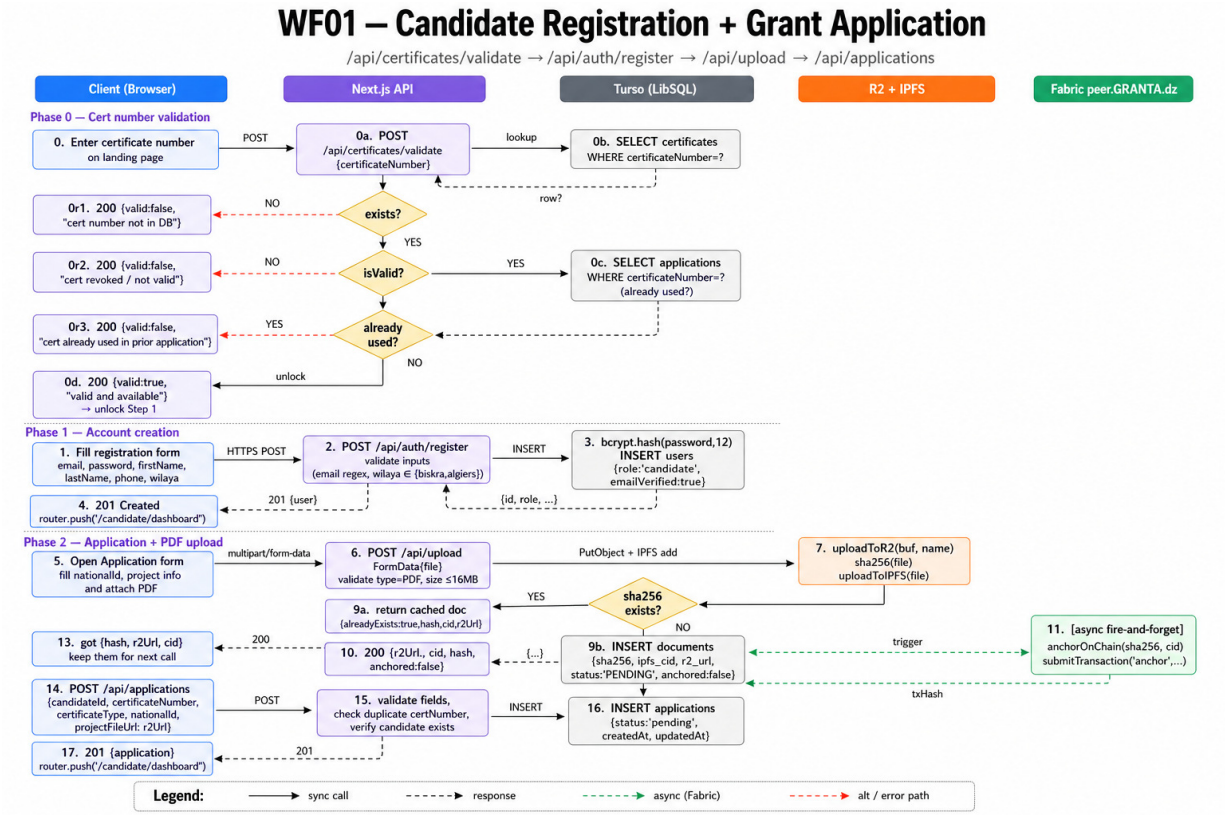


Figure 3.4: WF01 — Candidate Registration and Grant Application sequence.

Purpose

This workflow serves as the primary entry point of the system. It ensures that only eligible candidates can submit applications while establishing strong guarantees of document integrity and traceability.

Design and Justification

A certificate-based validation gate is introduced as an early filtering mechanism. This design decision prevents unauthorized or duplicate applications at the earliest stage, reducing downstream processing overhead.

The adoption of SHA-256 hashing for document identification enables content-addressable storage. This approach ensures that identical documents are stored only once, improving storage efficiency and consistency.

The integration of asynchronous blockchain anchoring reflects a deliberate trade-off between performance and consistency. By decoupling blockchain operations from the request-response cycle, the system maintains low latency for users while ensuring eventual integrity guarantees.

Design rationale

WF01 deliberately separates document storage (R2 + IPFS), business persistence (Turso), and blockchain anchoring (Fabric). The blockchain call is asynchronous so that the user receives an immediate `201 Created` response while the anchor is finalised in the background.

3.6 WF02 — Supervisor Multi-Level Review

3.6.1 Overview

WF02 is the parallel review phase. After WF01, an application is reviewed independently by several supervisors. Each supervisor produces one review per application; the final decision is taken later in WF03 once the aggregate of reviews is available. Figure 3.5 summarises the interaction between the supervisor UI, the `/api/reviews` endpoint, the Turso database, and the Fabric ledger.

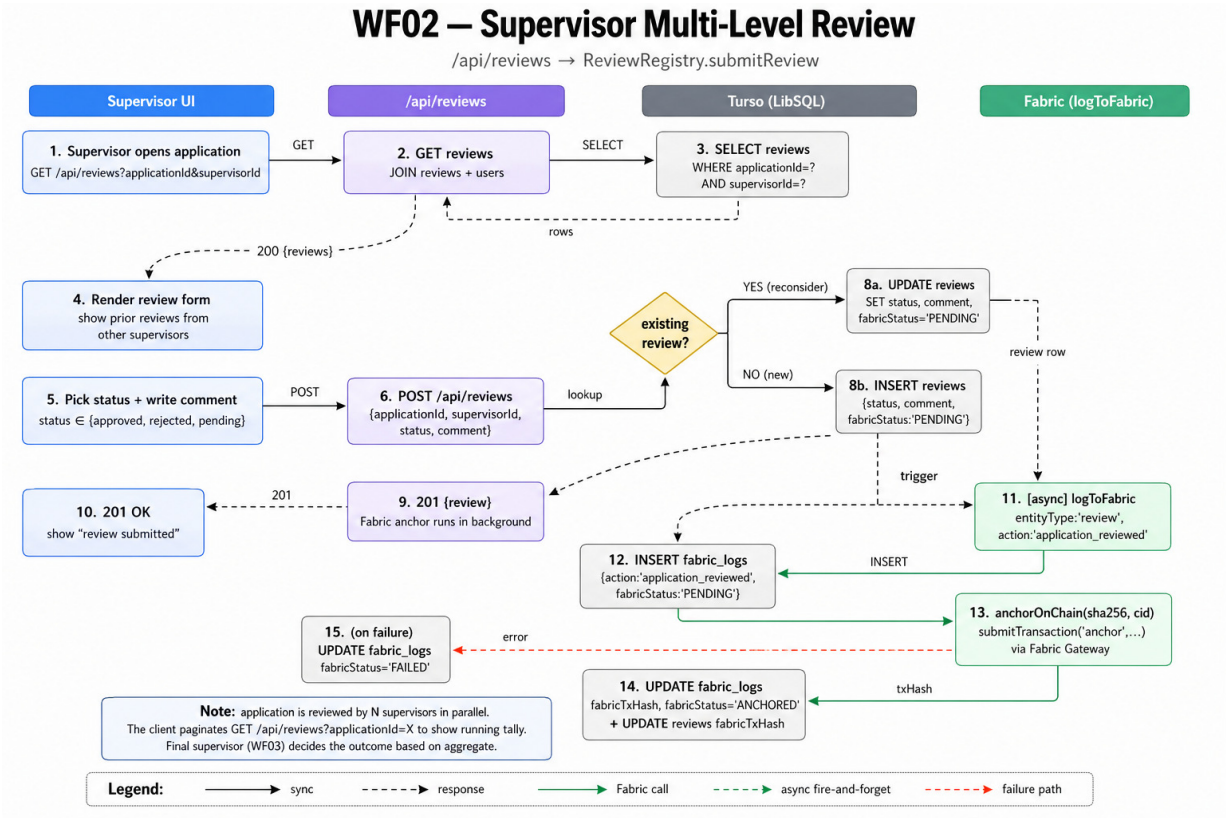


Figure 3.5: WF02 — Supervisor Multi-Level Review sequence.

Purpose

This workflow implements a decentralized evaluation mechanism by allowing multiple supervisors to independently review applications.

Design and Justification

The system adopts a parallel review model, where supervisors operate independently without synchronization constraints. This design increases throughput and avoids bottlenecks associated with sequential approval systems.

The enforcement of one review per supervisor ensures data consistency and prevents conflicting inputs from the same actor.

Blockchain logging is performed asynchronously to maintain responsiveness while ensuring auditability.

3.7 WF03 — Final Supervisor: Decision and Invitation

3.7.1 Overview

WF03 is the decision phase. Once enough reviews have been produced in WF02, the final supervisor opens the application detail, examines the aggregate of approved/rejected/pending reviews, and either rejects the application or approves it and sends an interview invitation. Figure 3.6 presents the full sequence.

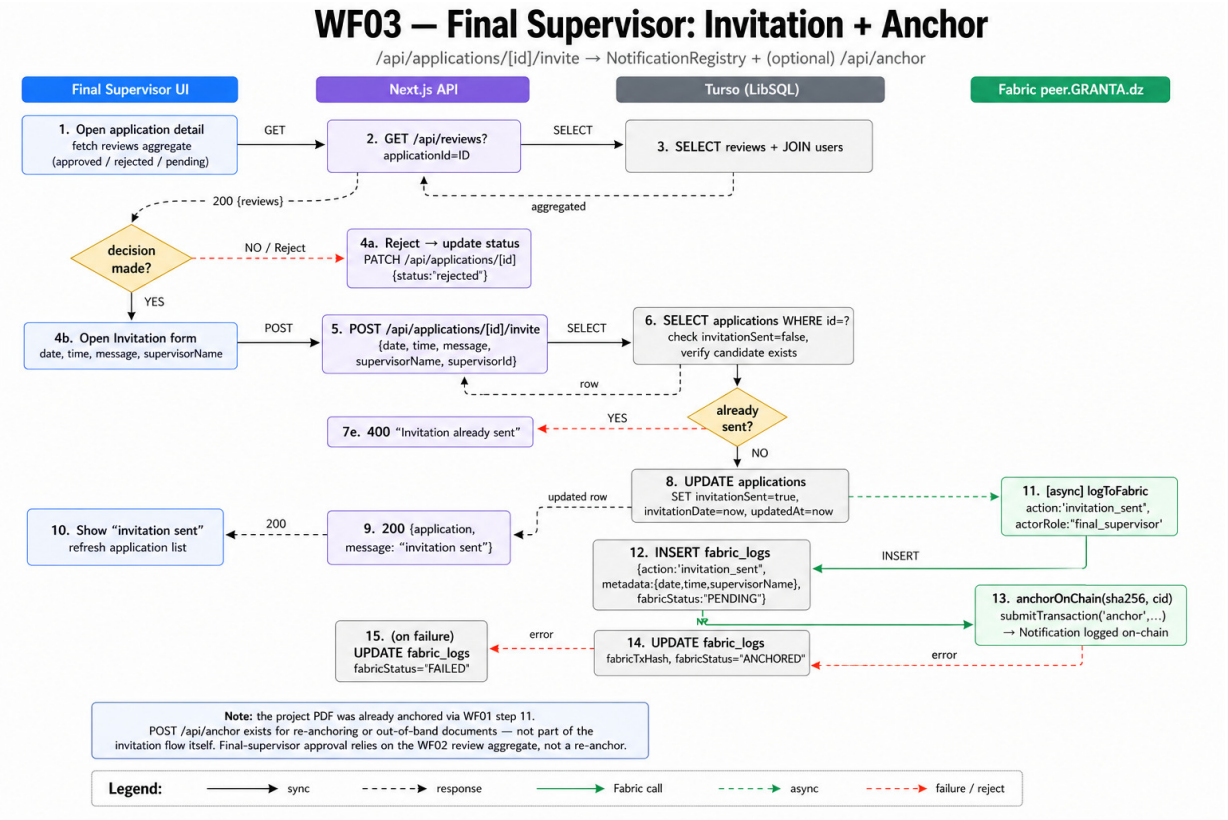


Figure 3.6: WF03 — Final Supervisor: Decision and Invitation sequence.

Purpose

This workflow consolidates distributed reviews into a single decision and initiates the invitation process for successful candidates.

Design and Justification

A centralized decision-making role (final supervisor) is introduced to resolve potential inconsistencies arising from parallel reviews. This ensures a clear and authoritative outcome.

The system enforces a single-invitation constraint to prevent redundant or conflicting actions.

As with previous workflows, blockchain logging is handled asynchronously to preserve system responsiveness.

3.8 WF04 — Cert Supervisor: Add CDE Certificate and NFT Mint

3.8.1 Overview

WF04 is the certificate issuance workflow. Once a candidate has been validated through WF01–WF03, the certificate supervisor adds the CDE certificate, which simultaneously generates a unique NFT token and anchors the certificate on the Fabric ledger. Figure 3.7 details the sequence.

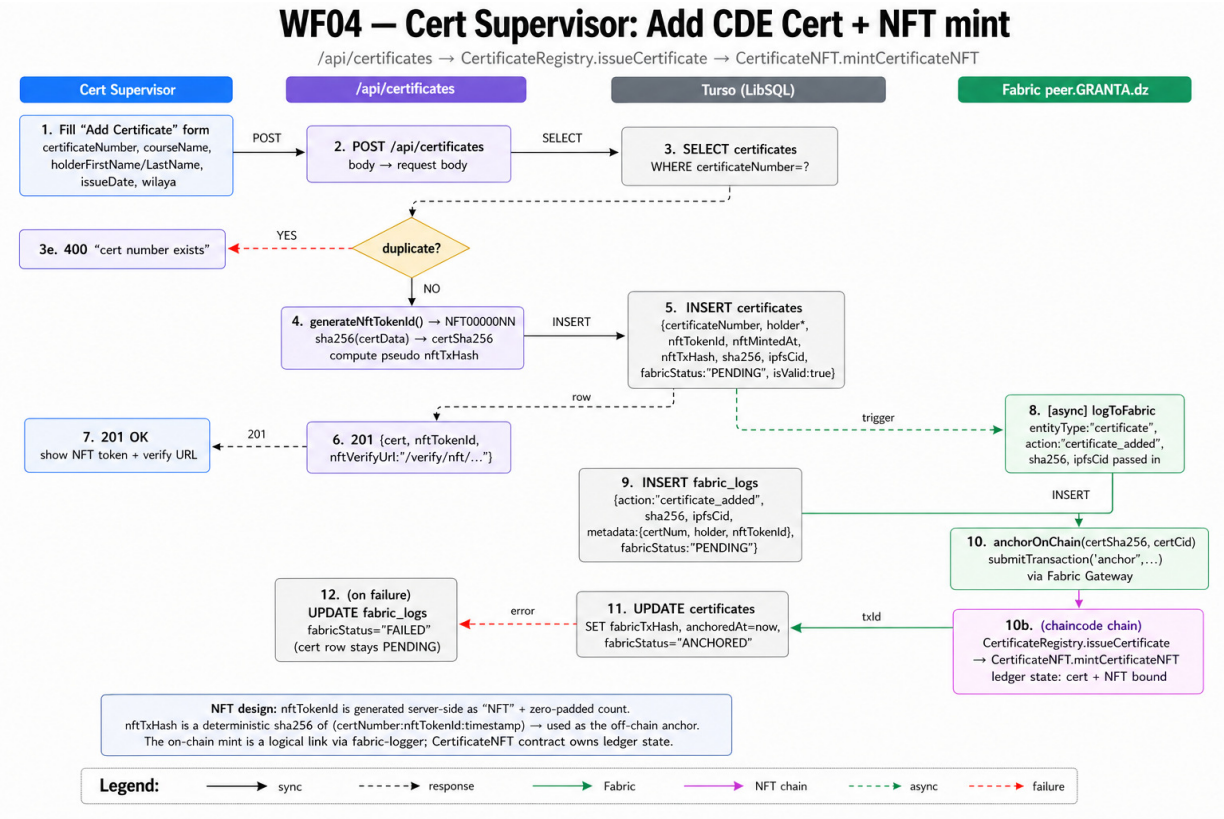


Figure 3.7: WF04 — Cert Supervisor: Add CDE Certificate and NFT Mint sequence.

Purpose

This workflow formalizes the issuance of certificates and associates them with NFT-like identifiers to ensure authenticity and traceability.

Design and Justification

The generation of a deterministic NFT token identifier provides a unique and verifiable representation of each certificate.

By linking certificate issuance with blockchain anchoring and NFT minting, the system establishes a hybrid trust model combining centralized control with decentralized verification.

The use of asynchronous blockchain interaction minimizes latency while preserving auditability.

3.9 WF05 — Public Dual Verification

3.9.1 Overview

WF05 is the public verification workflow. Any external visitor (employer, institution, candidate) can independently verify a certificate without an authenticated account. Two paths are exposed: a SHA-256 hash path on `/api/verify`, and an NFT path on `/api/certificates/verify-nft`. Figure 3.8 illustrates both paths.

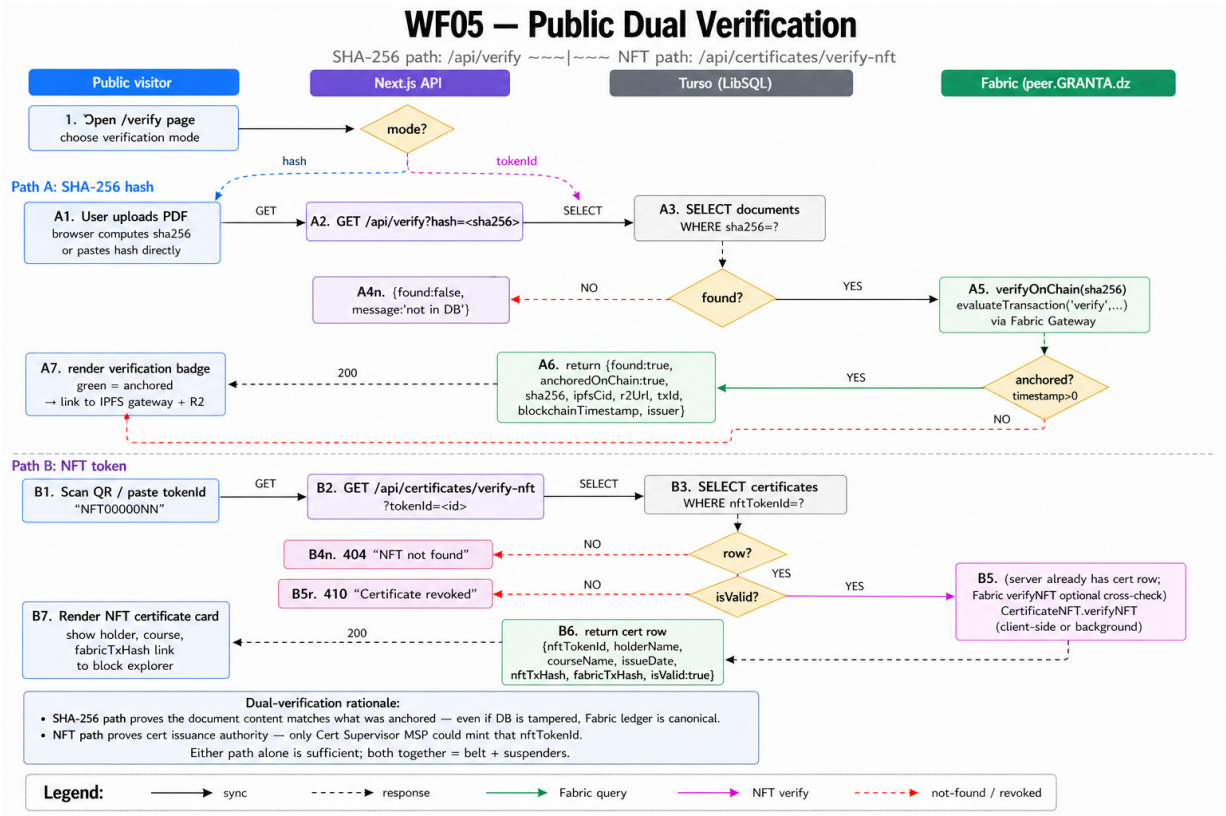


Figure 3.8: WF05 — Public Dual Verification (SHA-256 path and NFT path).

Purpose

This workflow provides external users with the ability to verify documents and certificates independently of the system.

Design and Justification

A dual-verification model is implemented:

- SHA-256 verification ensures document integrity
- NFT verification ensures certificate authenticity

This redundancy strengthens trust by allowing verification through multiple independent mechanisms.

3.10 WF06 — CDE Training Registration (4-step Form)

3.10.1 Overview

WF06 is the training registration workflow. A candidate registers for a CDE training session through a four-stage form, and the certificate supervisor approves or rejects each registration. Figure 3.9 presents the complete sequence including both the candidate submission phase and the approval phase.

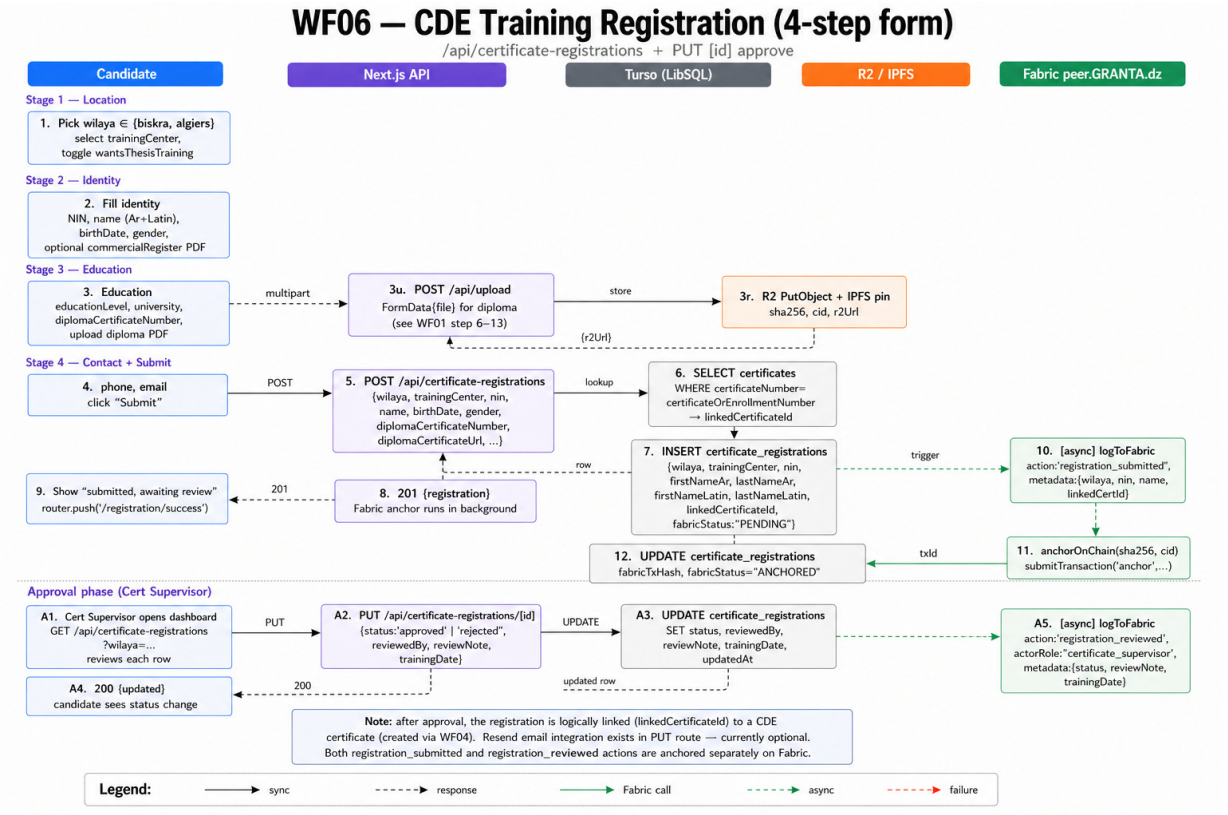


Figure 3.9: WF06 — CDE Training Registration (4-step form) and approval phase.

Candidate submission (Stages 1–4)

- **Stage 1 — Location.** The candidate picks the wilaya in {biskra, algiers}, selects the training center, and toggles `wantsThesisTraining`.
- **Stage 2 — Identity.** The candidate fills NIN, name (Arabic + Latin), birth date, gender, and an optional commercial register PDF.
- **Stage 3 — Education.** The candidate provides education level, university, diploma certificate number, and uploads the diploma PDF. The PDF is sent to `POST /api/upload` (same logic as WF01 steps 6–13), stored to R2 and IPFS, and the resulting URL is kept.
- **Stage 4 — Contact and Submit.** The candidate enters phone and email and clicks *Submit*.

Purpose

This workflow manages candidate enrollment in training programs and enables supervisor-based approval processes.

Design and Justification

A multi-step form is used to improve user experience and data accuracy. This approach reduces input errors by segmenting information into logical stages.

The system attempts to link registrations with existing certificates, promoting data reuse and consistency across workflows.

Approval is handled by a supervisor, ensuring human oversight in the decision process.

Important note

After approval, the registration is logically linked (`linkedCertificateId`) to a CDE certificate created via WF04. Re-send-email integration exists in the PUT route currently optional. Both `registration_submitted` and `registration_reviewed` actions are anchored separately on Fabric.

3.11 Hyperledger Fabric Deployment Architecture

3.11.1 Complete Fabric network diagram

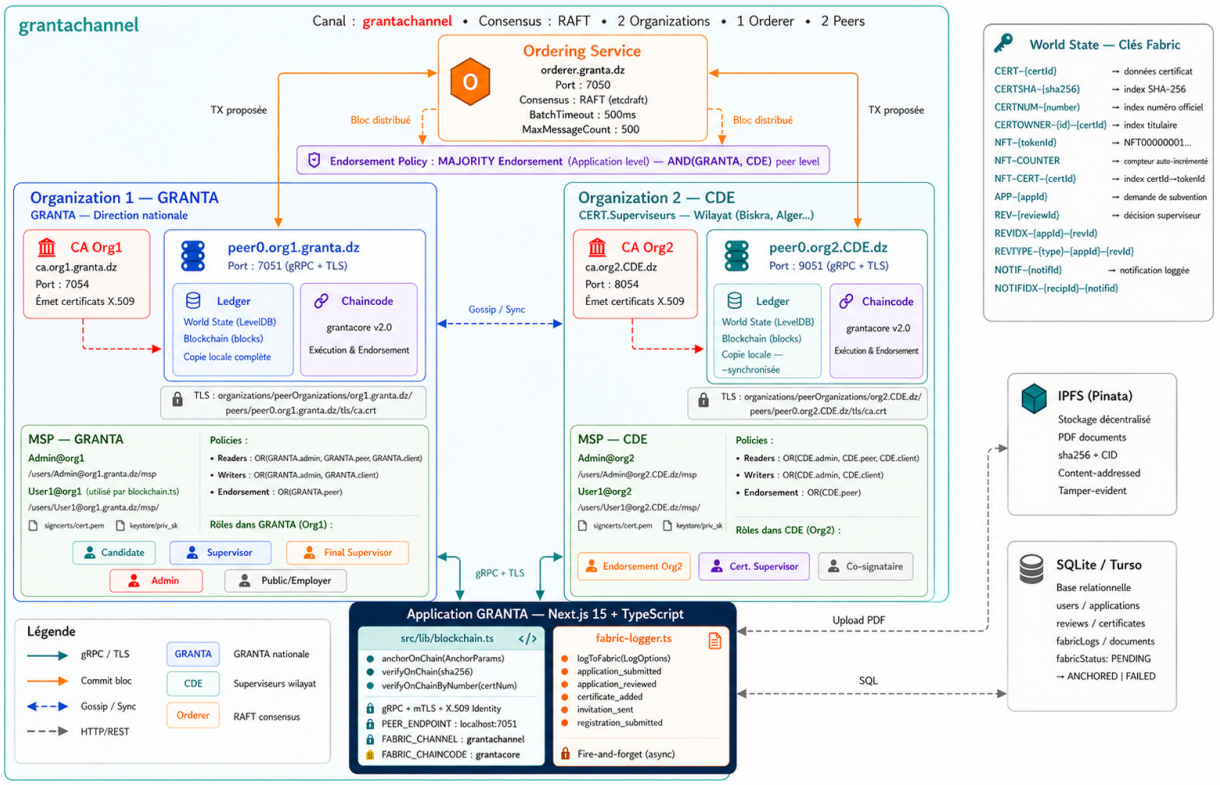


Figure 3.10: Complete Fabric network diagram GRANTA.

The architecture is built around a permissioned blockchain network using Hyperledger Fabric, designed to ensure trust, transparency, and decentralized validation across authorized organizations involved in GRANTA processes.

At its core, the system is organized into a single channel (grantachannel) that connects two main organizations: GRANTA (national authority) and CDE (certification supervisors at the wilaya level). Each organization operates its own peer node, maintaining a synchronized copy of the ledger and executing the shared smart contract (chaincode: grantacore), with the World State database (LevelDB) enabling efficient data queries.

A RAFT-based ordering service ensures consistency across the network by ordering endorsed transactions into blocks and distributing them to all peers, guaranteeing a deterministic and tamper-resistant blockchain state.

The endorsement policy, based on a majority mechanism and requiring participation from both GRANTA and CDE peers, enforces collaborative validation. This prevents any single organization from unilaterally approving transactions and strengthens trust among stakeholders.

The application layer, implemented in Next.js and TypeScript, acts as the client gateway to the blockchain network through blockchain.ts, invoking smart contract operations such as certificate anchoring and verification over secure gRPC connections using TLS and X.509 identities. In parallel, fabric-logger.ts records application events asynchronously on-chain, ensuring full auditability of system activities.

A hybrid storage model is adopted: large certificate documents are stored off-chain in IPFS (Pinata), while their cryptographic hashes and content identifiers (CID) are anchored on-chain to ensure integrity and reduce storage overhead. Relational metadata and application state are managed in SQLite/Turso, enabling efficient off-chain data processing synchronized with blockchain state transitions.

Security and governance are enforced through Fabric MSP policies, TLS authentication, endorsement constraints, and multi-level organizational roles. This architecture combines consortium-based trust, cryptographic immutability, scalable storage, and multi-party governance to support secure, auditable, and efficient digital certification and grant management.

3.11.2 Certificate registration workflow over Fabric

The certificate registration workflow over the Fabric network follows logical stages:

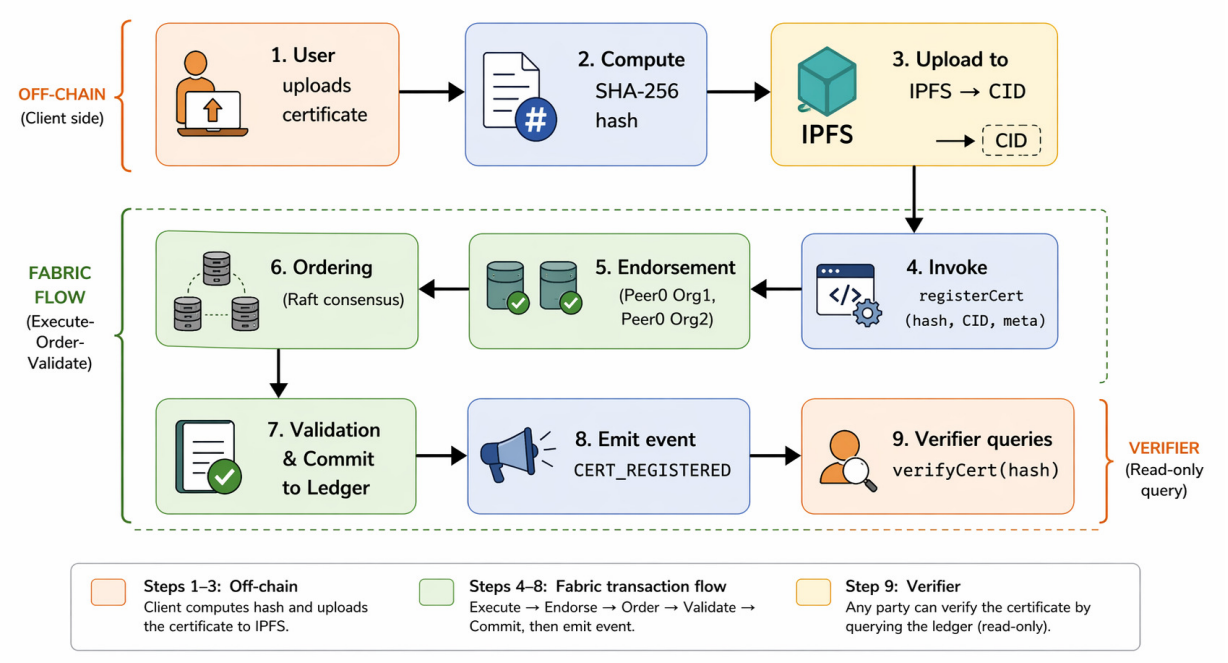


Figure 3.11: End-to-end certificate registration and verification workflow. Steps 1–3 are executed off-chain (hashing and IPFS upload). Steps 4–8 correspond to the Fabric execute-order-validate transaction flow. Step 9 illustrates a verifier independently checking the authenticity of a certificate through a read-only chaincode query.

The nine steps of the workflow are:

1. **User upload.** The certificate holder or issuing authority uploads the raw certificate document (e.g., PDF) through the client web application.
2. **SHA-256 hashing.** The client application computes the SHA-256 hash of the document. This 256-bit fingerprint serves as the cryptographic identity of the certificate.
3. **IPFS storage.** The document is uploaded to the IPFS network, which returns a content identifier (CID) derived from the content itself. IPFS provides decentralized, content-addressed storage that cannot be silently modified.
4. **Chaincode invocation.** The client invokes the `registerCertificate` function of the `CertificateRegistry` chaincode with three arguments: the SHA-256 hash, the IPFS CID, and a JSON metadata object (issuer, holder, certificate type, issue date, etc.).
5. **Endorsement.** The transaction proposal is sent to the endorsing peers of both organizations (at minimum Peer0 of Org1MSP and Peer0 of Org2MSP) according to the endorsement policy `AND('Org1MSP.peer', 'Org2MSP.peer')`. Each peer executes the chaincode against its local world state and returns a signed endorsement.
6. **Ordering.** The client collects the endorsements and submits the signed transaction to the Raft ordering service, which orders it relative to other concurrent transactions and includes it in a new block.
7. **Validation and commit.** All peers receive the new block, validate that each transaction satisfies its endorsement policy and has no read-write set conflicts, and commit the valid transactions to the ledger and world state.
8. **Event emission.** The chaincode emits a `CERT_REGISTERED` event that is propagated to subscribed client applications via gRPC streams, enabling real-time notification.
9. **Verification.** Any authorized verifier (employer, institution, candidate) can subsequently query the chaincode function `verifyCertificate(hash)` to confirm the authenticity of a certificate. This operation is read-only (evaluated locally on a single peer) and therefore offers significantly higher throughput and lower latency than write operations.

3.11.3 NFT Minting Workflow

Figure 3.12 illustrates the end-to-end flow from certificate registration to NFT minting and public verification.

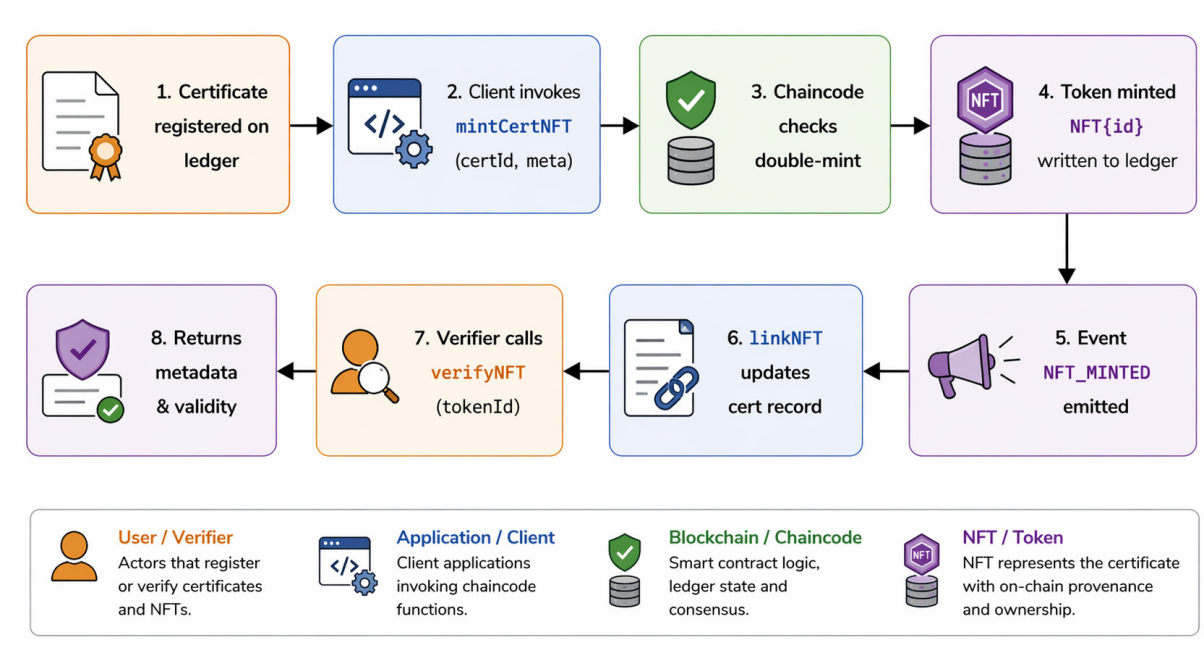


Figure 3.12: NFT minting and verification workflow. Steps 1–4 cover the minting path: after the certificate is registered on the ledger, the client invokes mintCertificateNFT; the chaincode enforces the double-mint guard and writes the token atomically. Steps 5–6 propagate the token identifier back to the certificate record via linkNFT. Steps 7–8 illustrate a verifier retrieving the full OpenSea-compatible metadata and validity status through a single read-only peer query.

3.12 On-chain Transactions and Their Meaning

The extracted chaincode modules define a rich set of transactions that support the full application and certificate lifecycle. Table 3.3 summarises the most relevant operations.

Table 3.3: Main blockchain transactions used in the platform

Transaction	Type	Purpose
submitApplication	Invoke	Creates an on-chain representation of an application submission and preserves its first auditable state.
updateApplicationStat	Invoke	Updates the application status after review or final decision.
submitReview	Invoke	Logs a supervisor review and its associated decision metadata.
issueCertificate	Invoke	Registers a newly issued certificate on the ledger.
verifyCertificate	Query	Reads a certificate record from the ledger for verification.
verifyCertificateBySH	Query	Verifies a certificate directly from its SHA-256 fingerprint.
verifyCertificateByNu	Query	Retrieves a certificate using its official certificate number.
linkNFT	Invoke	Associates an issued certificate with an NFT token identifier.
mintCertificateNFT	Invoke	Creates the NFT representation attached to a certificate.
getCertificateHistory	Query	Returns the full modification history of a certificate record.
logNotification	Invoke	Stores a notification event related to a business action.
getNotificationsByRec	Query	Retrieves notifications addressed to a given user.

The semantics of these transactions can be grouped into four families. The first family concerns application management, where the ledger acts as an immutable audit trail for the lifecycle of a candidate request. The second family concerns certificate issuance, where a validated record becomes a verifiable digital certificate. The third family concerns NFT linkage, which adds a token-based representation suitable for advanced ownership or public showcase scenarios. The fourth family concerns traceability and notification, which allows the platform to reconstruct who did what and when.

3.12.1 Explanation of the most important transactions

`issueCertificate`. This is the core issuance operation. It creates the canonical ledger record associated with a certificate after the off-chain validations have been completed. Once committed, the certificate becomes tamper-evident and can be checked by any authorised verifier.

`verifyCertificate` and `verifyCertificateBySHA256`. These are read-only queries. They do not modify the ledger state, but they provide strong verification capabilities by retrieving the authoritative blockchain record either from a business identifier or directly from a cryptographic hash.

`mintCertificateNFT`. This function extends the certificate model with a token representation. In practice, it is useful when the platform needs a compact public-facing identifier or when certificate ownership and visibility must be materialised through an NFT abstraction.

`getCertificateHistory`. This transaction is essential for auditability. It allows the institution or the verifier to reconstruct the full history of a certificate and detect any revocation, update, or administrative event associated with its lifecycle.

3.13 Conclusion

This redesigned chapter has provided a comprehensive and structured representation of the GRANTA platform through complementary UML, workflow, and architectural perspectives. The use case diagram delineated the functional scope and system boundaries, while the class diagram formalised the core domain entities and their interrelationships. The cross-workflow architectural view (big picture) then placed every API route, service, table, and chaincode in its precise layer, and the six per-workflow sections (WF01 through WF06) decomposed the dynamic behaviour of the platform into autonomous, independently deployable scenarios — from candidate registration up to public dual verification and CDE training registration. The deployment architecture was finally presented through detailed diagrammatic representations capturing the configuration of the Hyperledger Fabric network that underpins certificate anchoring and verification processes.

The overall design validates the adoption of a hybrid trust model, in which multiple technological layers collaboratively ensure system robustness and reliability. The web application layer enhances usability and user interaction, the database layer guarantees operational consistency and structured data management, IPFS and R2 enable decentralised and content-addressable storage, and Hyperledger Fabric provides immutability, traceability, and auditable certification events. Collectively, these components establish a secure, scalable, and trustworthy architecture for certificate registration, verification, and long-term traceability. The seven-workflow decomposition adopted in this chapter — one

cross-workflow big picture and six independent per-workflow views — proved to be a powerful structuring principle: it makes the platform easier to reason about, easier to test, and easier to evolve in isolation, while still preserving the strong cross-cutting guarantees of integrity, auditability, and public verifiability that define the GRANTA system.

Chapter 4

Implementation

GRANTA — Blockchain-Based Document Verification Platform

Hyperledger Fabric 2.5 • Next.js 15 • TypeScript • SQLite • IPFS

4.1 Introduction

This chapter presents the concrete implementation of the GRANTA platform, covering all components designed in the previous chapter. Each section describes the technical realisation of a specific module, accompanied by code excerpts, configuration details, and implementation decisions.

The implementation is organised into six main modules: (1) the development environment setup, (2) the database layer, (3) the Hyperledger Fabric blockchain network and chaincode, (4) the IPFS integration, (5) the Next.js web application with its API routes, and (6) the user interface for each actor role.

4.2 Development Environment

4.2.1 Technologies and Versions

Table 4.1 lists the full technology stack used in the GRANTA platform.

Table 4.1: Technology stack and versions

Technology	Version	Role
Next.js	15.3.5	Full-stack web framework (App Router + API Routes)
React	19.0.0	UI component library
TypeScript	5.x	Static typing for all application code
Hyperledger Fabric	2.5.0	Permissioned blockchain network
fabric-gateway	1.4.0	Node.js SDK for Fabric transaction submission
@grpc/grpc-js	1.10.0	gRPC transport for Fabric peer communication
Drizzle ORM	0.41.0	Type-safe SQLite query builder
SQLite / libsql	0.15.15	Embedded relational database
IPFS / Pinata	API v1	Decentralised file storage
bcrypt	6.0.0	Password hashing
Tailwind CSS	3.x	Utility-first CSS framework
Radix UI	latest	Accessible UI component primitives
Docker	24+	Fabric network containerisation
Node.js	18+	JavaScript runtime

4.2.2 Project Structure

```
GRANTA/
|-- src/
|   |-- app/                          # Next.js App Router pages
|   |   |-- candidate/dashboard/      # Candidate interface
|   |   |-- supervisor/dashboard/     # Supervisor interface
|   |   |-- final-supervisor/         # Final supervisor + blockchain
|   |   |-- admin/                    # Admin dashboard + live monitor
|   |   |-- verify/                   # Public verification page
|   |   `-- api/                      # REST API routes
|   |       |-- anchor/route.ts       # Blockchain anchoring endpoint
|   |       |-- verify/route.ts       # Verification endpoint
|   |       |-- applications/         # Application CRUD
|   |       `-- auth/                 # Login / Register
|   |-- db/
|   |   |-- index.ts                  # Drizzle client
|   |   `-- schema.ts                 # All table definitions
|   `-- lib/
|       |-- blockchain.ts             # Fabric Gateway client
```

```
|       |-- hash.ts                    # SHA-256 utility
|       |-- ipfs.ts                    # Pinata upload
|       |-- auth.ts                    # Auth helpers
|       `-- websocket-notify.ts        # Live dashboard notifier
|-- fabric/
|   `-- fabric-samples/
|       `-- test-network/
|           |-- chaincode/
|           |   `-- grantacore/index.js
|           `-- organizations/         # Crypto material
`-- caliper/                           # Benchmark configuration
```

Listing 4.1: Project directory structure

4.2.3 Environment Configuration

The platform uses environment variables for all sensitive configuration. The `.env` file contains:

```
# Database
DATABASE_URL=file:./GRANTAplatform.db

# IPFS Storage (Pinata)
PINATA_API_KEY=<your_pinata_key>
PINATA_SECRET=<your_pinata_secret>

# Hyperledger Fabric
FABRIC_CHANNEL=GRANTAchannel
FABRIC_CHAINCODE=grantacore
FABRIC_PEER=localhost:7051
FABRIC_PEER_HOST=peer0.org1.example.com
FABRIC_MSP_ID=Org1MSP
FABRIC_CRYPTO_PATH=./fabric/fabric-samples/test-network/organizations/...

# Email (Resend)
RESEND_API_KEY=<your_resend_key>

# WebSocket Live Dashboard
WS_NOTIFY_URL=http://localhost:8081/notify
```

Listing 4.2: Environment variables (.env)

4.3 Database Implementation

The database layer uses SQLite as the storage engine with Drizzle ORM providing type-safe query construction. SQLite was chosen for its zero-configuration setup and embedded deployment model, appropriate for a single-server application.

4.3.1 Schema Definition

All tables are defined in `src/db/schema.ts` using Drizzle's declarative API. Listing 4.3 shows the core table definitions.

```
import { sqliteTable, integer, text } from 'drizzle-orm/sqlite-core';

export const users = sqliteTable('users', {
  id:            integer('id').primaryKey({ autoIncrement: true }),
  email:         text('email').notNull().unique(),
  password:      text('password').notNull(),   // bcrypt hash
  role:          text('role').notNull(),        // candidate | supervisor
      | ...
  firstName:     text('first_name'),
  lastName:      text('last_name'),
  wilaya:        text('wilaya'),                // biskra | algiers
  emailVerified: integer('email_verified',
                   { mode: 'boolean' }).default(false),
  createdAt:     text('created_at').notNull(),
});

export const documents = sqliteTable('documents', {
  id:         text('id').primaryKey(),           // UUID
  fileName:   text('file_name').notNull(),
  sha256:     text('sha256').notNull().unique(),  // Document fingerprint
  ipfsCid:    text('ipfs_cid').notNull(),        // IPFS content address
  r2Url:      text('r2_url').notNull(),          // Cloudflare R2 backup
  status:     text('status').notNull(),          // PENDING|ANCHORED|
     FAILED
  anchored:   integer('anchored',
                { mode: 'boolean' }).notNull().default(false),
  txHash:     text('tx_hash'),                   // Fabric transaction ID
  createdAt: text('created_at'),
});
```

Listing 4.3: Core database schema (src/db/schema.ts)

4.3.2 Database Client

```
import { drizzle }      from 'drizzle-orm/libsql';
import { createClient } from '@libsql/client';
import * as schema      from './schema';

const client = createClient({
  url: process.env.DATABASE_URL || 'file:./GRANTAplatform.db',
});

export const db = drizzle(client, { schema });
```

Listing 4.4: Drizzle database client (src/db/index.ts)

4.4 Blockchain Implementation

4.4.1 Hyperledger Fabric Network Setup

The Fabric network uses the `test-network` scaffold from `fabric-samples`, configured with a custom channel (`GRANTAchannel`) and two organisations (`Org1MSP` and `Org2MSP`). The network is launched using Docker Compose:

```
# Start the Fabric network with CA support
cd fabric/fabric-samples/test-network
./network.sh up createChannel -c GRANTAchannel -ca

# Deploy the grantacore chaincode
./network.sh deployCC -c GRANTAchannel       \
  -ccn grantacore                    \
  -ccp ../chaincode/grantacore       \
  -ccl javascript                          \
  -ccv 1.0

# Verify deployment
peer lifecycle chaincode querycommitted -C GRANTAchannel
# Output: grantacore v1.0
```

Listing 4.5: Network startup and chaincode deployment

4.4.2 Chaincode Implementation

The `grantacore` chaincode is implemented in JavaScript using the `fabric-contract-api` library. It exposes three functions: `ApplicationRegistry`, `grantacore`, and `ReviewRegistry`,`CertificateRegistry`,`CertificateNFT`,`NotificationRegistry`.

```
'use strict';
const shim = require('fabric-shim');

const ApplicationRegistry  = require('./contracts/ApplicationRegistry');
const ReviewRegistry       = require('./contracts/ReviewRegistry');
const CertificateRegistry  = require('./contracts/CertificateRegistry');
const CertificateNFT       = require('./contracts/CertificateNFT');
const NotificationRegistry = require('./contracts/NotificationRegistry');

// Instantiate contracts
const appRegistry   = new ApplicationRegistry();
const revRegistry   = new ReviewRegistry();
const certRegistry  = new CertificateRegistry();
const nftRegistry   = new CertificateNFT();
const notifRegistry = new NotificationRegistry();

// ---- Route table
    ------------------------------------------------------------------
// Maps "FunctionName" → handler(stub, params)
```

```
const ROUTES = {

  // ---- Application
  submitRequest:          (s, p) => appRegistry.submitRequest(s, p),
  updateRequestStatus:    (s, p) => appRegistry.updateRequestStatus(s, p),
  getRequest:             (s, p) => appRegistry.getRequest(s, p),
  getRequestHistory:      (s, p) => appRegistry.getRequestHistory(s, p),
  getRequestsByCandidate:(s, p) => appRegistry.getRequestsByCandidate(s,
     p),
  verifyRequest:          (s, p) => appRegistry.verifyRequest(s, p),

  // ---- Review ----
  submitReview:               (s, p) => revRegistry.submitReview(s, p),
  getReview:                  (s, p) => revRegistry.getReview(s, p),
  getReviewsByRequest:    (s, p) => revRegistry.getReviewsByRequest(s, p),
  getReviewsByType:           (s, p) => revRegistry.getReviewsByType(s, p)
     ,
  getReviewsBySupervisor:     (s, p) => revRegistry.getReviewsBySupervisor
     (s, p),
  verifyReview:               (s, p) => revRegistry.verifyReview(s, p),

  // ---- Certificate ----
  issueCertificate:           (s, p) => certRegistry.issueCertificate(s, p
     ),
  linkNFT:                    (s, p) => certRegistry.linkNFT(s, p),
  revokeCertificate:          (s, p) => certRegistry.revokeCertificate(s,
     p),
  getCertificate:             (s, p) => certRegistry.getCertificate(s, p),
  verifyCertificate:          (s, p) => certRegistry.verifyCertificate(s,
     p),
  verifyCertificateBySHA256: (s, p) => certRegistry.
     verifyCertificateBySHA256(s, p),
  verifyCertificateByNumber: (s, p) => certRegistry.
     verifyCertificateByNumber(s, p),
  getCertificatesByHolder:    (s, p) => certRegistry.
     getCertificatesByHolder(s, p),
  getCertificateHistory:      (s, p) => certRegistry.getCertificateHistory
     (s, p),

  // ---- NFT ----
  mintCertificateNFT:         (s, p) => nftRegistry.mintCertificateNFT(s,
     p),
  getNFT:                     (s, p) => nftRegistry.getNFT(s, p),
  getNFTByCertificate:        (s, p) => nftRegistry.getNFTByCertificate(s,
      p),
  getNFTsByOwner:             (s, p) => nftRegistry.getNFTsByOwner(s, p),
  verifyNFT:                  (s, p) => nftRegistry.verifyNFT(s, p),
  burnNFT:                    (s, p) => nftRegistry.burnNFT(s, p),
  totalSupply:                (s, p) => nftRegistry.totalSupply(s, p),

  // ---- Notification ----
```

```
  logNotification:              (s, p) => notifRegistry.logNotification(s, p),
  checkInvitationExpiry:        (s, p) => notifRegistry.checkInvitationExpiry(s, p),
  getNotification:              (s, p) => notifRegistry.getNotification(s, p),
  getNotificationsByRecipient:  (s, p) => notifRegistry.getNotificationsByRecipient(s, p),
  getNotificationsByEntity:     (s, p) => notifRegistry.getNotificationsByEntity(s, p),
};

// ---- Main Chaincode Class ----------------------------------------------------
class GRANTACore {

  async Init(stub) {
    const args = stub.getFunctionAndParameters();
    console.log('GRANTACore chaincode initialized', args);
    return shim.success(Buffer.from('GRANTACore initialized'));
  }

  async Invoke(stub) {
    const { fcn, params } = stub.getFunctionAndParameters();
    console.log(`[GRANTACore] Invoke: ${fcn}`, params);

    const handler = ROUTES[fcn];

    if (!handler) {
      const errMsg = `Function "${fcn}" not found. Available: ${Object.keys(ROUTES).join(', ')}`;
      console.error(errMsg);
      return shim.error(Buffer.from(errMsg));
    }

    try {
      const result = await handler(stub, params);
      return shim.success(Buffer.from(result || '{}'));
    } catch (err) {
      console.error(`[GRANTACore] Error in ${fcn}:`, err.message);
      return shim.error(Buffer.from(err.message));
    }
  }
}

shim.start(new GRANTACore());
```

Listing 4.6: grantacore chaincode (grantacore/index.js)

4.4.3 Fabric Gateway Client

The Next.js API communicates with the Fabric network through the `@hyperledger/fabric-gateway` SDK. The `blockchain.ts` module provides two exported functions used by the API routes:

```
import * as grpc from '@grpc/grpc-js';
import { connect } from '@hyperledger/fabric-gateway';
import * as fs from 'fs';

const CHANNEL_NAME   = process.env.FABRIC_CHANNEL   || 'GRANTAchannel';
const CHAINCODE_NAME = process.env.FABRIC_CHAINCODE  || 'grantacore';
const PEER_ENDPOINT  = process.env.FABRIC_PEER       || 'localhost:7051';

function newGrpcConnection(): grpc.Client {
  const tlsRootCert    = fs.readFileSync(TLS_CERT_PATH);
  const tlsCredentials = grpc.credentials.createSsl(tlsRootCert);
  return new grpc.Client(PEER_ENDPOINT, tlsCredentials, {
    'grpc.ssl_target_name_override': PEER_HOST,
  });
}

// Anchor a document on the Fabric ledger
export async function anchorOnChain(
    sha256: string, cid: string): Promise<string> {
  const client  = newGrpcConnection();
  const gateway = await newGateway(client);
  try {
    const contract = gateway
      .getNetwork(CHANNEL_NAME).getContract(CHAINCODE_NAME);
    const result = await contract.submitTransaction(
      'anchor', sha256, cid);
    return JSON.parse(Buffer.from(result).toString()).txId;
  } finally { gateway.close(); client.close(); }
}

// Verify a document on the Fabric ledger (read-only)
export async function verifyOnChain(sha256: string) {
  const client  = newGrpcConnection();
  const gateway = await newGateway(client);
  try {
    const contract = gateway
      .getNetwork(CHANNEL_NAME).getContract(CHAINCODE_NAME);
    const result = await contract.evaluateTransaction(
      'verify', sha256);
    return JSON.parse(Buffer.from(result).toString());
  } catch { return null; }
  finally { gateway.close(); client.close(); }
}
```

Listing 4.7: Fabric Gateway client (src/lib/blockchain.ts)

4.5 IPFS Integration and Document Hashing

4.5.1 SHA-256 Hash Generation

Each uploaded document is fingerprinted using the SHA-256 algorithm via the Web Crypto API. This produces a unique 64-character hexadecimal string that serves as the document's identifier on the blockchain:

```typescript
export async function sha256(file: File): Promise<string> {
  const buffer     = await file.arrayBuffer();
  const hashBuffer = await crypto.subtle.digest('SHA-256', buffer);
  return Array.from(new Uint8Array(hashBuffer))
    .map(b => b.toString(16).padStart(2, '0'))
    .join('');
}
// Example output: 'a3f2c1...d8e7b9'  (64 hex characters)
```

Listing 4.8: SHA-256 hash utility (src/lib/hash.ts)

4.5.2 IPFS Upload via Pinata

Uploaded PDF files are stored on IPFS through the Pinata pinning service. Pinata returns an IPFS Content Identifier (CID) that permanently addresses the file content:

```typescript
import axios    from 'axios';
import FormData from 'form-data';

export async function uploadToIPFS(file: File): Promise<string> {
  const data   = new FormData();
  const buffer = Buffer.from(await file.arrayBuffer());
  data.append('file', buffer, file.name);

  const res = await axios.post(
    'https://api.pinata.cloud/pinning/pinFileToIPFS',
    data,
    { headers: {
        pinata_api_key:        process.env.PINATA_API_KEY,
        pinata_secret_api_key: process.env.PINATA_SECRET,
        ...data.getHeaders(),
    }}
  );
  return res.data.IpfsHash;  // e.g. 'QmXyz...abc'
}
// Accessible at: https://gateway.pinata.cloud/ipfs/{CID}
```

Listing 4.9: IPFS upload via Pinata (src/lib/ipfs.ts)

The combination of SHA-256 hashing and IPFS storage provides a two-layer integrity guarantee: the SHA-256 hash is stored on the blockchain and changes if any byte of the document is modified, while IPFS content addressing ensures the stored file matches its CID exactly.

4.6 API Routes Implementation

The Next.js App Router provides API routes as TypeScript files under `src/app/api/`. Each route handles a specific resource and HTTP method.

4.6.1 Authentication — POST /api/auth/login

```
export async function POST(request: NextRequest) {
  const { email, password } = await request.json();

  // 1. Find user by email
  const userResult = await db.select().from(users)
    .where(eq(users.email, email.toLowerCase().trim()))
    .limit(1);
  if (userResult.length === 0)
    return NextResponse.json(
      { error: 'Invalid credentials' }, { status: 401 });

  // 2. Verify password with bcrypt
  const match = await bcrypt.compare(
    password, userResult[0].password);
  if (!match)
    return NextResponse.json(
      { error: 'Invalid credentials' }, { status: 401 });

  // 3. Return user without password field
  const { password: _, ...safeUser } = userResult[0];
  return NextResponse.json(safeUser, { status: 200 });
}
```

Listing 4.10: Login endpoint (src/app/api/auth/login/route.ts)

4.6.2 Blockchain Anchoring — POST /api/anchor

```
export async function POST(req: Request) {
  const { hash, cid } = await req.json();

  const doc = await db.query.documents.findFirst({
    where: eq(documents.sha256, hash),
  });
  if (!doc)
    return Response.json(
      { error: 'Document not found' }, { status: 404 });
  if (doc.status === 'ANCHORED')
    return Response.json(
      { ok: true, alreadyAnchored: true, txHash: doc.tx_hash });

  try {
```

```
    const txHash = await anchorOnChain(hash, cid);
    await db.update(documents)
      .set({ status: 'ANCHORED', tx_hash: txHash })
      .where(eq(documents.sha256, hash));
    return Response.json({ ok: true, txHash });
  } catch (err) {
    await db.update(documents)
      .set({ status: 'FAILED' })
      .where(eq(documents.sha256, hash));
    return Response.json(
      { error: 'Blockchain anchoring failed' }, { status: 500 });
  }
}
```

Listing 4.11: Anchoring endpoint (src/app/api/anchor/route.ts)

4.6.3 Document Verification — GET /api/verify

```
export async function GET(req: NextRequest) {
  const hash = req.nextUrl.searchParams.get('hash');

  const doc = await db.query.documents.findFirst({
    where: eq(documents.sha256, hash!),
  });
  if (!doc) return NextResponse.json({ found: false });

  let anchoredOnChain = false, blockchainTimestamp = null;
  let issuer = null, txId = null;
  try {
    const result = await verifyOnChain(hash!);
    if (result?.timestamp > 0) {
      anchoredOnChain     = true;
      blockchainTimestamp = result.timestamp;
      issuer = result.issuer;
      txId   = result.txId;
    }
  } catch { /* not anchored yet */ }

  return NextResponse.json({
    found:    true,
    sha256:   doc.sha256,
    ipfsCid:  doc.ipfsCid,
    status:   doc.status,
    txHash:   txId ?? doc.txHash,
    anchoredOnChain, blockchainTimestamp, issuer,
  });
}
```

Listing 4.12: Verification endpoint (src/app/api/verify/route.ts)

4.6.4 API Routes Summary

Table 4.2 provides a complete reference of all API endpoints.

Table 4.2: Complete API routes reference

Method	Group	Endpoint	Description
POST	Auth	/api/auth/register	Register user with bcrypt hashing
POST	Auth	/api/auth/login	Login, returns user object
GET	Auth	/api/auth/verify-email	Verify email, activate account
GET	Applications	/api/applications	List applications with filters
POST	Applications	/api/applications	Submit new application
GET	Applications	/api/applications/[id]	Get single application + reviews
PATCH	Applications	/api/applications/[id]	Update (supervisor decision)
POST	Applications	/api/applications/[id]/in	Send interview invitation email
POST	Upload	/api/upload	Upload to IPFS + R2, get SHA-256
POST	Blockchain	/api/anchor	Anchor SHA-256 on Fabric ledger
GET	Blockchain	/api/verify	Verify document (DB + Fabric)
POST	Reviews	/api/reviews	Submit supervisor review
GET	Certificates	/api/certificates	List training certificates
POST	Certificates	/api/certificates/validate	Validate certificate eligibility
GET	Notif.	/api/notifications	Get user notifications
PATCH	Notif.	/api/notifications/[id]	Mark notification as read

4.7 User Interface Implementation

The GRANTA platform provides a dedicated dashboard interface for each actor role. All interfaces are implemented using Next.js App Router pages with React 19 components, Tailwind CSS for styling, and Radix UI for accessible component primitives.

4.7.1 Page Routing Structure

```
src/app/
|-- page.tsx                        # Landing page (login/register)
|-- candidate/
|   |-- register/page.tsx           # Application submission form
|   `-- dashboard/page.tsx          # Application status tracking
```

```
|-- supervisor/
|   `-- dashboard/page.tsx          # Application review interface
|-- final-supervisor/
|   `-- dashboard/page.tsx          # Final validation + anchor button
|-- certificate-supervisor/
|   `-- dashboard/page.tsx          # Certificate registration
|-- admin/
|   |-- GRANTA/page.tsx              # Full document management and Live
    blockchain activity and  Caliper benchmark
 monitor
|-- verify/
|   |-- page.tsx                    # Public hash verification form
|   `-- [hash]/page.tsx             # Verification result page
`-- certificate/
    |-- register/page.tsx           # Certificate registration
    `-- verify/page.tsx             # Certificate verification
```

Listing 4.13: App Router page structure

4.7.2 Landing Page and Authentication

The platform entry point is a unified login and registration page. Users are redirected to their role-specific dashboard after successful authentication.

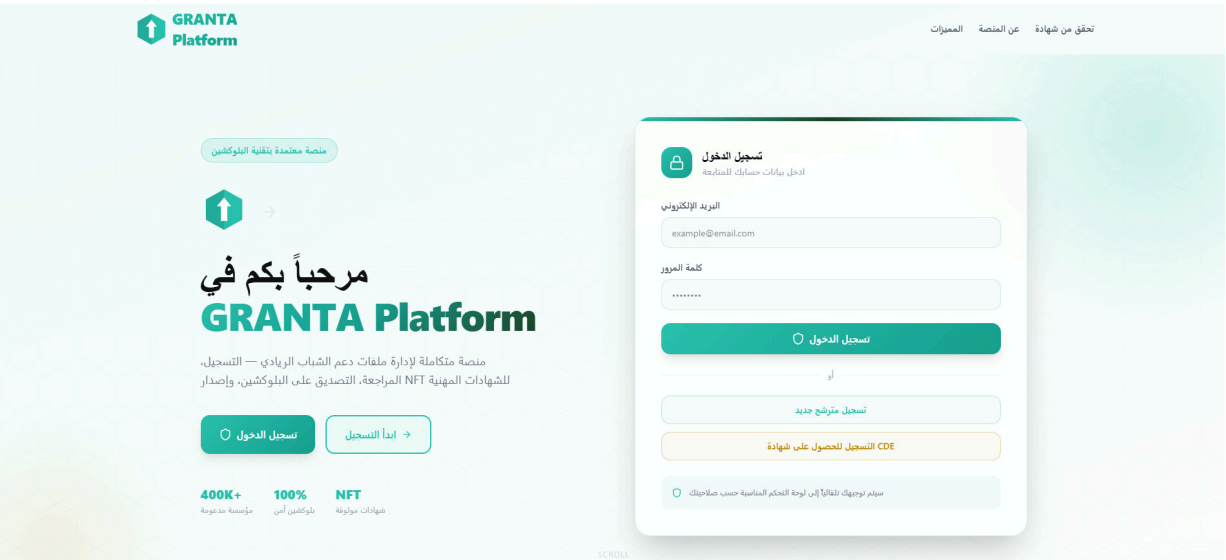


Figure 4.1: GRANTA platform landing page

4.7.3 Candidate Register for CDE training certificate

New candidates complete a multi-step registration process before submitting their professional certificate application. Figures 4.2 to 4.5 show the four steps of the registration flow.

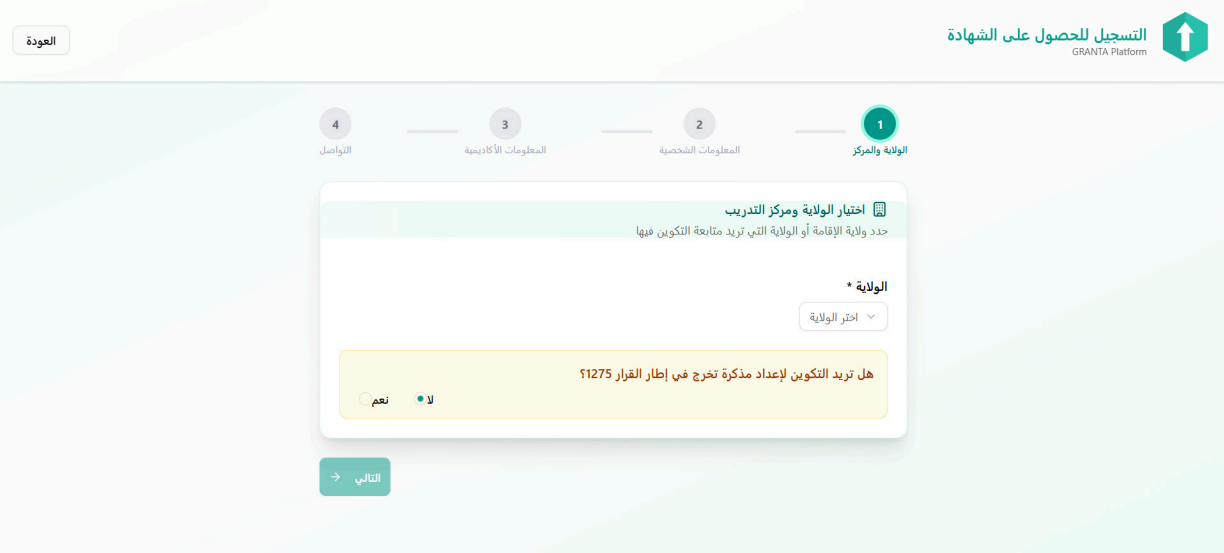


Figure 4.2: Candidate registration — Step 1: personal information. The candidate selects their wilaya (region) and universtiy and Decision 1275 (yes or no). All fields are validated client-side before the form advances to the next step.

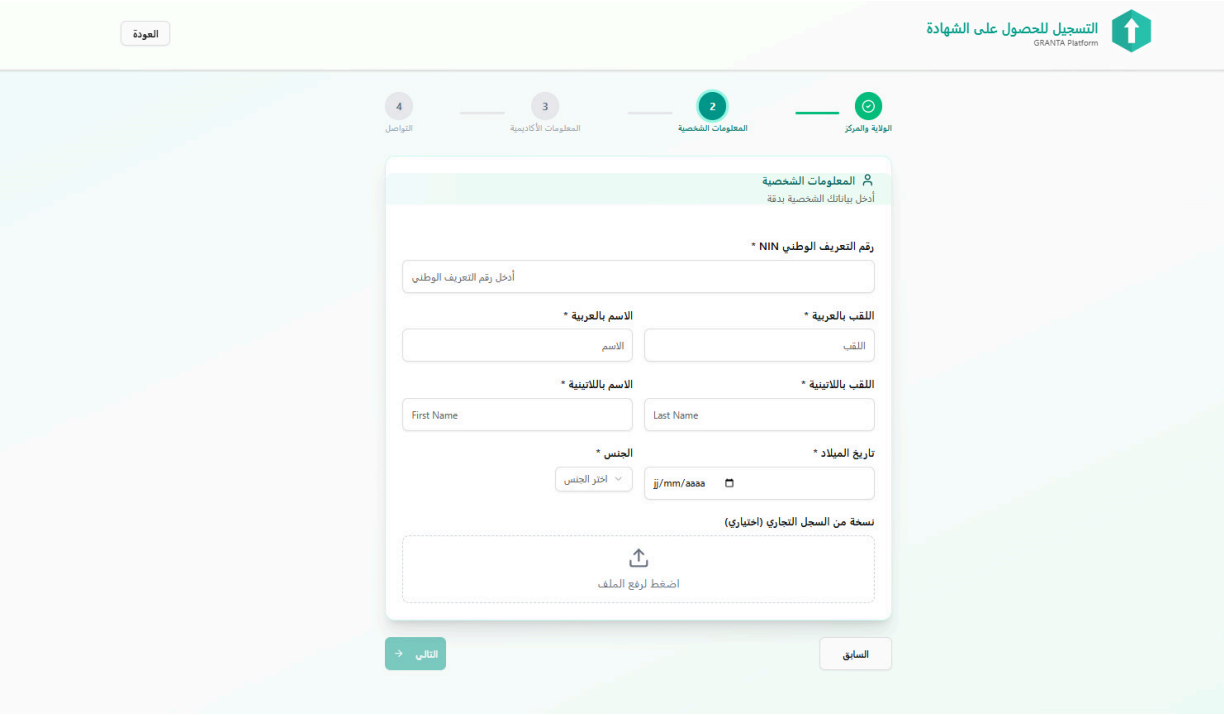


Figure 4.3: Candidate registration — Step 2: certificate and project details. The candidate selects the certificate type (university or institute), enters the certificate number, and writes the project description. This information is stored in the `applications` table upon submission.

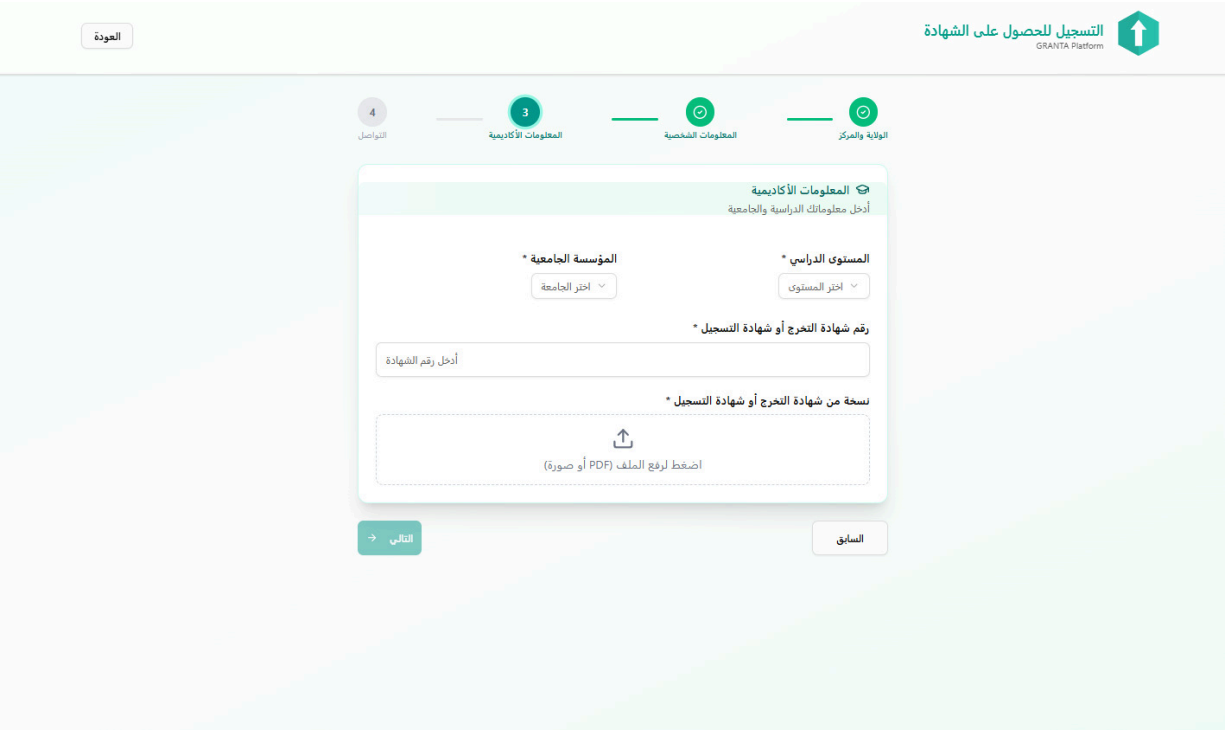


Figure 4.4: Candidate registration — Step 3: project file upload. The candidate uploads their project PDF. The system calls `POST /api/upload`, which sends the file to Pinata (IPFS), generates a SHA-256 fingerprint using the Web Crypto API, and stores both the IPFS CID and the hash in the `documents` table with status `PENDING`.

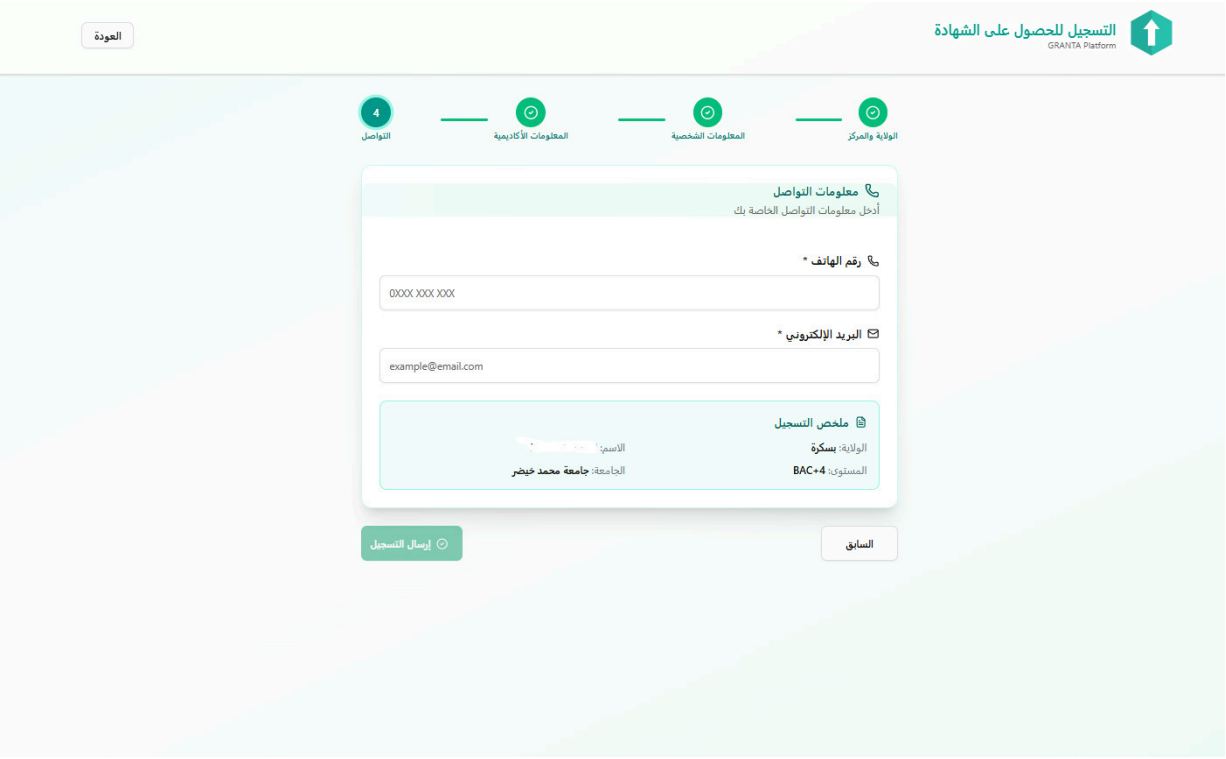


Figure 4.5: Candidate registration — Step 4: review and submit. The candidate reviews all entered information before final submission. Clicking *Submit* calls `POST /api/applications`, creates the application record in the database with status `pending`, and dispatches a notification to the assigned supervisor.

4.7.4 Register as a candidate

step flow: certificate validation → complete form → auto-create account + application

- Certificate number validation.
- Personal information + account creation.
- The candidate must have a valid CDE certificate number before they can register the form cannot be submitted without passing step 1. Email verification is sent automatically but does not block dashboard access.

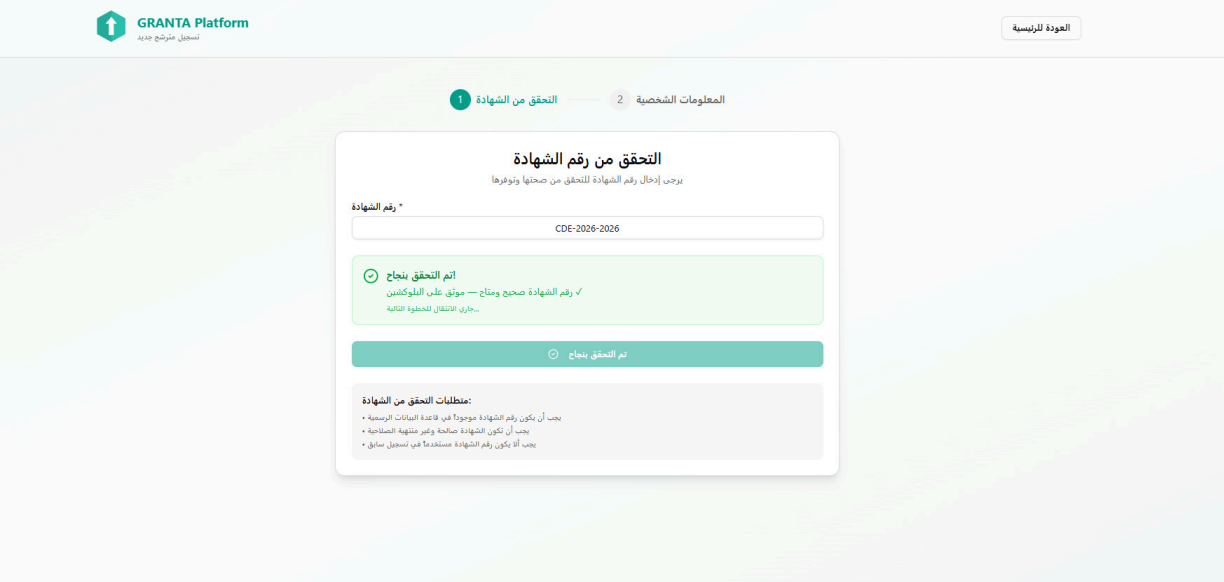


Figure 4.6: Certificate number validation

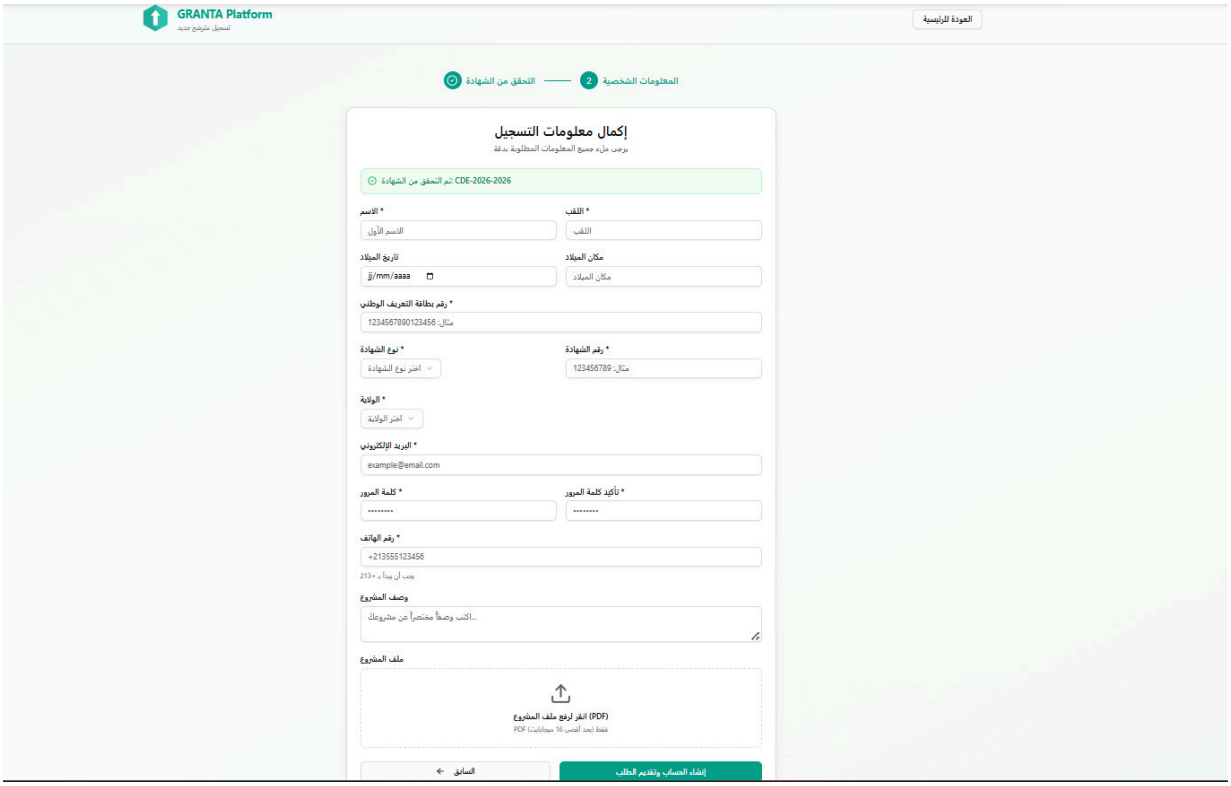


Figure 4.7: Candidate register

4.7.5 Candidate Dashboard

The candidate dashboard allows applicants to view their application status, track review progress. The dashboard fetches data from `/api/applications` filtered by the logged-in candidate's ID.

Key features implemented:

- View personal profile (read-only — cannot edit after registration).
- Track the status of their single submitted application
- See all supervisor decisions on their application (accepted / rejected / reconsider).
- Receive and read real-time notifications (review decisions, interview invitations).

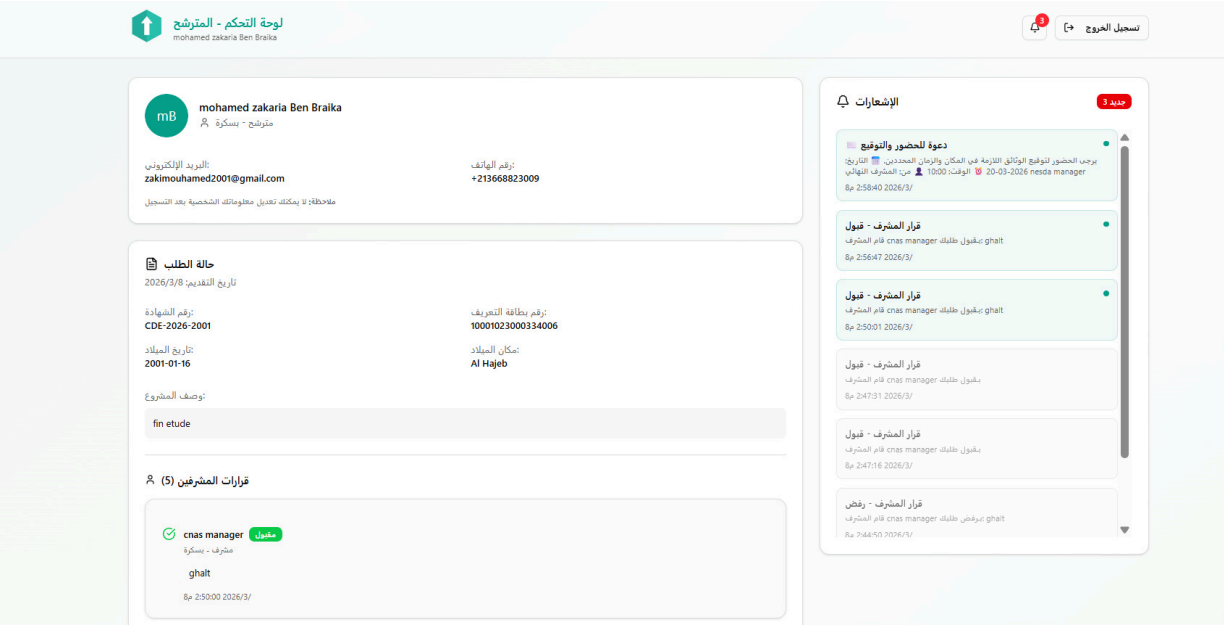


Figure 4.8: Candidate dashboard

4.7.6 Supervisor Dashboard

The supervisor dashboard presents all applications assigned to the logged-in supervisor. It allows the supervisor to read the application details, write a review decision.

- Review grant applications assigned to their wilaya.
- Accept, reject, or request reconsideration on each application.
- Add a mandatory comment when rejecting an application, Reverse a previous decision (reconsider accepted/rejected).

- Search applications by name, certificate number, or national ID

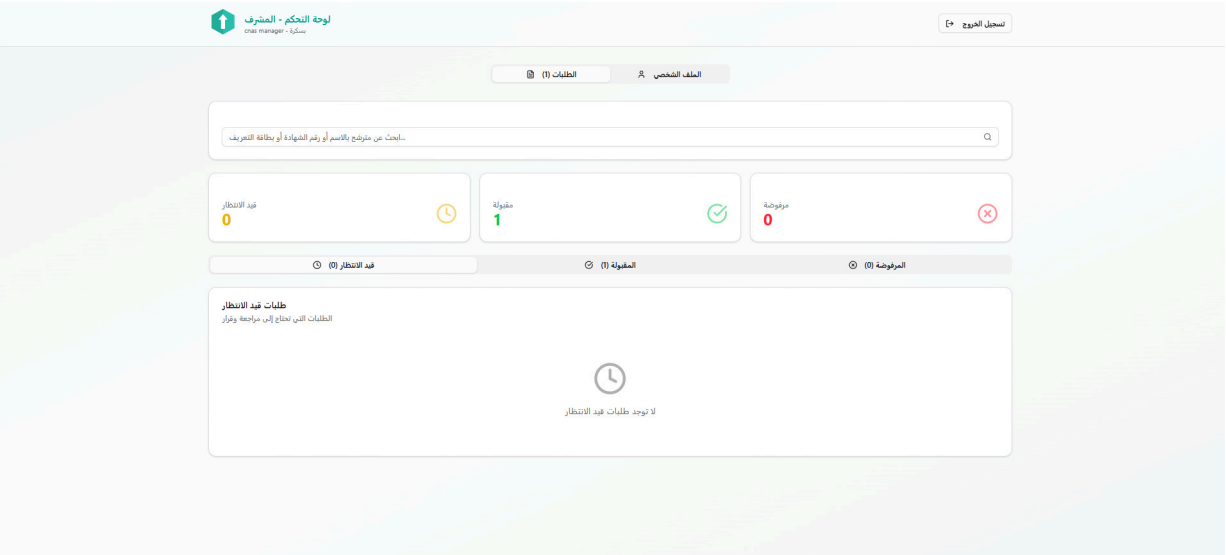


Figure 4.9: Supervisor dashboard — application review interface.

4.7.7 Final Supervisor Dashboard

The final supervisor dashboard is It displays all accepted applications

- Access only fully-accepted applications (approved by both supervisor + all reviews)
- Send an official interview invitation to the candidate (email + notification)
- Separate view: uninvited vs already-invited applicants
- Download or preview the candidate's uploaded project file

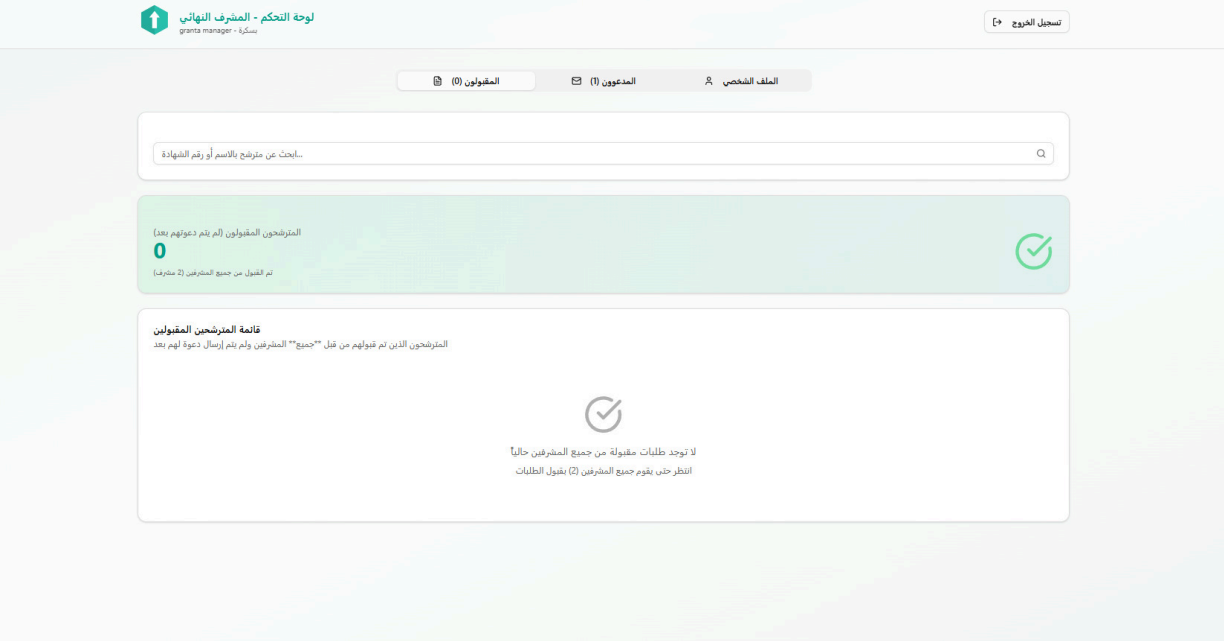


Figure 4.10: Final supervisor dashboard

4.7.8 Certificate Supervisor Interface

The certificate supervisor manages training certificates issued by CDE centres.

- Review candidate registration requests for CDE training programs.
- Approve or reject a registration, with a required review note and training date.
- Add a new CDE training certificate (triggers NFT + Fabric automatically).
- Automatically link the certificate to a pending registration by certificate number.
- Send an email invitation to a candidate to use their NFT token ID and CDE ID
- Monitor certificate stats: total issued, NFT-minted, Fabric-anchored, invitations sent

Figures 4.11 to 4.13 show the certificate registration and validation screens.

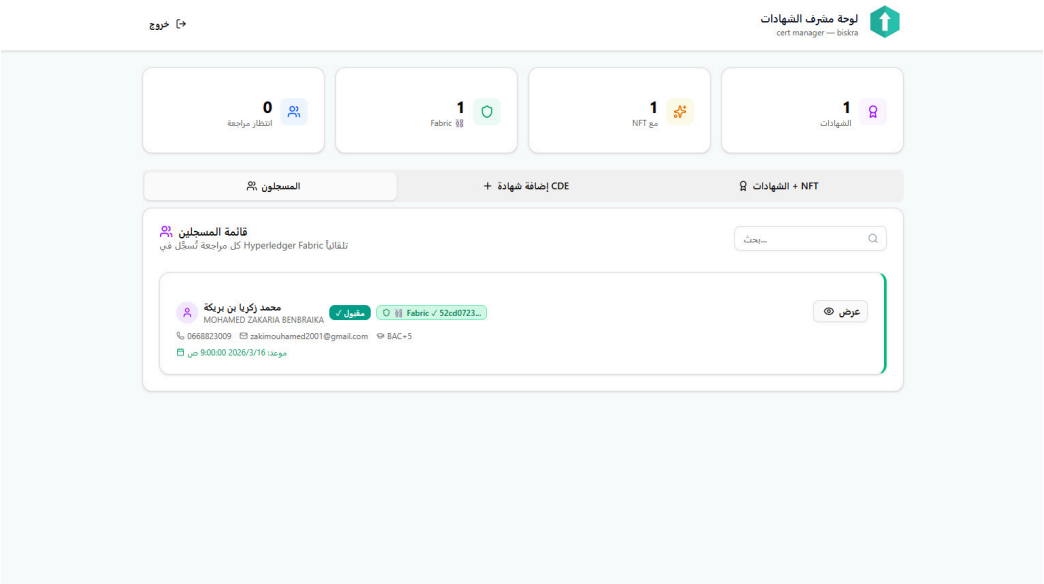


Figure 4.11: Certificate supervisor — certificate list view (Step 1).

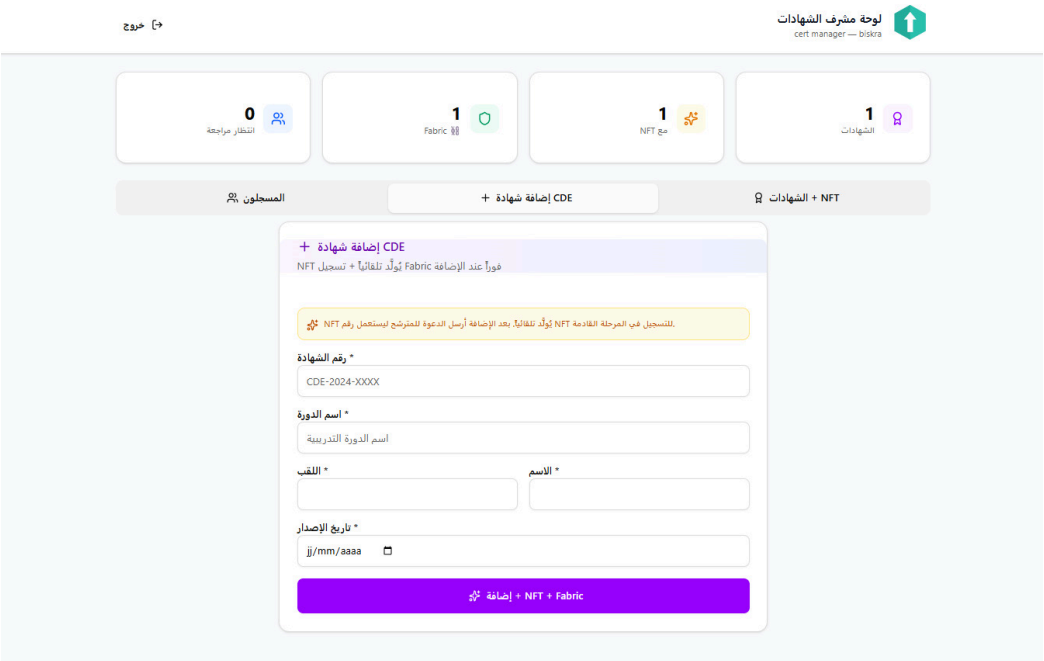


Figure 4.12: Certificate supervisor — certificate detail view (Step 2).

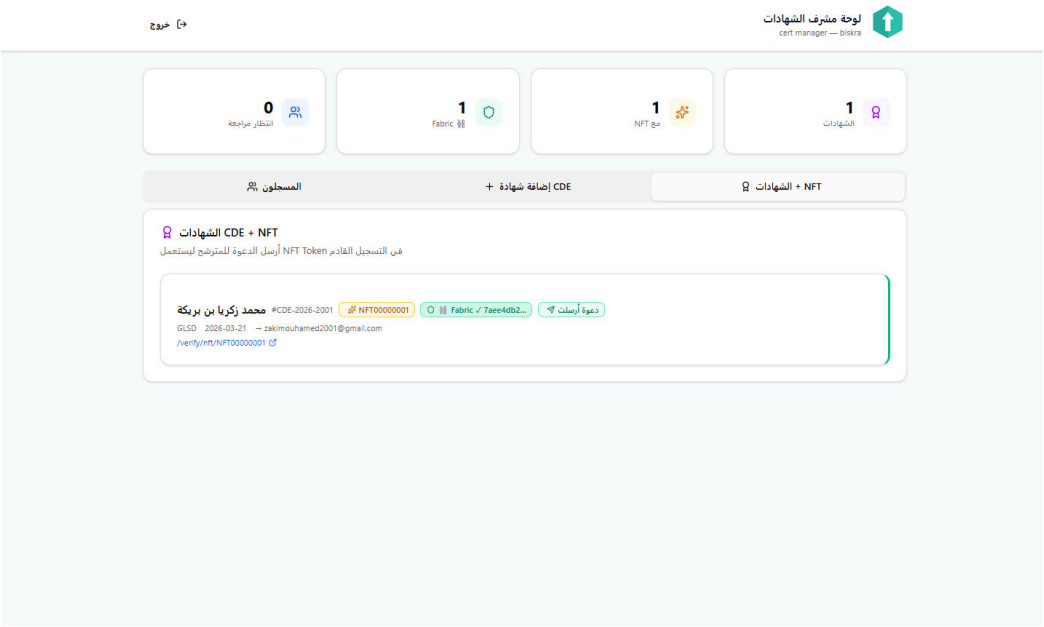


Figure 4.13: Certificate supervisor — validation confirmation (Step 3).

4.7.9 Admin Dashboard

The administrator has unrestricted access to the full platform. The admin dashboard aggregates all operational views: user management, document oversight, live blockchain logs, verification tools, certificate management, and the Caliper benchmark monitor.

- View all uploaded documents across the platform with full metadata
- Filter documents by status (All / Anchored / Pending) and search by filename or hash.
- Manually verify any document by uploading the file and computing its SHA-256.
- Monitor live blockchain events via WebSocket (upload, anchored, verified, error).
- View analytics charts: weekly anchoring trend, daily volumes, status distribution.
- See key platform stats: total documents, anchored count, pending count, today's uploads, weekly uploads
- Copy document SHA-256 hash or IPFS CID to clipboard with one click.
- Access the live Caliper benchmark monitor and the WebSocket log stream .

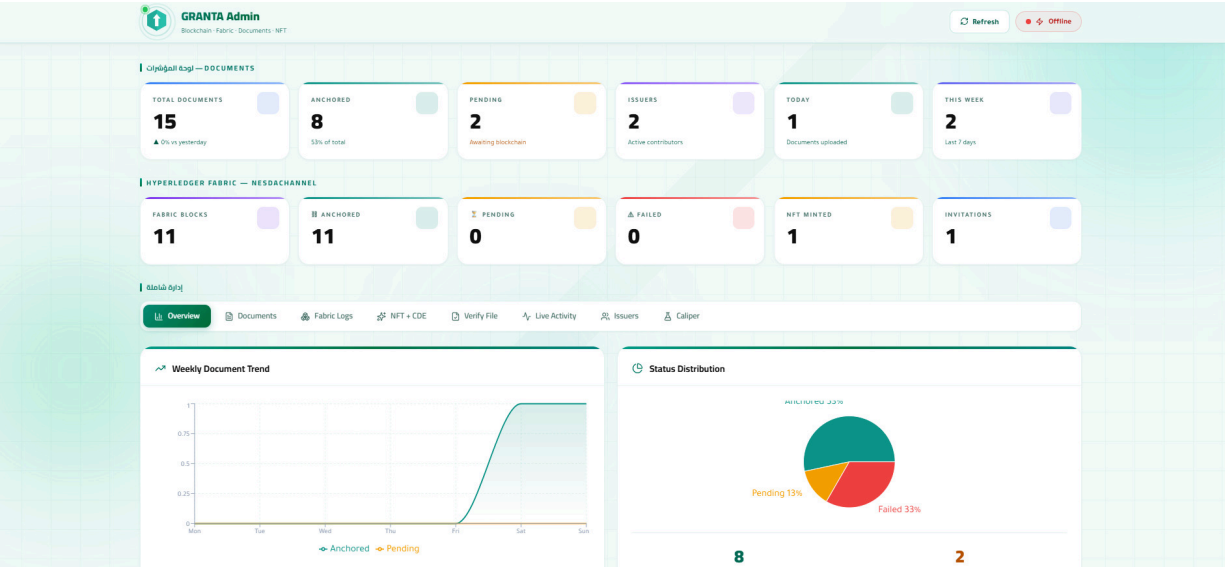


Figure 4.14: Admin dashboard — main overview panel. The main admin panel displays platform-wide statistics: total number of registered users, submitted applications, anchored documents, and active certificates. Navigation links provide direct access to each submodule. The sidebar shows role-based menu items exclusive to the administrator.

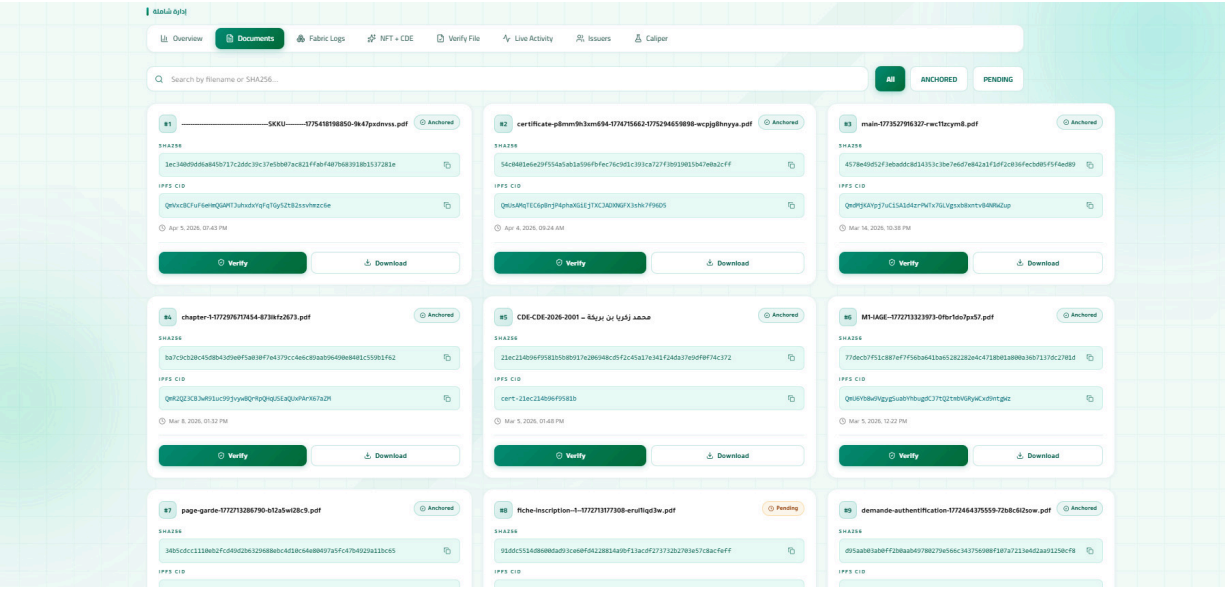


Figure 4.15: Admin dashboard — document management panel (`/admin/GRANTA`). This panel lists all documents stored in the platform with their SHA-256 hash, IPFS CID, Cloudflare R2 backup URL, blockchain status (`PENDING` / `ANCHORED` / `FAILED`), and transaction ID. The admin can filter by status, search by hash, and trigger manual re-anchoring for failed documents.

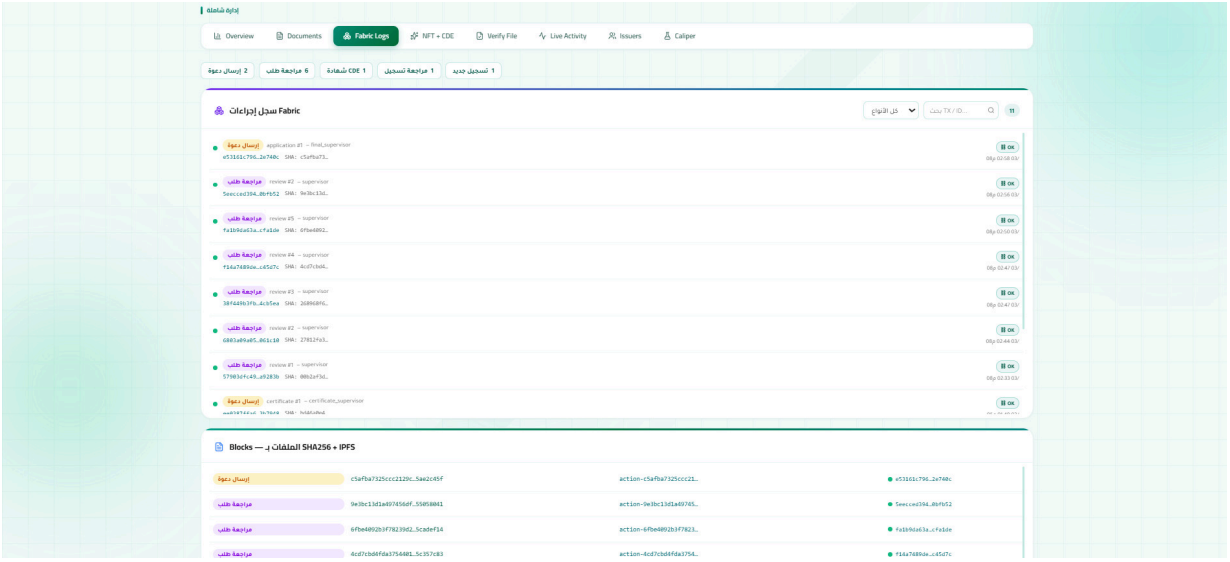


Figure 4.16: Admin dashboard — Hyperledger Fabric live logs panel. This view streams real-time Docker container logs from the Fabric peer and orderer nodes directly into the browser via WebSocket. The admin can observe block commit events, chaincode invocation results, and endorsement policy validations as they happen, enabling immediate diagnosis of any blockchain transaction issue.

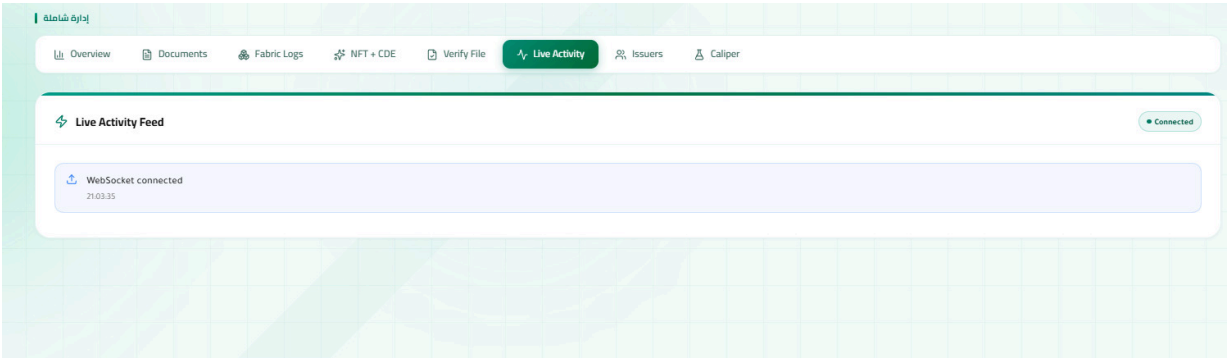


Figure 4.17: Admin dashboard — live blockchain activity monitor (`/admin/live`). The live activity panel displays a real-time event feed of all blockchain operations: document uploads, anchoring transactions, and verification queries. Each event card shows the event type, document filename, SHA-256 hash (truncated), IPFS CID, and timestamp. Events are pushed to the browser via the WebSocket notification module (`websocket-notify.ts`).

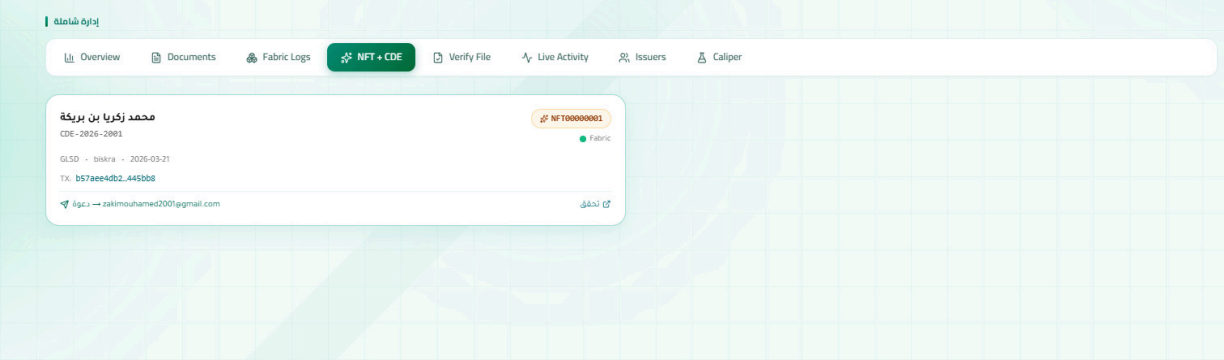


Figure 4.18: Admin dashboard — certificate NFT/CDE management panel. This panel provides the administrator with an overview of all training certificates registered by Certificate Supervisors. It displays the certificate holder, issuing CDE centre, course name, validity flag, and wilaya. The admin can search, filter, and manually override certificate validity status when required.

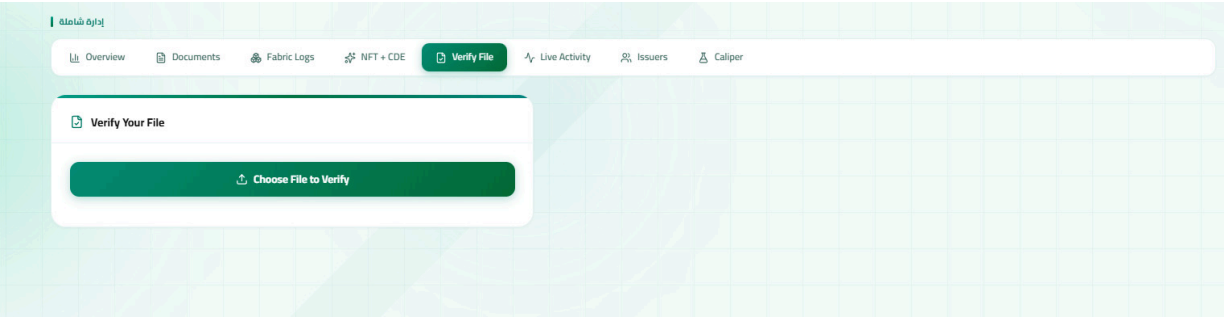


Figure 4.19: Admin dashboard — document verification tool (`/admin/verify`). The admin verification panel allows the administrator to query any document by its SHA-256 hash, combining the database record with the live Hyperledger Fabric ledger state. The result shows whether the document is anchored on-chain, the blockchain timestamp, the issuer X.509 identity, and the IPFS CID — the same data returned by the public `/api/verify` endpoint.

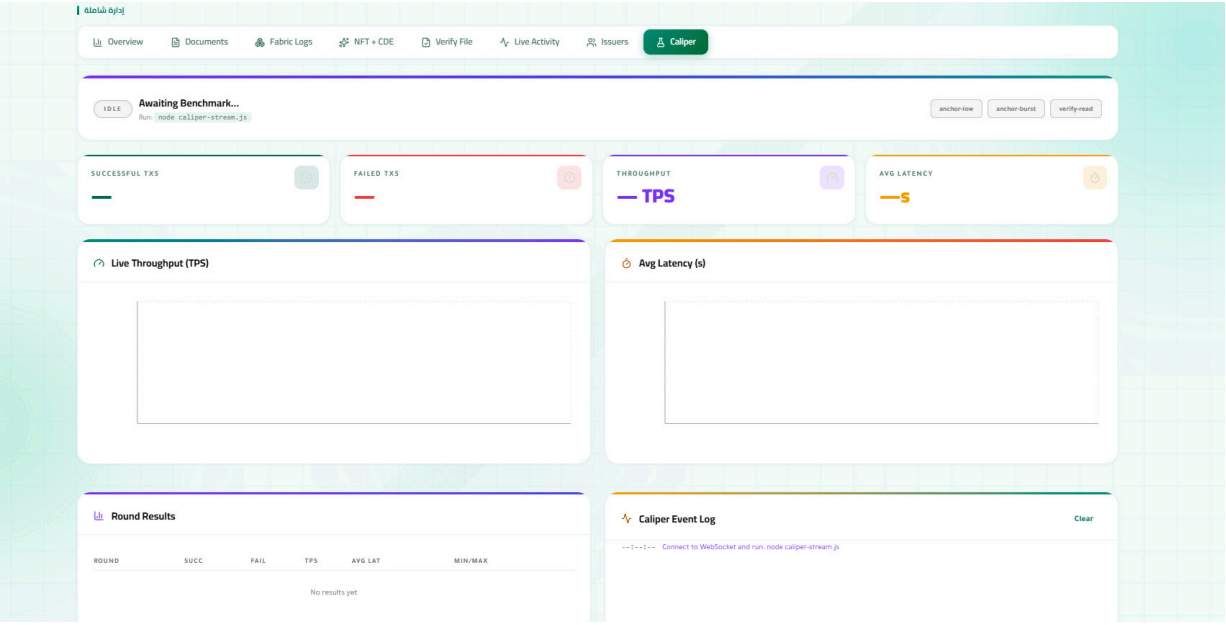


Figure 4.20: Admin dashboard — Hyperledger Caliper benchmark monitor (`/admin/live2`). This panel streams live results from the Caliper benchmark tool as it executes the six configured rounds (R1–R6). For each round it displays the target TPS, the number of successful and failed transactions, the average latency, and the effective throughput. The R2 normal-load result of 45.5 TPS with zero failures is highlighted, confirming production-grade performance.

4.7.10 Public Verification Page

The `/verify` page is publicly accessible without authentication. Any user can enter a SHA-256 hash to check whether a document has been anchored on the Fabric ledger:

```
const handleVerify = async () => {
  const res  = await fetch('/api/verify?hash=' + inputHash);
  const data = await res.json();
  if (!data.found) { setResult({ status: 'NOT_FOUND' }); return; }
  setResult({
    status:         data.anchoredOnChain ? 'VERIFIED' : 'PENDING',
    sha256:         data.sha256,
    ipfsCid:        data.ipfsCid,
    blockchainTime: new Date(data.blockchainTimestamp * 1000),
    issuer:         data.issuer,
    txHash:         data.txHash,
  });
};
```

Listing 4.14: Public document verification flow (simplified)

4.8 Hyperledger Fabric and Caliper Setup

4.8.1 Fabric network configuration

The experimental blockchain environment is based on a two-organisation Hyperledger Fabric deployment. Each organisation contributes endorsing peers, certificate authorities, and client identities. The ordering service relies on Raft. The benchmark driver connects to the network through a dedicated Caliper network descriptor and drives the `registerCertificate` workload using 5 workers.

Table 4.3: Benchmark environment summary

Item	Value
Operating environment	Windows 11 with WSL 2 (Ubuntu-based local environment)
Benchmark driver	Hyperledger Caliper 0.6.0
Blockchain platform	Hyperledger Fabric 2.5.0
Fabric CA	1.5.7
Caliper workers	5 workers
Primary workload	Certificate registration workload (renamed from legacy registerCDE benchmark)
Measured indicators	Success count, failure count, send rate, throughput, max latency, min latency, average latency

4.8.2 Orderer parameters used in the evaluation write-up

The benchmark analysis is discussed using the orderer envelope retained for the evaluation campaign. These parameters determine how fast endorsed transactions are grouped into blocks and forwarded to the peers for commit.

Table 4.4: Orderer parameters considered in the benchmark discussion

Parameter	Value	Role in performance
`BatchTimeout`	500 ms	Maximum waiting time before the orderer cuts a block.
`MaxMessageCount`	500	Maximum number of transactions grouped in one block.
`AbsoluteMaxBytes`	99 MB	Hard upper bound on the total block payload size.
`PreferredMaxBytes`	2 MB	Soft target for block size before early cut.
Ordering protocol	Raft	Crash-fault-tolerant ordering and deterministic block sequencing.

4.9 Performance Evaluation with Hyperledger Caliper

4.9.1 Benchmark methodology

Two benchmark families were retained. In the first family, the number of submitted transactions is fixed at 1000 and the target send rate increases from 100 TPS to 1000 TPS. This reveals the sustainable throughput of the certificate registration path under rate pressure. In the second family, the transaction volume grows from 1000 to 10000 operations. This evaluates scalability and queueing behaviour when the network receives progressively larger batches.

In the original Caliper reports the workload labels use the legacy name `registercde`. In this chapter, the labels are interpreted as the certificate registration transaction and are renamed conceptually to `registerCertificate` for consistency with the manuscript.

4.9.2 Rate-controlled benchmark results

Table 4.5: Rate-controlled results for `registerCertificate` (1000 transactions per round)

Round	Succ.	Fail	Send (TPS)	Max L. (s)	Min L. (s)	Avg L. (s)	Thr. (TPS)
registerCertificate-100tps	1000	0	100.5	2.16	0.14	0.76	97.1
registerCertificate-200tps	1000	0	184.9	4.04	0.70	2.41	137.4
registerCertificate-300tps	1000	0	254.1	2.81	0.56	1.14	185.7
registerCertificate-400tps	1000	0	199.4	6.85	2.61	5.55	130.4
registerCertificate-500tps	1000	0	250.4	6.25	3.03	5.14	138.4
registerCertificate-600tps	1000	0	253.4	5.29	0.71	2.65	187.4
registerCertificate-700tps	1000	0	353.1	5.89	3.75	5.05	151.1
registerCertificate-800tps	1000	0	528.8	5.99	4.64	5.30	149.8
registerCertificate-900tps	1000	0	485.7	5.91	3.94	5.14	149.0
registerCertificate-1000tps	1000	0	391.5	5.55	3.09	4.85	160.3

The rate-controlled campaign shows that the workload remains stable while the client-side send rate increases aggressively. Throughput rises quickly at the beginning and then plateaus around the 150–190 TPS range, which is consistent with a network that is saturating on endorsement and ordering rather than failing. The average latency stays bounded and remains below six seconds in all high-load rounds.

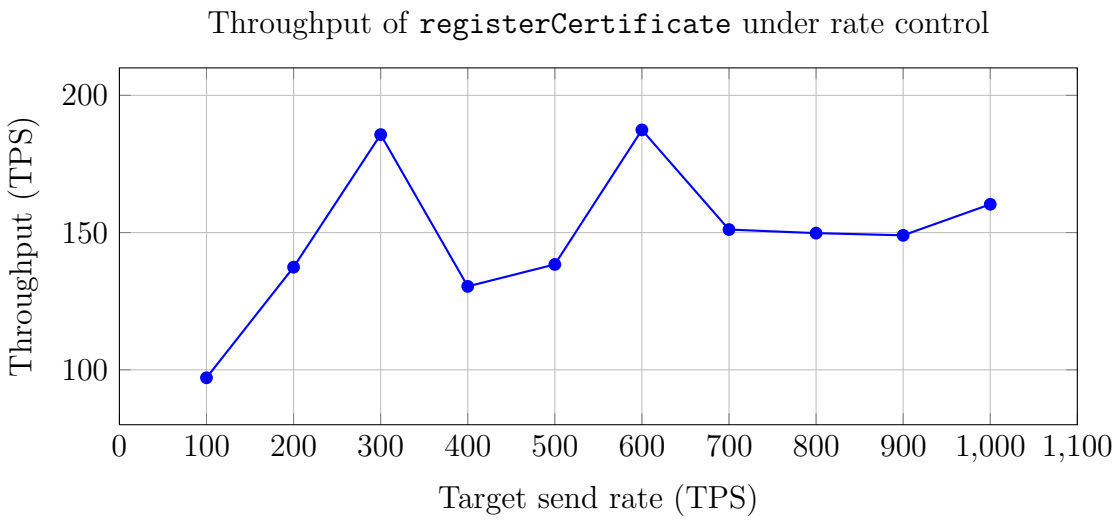


Figure 4.21: Rate-controlled throughput for `registerCertificate`.

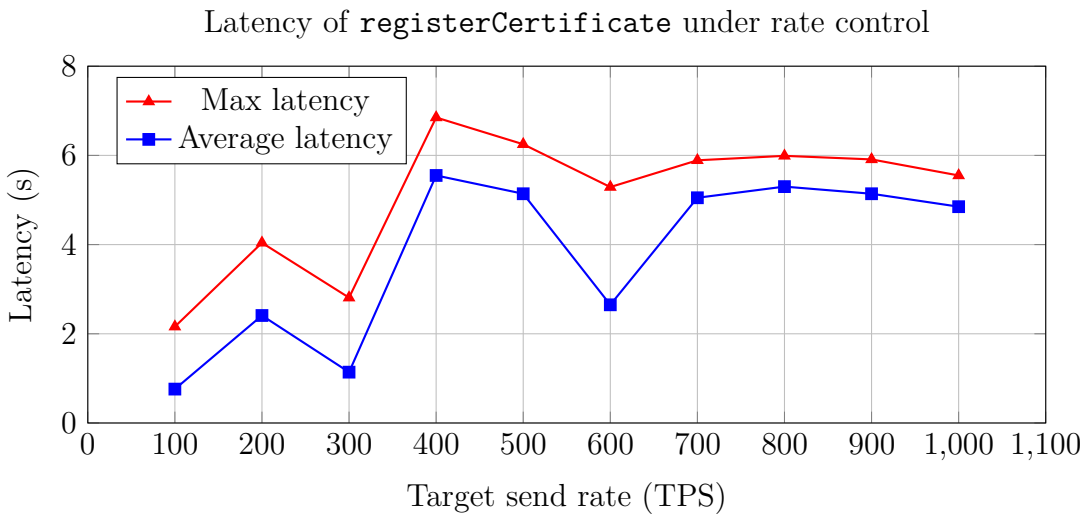


Figure 4.22: Rate-controlled latency for `registerCertificate`.

4.9.3 Volume-controlled benchmark results

Table 4.6: Volume-controlled results for `registerCertificate`

Round	Succ.	Fail	Send (TPS)	Max L. (s)	Min L. (s)	Avg L. (s)	Thr. (TPS)
registerCertificate-1000tx	1000	0	495.0	5.56	3.85	4.62	157.8
registerCertificate-2000tx	2000	0	496.9	11.54	6.57	8.61	169.2
registerCertificate-3000tx	3000	0	279.9	14.59	5.53	11.27	184.2
registerCertificate-4000tx	4000	0	424.5	17.25	2.55	13.47	199.0
registerCertificate-5000tx	5000	0	413.6	19.43	3.29	14.94	223.4
registerCertificate-6000tx	6000	0	459.8	22.35	2.75	17.47	228.4
registerCertificate-7000tx	7000	0	487.2	26.33	2.67	20.16	227.2
registerCertificate-8000tx	8000	0	466.5	32.98	4.90	25.85	210.1
registerCertificate-9000tx	9000	0	500.1	44.18	2.81	36.48	176.0
registerCertificate-10000tx	10000	0	430.7	40.83	6.83	32.12	209.5

The volume-controlled campaign confirms that the platform remains operational even with very large transaction sets. Throughput grows progressively and peaks around 228.4 TPS at 6000 transactions. As expected, latency increases with queue accumulation, but the system still preserves full completion for all large rounds, which is a good indicator of robustness for batch-oriented certificate registration.

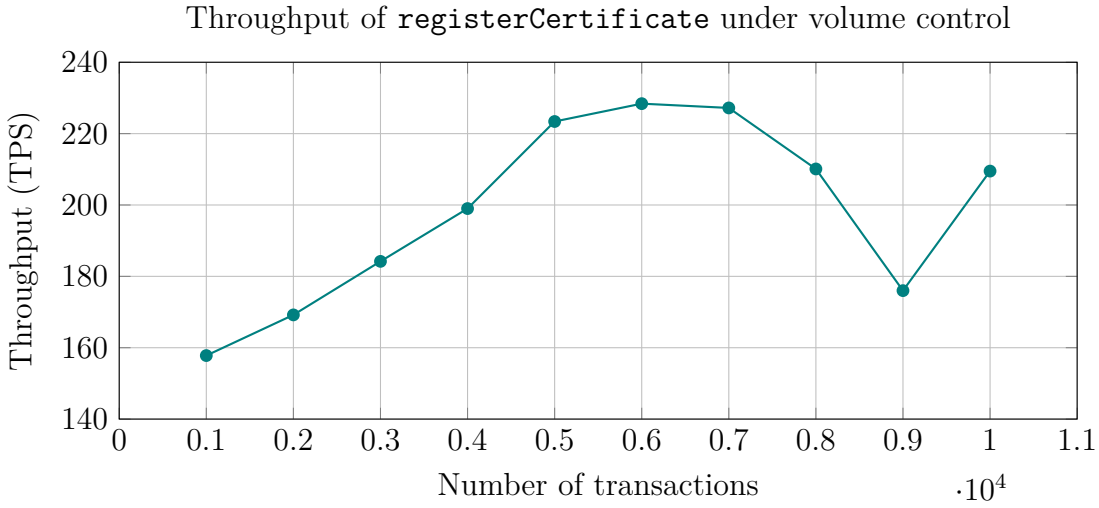


Figure 4.23: Volume-controlled throughput for `registerCertificate`.

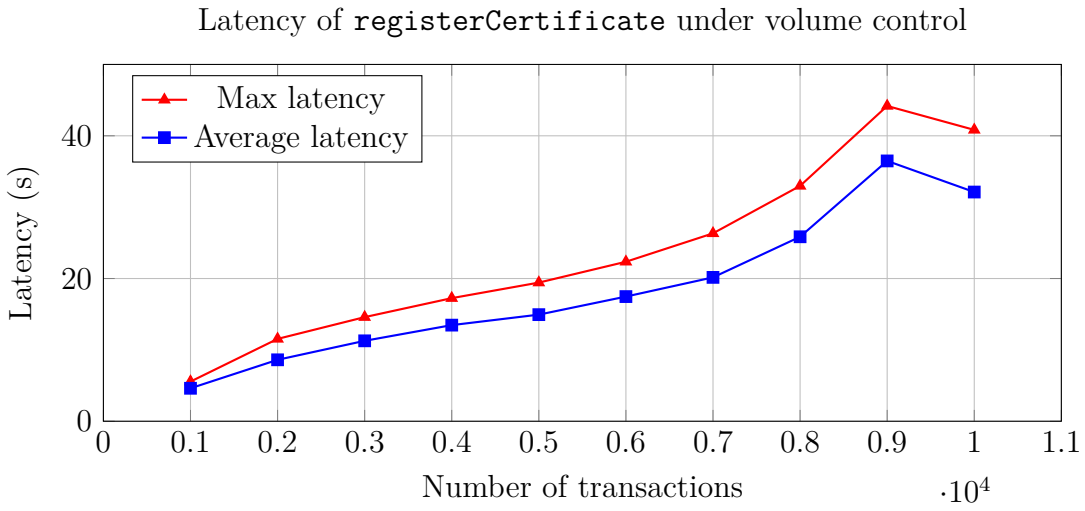


Figure 4.24: Volume-controlled latency for `registerCertificate`.

4.9.4 Unified comparison chart

The final comparison chart combines the two benchmark families on a common experiment index. This makes it easier to compare how the platform reacts to higher target rates on the one hand and larger transaction volumes on the other hand.

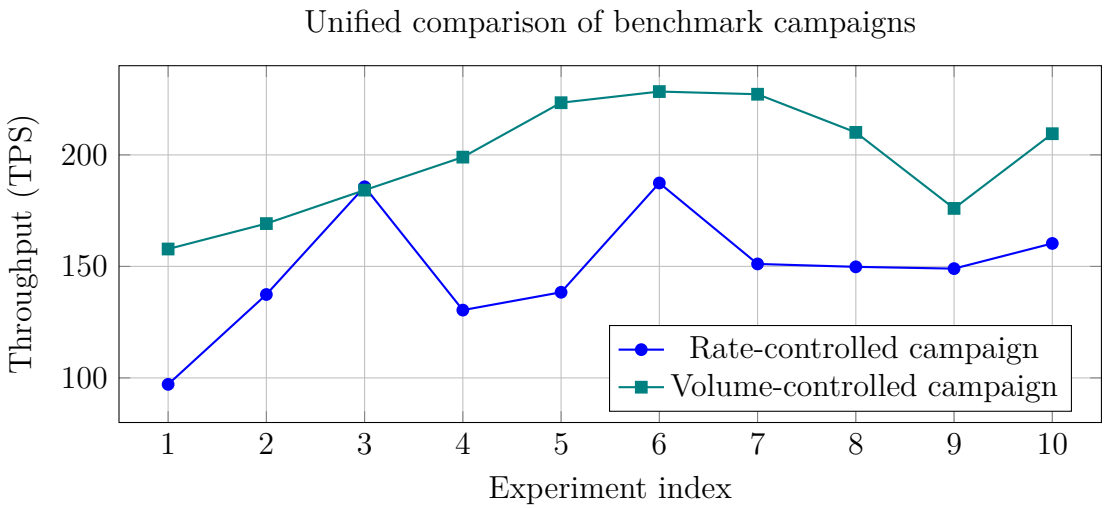


Figure 4.25: Unified comparison of throughput across all benchmark rounds.

4.9.5 Discussion of benchmark results

The benchmark results support three main observations. First, the certificate registration transaction is stable: the failure count is effectively zero across almost all rounds. Second,

the system reaches a practical throughput plateau once endorsement and ordering become the dominant constraints, which is normal in a permissioned Fabric network with multiple peers and a conservative block cutting policy. Third, the increase in latency under large transaction volumes does not translate into transaction loss, which means the platform remains suitable for institutional workloads where consistency and auditability are more important than sub-second confirmation time.

4.10 Conclusion

This chapter showed that the GRANTA platform is not only functionally complete but also technically coherent and experimentally validated. The implementation integrates a modern web stack, a structured relational data layer, decentralised file storage, and a permissioned blockchain network without fragmenting the business workflow. Certificate-related actions remain traceable from the user interface down to the ledger commit stage.

The Hyperledger Caliper evaluation strengthens this conclusion. The benchmark results demonstrate that the certificate registration pipeline can sustain non-trivial loads while preserving reliability and auditability. Throughput remains in a practically usable range under both rate-controlled and volume-controlled conditions, while latency evolves predictably as workload increases. These observations confirm that the selected implementation choices are appropriate for a real certificate registration and verification platform.

Overall, the chapter establishes that the GRANTA platform is ready to be presented not merely as a conceptual prototype, but as an implemented and measurable system whose blockchain layer has been evaluated under reproducible experimental conditions.

General Conclusion

The work presented in this thesis was motivated by a concrete and persistent challenge: the modernisation of digital certification within the Algerian youth entrepreneurship support ecosystem. As the country pursues economic diversification beyond hydrocarbons, public agencies such as NESDA, CNAC, ANGEM, and the Ministry of Knowledge Economy, Startups, and Micro-Enterprises have multiplied financial, legal, and institutional instruments to encourage business creation by young people. However, despite this remarkable institutional effort, the entrepreneurial ecosystem still suffers from bureaucratic complexity, fragmented digital procedures, vulnerable document handling, and the absence of a tamper-evident infrastructure connecting applicants, supervisors, and verifying institutions. The central problem addressed in this work was therefore the design, implementation, and quantitative evaluation of a permissioned blockchain platform capable of providing secure, transparent, and auditable digital certificate management within this specific institutional context.

Summary of the Work

To answer this problem, the thesis was organised around four complementary chapters whose contributions can now be synthesised.

Chapter 1 laid the contextual and socio-economic foundation of the study. It examined the structure of youth entrepreneurship in Algeria, the extent of youth unemployment, the public policies and dedicated agencies that support project holders, the legal reforms introduced since 2019 (notably Executive Decree 20-254 and the recent self-employed entrepreneur status), and the role of insurance and digitalisation in stabilising emerging firms. This chapter showed that, while a comprehensive support architecture exists on paper, persistent barriers limited access to finance, opaque administrative procedures, weak interoperability, and uneven digital transformation still restrict the practical impact of these mechanisms. The chapter therefore justified the need for a trusted digital infrastructure capable of bridging the gap between legal design and operational reality.

Chapter 2 provided the technical and theoretical foundation required to address this gap. It analysed the fundamentals of blockchain technology distributed ledger structure, cryptographic hash functions, digital signatures, and core principles of decentralisation, immutability, and auditability followed by a detailed comparison of the main consensus

mechanisms (PoW, PoS, PBFT, and Raft). Particular attention was given to Raft as used in Hyperledger Fabric, since it offers high throughput, low latency, and crash-fault tolerance suitable for permissioned institutional networks. The chapter also examined smart contracts, the architecture of Hyperledger Fabric, hybrid on-chain/off-chain storage models with IPFS, and the state of the art in blockchain-based certification. This survey confirmed that a permissioned consortium blockchain coupled with content-addressable off-chain storage is the most appropriate architecture for the institutional certification problem identified in Chapter 1.

Chapter 3 translated this theoretical foundation into a complete system design. Through complementary UML and architectural views a global use case diagram, a class diagram, a cross-workflow integration view, six independent per-workflow sequence diagrams (WF01–WF06), and a full Hyperledger Fabric deployment diagram the GRANTA platform was specified as a coherent, reproducible artifact. Six actor categories (Candidate, Supervisor, Final Supervisor, Certificate Supervisor, Administrator, and Public Verifier) were identified, the persistent data model (User, Application, Review, Document, Certificate, CertificateRegistration, Notification, and FabricLog) was formalised, and the chaincode transaction surface (`submitApplication`, `issueCertificate`, `verifyCertificate`, `verifyCertificateBySHA256`, `mintCertificateNFT`, `linkNFT`, `getCertificateHistory`, etc.) was explained in detail. The seven-view decomposition adopted in this chapter proved to be a powerful structuring principle, treating each workflow as an autonomous, independently testable unit while preserving the cross-cutting integrity guarantees of the system.

Chapter 4 reported the concrete implementation and the experimental evaluation of the platform. The technological stack combined Hyperledger Fabric 2.5, Next.js 15, TypeScript, SQLite, and IPFS. The chapter detailed the project structure, the database schema, the Fabric network and chaincode, the IPFS integration, the Next.js API routes, and the role-based user interfaces exposed to candidates, supervisors, certificate supervisors, administrators, and public verifiers. A Hyperledger Caliper benchmark campaign with five workers, a 500 ms BatchTimeout, and a 500-transaction MaxMessageCount was then executed against the certificate registration transaction (`registerCertificate`). Two complementary families of experiments were conducted: a rate-controlled campaign (1000 transactions per round, send rate from 100 to 1000 TPS) and a volume-controlled campaign (transaction volume from 1000 to 10000). The results demonstrated full operational stability with zero failures across all rounds, a sustainable throughput plateau in the 150–190 TPS range under rate pressure, a peak throughput of about 228 TPS at 6000 transactions under volume pressure, and a predictable, queue-driven evolution of latency. These quantitative results show that the platform is not a conceptual prototype but a measurable, deployable system whose performance profile is compatible with realistic institutional workloads.

Main Contributions

The contributions of this thesis lie at the intersection of applied blockchain engineering and the study of institutional support for youth entrepreneurship in Algeria. They can be

summarised as follows.

First, the work provides an **up-to-date contextual synthesis** of youth entrepreneurship in Algeria, articulating socio-economic indicators, support agencies, legal reforms, and digital transformation challenges into a coherent analytical framework that motivates the need for trusted digital infrastructures.

Second, the thesis offers a **rigorous architectural specification** of a permissioned certificate management platform. The combination of UML modelling, workflow-based decomposition (WF01–WF06), and a fully detailed Hyperledger Fabric topology produces an architecture that is reproducible, testable, and extensible.

Third, the thesis delivers **GRANTA**, a fully functional implementation that integrates a modern web stack, a structured relational layer, decentralised content-addressable storage (IPFS), and a permissioned blockchain network. Certificate-related actions are traceable end-to-end, from user submission through SHA-256 anchoring, IPFS storage, chaincode endorsement, Raft ordering, and ledger commit, up to public read-only verification and optional NFT minting.

Fourth, the work contributes **quantitative experimental evidence** of the platform's performance through a structured Caliper benchmark campaign. The zero-failure profile, the bounded latency, and the consistent throughput plateau provide an empirical argument that permissioned blockchain technology is mature enough to support institutional certification workloads in the Algerian context.

Future Work and Perspectives

Several promising research and engineering directions emerge naturally from this work.

A first direction is the **extension of the consortium** to additional Algerian institutions, including NESDA, CNAC, ANGEM, universities, and major employers. This would transform GRANTA from a two-organisation prototype into a true national certification fabric, and would require dedicated work on multi-organisation membership service providers, channel governance, and policy evolution.

A second direction concerns the **evolution of the consensus layer**. While Raft is sufficient for the current crash-fault-tolerant deployment, future iterations could explore Byzantine-tolerant variants (hhRaft, VSSB-Raft, PB-Raft, X-RAFT, and Raft/PBFT hybrids) to strengthen the trust model in scenarios where participating organisations cannot be fully trusted.

A third direction is the **enrichment of the certificate model** with selective disclosure mechanisms based on zero-knowledge proofs, decentralised identifiers (DIDs), and verifiable credentials. This would enable candidates to prove specific attributes (e.g., diploma possession, training completion) without revealing the full content of their certificates, addressing both privacy and regulatory concerns.

A fourth direction involves **interoperability** with existing Algerian e-government platforms, the national startup labelling portal, and university information systems, so that GRANTA acts as a transparent trust layer rather than an isolated silo. Standardised APIs,

OpenAPI contracts, and event-driven synchronisation would be central to this direction.

A fifth direction is a **longitudinal field study**, in which GRANTA is deployed within at least one entrepreneurship support agency or university incubator over an extended period. Such a study would measure usability, verification time, fraud reduction, and trust perception among real stakeholders, complementing the synthetic benchmark results presented in this thesis.

A final direction concerns **advanced analytics and machine learning** on the on-chain audit trail and off-chain operational data, to detect anomalies in application processing, predict bottlenecks, and provide data-driven insights to public decision-makers.

Closing Remarks

By bringing together a contextual study of Algerian youth entrepreneurship, a rigorous theoretical foundation in blockchain technology, a complete system design, a working implementation, and a reproducible experimental evaluation, this thesis demonstrates that decentralised trust infrastructure is no longer a purely speculative tool but a practical building block for the modernisation of public entrepreneurship support. The GRANTA platform shows, in concrete terms, how Hyperledger Fabric, IPFS, and a modern web stack can be combined into a coherent, auditable, and measurable certification system aligned with the institutional realities of Algeria. We hope that the architectural decisions, the workflow decomposition, and the experimental methodology presented in this work will provide a useful basis for further research and for real deployments contributing to a more transparent, more efficient, and more trustworthy entrepreneurial ecosystem.

Bibliography

[1] In: (). URL: https://www.oecd.org/en/publications/the-missing-entrepreneurs-2019_3ed84801-en/full-report/reader-s-guide_81440ecf.html?.

[2] In: (). URL: https://link.springer.com/article/10.1007/s11301-020-00193-7.

[3] K. Abdellaoui. "Does the Legal Framework Governing Emerging Institutions in Algeria Differ from That of Traditional Projects?" In: *Law and World* (2024). DOI: 10.36475/10.2.2. URL: https://www.trade.gov/country-commercial-guides/algeria-digital-economy.

[4] K. Abdellaoui. "Does the Legal Framework Governing Emerging Institutions in Algeria Differ from That of Traditional Projects?" In: *Law and World* (2024). DOI: 10.36475/10.2.2. URL: :https://www.nesda.dz/.

[5] K. Abdellaoui. "Does the Legal Framework Governing Emerging Institutions in Algeria Differ from That of Traditional Projects?" In: *Law and World* (2024). DOI: 10.36475/10.2.2. URL: https://www.cnac.dz/Ar/accueil.

[6] K. Abdellaoui. "Does the Legal Framework Governing Emerging Institutions in Algeria Differ from That of Traditional Projects?" In: *Law and World* (2024). DOI: 10.36475/10.2.2. URL: https://www.angem.dz/fr.

[7] K. Abdellaoui. "Does the Legal Framework Governing Emerging Institutions in Algeria Differ from That of Traditional Projects?" In: *Law and World* (2024). DOI: 10.36475/10.2.2. URL: https://doi.org/10.36475/10.2.2.

[8] Ali Ibrahim Ahmed et al. "Blockchain and Smart Contracts: Enabling Trustworthy and Decentralized Digital Transactions." In: *36th Conference of Open Innovations Association (FRUCT)*. 2024, pp. 262–271. DOI: 10.23919/fruct64283.2024.10749910. URL: https://doi.org/10.23919/fruct64283.2024.10749910.

[9] Pr Chaib Baghdad. "The Question of Entrepreneurship in Algeria Between the Will of the State and Realities on the Ground." In: *EPH - International Journal of Business & Management Science* (2019). DOI: 10.53555/eijbms.v5i1.78. URL: World%20Bank,%20World%20Development%20Indicators%20/%20Algeria%20Economic%20Data.

[10] Mokrane Nait Bahloul and Khouloud Abderrahmane. "Student Start-Up." In: *International Journal of Public Administration, Management and Economic Development* (2024). DOI: 10.60026/ijpamed.v9i2.191. URL: World%20Bank%20(2023),%20Algeria%20Economic%20Update.

[11] Farid Belgoum and Nawel Benessalah. "Start-Up and Patent Degrees Initiative in Algeria: Supporting Business Innovation and Creation Among University Students." In: *Entrepreneurship* (2023). DOI: 10.37708/ep.swu.v11i1.2. URL: https://doi.org/10.37708/ep.swu.v11i1.2.

[12] Narimane Benabderrahmane. "The Impact of Digital Challenges On the Development of Startups In the Tourism Sector In Algeria." In: *SocioEconomic Challenges* (2024). DOI: 10.61093/sec.8(4).177-190.2024. URL: https://doi.org/10.61093/sec.8(4).177-190.2024.

[13] Juan Benet. "IPFS - Content Addressed, Versioned, P2P File System." In: *arXiv preprint arXiv:1407.3561* (2014). DOI: 10.48550/arxiv.1407.3561. URL: https://doi.org/10.48550/arxiv.1407.3561.

[14] Muhammad Nasir Mumtaz Bhutta et al. "A Survey on Blockchain Technology: Evolution, Architecture and Security." In: *IEEE Access* 9 (2021), pp. 61048–61073. DOI: 10.1109/access.2021.3072849. URL: https://doi.org/10.1109/access.2021.3072849.

[15] Hamid Chaouche. "The Reality of Youth Entrepreneurship under the National Agency for Supporting Youth Employment: Evaluation of the Agency's Activity in the Field of Establishing Micro-enterprises in the State of Tizi Ouzou, Algeria, from 1996 to 2018." In: *Al-Anbar University Journal For Humanities* (2020). DOI: 10.37653/juah.2020.171088. URL: https://ijame.com/index.php/IJAME/article/view.

[16] Shi Dong et al. "Blockchain Technology and Application: An Overview." In: *PeerJ Computer Science* 9 (2023). DOI: 10.7717/peerj-cs.1705. URL: https://doi.org/10.7717/peerj-cs.1705.

[17] Weihang Feng. "A Comprehensive Survey on Blockchain Technology: Consensus Algorithms, Data Storage Mechanisms, and Architectures." In: *Applied and Computational Engineering* (May 2025). DOI: 10.54254/2755-2721/2025.CH23272. URL: https://doi.org/10.54254/2755-2721/2025.ch23272.

[18] Younes Ferdj. "Entrepreneurship and territorial economic development in Algeria." In: *Management and Entrepreneurship: Trends of Development* (2024). DOI: 10.26661/2522-1566/2024-1/27-01. URL: https://doi.org/10.26661/2522-1566/2024-1/27-01.

[19] Wei Fu, Xuefeng Wei, and Shihua Tong. "An Improved Blockchain Consensus Algorithm Based on Raft." In: *Arabian Journal for Science and Engineering* 46 (2021), pp. 8137–8149. DOI: 10.1007/s13369-021-05427-8. URL: https://doi.org/10.1007/s13369-021-05427-8.

[20] A. Hassanein, Neamat El-Tazi, and N. Mohy. "Blockchain, Smart Contracts, and Decentralized Applications: An Introduction." In: *Blockchain Technologies* (2022). DOI: 10.1007/978-981-16-3412-3_6. URL: https://doi.org/10.1007/978-981-16-3412-3_6.

[21] Md. Rifat Hossain et al. "A Comprehensive Analysis of Blockchain Technology and Consensus Protocols Across Multilayered Framework." In: *IEEE Access* 12 (2024), pp. 63087–63129. DOI: 10.1109/access.2024.3395536. URL: https://doi.org/10.1109/access.2024.3395536.

[22] Dongyan Huang, Xiaoli, and Shengli Zhang. "Performance Analysis of the Raft Consensus Algorithm for Private Blockchains." In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50 (2018), pp. 172–181. DOI: 10.1109/tsmc.2019.2895471. URL: https://doi.org/10.1109/tsmc.2019.2895471.

[23] A. Jain, Nishant Gupta, and Brij Gupta. "A Survey on Scalable Consensus Algorithms for Blockchain Technology." In: *Cyber Security and Applications* (2024). DOI: 10.1016/j.csa.2024.100065. URL: https://doi.org/10.1016/j.csa.2024.100065.

[24] S. Khan et al. "Blockchain Smart Contracts: Applications, Challenges, and Future Trends." In: *Peer-to-Peer Networking and Applications* 14 (2021), pp. 2901–2925. DOI: 10.1007/s12083-021-01127-0. URL: https://doi.org/10.1007/s12083-021-01127-0.

[25] Aboubaker Khoualed et al. "A new approach to developing startups in Algeria through university engagement: an examination of the implementation of Ministerial Decision no. 1275 at Annaba University and The Higher School of Management Sciences (ESSG)." In: *Management and Entrepreneurship: Trends of Development* (2024). DOI: 10.26661/2522-1566/2024-3/29-13. URL: https://doi.org/10.26661/2522-1566/2024-3/29-13.

[26] N. Suresh Kumar et al. "Navigating the Blockchain Landscape with a Comprehensive Survey of Smart Contracts, Ethereum, and Consensus Mechanisms." In: *4th International Conference on Data Engineering and Communication Systems (ICDECS)*. 2024, pp. 1–6. DOI: 10.1109/icdecs59733.2023.10503046. URL: https://doi.org/10.1109/icdecs59733.2023.10503046.

[27] Bahareh Lashkari and Petr Musilek. "A Comprehensive Review of Blockchain Consensus Mechanisms." In: *IEEE Access* 9 (2021), pp. 43620–43652. DOI: 10.1109/access.2021.3065880. URL: https://doi.org/10.1109/access.2021.3065880.

[28] Fengqi Li et al. "X-RAFT: Improve RAFT Consensus to Make Blockchain Better Secure EdgeAI-Human-IoT Data." In: *IEEE Transactions on Emerging Topics in Computing* 13 (2025), pp. 22–33. DOI: 10.1109/tetc.2024.3472059. URL: https://doi.org/10.1109/tetc.2024.3472059.

[29] Jiajun Liu and Junhao Wu. "A Comprehensive Survey on Blockchain Technology and Its Applications." In: *Highlights in Science, Engineering and Technology* (2024). DOI: 10.54097/r0ggyr24. URL: https://doi.org/10.54097/r0ggyr24.

[30] Shaofei Lu et al. "P-Raft: An Efficient and Robust Consensus Mechanism for Consortium Blockchains." In: *Electronics* (2023). DOI: 10.3390/electronics12102271. URL: https://doi.org/10.3390/electronics12102271.

[31] Haoxiang Luo, Hongfang Yu, and Jian Luo. "PRAFT and RPBFT: A Class of Blockchain Consensus Algorithm and Their Applications in Electric Vehicles Charging Scenarios for V2G Networks." In: *Internet of Things and Cyber-Physical Systems* (2023). DOI: 10.1016/j.iotcps.2023.02.003. URL: https://doi.org/10.1016/j.iotcps.2023.02.003.

[32] Abbas Fadhil Mahdi and Furkan Rabee. "An Overview of Blockchain Technology in Architecture and Consensus with Key Advances." In: *Al-Salam Journal for Engineering and Technology* (Jan. 2025). DOI: 10.55145/ajest.2025.04.01.020. URL: https://doi.org/10.55145/ajest.2025.04.01.020.

[33] Mohit Marwaha, R. Bedi, and Sunil Kumar Gupta. "An Analysis of Blockchain Ecosystems: Understanding Types, Consensus Models and Security." In: *12th International Conference on Computing for Sustainable Global Development (INDIACom)*. 2025, pp. 1–6. DOI: 10.23919/indiacom66777.2025.11115579. URL: https://doi.org/10.23919/indiacom66777.2025.11115579.

[34] Megharani Mayani. "Blockchain Technology: Revolutionizing Data Security and Decentralized Applications." In: *International Scientific Journal of Engineering and Management* (Apr. 2025). DOI: 10.55041/isjem00116. URL: https://doi.org/10.55041/isjem00116.

[35] Abderrezak Osmani and Ahmed Beloucif. "Challenges of Algeria's Economic Development: A Youth Entrepreneurship Perspective." In: (2021). DOI: 10.1108/978-1-80071-322-220211009. URL: https://educate.sohayota.gov.bd/entrepreneurship-and-innovation-education/youth-entrepreneurship-in-developing-economies.

[36] Dattaprasad Patil and Vijaya Bhosale. "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends." In: *International Journal of Advanced Research in Science, Communication and Technology* (Jan. 2023). DOI: 10.48175/ijarsct-8158. URL: https://doi.org/10.48175/ijarsct-8158.

[37] Huatao Peng and L'hocine Walid. "The Effects of Entrepreneurs' Perceived Risks and Perceived Barriers on Sustainable Entrepreneurship in Algeria's SMEs: The Mediating Role of Government Support." In: *Sustainability* (2022). DOI: 10.3390/su141711067. URL: Organisation%20for%20Economic%20Co-operation%20and%20Development.

[38] Ahmad Al-Qerem. "Using Raft as Consensus Algorithm for Blockchain Application of Roaming Services for Mobile Network." In: *International Journal Artificial Intelligent and Informatics* (2022). DOI: 10.33292/ijarlit.v3i1.46. URL: https://doi.org/10.33292/ijarlit.v3i1.46.

[39] Ferron Regis. "Facilitating a soft landing for new start-up companies in Algeria developed by innovative entrepreneurs." In: *International Journal of Technology Management & Sustainable Development* (2019). DOI: 10.1386/tmsd_00012_1. URL: https://doi.org/10.1386/tmsd_00012_1.

[40] Abdelwaheb Sakhri. "The Effectiveness of Business Incubators in Empowering Young Entrepreneurs in Algeria: Policies and Practices." In: *Valahian Journal of Economic Studies* 16 (2025), pp. 25–34. DOI: 10.2478/vjes-2025-0002. URL: https://doi.org/10.2478/vjes-2025-0002.

[41] Haoran Shi et al. "PB-Raft: A Byzantine Fault Tolerance Consensus Algorithm Based on Weighted PageRank and BLS Threshold Signature." In: *Peer-to-Peer Networking and Applications* 18 (2024). DOI: 10.1007/s12083-024-01876-8. URL: https://doi.org/10.1007/s12083-024-01876-8.

[42] B. Shrimali and H. Patel. "Blockchain State-of-the-Art: Architecture, Use Cases, Consensus, Challenges and Opportunities." In: *Journal of King Saud University - Computer and Information Sciences* 34 (2021), pp. 6793–6807. DOI: 10.1016/j.jksuci.2021.08.005. URL: https://doi.org/10.1016/j.jksuci.2021.08.005.

[43] Aditya Pratap Singh. *Blockchain Technology: Core Mechanisms, Evolution, and Future Implementation Challenges*. arXiv preprint arXiv:2505.08772. 2025. DOI: 10.48550/arxiv.2505.08772. URL: https://doi.org/10.48550/arxiv.2505.08772.

[44] H'oriya Souiki. "Startups in Algeria - Towards the promotion of the economic development movement as an alternative to wealth creation without depending on economic rents." In: *Indonesian Journal of Social Science Research* (2023). DOI: 10.11594/ijssr.04.01.05. URL: https://waya.media/algerias-nesda-rolls-out-3-startup-funding-schemes-offering-up-to-usd-75k.

[45] Yufang Sun et al. "Improving Raft Consensus Algorithm with Relay and Lease Mechanism." In: (2024), pp. 84–94. DOI: 10.1007/978-3-031-77095-1_6. URL: https://doi.org/10.1007/978-3-031-77095-1_6.

[46] Siben Tian et al. "VSSB-Raft: A Secure and Efficient Zero Trust Consensus Algorithm for Blockchain." In: *ACM Transactions on Sensor Networks* 20 (2023), pp. 1–22. DOI: 10.1145/3611308. URL: https://doi.org/10.1145/3611308.

[47] Nikita Tiwari et al. "Survey and Analysis of Blockchain Technologies with Respect to: Properties, Algorithms, Architecture, Models, Evolution and Framework." In: *Multimedia Tools and Applications* 84 (2024), pp. 13571–13615. DOI: 10.1007/s11042-024-19580-3. URL: https://doi.org/10.1007/s11042-024-19580-3.

[48] Gautami Tripathi, M. Ahad, and Gabriella Casalino. "A Comprehensive Review of Blockchain Technology: Underlying Principles and Historical Background with Future Challenges." In: *Decision Analytics Journal* (2023). DOI: 10.1016/j.dajour.2023.100344. URL: https://doi.org/10.1016/j.dajour.2023.100344.

[49] Rihong Wang et al. "K-Bucket Based Raft-Like Consensus Algorithm for Permissioned Blockchain." In: *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. 2019, pp. 996–999. DOI: 10.1109/icpads47876.2019.00152. URL: https://doi.org/10.1109/icpads47876.2019.00152.

[50] Shuai Wang et al. "Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends." In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49 (2019), pp. 2266–2277. DOI: 10.1109/tsmc.2019.2895123. URL: https://doi.org/10.1109/tsmc.2019.2895123.

[51] Yuchen Wang et al. "Improved Raft Consensus Algorithm in High Real-Time and Highly Adversarial Environment." In: (2021), pp. 718–726. DOI: 10.1007/978-3-030-87571-8_62. URL: https://doi.org/10.1007/978-3-030-87571-8_62.

[52] A. Yadav, Nikita Singh, and D. Kushwaha. "Evolution of Blockchain and Consensus Mechanisms and its Real-World Applications." In: *Multimedia Tools and Applications* (2023). DOI: 10.1007/s11042-023-14624-6. URL: https://doi.org/10.1007/s11042-023-14624-6.

[53] A. Yadav et al. "A Comparative Study on Consensus Mechanism with Security Threats and Future Scopes: Blockchain." In: *Computer Communications* 201 (2023), pp. 102–115. DOI: 10.1016/j.comcom.2023.01.018. URL: https://doi.org/10.1016/j.comcom.2023.01.018.

[54] Dana Yang, Inshil Doh, and K. Chae. "Cell Based Raft Algorithm for Optimized Consensus Process on Blockchain in Smart Data Market." In: *IEEE Access* 10 (2022), pp. 85199–85212. DOI: 10.1109/access.2022.3197758. URL: https://doi.org/10.1109/access.2022.3197758.

[55] Sijia Yang, Pengliu Tan, and Haowei Fu. "Improved Raft Consensus Algorithm Based on NSGA-II and K-Means++." In: *2024 10th International Symposium on System Security, Safety, and Reliability (ISSSR)*. 2024, pp. 383–390. DOI: 10.1109/isssr61934.2024.00055. URL: https://doi.org/10.1109/isssr61934.2024.00055.

[56] Haotian Yuan et al. "Double-layer Byzantine Fault-tolerant Grouping Consensus Algorithm Based on Raft." In: *IET Blockchain* 4 (2024), pp. 555–569. DOI: 10.1049/blc2.12073. URL: https://doi.org/10.1049/blc2.12073.

[57] Peiyun Zhang and Mengchu Zhou. "Security and Trust in Blockchains: Architecture, Key Technologies, and Open Issues." In: *IEEE Transactions on Computational Social Systems* 7 (2020), pp. 790–801. DOI: 10.1109/tcss.2020.2990103. URL: https://doi.org/10.1109/tcss.2020.2990103.

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة محمد خيضر -بسكرة -
كلية العلوم الدقيقة وعلوم الطبيعة والحياة
قسم الإعلام الآلي

إسم المشروع

منصة GRANTA لإدارة و توثيق الملفات لدعم رواد الأعمال الشباب باستخدام تقنية البلوكشين

مشروع لنيل شهادة مؤسسة ناشئة في اطار القرار الوزاري 1275

الإسم التجاري
GRANTA

الموسم الدراسي: 2025 / 2026

بطاقة معلومات:

حول فريق الإشراف وفريق العمل

1- فريق الإشراف:

فريق الإشراف	
المشرف الرئيسي: لعناني الصادق	**التخصص:** إعلام آلي
المشرف المساعد: كحلول لعيد	**التخصص:** إعلام آلي

2- فريق العمل:

فريق المشروع:	التخصص:	الكلية
الطالبة: لحول هاجر	إعلام آلي	كلية العلوم الدقيقة
الطالب: بن بريكة محمد زكريا	إعلام آلي	كلية العلوم الدقيقة
الطالب: عبد الحي رغيوة	إعلام آلي	كلية العلوم الدقيقة


بسم الله الرحمن الرحيم

المحتويات

1 تقديم المشروع **1**

1.1 فكرة المشروع (الحل المقترح): 2

2.1 القيم المقترحة: 2

3.1 فريق العمل: 3

4.1 أهداف المشورع 4

2 الجوانب الإبتكارية **6**

1.2 طبيعة الإبتكارات في المشروع 7

2.2 مجالات الإبتكار في المشروع 8

3 نموذج الأعمال - Canvas Model Business **10**

1.3 شرائح العملاء (Customer Segments) 11

2.3 علاقات العملاء (Customer Relationships) 12

3.3 القيمة المقترحة (Value Propositions) 12

4.3 الأنشطة الرئيسية (Key Activities) 13

5.3 الشراكات الرئيسية (Key Partners) 14

6.3 الموارد الرئيسية (Key Resources) 15

7.3 القنوات (Channels) 16

8.3 هيكل التكاليف (Cost Structure) 16

9.3 مصادر الإيرادات (Revenue Streams) 17

4 التحليل الاستراتيجي للسوق **19**

1.4 تحليل المتغيرات الكلية :(PESTEL) 20

2.4 تحليل القوى التنافسية :(PORTER) 21

3.4 تحليل :SWOT 22

4.4 المزيج التسويقي 24

5.4 الإستراتيجيات التسويقية 26

5 خطة الإنتاج و التسويق **28**

1.5 خطة الإنتاج 29

1.1.5 موقع المشروع 29

2.1.5 احتياجات الإنتاج 29

3.1.5 مراحل تطوير المنصة 30

2.5 الخطة التنظيمية 30

1.2.5 الهيكل التنظيمي للمشروع 30

2.2.5 توزيع المهام داخل الفريق 31

3.2.5 إدارة المشروع 31

6 الخطة المالية **32**

1.6 تكاليف المشروع 33

1.1.6 التكاليف الاستثمارية 33

2.1.6 التكاليف التشغيلية 33

2.6 الهيكل التمويلي للمشروع 34

7 النموذج الأولي التجريبي **35**

1.7 المعمارية التقنية للنظام 36

2.7 تسلسل العمل (Worfklow) داخل المنصة 37

3.7 لقطات من المنصة (Screenshots) 38

1.3.7 الصفحة الرئيسية و التوجيه 38

2.3.7 واجهة المترشح (Candidate Dashboard) 39
3.3.7 واجهة المشرف الأولي و النهائي 41
4.3.7 إصدار شهادة NFT و تخزينها على البلوكشين 42
5.3.7 التحقق العمومي من الشهادة 43
6.3.7 اللوحة الإدارية و المراقبة 43
4.7 قياس الأداء (Performance Benchmarking) 44
5.7 خارطة طريق التطوير (Roadmap) 46

خاتمة عامة **47**

قائمة الجداول

1.4 تحليل SWOT للمشروع 23

2.4 التغلب على نقاط الضعف و تجنب التهديدات 24

1.6 التكاليف الاستثمارية للمشروع 33

2.6 التكاليف التشغيلية للمشروع 34

3.6 الهيكل التمويلي للمشروع 34

1.7 التقنيات المستخدمة في تطوير منصة GRANTA 37

2.7 مؤشرات أداء منصة GRANTA المُقاسة بـ Caliper Hyperledger 45

3.7 خارطة طريق تطوير و توسع منصة GRANTA 46

المحور 1

تقديم المشروع

1.1 فكرة المشروع (الحل المقترح):

يهدف هذا المشروع إلى تطوير المنصة الرقمية **GRANTA** لإدارة و توثيق ملفات طلبات الدعم الموجهة للشباب المقاول في الجزائر، باستخدام تقنية البلوكشين Hyperledger) ،(Fabric تخزين الملفات اللامركزي ،(IPFS) و الشهادات الرقمية على شكل .NFT تسمح هذه المنصة برقمنة الكاملة لعملية إيداع، مراجعة، توثيق و التحقق من الملفات الخاصة بالمؤسسات الناشئة و المشاريع الصغيرة.

حالياً، تعتمد العديد من الهيئات الداعمة مثل NESDA و CNAC و ANGEM على إجراءات إدارية تقليدية تتطلب تقديم ملفات ورقية، مما يؤدي إلى تعقيد الإجراءات وطول مدة المعالجة، إضافة إلى مخاطر التزوير و فقدان الوثائق و انعدام الشفافية في تتبع حالة الملف.

تُقدّم **GRANTA** حلاً تقنياً متكاملاً يستجيب لهذه التحديات من خلال:

- **تسلسل عمل (Worfklow) متعدد المستويات** يربط المترشح بمشرف أولي ثم مشرف نهائي، مع تتبع لحظي لحالة الملف.
- **تثبيت بصمة الشهادات Hash) (Anchoring** على دفتر سجلات بلوكشين مُرخَّص باستخدام Hyperledger Fabric 5.2 و خوارزمية التوافق .Raft
- **إصدار شهادات NFT** غير قابلة للتزوير، يمكن التحقق منها علنياً عبر رابط أو رمز .QR
- **تخزين الوثائق على IPFS** يضمن التوزيع و عدم الاعتماد على خادم مركزي وحيد.
- **واجهة ويب حديثة** مبنية على Next.js 15 و TypeScript تُقدّم تجربة استخدام بسيطة لجميع الأطراف.

2.1 القيم المقترحة:

ترتكز القيمة المقترحة لمشروع **GRANTA** على تقديم حل تكنولوجي مبتكر يجمع بين رقمنة الإجراءات الإدارية التقليدية و ضمان ثبوتية الوثائق عبر تقنية البلوكشين، لفائدة رواد الأعمال الشباب و الهيئات الداعمة على حد سواء. و تتمثل أهم المزايا التي توفرها المنصة فيما يلي:

1. **رقمنة الإجراءات الإدارية**

توفر المنصة إمكانية إيداع ملفات طلب الدعم بشكل إلكتروني، مما يلغي الحاجة إلى التعامل مع الوثائق الورقية ويقلل من التعقيدات الإدارية التقليدية.

2. **تسريع معالجة الملفات**

من خلال اعتماد نظام رقمي لإدارة الطلبات، يمكن تقليل مدة دراسة الملفات وتسريع عملية اتخاذ القرار، الأمر الذي يساعد رواد الأعمال على إطلاق مشاريعهم في وقت أقصر.

3. **تحسين الشفافية والمتابعة**

تسمح المنصة للمستخدمين بمتابعة حالة ملفاتهم في الوقت الحقيقي، مما يعزز الشفافية في معالجة الطلبات ويقلل من حالات الغموض أو التأخير غير المبرر.

4. **مركزية المعلومات**

تعمل المنصة على تجميع جميع المعلومات والوثائق المتعلقة بطلبات الدعم في نظام موحد، مما يسهل على الهيئات المعنية إدارة الملفات والرجوع إليها عند الحاجة.

5. **تعزيز أمن البيانات و عدم قابليتها للتزوير**

تعتمد المنصة على آليات حماية متقدمة، أبرزها تسجيل بصمة كل وثيقة (Hash) على دفتر البلوكشين بشكل غير قابل للتعديل، إضافة إلى تشفير البيانات و إدارة دقيقة لصلاحيات المستخدمين ،(RBAC) مما يلغي عملياً إمكانية التلاعب بالوثائق بعد تسجيلها.

6. **دعم التحول الرقمي**

يساهم المشروع في دعم جهود التحول الرقمي في الإدارة الجزائرية، من خلال إدخال حلول رقمية حديثة لتحسين جودة الخدمات المقدمة للمواطنين.

7. **تحسين تجربة المستخدم**

تم تصميم المنصة بواجهة استخدام بسيطة وسهلة، مما يسمح لرواد الأعمال باستخدامها بسهولة حتى في حال عدم امتلاكهم خبرة تقنية متقدمة.

3.1 فريق العمل:

يتكون فريق عمل مشروع المنصة الرقمية لإدارة ملفات دعم الشباب المقاول من مجموعة من الطلبة في تخصص الإعلام الآلي، تجمعهم الرغبة في تطوير حلول تكنولوجية تساهم في تحسين الخدمات الإدارية ودعم التحول الرقمي.

يعمل أعضاء الفريق بشكل تعاوني من أجل تصميم وتطوير منصة رقمية قادرة على تسهيل عملية إدارة ملفات طلبات الدعم الموجهة للشباب المقاول، وذلك من خلال توظيف معارفهم في مجالات تطوير البرمجيات، قواعد البيانات، وتصميم واجهات المستخدم.

كما يتم توزيع المهام داخل الفريق بطريقة منظمة تضمن التكامل بين مختلف الجوانب التقنية للمشروع، حيث يشمل العمل:

- تحليل متطلبات النظام وتحديد الوظائف الأساسية للمنصة.
- تصميم بنية النظام وقاعدة البيانات الخاصة بإدارة الملفات.
- تطوير واجهة المستخدم الخاصة بالمنصة لضمان سهولة الاستخدام.
- برمجة وظائف النظام المتعلقة بإدارة الطلبات والوثائق.
- اختبار النظام والتأكد من كفاءته قبل إطلاق النسخة التجريبية.

ويهدف هذا التعاون بين أعضاء الفريق إلى تحقيق أفضل النتائج الممكنة وتطوير منصة عملية تلبي احتياجات رواد الأعمال والهيئات الداعمة، مع التركيز على الابتكار والجودة في تنفيذ المشروع.

كما يستفيد الفريق من توجيهات الأستاذ المشرف الذي يقدم الدعم العلمي والمنهجي خلال مختلف مراحل إنجاز المشروع.

4.1 أهداف المشروع

1. **يهدف المشروع إلى توفير منصة إلكترونية** تسمح لرواد الأعمال بإيداع ملفات طلب الدعم بشكل رقمي، مما يقلل من الاعتماد على الوثائق الورقية ويُبسط الإجراءات الإدارية.

2. **تسهيل إدارة الطلبات**

   تمكين الهيئات المعنية من إدارة ومراجعة ملفات طلبات الدعم بطريقة منظمة من خلال نظام مركزي يسمح بتخزين ومعالجة البيانات بشكل فعال.

3. **تسريع معالجة الملفات**

   المساهمة في تقليل مدة دراسة الملفات من خلال أتمتة بعض العمليات الإدارية وتسهيل عملية تبادل المعلومات بين مختلف الجهات المعنية.

4. **تحسين الشفافية**

توفير نظام يسمح للمستخدمين بمتابعة حالة ملفاتهم في مختلف مراحل المعالجة، مما يعزز الشفافية ويزيد من ثقة رواد الأعمال في النظام.

5. **ضمان أمن البيانات**

توفير آليات حماية متقدمة لضمان سلامة البيانات المخزنة داخل المنصة، مثل تشفير المعلومات وإدارة صلاحيات الوصول للمستخدمين.

6. **دعم التحول الرقمي**

المساهمة في دعم التحول الرقمي للإدارة من خلال تقديم حل تقني حديث يساعد على تحسين جودة الخدمات الإدارية المقدمة للمواطنين.

7. **تطوير نموذج أولي قابل للتطوير**

تصميم وتنفيذ نموذج أولي للمنصة يمكن تطويره مستقبلاً ليصبح نظاماً متكاملاً يُستخدم من قبل الهيئات الداعمة لرواد الأعمال.

المحور 2

الجوانب الإبتكارية

1.2 طبيعة الإبتكارات في المشروع

يعتمد مشروع **GRANTA** على دمج عدة تقنيات حديثة لم تسبق تجربتها مجتمعةً في سياق دعم رواد الأعمال الشباب في الجزائر، حيث يجمع بين سلاسة الويب الحديث، و ثبوتية البلوكشين المُرخَّص، و لامركزية تخزين IPFS، و رمزية الشهادات NFT.

و يمكن تحديد طبيعة الإبتكار في المشروع من خلال الجوانب التالية:

1. **ابتكار في الخدمة**

يقدّم المشروع خدمة رقمية غير مسبوقة وطنياً، تجمع بين إدارة دورة حياة ملف طلب الدعم و التوثيق غير القابل للتزوير للوثائق عبر بلوكشين مُرخَّص، إضافة إلى توفير خدمة تحقق عمومية عبر رمز QR أو رابط عمومي.

2. **ابتكار تكنولوجي**

يعتمد النظام على منظومة متكاملة من التقنيات الحديثة:

- **Fabric Hyperledger 5.2**: بلوكشين مُرخَّص داخل المؤسسة يستخدم خوارزمية التوافق Raft.
- **IPFS**: تخزين لامركزي للوثائق بعناوين محتوى (Content-Addressed Storage).
- **Next.js 15 + TypeScript**: واجهة ويب سريعة و مستجيبة.
- **SQLite**: قاعدة بيانات تشغيلية خفيفة للبيانات غير الحساسة.
- **Caliper Hyperledger**: أداة قياس الأداء وثّقت إنتاجية تصل إلى إنتاجية 228 معاملة في الثانية (TPS) بدون أي فشل في المعاملات.

3. **ابتكار في إدارة البيانات (نموذج هجين)**

يعتمد المشروع على نموذج ثقة هجين: طبقة الويب تضمن سهولة الاستخدام، القاعدة العلائقية تضمن التناسق التشغيلي، IPFS يوفر التخزين اللامركزي، و Fabric Hyperledger يثبت بصمات الشهادات بشكل غير قابل للتعديل.

4. **ابتكار في تجربة المستخدم**

تمّ تصميم المنصة بواجهات متخصصة حسب الدور (مترشح، مشرف أولي، مشرف نهائي، إداري) مع توجيه بصري واضح لجعل تكنولوجيا معقدة (البلوكشين، العقود الذكية، IPFS شفافة للمستخدم النهائي.

5. **ابتكار في الشفافية و التحقق العمومي**

يتيح النظام لأي طرف ثالث (بنوك، شركاء، صناديق الدعم) التحقق من أصالة الشهادة في ثوانٍ دون الحاجة للتواصل مع الجهة المصدرة، عبر مقارنة بصمة الوثيقة بالبصمة المثبتة على البلوكشين.

2.2 مجالات الإبتكار في المشروع

يتضمن مشروع المنصة الرقمية لإدارة ملفات دعم الشباب المقاول عدة مجالات للابتكار تساهم في تطوير حل رقمي حديث يسهل عملية إدارة طلبات الدعم ويعزز كفاءة الخدمات الإدارية. وتتمثل أهم مجالات الابتكار في هذا المشروع فيما يلي:

1. **الابتكار في الرقمنة وإدارة الخدمات**

يقدم المشروع نموذجاً حديثاً لإدارة الخدمات الإدارية من خلال رقمنة عملية تقديم ملفات طلب الدعم، مما يسمح بالانتقال من النظام التقليدي المعتمد على الوثائق الورقية إلى نظام رقمي يعتمد على معالجة البيانات إلكترونياً.

2. **الابتكار في إدارة الوثائق الرقمية**

توفر المنصة نظاماً متكاملاً لإدارة الوثائق يسمح للمستخدمين برفع الملفات المطلوبة وتنظيمها بطريقة آمنة، مع إمكانية الوصول إليها بسهولة من قبل الجهات المعنية، مما يساهم في تسهيل عملية مراجعة الملفات وتقليل الأخطاء الإدارية.

3. **الابتكار في تصميم المنصات الرقمية**

يعتمد المشروع على تطوير منصة ويب حديثة مصممة بواجهة استخدام سهلة ومرنة، مما يسمح للمستخدمين بالتفاعل مع النظام بطريقة بسيطة وسرعة، ويضمن تجربة استخدام فعالة لمختلف الفئات المستفيدة.

4. **الابتكار في أمن المعلومات**

يولي المشروع أهمية كبيرة لحماية البيانات، حيث يتم اعتماد آليات مختلفة لضمان أمن المعلومات مثل إدارة صلاحيات المستخدمين، تشفير البيانات الحساسة، وضمان سلامة الوثائق المخزنة داخل النظام.

5. **الابتكار في تكامل الأنظمة**

تتيح المنصة إمكانية التكامل مع أنظمة أخرى مستقبلاً مثل أنظمة المؤسسات الداعمة أو الأنظمة البنكية، مما يسمح بتبادل المعلومات بطريقة آمنة ويساهم في تحسين كفاءة إدارة ملفات الدعم.

6. الابتكار في دعم التحول الرقمي

يساهم المشروع في دعم التحول الرقمي داخل المؤسسات من خلال تقديم نموذج عملي لاستخدام التكنولوجيا في تحسين الخدمات الإدارية، مما يساعد على تطوير بيئة أكثر كفاءة وشفافية في معالجة طلبات الدعم.

المحور 3

نموذج الأعمال - Canvas Model Business

يُعد نموذج العمل التجاري **Canvas Model Business (BMC)** أداةً استراتيجيةً تسمح بتلخيص الكيفية التي يخلق بها المشروع القيمة و يقدّمها و يحقّق منها عوائد. يعتمد هذا النموذج على تسعة مكوّنات أساسية وفق منهجية Osterwalder، تُغطّي جميع جوانب المشروع من العملاء إلى الموارد و التكاليف و الإيرادات. و فيما يلي العرض التفصيلي لكل مكوّن من مكوّنات نموذج الأعمال لمشروع **GRANTA**:

1.3 شرائح العملاء (Customer Segments)

تُحدّد هذه الشريحة الفئات المستهدفة التي يسعى المشروع لخدمتها و تلبية احتياجاتها:

1. **حاملو أفكار مشاريع و رواد الأعمال الشباب:**
يمثّلون الفئة الأساسية المستهدفة، و هم أشخاص يمتلكون أفكاراً ريادية و يحتاجون إلى دعم مالي و إداري لتحويل أفكارهم إلى مشاريع فعلية. تُوفّر لهم المنصة آلية لتقديم ملفاتهم و متابعة مسار معالجتها بشكل رقمي و شفاف.

2. **مؤسسات مصغّرة و مؤسسات ناشئة:**
تشمل المؤسسات الصغيرة التي تمّ إنشاؤها حديثاً أو التي لا تزال في طور النمو، و تحتاج إلى مرافقة إدارية و تقنية و تمويل لضمان استمراريتها و تطوّرها في السوق.

3. **خرّيجو الجامعات و مراكز التكوين المهني:**
يُشكّلون فئة مهمة من الشباب الذين أنهوا تكوينهم الأكاديمي أو المهني و يبحثون عن فرص لإنشاء مشاريعهم الخاصة. تُساعدهم المنصة في الحصول على التمويل و المرافقة اللازمة لبدء نشاطهم.

4. **شباب بطّالين:**
يُعدّ إدماج الشباب البطّالين في سوق العمل من أولويات المشروع، حيث يُوفّر لهم فرصة الحصول على تمويل و مرافقة لإنشاء مشاريعهم الخاصة بدلاً من انتظار فرص التوظيف التقليدية.

5. **الهيئات الداعمة العمومية:**
تشمل الوكالات الحكومية مثل NESDA و CNAC و ANGEM و الوزارة المكلّفة بالاقتصاد المعرفي و المؤسسات الناشئة، التي تحتاج إلى نظام مركزي لمعالجة ملفات الدعم و إصدار الشهادات و متابعة المشاريع الممولة.

2.3 علاقات العملاء (Customer Relationships)

يُحدّد هذا المكوّن طبيعة العلاقة التي تربط المشروع بعملائه و كيفية التفاعل معهم:

1. **استشارات إدارية و تقنية:**

تقديم خدمات استشارية متخصّصة لمساعدة حاملي المشاريع في إعداد ملفاتهم الإدارية و التقنية بشكل صحيح، بما يشمل دراسات الجدوى و خطط الأعمال و الوثائق القانونية المطلوبة.

2. **المرافقة و التكوين:**

توفير برامج تكوينية و ورشات عمل لتأهيل رواد الأعمال و تزويدهم بالمهارات اللازمة لإدارة مشاريعهم، مع مرافقة مستمرة خلال مختلف مراحل إنشاء المؤسسة.

3. **الدعم التقني المباشر:**

توفير قناة دعم مباشرة عبر البريد الإلكتروني و نظام تذاكر داخل المنصة للرد على استفسارات المستخدمين و حل المشكلات التقنية التي قد تواجههم.

4. **المتابعة الدورية:**

متابعة دورية للمشاريع الممولة لضمان حسن سير العمل و تقديم التوجيهات اللازمة عند الحاجة، مع إعداد تقارير دورية عن تقدّم المشاريع.

3.3 القيمة المقترحة (Value Propositions)

تُمثّل القيمة المقترحة مجموعة المنافع التي يقدّمها المشروع لعملائه و التي تميّزه عن غيره:

1. **التمويل:**

توفير آليات تمويل مناسبة لحاملي المشاريع، سواء عبر التمويل المباشر أو عبر التنسيق مع البنوك و المؤسسات المالية لتسهيل الحصول على القروض و الدعم المالي اللازم لإطلاق المشاريع.

2. **المرافقة:**

تقديم خدمة مرافقة شاملة تبدأ من مرحلة دراسة فكرة المشروع إلى غاية تجسيده على أرض الواقع، مع توفير خبراء و مستشارين متخصّصين في مختلف المجالات.

3. **محاربة البطالة:**

المساهمة الفعّالة في تقليص نسبة البطالة من خلال تشجيع المبادرات الفردية و تحويل الشباب البطّالين إلى رواد أعمال منتجين، مما يُسهم في خلق فرص عمل جديدة.

4. **دعم الإدماج المهني:**

تسهيل عملية إدماج الشباب في الحياة المهنية عبر توفير الأدوات و الموارد اللازمة لإنشاء مؤسساتهم الخاصة، مع ضمان مطابقة المشاريع للمتطلبات القانونية و التنظيمية.

5. **تسهيل الإجراءات الإدارية:**

تبسيط و رقمنة الإجراءات الإدارية المعقّدة المتعلّقة بإنشاء المؤسسات و الحصول على الدعم، مما يُقلّص الوقت و الجهد المبذولين من طرف حاملي المشاريع و يُسرّع عملية المعالجة.

6. **التوثيق الرقمي الموثوق:**

تقديم نظام توثيق رقمي يضمن أصالة الشهادات و الوثائق الصادرة عبر المنصة، مع إمكانية التحقق الفوري من صحتها من طرف أي جهة معنية.

4.3 الأنشطة الرئيسية (Key Activities)

تُحدّد الأنشطة الرئيسية العمليات الأساسية التي يقوم بها المشروع لتقديم قيمته المقترحة:

1. **دراسة المشاريع:**

استقبال ملفات المشاريع المقدّمة من طرف رواد الأعمال و تقييمها وفق معايير محدّدة تشمل الجدوى الاقتصادية و التقنية و الأثر الاجتماعي، مع إعداد تقارير تقييم مفصّلة لكل مشروع.

2. **تمويل المشاريع:**

إدارة عملية تمويل المشاريع المقبولة، بما يشمل تحديد مبلغ التمويل المناسب و آليات الصرف و شروط الاسترداد، مع ضمان الشفافية في جميع المعاملات المالية.

3. **التنسيق مع البنوك:**

بناء علاقات شراكة مع المؤسسات البنكية لتسهيل حصول رواد الأعمال على القروض و الخدمات المالية، مع التفاوض للحصول على شروط تمويل تفضيلية للمشاريع المدعومة.

4. **متابعة المشاريع:**

تتبع سير المشاريع الممولة بشكل دوري للتأكد من التزام أصحابها بالخطط المتفق عليها، مع تقديم الدعم و التوجيه عند مواجهة صعوبات أو عراقيل.

5. **التطوير و الصيانة التقنية:**

تطوير و تحسين المنصة الرقمية بشكل مستمر، بما يشمل تحديث الواجهات و إضافة ميزات جديدة و تصحيح الأخطاء و ضمان استقرار النظام و أمنه.

6. **التسويق و بناء الشراكات:**

الترويج لخدمات المنصة و إبرام عقود شراكة مع المؤسسات الداعمة و الجامعات و حاضنات الأعمال لتوسيع قاعدة المستفيدين و تعزيز تأثير المشروع.

5.3 الشراكات الرئيسية (Key Partners)

تُحدّد الشراكات الرئيسية الجهات الفاعلة التي يتعاون معها المشروع لتحقيق أهدافه:

1. **وزارة العمل و التشغيل:**

شريك استراتيجي أساسي يوفّر الإطار التنظيمي و القانوني للمشروع، و يُسهم في توجيه السياسات العمومية المتعلّقة بدعم التشغيل و ريادة الأعمال.

2. **مراكز التكوين المهني:**

تُعدّ نقاط إدخال مهمة للمستفيدين المحتملين، حيث توفّر قاعدة من الشباب المؤهّلين مهنياً الذين يمكنهم الاستفادة من خدمات التمويل و المرافقة.

3. **الجامعات و حاضنات الأعمال:**

تُساهم في تحديد المشاريع الواعدة و توجيه الطلبة و الخرّيجين نحو ريادة الأعمال، مع توفير فضاءات حاضنة و برامج تكوينية مكمّلة.

4. **البنوك و المؤسسات المالية:**

شركاء أساسيون في عملية التمويل، حيث توفّر القروض و الخدمات المالية لرواد الأعمال بشروط تفضيلية في إطار اتفاقيات الشراكة.

5. **غرف الصناعة التقليدية:**

تُساهم في دعم المشاريع المتعلّقة بالحرف و الصناعات التقليدية، و تُوفّر خبرة ميدانية في هذا المجال و شبكة علاقات مع الحرفيين و المنتجين المحليين.

6. **غرف التجارة:**

تُوفّر بيئة أعمال مناسبة و تُسهّل التواصل بين رواد الأعمال و الفاعلين الاقتصاديين، مع تقديم خدمات التسجيل و المرافقة التجارية.

6.3 الموارد الرئيسية (Key Resources)

تُحدّد الموارد الرئيسية الأصول الضرورية التي يحتاجها المشروع لتنفيذ أنشطته:

1. **التمويل:**

يُمثّل العنصر الأساسي لتشغيل المشروع، و يشمل رأس المال الأولي و الأموال المخصّصة لتمويل المشاريع المدعومة، بالإضافة إلى ميزانية التسيير اليومي.

2. **الموارد البشرية:**

فريق عمل متعدّد الكفاءات يشمل مستشارين في ريادة الأعمال و خبراء ماليين و مطوّرين تقنيين و مسؤولي متابعة المشاريع، يُشكّلون العمود الفقري لنشاط المشروع.

3. **المكاتب و البنية التحتية:**

فضاءات عمل مجهّزة لاستقبال رواد الأعمال و عقد الاجتماعات و تقديم الاستشارات، بالإضافة إلى البنية التحتية التقنية (خوادم، شبكات، أنظمة معلوماتية).

4. **المرافقة:**

منظومة مرافقة متكاملة تشمل أدوات التقييم و نماذج دراسات الجدوى و برامج التكوين و شبكة الخبراء و المستشارين المتخصّصين.

5. **التنظيمات و الإطار القانوني:**

مجموعة النصوص التنظيمية و القانونية التي تُؤطّر نشاط المشروع، بما فيها القرار الوزاري 1275 و القوانين المتعلّقة بدعم المؤسسات الناشئة و حماية البيانات.

7.3 القنوات (Channels)

تُحدّد القنوات الوسائل التي يستخدمها المشروع للوصول إلى عملائه و تقديم خدماته:

1. **الوكالات المحلية:**

شبكة من الوكالات المنتشرة عبر التراب الوطني تُوفّر خدمة القرب لرواد الأعمال، و تُمكّنهم من تقديم ملفاتهم و الحصول على الاستشارات بشكل مباشر.

2. **منصات التواصل الاجتماعي:**

استخدام شبكات التواصل الاجتماعي (فيسبوك، لينكد إن، إنستغرام) للترويج لخدمات المشروع و التفاعل مع الجمهور المستهدف و نشر قصص نجاح المستفيدين.

3. **أيام تحسيسية:**

تنظيم فعاليات و أيام إعلامية و تحسيسية لتعريف الشباب بفرص الدعم المتاحة و تشجيعهم على المبادرة و إنشاء مشاريعهم الخاصة.

4. **شراكات مع الجامعات و مراكز التكوين:**

التواصل المباشر مع الطلبة و الخرّيجين عبر اتفاقيات شراكة مع المؤسسات الجامعية و مراكز التكوين المهني، لتوجيههم نحو ريادة الأعمال.

5. **الموقع الإلكتروني:**

منصة رقمية متكاملة تُتيح للمستخدمين الاطلاع على الخدمات المتاحة و تقديم ملفاتهم و متابعة مسار معالجتها عن بُعد.

6. **الملتقيات و الفعاليات:**

المشاركة في صالونات و ملتقيات ريادة الأعمال و المعارض الاقتصادية للتعريف بالمشروع و استقطاب مستفيدين و شركاء جُدد.

8.3 هيكل التكاليف (Cost Structure)

يُحدّد هيكل التكاليف مجموع المصاريف اللازمة لتشغيل المشروع و تقديم خدماته:

1. **أجور الموظفين:**

تُمثّل أكبر بند في هيكل التكاليف، و تشمل رواتب الفريق التقني و الإداري و المستشارين و مسؤولي متابعة المشاريع، بالإضافة إلى الأعباء الاجتماعية المرتبطة بها.

2. **تكاليف التكوين:**

المصاريف المتعلّقة بتنظيم برامج التكوين و ورشات العمل لفائدة رواد الأعمال، بما يشمل أتعاب المكوّنين و تجهيز قاعات التكوين و إعداد المحتوى التعليمي.

3. **مصاريف التسيير:**

التكاليف التشغيلية اليومية التي تشمل إيجار المكاتب و فواتير الطاقة و الاتصالات و اللوازم المكتبية و صيانة التجهيزات و التنقلات.

4. **تكاليف التمويل:**

المصاريف المالية المرتبطة بإدارة عمليات التمويل، بما فيها تكاليف معالجة الملفات و التأمينات و الضمانات و العمولات البنكية.

5. **تكاليف التطوير و الصيانة:**

المصاريف المتعلّقة بتطوير المنصة الرقمية و صيانتها و استضافتها، بما يشمل تكاليف الخوادم و تراخيص البرمجيات و خدمات الأمن السيبراني.

6. **تكاليف التسويق:**

ميزانية الأنشطة الترويجية و الإشهارية و تنظيم الفعاليات و المشاركة في الملتقيات، بالإضافة إلى تكاليف إدارة منصات التواصل الاجتماعي.

9.3 مصادر الإيرادات (Revenue Streams)

تُحدّد مصادر الإيرادات الطرق التي يُحقّق من خلالها المشروع عوائده المالية:

1. **التمويل الحكومي:**

يُعدّ المصدر الرئيسي للإيرادات، ويشمل المنح و الإعانات المقدّمة من طرف الدولة في إطار برامج دعم التشغيل و ريادة الأعمال و المؤسسات الناشئة، وفق الأطر القانونية المعمول بها.

2. **الشراكات مع البنوك:**

تحقيق إيرادات من خلال اتفاقيات الشراكة مع المؤسسات البنكية، التي تشمل عمولات الوساطة في عمليات التمويل و رسوم إدارة الملفات و خدمات التحقق من الوثائق.

3. **الخدمات الاستشارية:**

تقديم خدمات استشارية مدفوعة للمؤسسات و الهيئات التي تحتاج إلى خبرة في مجال دعم ريادة الأعمال و رقمنة الإجراءات الإدارية و إعداد دراسات الجدوى.

4. **رسوم الخدمات:**

تحصيل رسوم مقابل خدمات محدّدة مثل إصدار الشهادات و التحقق من الوثائق و معالجة الملفات العاجلة، وفق جدول تعريفي محدّد و شفاف.

المحور 4

التحليل الاستراتيجي للسوق

1.4 تحليل المتغيرات الكلية :(PESTEL)

يهدف هذا الجزء إلى دراسة الجوانب المختلفة التي تؤثر على بيئة المشروع:

1. **العوامل السياسية**

تلعب السياسات الحكومية دوراً مهماً في دعم مشاريع التحول الرقمي وريادة الأعمال في الجزائر. حيث تعمل الدولة على تشجيع إنشاء المؤسسات الناشئة من خلال برامج الدعم والتمويل الموجهة للشباب، إضافة إلى إطلاق مبادرات تهدف إلى تطوير الخدمات الرقمية داخل الإدارات العمومية.

كما أن وجود هيئات حكومية مثل NESDA وCNAC وANGEM يساهم في دعم المشاريع الصغيرة، مما يوفر بيئة مناسبة لتطبيق حلول رقمية تساعد على تحسين إدارة طلبات الدعم.

2. **العوامل الاقتصادية**

يشهد الاقتصاد الجزائري اهتماماً متزايداً بتطوير قطاع المؤسسات الصغيرة والمتوسطة وتشجيع الابتكار لدى الشباب. ويعد توفير منصات رقمية لتسهيل الإجراءات الإدارية عاملاً مهماً في تقليل التكاليف التشغيلية وتحسين كفاءة الخدمات.

كما أن رقمنة الخدمات يمكن أن تساعد على تقليل الوقت والموارد المستهلكة في معالجة الملفات الإدارية، مما يساهم في تحسين الأداء الاقتصادي للمؤسسات.

3. **العوامل الاجتماعية**

يشهد المجتمع الجزائري تزايداً في اهتمام الشباب بريادة الأعمال وإنشاء المشاريع الخاصة، خاصة مع انتشار الثقافة الرقمية واستخدام الإنترنت بشكل واسع. لذلك فإن توفير منصات إلكترونية تسهل الوصول إلى خدمات الدعم يمكن أن يشجع المزيد من الشباب على إطلاق مشاريعهم.

كما أن المستخدمين أصبحوا يفضلون الخدمات الرقمية التي توفر الوقت والجهد بدلاً من الإجراءات التقليدية.

4. **العوامل التكنولوجية**

يشهد مجال التكنولوجيا تطوراً سريعاً، خاصة في تقنيات تطوير تطبيقات الويب والخدمات السحابية وإدارة البيانات الرقمية. ويساهم توفر هذه التقنيات في إمكانية تطوير منصات رقمية متقدمة قادرة على معالجة البيانات بكفاءة عالية.

كما أن انتشار الإنترنت والبنية التحتية الرقمية يسمح بتوفير خدمات إلكترونية يمكن الوصول إليها من مختلف المناطق.

5. **العوامل البيئية**

يساهم التحول نحو استخدام الأنظمة الرقمية في تقليل الاعتماد على الوثائق الورقية، مما يقلل من استهلاك الموارد الطبيعية ويحد من التأثيرات البيئية المرتبطة بإنتاج الورق.

كما أن رقمنة الخدمات تساعد على تقليل التنقلات الإدارية، مما يساهم في تقليل استهلاك الطاقة والانبعاثات المرتبطة بالنقل.

6. **العوامل القانونية**

يجب أن يلتزم المشروع بالقوانين والتشريعات المتعلقة بحماية البيانات والخصوصية، خاصة فيما يتعلق بإدارة المعلومات الشخصية للمستخدمين.

كما ينبغي احترام القوانين المنظمة للخدمات الرقمية والتعاملات الإلكترونية، إضافة إلى ضمان أمن المعلومات وحماية الوثائق المخزنة داخل النظام.

2.4 تحليل القوى التنافسية :(PORTER)

يُستخدم نموذج Porter للقوى التنافسية الخمس من أجل تحليل مستوى المنافسة داخل السوق وتحديد العوامل التي قد تؤثر على نجاح المشروع. ويساعد هذا التحليل على فهم البيئة التنافسية التي يعمل فيها المشروع وتحديد الفرص والتحديات المحتملة.

وفيما يلي تحليل القوى التنافسية بالنسبة لمشروع المنصة الرقمية لإدارة ملفات دعم الشباب المقاول:

1. **شدة المنافسة بين المنافسين الحاليين**

تعتبر المنافسة في هذا المجال محدودة نسبياً، خاصة في السوق المحلي، حيث لا توجد العديد من المنصات الرقمية المتخصصة في إدارة ملفات دعم المشاريع الموجهة للشباب. وغالباً ما تعتمد الهيئات المعنية على الإجراءات الإدارية التقليدية أو أنظمة داخلية محدودة.

وهذا يوفر فرصة للمشروع لتقديم حل رقمي مبتكر يمكن أن يساهم في تحسين طريقة إدارة الملفات الإدارية.

2. **قوة تفاوض العملاء**

يمثل رواد الأعمال والجهات المستفيدة الفئة الرئيسية من مستخدمي المنصة، وتعتبر قدرتهم على التفاوض متوسطة، حيث يعتمد نجاح المشروع على مدى توفير خدمة فعالة وسهلة الاستخدام تلبي احتياجاتهم.

كما أن توفير منصة رقمية موثوقة يمكن أن يزيد من ثقة المستخدمين ويشجعهم على اعتماد النظام في إدارة طلباتهم.

3. **قوة تفاوض الموردين**

تمثل الموارد الأساسية للمشروع في التقنيات البرمجية والبنية التحتية الرقمية مثل خدمات الاستضافة وقواعد البيانات. وبما أن العديد من هذه الخدمات متوفرة في السوق، فإن قوة الموردين تعتبر محدودة نسبياً، مما يمنح المشروع مرونة في اختيار التقنيات المناسبة.

4. **تهديد المنتجات البديلة**

تتمثل البدائل المتاحة حالياً في:

- تقديم الملفات الورقية و التنقل الفعلي بين الإدارات.
- أنظمة داخلية بسيطة داخل المؤسسات تعتمد على الجداول الإلكترونية أو برمجيات إدارة تقليدية (مثل ERP عامة لا تغطي موضوع التوثيق).
- أدوات عامة لتبادل الوثائق مثل Drive Google أو البريد الإلكتروني، و هي لا تقدم أي ضمان تقني لعدم التزوير.

تبقى هذه البدائل عاجزة عن تقديم ثبوتية رقمية غير قابلة للتزوير، و هو العنصر الذي تفردت به منصة GRANTA.

5. **تهديد الداخلين الجدد**

دخول فاعلين جدد في سوق حلول التوثيق الرقمي يبقى محدوداً جزائرياً بسبب المعرفة المتخصصة المطلوبة في البلوكشين و العقود الذكية، إلا أن الدور الداعم للوزارة المكلفة بالإقتصاد المعرفي و المؤسسات الناشئة قد يجلب فاعلين جدد.

3.4 تحليل :SWOT

يُعد تحليل SWOT من الأدوات المهمة في التحليل الاستراتيجي، حيث يسمح بدراسة البيئة الداخلية والخارجية للمشروع من خلال تحديد نقاط القوة ونقاط الضعف والفرص والتهديدات التي قد تؤثر على نجاحه. ويساعد هذا التحليل على فهم وضعية المشروع في السوق ووضع استراتيجيات مناسبة لتطويره.

نقاط القوة (Strengths)	نقاط الضعف (Weaknesses)
- تقديم حل رقمي حديث يسهل عملية إدارة ملفات طلبات الدعم. - تقليل الإجراءات الورقية وتعقيدات المعاملات الإدارية. - إمكانية متابعة حالة الطلب بشكل مباشر من خلال المنصة. - تصميم منصة سهلة الاستخدام ومناسبة لمختلف فئات المستخدمين. - قابلية النظام للتطوير وإضافة خدمات جديدة مستقبلاً. - المساهمة في دعم التحول الرقمي داخل المؤسسات.	- محدودية الموارد المالية والتقنية في المراحل الأولى من المشروع. - حاجة المستخدمين إلى التكيف مع النظام الرقمي الجديد. - الحاجة إلى تطوير مستمر للنظام لمواكبة التطورات التكنولوجية. - اعتماد نجاح المشروع على مدى قبول المؤسسات باستخدام المنصة.
الفرص (Opportunities)	**التهديدات (Threats)**
- تزايد الاهتمام بريادة الأعمال وتشجيع إنشاء المؤسسات الناشئة. - دعم الحكومة لمشاريع التحول الرقمي والخدمات الإلكترونية. - إمكانية التعاون مع الهيئات الداعمة للمشاريع مثل برامج تمويل الشباب. - زيادة استخدام الإنترنت والتكنولوجيا الرقمية في المجتمع. - إمكانية توسيع المنصة لتشمل خدمات أخرى مرتبطة بريادة الأعمال.	- ظهور منصات أو أنظمة رقمية منافسة تقدم خدمات مشابهة. - التغيرات في القوانين أو السياسات المتعلقة بإدارة البيانات الرقمية. - مخاطر الأمن السيبراني ومحاولات اختراق الأنظمة الرقمية. - مقاومة بعض المؤسسات أو المستخدمين للتحول من النظام التقليدي إلى النظام الرقمي.

جدول 1.4: تحليل SWOT للمشروع

المصدر : من إعداد الطالب

لكن سيتم إعتماد بعض الطرق للتغلب على نقاط الضعف و تجنب التهديدات المحتملة :

التغلب على نقاط الضعف	تجنب التهديدات المحتملة
- البحث عن دعم مالي من برامج دعم المؤسسات الناشئة أو التعاون مع حاضنات الأعمال لتوفير الموارد اللازمة لتطوير المشروع	- التركيز على تطوير ميزات مبتكرة وتقديم خدمات ذات جودة عالية لضمان التميز في السوق
- تنظيم شروحات مبسطة وفيديوهات توضيحية تشرح كيفية استخدام المنصة وتسهيل عملية التعامل معها	- اعتماد تقنيات حماية متقدمة مثل التشفير وإدارة صلاحيات المستخدمين لحماية المعلومات
- العمل على تحديث المنصة بشكل دوري وإضافة تحسينات جديدة لمواكبة التطورات التكنولوجية	-الالتزام بالتشريعات والقوانين الخاصة بحماية البيانات والتعاملات الرقمية
التواصل مع المؤسسات والهيئات المعنية وتقديم عروض تجريبية لإثبات فعالية النظام	- نشر الوعي بأهمية الخدمات الرقمية وإبراز الفوائد التي توفرها المنصة للمستخدمين

جدول 2.4: التغلب على نقاط الضعف و تجنب التهديدات

المصدر : من إعداد الطالب

4.4 المزيج التسويقي

يُعد المزيج التسويقي من أهم الأدوات المستخدمة في دراسة السوق، حيث يساعد على تحديد الطريقة المناسبة لتقديم المنتج أو الخدمة للعملاء. ويتكون المزيج التسويقي من أربعة عناصر أساسية تعرف بـ **4P** وهي: المنتج، السعر، المكان، والترويج.

وفيما يلي تحليل عناصر المزيج التسويقي بالنسبة لمشروع المنصة الرقمية لإدارة ملفات دعم الشباب المقاول:

1. المنتج (Product)

يتمثل المنتج في منصة رقمية عبر الويب تسمح بإدارة ملفات طلبات الدعم الموجهة للشباب المقاول بطريقة إلكترونية. توفر المنصة مجموعة من الوظائف الأساسية مثل:

- إنشاء حسابات للمستخدمين.

- إيداع ملفات طلب الدعم بشكل إلكتروني.
- تحميل وإدارة الوثائق المطلوبة.
- متابعة حالة الطلب في مختلف مراحل المعالجة.
- لوحة تحكم مخصصة للإدارة لمراجعة الملفات وإدارة الطلبات.

وتتميز المنصة بسهولة الاستخدام وإمكانية الوصول إليها عبر الإنترنت، مما يسهل على المستخدمين التعامل معها في أي وقت ومن أي مكان.

2. **السعر (Price)**

تعتمد استراتيجية التسعير في المشروع على تقديم الخدمة بطريقة مناسبة للمؤسسات والجهات المستفيدة. ويمكن أن تكون المنصة متاحة:

- من خلال اشتراك سنوي للمؤسسات التي ترغب في استخدام النظام.
- أو من خلال نموذج خدمة رقمية يتم تقديمها للهيئات الداعمة للمشاريع.
- كما يمكن توفير نسخة تجريبية من المنصة في البداية من أجل تشجيع المؤسسات على تجربتها قبل اعتمادها بشكل كامل.

3. **المكان (Place)**

يتم تقديم الخدمة عبر منصة إلكترونية على شبكة الإنترنت، مما يسمح للمستخدمين بالوصول إليها بسهولة من مختلف المناطق دون الحاجة إلى التنقل إلى المؤسسات الإدارية.

كما يمكن للمؤسسات استخدام المنصة من خلال أجهزة الحاسوب أو الأجهزة الذكية، مما يسهل عملية إدارة الملفات ومتابعة الطلبات.

4. **الترويج (Promotion)**

يعتمد الترويج للمشروع على مجموعة من الوسائل التي تهدف إلى التعريف بالمنصة وجذب المستخدمين، ومن أهمها:

- استخدام وسائل التواصل الاجتماعي للتعريف بالخدمة.
- المشاركة في الفعاليات والملتقيات المتعلقة بريادة الأعمال والتكنولوجيا.
- التعاون مع الجامعات والمؤسسات الداعمة للمشاريع الناشئة.
- إعداد عروض توضيحية وفيديوهات تشرح كيفية استخدام المنصة وفوائدها.

5.4 الإستراتيجيات التسويقية

من أجل ضمان نجاح مشروع المنصة الرقمية لإدارة ملفات دعم الشباب المقاول وتحقيق ميزة تنافسية في السوق، من الضروري اعتماد مجموعة من الاستراتيجيات التسويقية التي تساعد على التعريف بالمنصة وجذب المستخدمين وتشجيع المؤسسات على استخدامها.

وفيما يلي أهم الاستراتيجيات التسويقية المقترحة للمشروع:

1. **استراتيجية التميز (Strategy Dffierentiation)**

تعتمد هذه الاستراتيجية على تقديم خدمة متميزة تختلف عن الحلول التقليدية، وذلك من خلال توفير منصة رقمية حديثة تسهل عملية إدارة ملفات طلبات الدعم بطريقة أكثر سرعة وشفافية.

كما يتميز المشروع بتقديم واجهة استخدام سهلة ومرنة تسمح للمستخدمين بالتعامل مع النظام بسهولة، إضافة إلى توفير أدوات تساعد على متابعة الطلبات وإدارة الوثائق بشكل فعال.

2. **استراتيجية التسويق الرقمي (Strategy Marketing Digital)**

يتم الترويج للمنصة عبر الإنترنت باستخدام وسائل التسويق الرقمي المختلفة، مثل:

- إنشاء صفحات خاصة بالمشروع على منصات التواصل الاجتماعي.
- نشر محتوى تعريفي حول المنصة وخدماتها.
- استخدام المواقع الإلكترونية والمنتديات التقنية للتعريف بالمشروع.
- نشر فيديوهات توضيحية تشرح كيفية استخدام المنصة وفوائدها.

3. **استراتيجية التسويق بالمحتوى (Strategy Marketing Content)**

تعتمد هذه الاستراتيجية على نشر محتوى مفيد يشرح أهمية التحول الرقمي في إدارة الخدمات الإدارية، ويبرز دور المنصة في تسهيل إجراءات طلب الدعم.

يمكن أن يشمل هذا المحتوى مقالات، شروحات، أو دراسات بسيطة حول كيفية تحسين إدارة الملفات الإدارية باستخدام الحلول الرقمية.

4. **استراتيجية الشراكات (Strategy Partnership)**

يمكن للمشروع تحقيق انتشار أكبر من خلال بناء شراكات مع جهات مختلفة مثل:

- المؤسسات الداعمة للمشاريع الناشئة.

- الجامعات ومراكز البحث.
- حاضنات الأعمال وبرامج دعم ريادة الأعمال.

وتساعد هذه الشراكات على توسيع نطاق استخدام المنصة والوصول إلى عدد أكبر من المستخدمين.

5. **استراتيجية التفاعل مع المستخدمين (Strategy Marketing Interactive)**

تعتمد هذه الاستراتيجية على التفاعل المباشر مع المستخدمين من خلال جمع آرائهم وملاحظاتهم حول المنصة، والعمل على تحسين الخدمات بناءً على احتياجاتهم.

كما يمكن توفير دعم تقني للمستخدمين والإجابة عن استفساراتهم لضمان تجربة استخدام جيدة وتعزيز الثقة في النظام.

المحور 5

خطة الإنتاج و التسويق

تهدف خطة الإنتاج والتنظيم إلى توضيح الكيفية التي سيتم من خلالها تطوير وتنفيذ مشروع المنصة الرقمية لإدارة ملفات دعم الشباب المقاول، وذلك من خلال تحديد الموارد اللازمة، مراحل العمل، وآلية تنظيم الفريق لضمان إنجاز المشروع بطريقة فعالة.

كما تسمح هذه الخطة بتحديد مختلف العناصر التقنية والبشرية الضرورية لتطوير المنصة وتشغيلها بشكل مناسب.

1.5 خطة الإنتاج

تتمثل خطة الإنتاج في مجموعة من المراحل التي يتم اتباعها من أجل تصميم وتطوير المنصة الرقمية وتنفيذ وظائفها الأساسية. وتشمل هذه المراحل تحديد موقع المشروع، توفير الموارد التقنية، وتنظيم عملية تطوير النظام.

1.1.5 موقع المشروع

يتم تطوير المشروع في بيئة عمل رقمية تعتمد أساساً على استخدام الحواسيب وشبكة الإنترنت، حيث يتم تصميم وتطوير المنصة باستخدام أدوات البرمجة الحديثة.

كما يمكن اعتبار موقع المشروع افتراضياً من خلال منصة إلكترونية يتم من خلالها نشر الخدمة وتمكين المستخدمين من الوصول إليها عبر الإنترنت.

2.1.5 احتياجات الإنتاج

يتطلب تطوير المنصة مجموعة من الموارد التقنية والبشرية التي تساهم في تنفيذ المشروع، ومن أهمها:

الموارد التقنية:

- حواسيب مخصصة لتطوير البرمجيات.
- أدوات وبرامج تطوير تطبيقات الويب.
- نظام إدارة قواعد البيانات لتخزين معلومات المستخدمين والملفات.
- خدمات استضافة إلكترونية لنشر المنصة على الإنترنت.
- أدوات اختبار البرمجيات لضمان جودة النظام.

الموارد البشرية:

- مطورو برمجيات مسؤولون عن تطوير المنصة.

- مصمم واجهات مستخدم لتحسين تجربة الاستخدام.
- مسؤول عن إدارة قواعد البيانات.
- فريق اختبار للتأكد من كفاءة النظام قبل إطلاقه.

3.1.5 مراحل تطوير المنصة

تمر عملية تطوير المنصة بعدة مراحل أساسية، وهي:

- **مرحلة تحليل المتطلبات:** تحديد احتياجات المستخدمين والوظائف الأساسية للنظام.
- **مرحلة تصميم النظام:** تصميم بنية المنصة وقاعدة البيانات وواجهة المستخدم.
- **مرحلة التطوير البرمجي:** برمجة مختلف وظائف النظام وتنفيذ خصائص المنصة.
- **مرحلة الاختبار:** اختبار النظام والتأكد من خلوه من الأخطاء التقنية.
- **مرحلة الإطلاق التجريبي:** تشغيل المنصة بشكل تجريبي قبل إطلاقها بشكل كامل.

2.5 الخطة التنظيمية

تهدف الخطة التنظيمية إلى تحديد كيفية تنظيم العمل داخل المشروع وتوزيع المهام بين أعضاء الفريق من أجل ضمان تنفيذ مختلف مراحل تطوير المنصة بطريقة فعالة ومنظمة.
كما تساعد هذه الخطة على تحديد المسؤوليات داخل الفريق وتوضيح الأدوار التي يقوم بها كل عضو، مما يساهم في تحسين التنسيق والتعاون بين أعضاء المشروع.

1.2.5 الهيكل التنظيمي للمشروع

يعتمد المشروع على هيكل تنظيمي بسيط يضم مجموعة من الأدوار الأساسية التي تساهم في تطوير وتشغيل المنصة الرقمية. ويمكن تقسيم هذه الأدوار إلى عدة مهام رئيسية، منها:

- **مدير المشروع:** مسؤول عن الإشراف العام على المشروع وتنظيم مراحل العمل وضمان تحقيق الأهداف المحددة.
- **مطورو البرمجيات:** يتولون مهمة تصميم وبرمجة مختلف وظائف المنصة وتطوير النظام.

- **مصمم واجهة المستخدم:** مسؤول عن تصميم واجهة المنصة بطريقة تسهل استخدامها من قبل المستخدمين.
- **مسؤول قاعدة البيانات:** يتولى إدارة البيانات وتنظيمها وضمان تخزينها بشكل آمن داخل النظام.
- **مسؤول الاختبار:** يقوم باختبار النظام والتأكد من سلامة وظائفه قبل إطلاقه للمستخدمين.

2.2.5 توزيع المهام داخل الفريق

يتم توزيع المهام بين أعضاء الفريق بطريقة تضمن التكامل بين مختلف الجوانب التقنية للمشروع، حيث يعمل كل عضو على إنجاز المهام المرتبطة بمجال تخصصه.
كما يتم التنسيق بين أعضاء الفريق من خلال اجتماعات دورية لمتابعة تقدم العمل، مناقشة التحديات التقنية، واقتراح الحلول المناسبة لتحسين أداء المنصة.

3.2.5 إدارة المشروع

تعتمد إدارة المشروع على التخطيط الجيد لمختلف مراحل العمل، حيث يتم تحديد جدول زمني لتطوير المنصة يشمل مراحل التحليل والتصميم والتطوير والاختبار.
كما يتم استخدام أدوات إدارة المشاريع لمتابعة تقدم العمل وتنظيم المهام بين أعضاء الفريق، مما يساعد على ضمان تنفيذ المشروع في الوقت المحدد وبالجودة المطلوبة.

المحور 6

الخطة المالية

تُعد الخطة المالية من العناصر الأساسية في دراسة أي مشروع، حيث تسمح بتقدير التكاليف اللازمة لتنفيذ المشروع وتحديد مصادر التمويل الممكنة. كما تساعد هذه الخطة على تقييم مدى جدوى المشروع من الناحية الاقتصادية. وفي هذا المحور سيتم عرض أهم التكاليف المرتبطة بتطوير وتشغيل المنصة الرقمية لإدارة ملفات دعم الشباب المقاول، بالإضافة إلى توضيح الهيكل التمويلي للمشروع.

1.6 تكاليف المشروع

تشمل تكاليف المشروع نوعين رئيسيين من التكاليف، وهما التكاليف الاستثمارية والتكاليف التشغيلية.

1.1.6 التكاليف الاستثمارية

تمثل التكاليف الاستثمارية المصاريف الأولية اللازمة لبدء المشروع وتطوير المنصة الرقمية. وتشمل هذه التكاليف شراء المعدات والبرمجيات الضرورية لتطوير النظام.

العنصر	التكلفة التقديرية
حواسيب لتطوير البرمجيات	300,000 دج
معدات مكتبية	50,000 دج
خدمات استضافة المنصة	100,000 دج
أدوات وبرامج تطوير البرمجيات	50,000 دج
مصاريف إعداد المشروع	100,000 دج
المجموع التقريبي	**600,000 دج**

جدول 1.6: التكاليف الاستثمارية للمشروع

المصدر: إعداد الطالب

2.1.6 التكاليف التشغيلية

تشمل التكاليف التشغيلية المصاريف المرتبطة بتشغيل المشروع بعد إطلاقه، مثل تكاليف الإنترنت والصيانة والتسويق.

العنصر	التكلفة التقديرية
خدمات الإنترنت	10,000 دج
صيانة وتطوير المنصة	50,000 دج
مصاريف التسويق والإشهار	30,000 دج
خدمات الاستضافة	20,000 دج
المجموع التقريبي	**110,000 دج**

جدول 2.6: التكاليف التشغيلية للمشروع

المصدر: إعداد الطالب

2.6 الهيكل التمويلي للمشروع

يمثل الهيكل التمويلي الطريقة التي يتم من خلالها تمويل المشروع، سواء من خلال التمويل الذاتي أو من خلال القروض أو برامج دعم المشاريع.

مصدر التمويل	النسبة	القيمة
التمويل الذاتي	30%	180,000 دج
تمويل خارجي أو دعم حكومي	70%	420,000 دج
المجموع	**100%**	**600,000 دج**

جدول 3.6: الهيكل التمويلي للمشروع

المصدر: إعداد الطالب

المحور 7

النموذج الأولي التجريبي

تم تطوير نموذج أولي تجريبي وظيفي كامل لمنصة **GRANTA** يُمكِّن من اختبار جميع الوظائف الأساسية للنظام في بيئة قريبة من الإنتاج. و قد تم بناء هذا النموذج بشكل نمطي لضمان قابلية التوسع و الصيانة.

1.7 المعمارية التقنية للنظام

تعتمد منصة GRANTA على معمارية ثلاثية الطبقات 3-Tier) (Architecture موسّعة بطبقة بلوكشين، تتضمّن:

1. **طبقة العرض (Presentation Layer)**: تطبيق ويب مبني على Next.js 15 و React 19 و TypeScript، مع تصميم مستجيب (Responsive) باستخدام Tailwind CSS.
2. **طبقة المنطق التطبيقي (Application Layer)**: واجهات برمجية API REST مبنية على Next.js API Routes، تتولى التحقق من الصلاحيات و معالجة المنطق التشغيلي.
3. **طبقة قاعدة البيانات (Data Layer)**: قاعدة SQLite للبيانات التشغيلية (المستخدمين، حالة الملفات، السجلّات) مع ORM Prisma.
4. **طبقة التخزين اللامركزي (IPFS Layer)**: خادم IPFS لتخزين الوثائق الفعلية و الشهادات الرقمية.
5. **طبقة البلوكشين (Blockchain Layer)**: شبكة Hyperledger Fabric 2.5 مع خوارزمية التوافق Raft و عقود ذكية (Chaincode) مكتوبة بـ Go و JavaScript.

الطبقة	التقنيات المستخدمة
العرض	CSS, Tailwind TypeScript, ,19 React ,15 Next.js shadcn/ui
المنطق	Zod NextAuth, Routes, API Next.js
قاعدة البيانات	ORM Prisma SQLite,
التخزين اللامركزي	(go-ipfs) IPFS
البلوكشين	Chaincode Raft, ,5.2 Fabric Hyperledger (Go/JS)
الاختبار و الأداء	Jest Caliper, Hyperledger

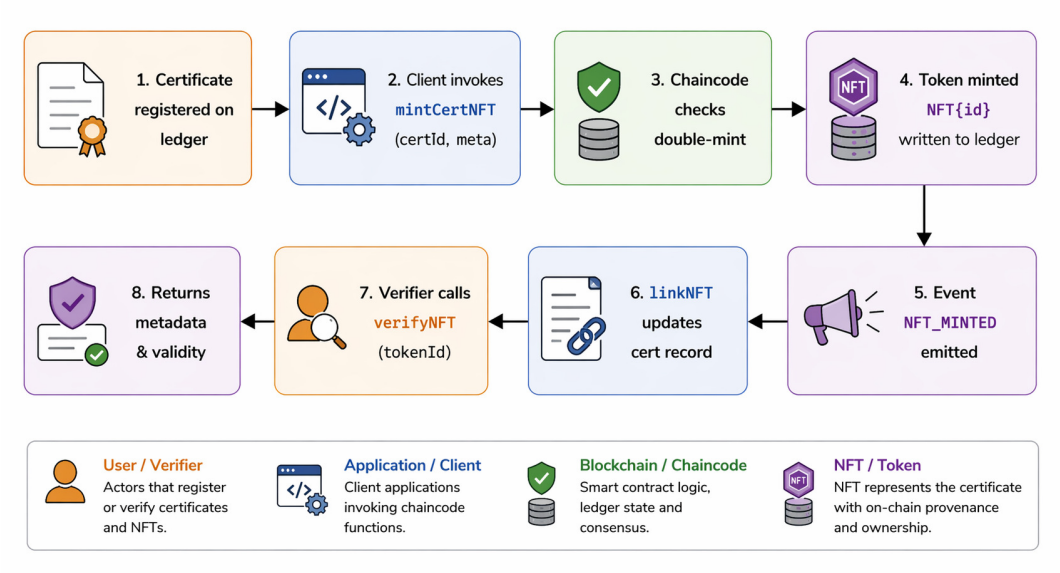
جدول 1.7: التقنيات المستخدمة في تطوير منصة GRANTA

المصدر : من إعداد الطالب

2.7 تسلسل العمل (Worfklow) داخل المنصة

تتضمّن المنصة سير عمل متكامل يضم خمسة أدوار رئيسية:

- **المترشح (Candidate)**: ينشئ حساباً، يرفع ملف طلب الدعم.
- **المشرف الأولي (Primary Supervisor)**: يفحص الملف و يقدّم ملاحظات أو يوافق.
- **المشرف النهائي (Final Supervisor)**: يصادق نهائياً على الملف و يطلق إصدار الشهادة.
- **الإداري (Admin)**: يدير المستخدمين و يراقب الأداء و سجلّات Fabric.
- **المتحقِّق العمومي (Public Verfiier)**: أي طرف ثالث يستطيع التحقق من شهادة عبر رمز QR أو رابط عمومي.



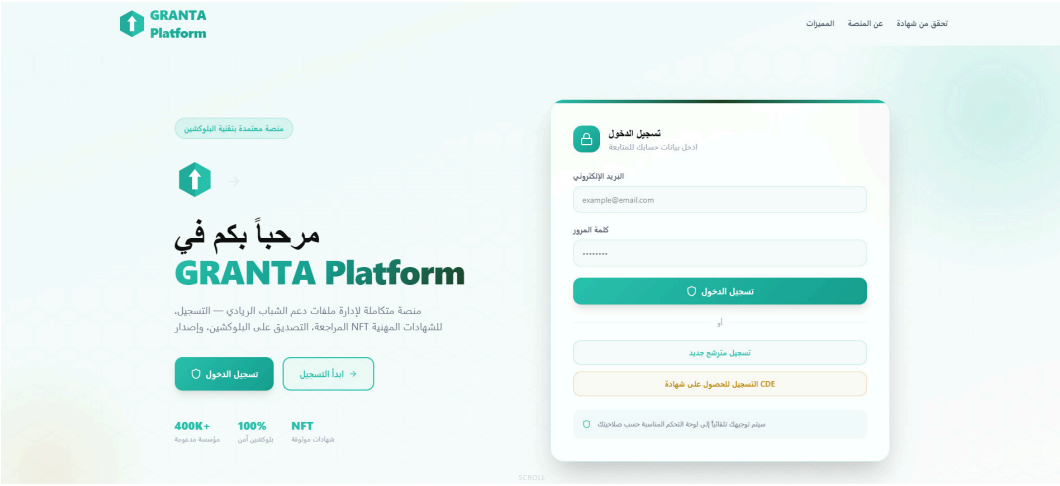
شكل 1.7: مخطط تسلسل العمل لإصدار شهادة NFT على Fabric Hyperledger

المصدر : من إعداد الطالب

3.7 لقطات من المنصة (Screenshots)

تعرض الفقرات التالية لقطات حقيقية من واجهات المنصة المُنفّذة، توضّح مختلف الأدوار و سير العمل.

1.3.7 الصفحة الرئيسية و التوجيه

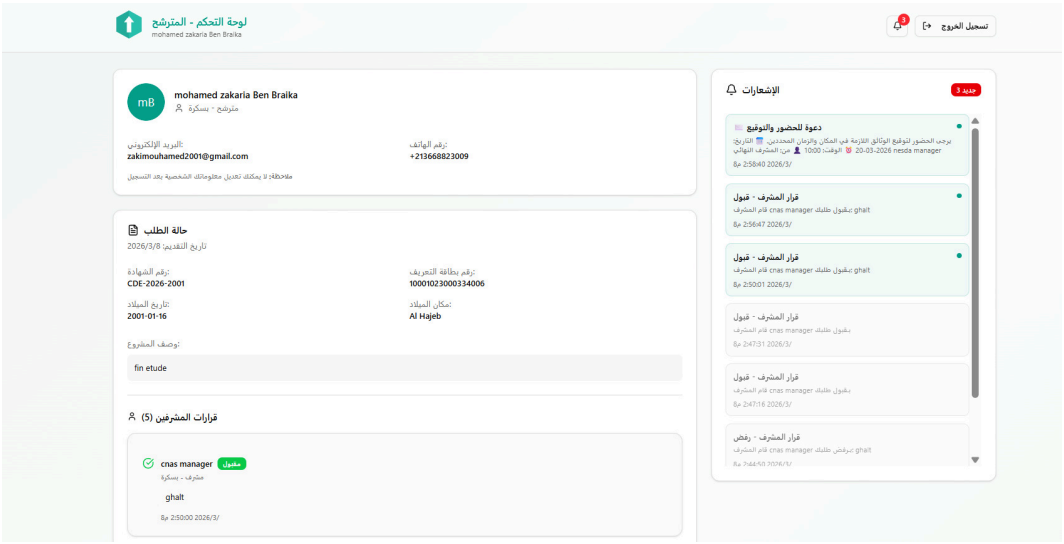


شكل 2.7: الصفحة الرئيسية للمنصة GRANTA

المصدر : من إعداد الطالب

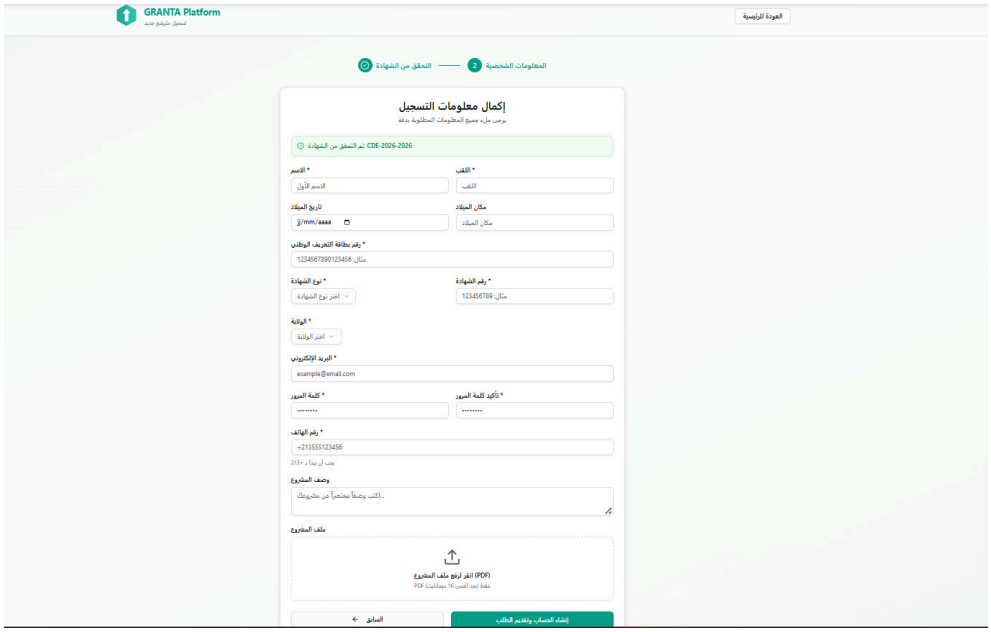
2.3.7 واجهة المترشح (Candidate) Dashboard

تتيح للمترشح إنشاء طلب جديد، رفع الوثائق المطلوبة، و متابعة حالة كل ملف بشكل لحظي.



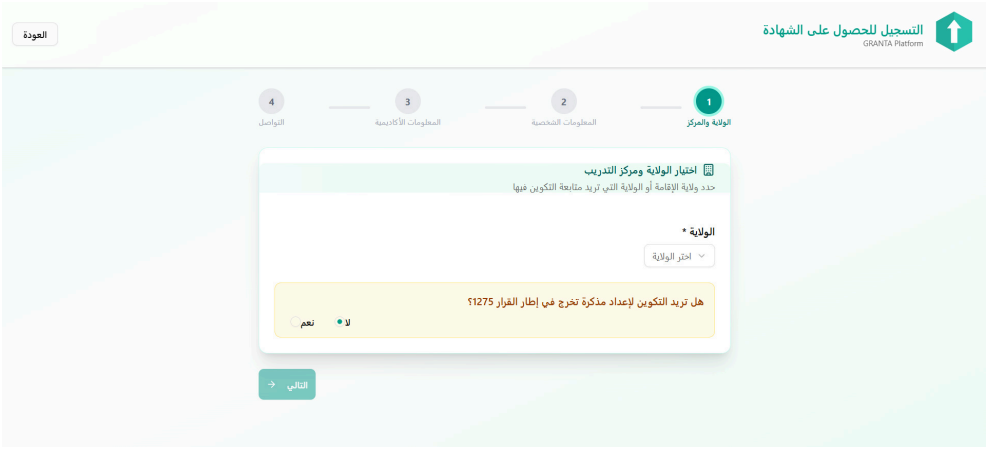
شكل 3.7: لوحة تحكم المترشح

المصدر : من إعداد الطالب



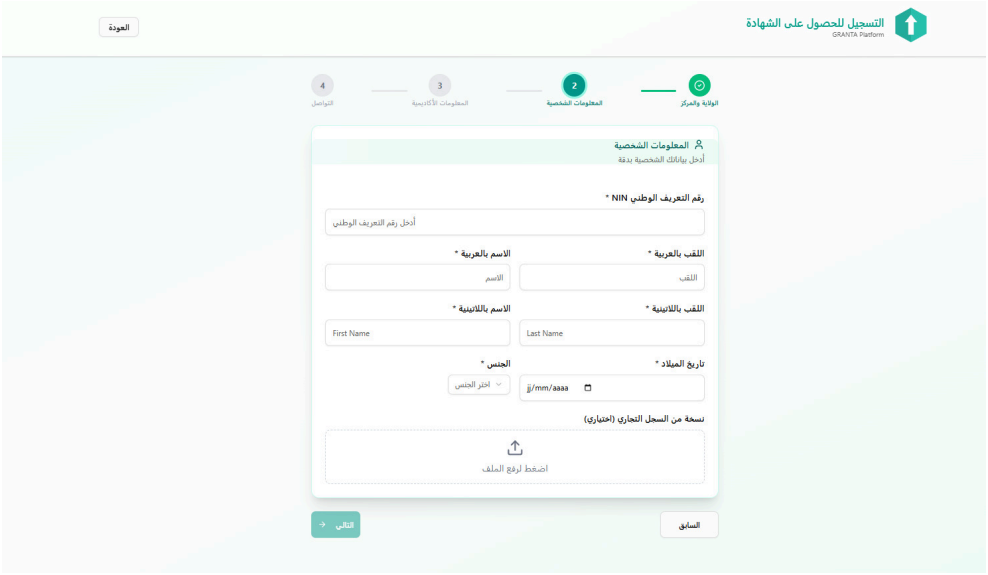
شكل 4.7: نموذج تسجيل ملف طلب الدعم

المصدر : من إعداد الطالب



شكل 5.7: نموذج تسجيل CDE (الخطوة 1)

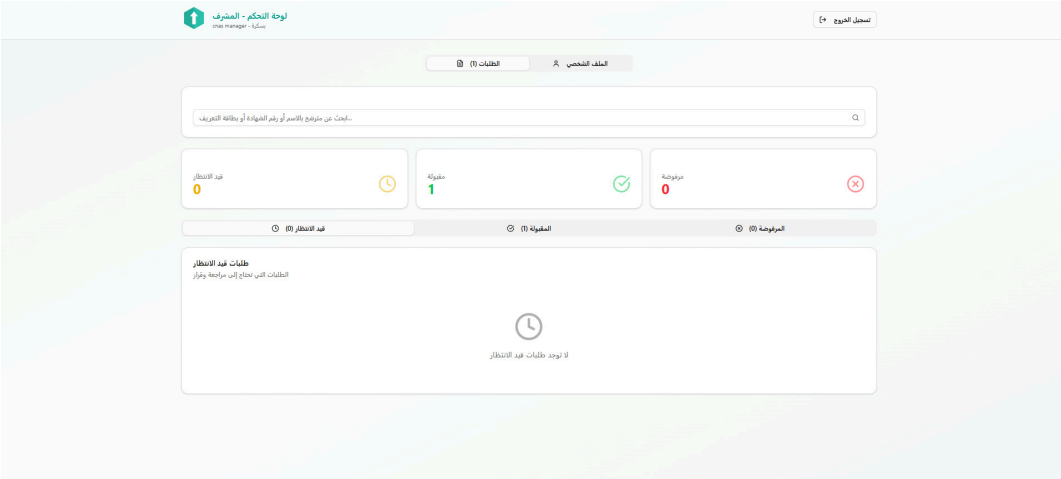
المصدر : من إعداد الطالب



شكل 6.7: نموذج تسجيل CDE (الخطوة 2)

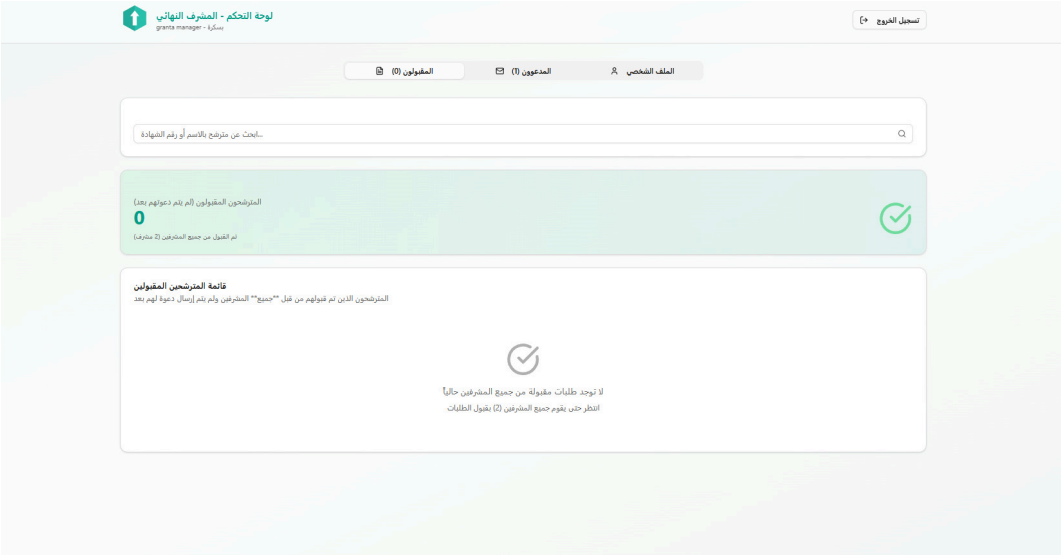
المصدر : من إعداد الطالب

3.3.7 واجهة المشرف الأولي و النهائي



شكل 7.7: لوحة تحكم المشرف الأولي - مراجعة الملفات

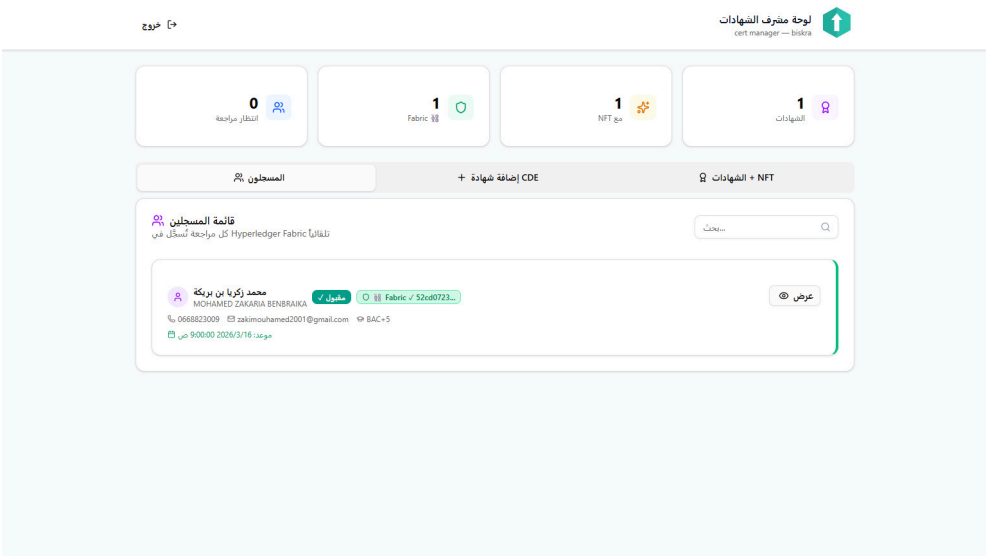
المصدر : من إعداد الطالب



شكل 8.7: لوحة تحكم المشرف النهائي - إرسال دعوة

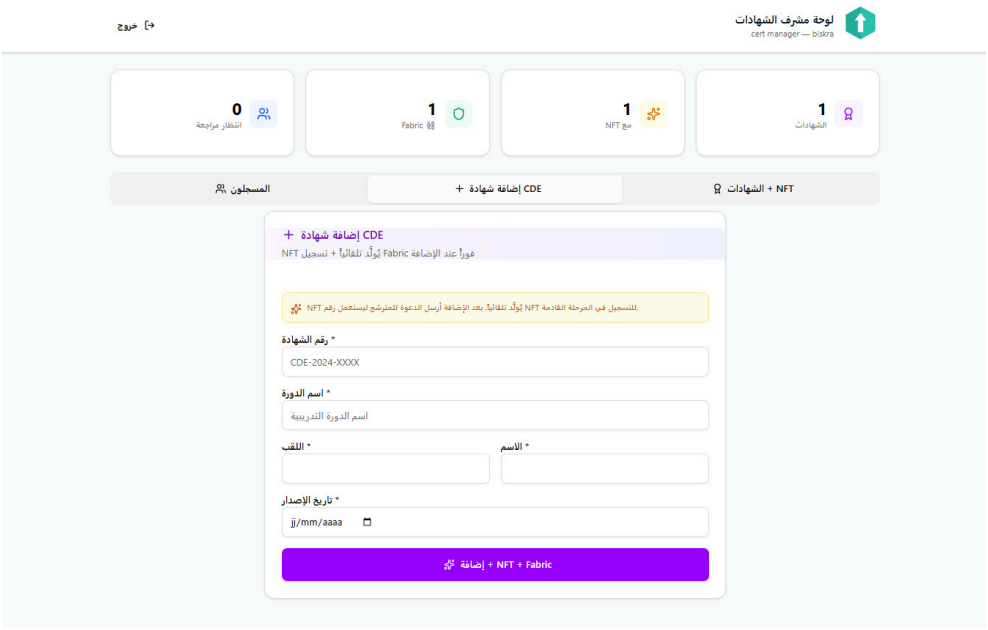
المصدر : من إعداد الطالب

4.3.7 إصدار شهادة NFT و تخزينها على البلوكشين



شكل 9.7: واجهة مشرف -CDE االرسال قبول المترشح لحضور الدورة

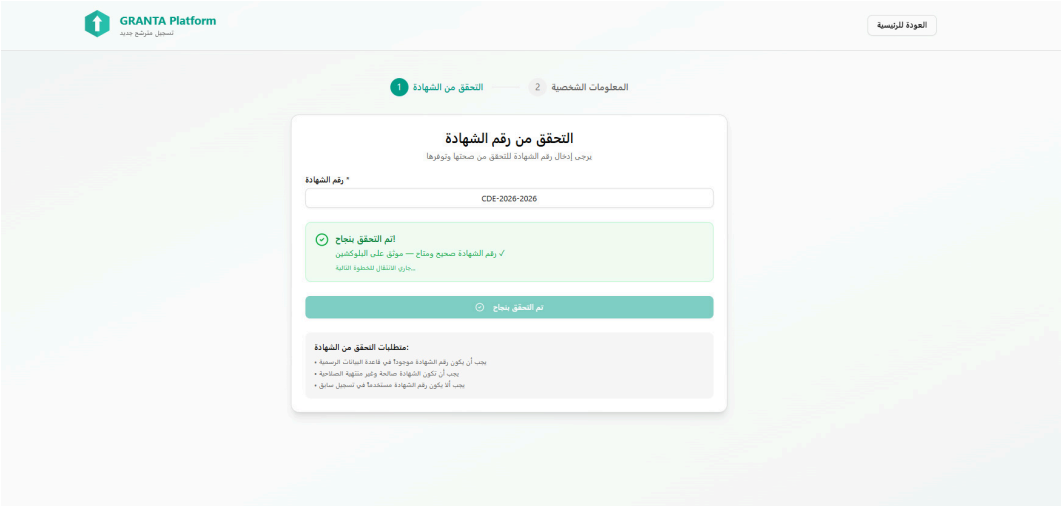
المصدر : من إعداد الطالب



شكل 10.7: واجهة مشرف -CDE االصدار شهادات

المصدر : من إعداد الطالب

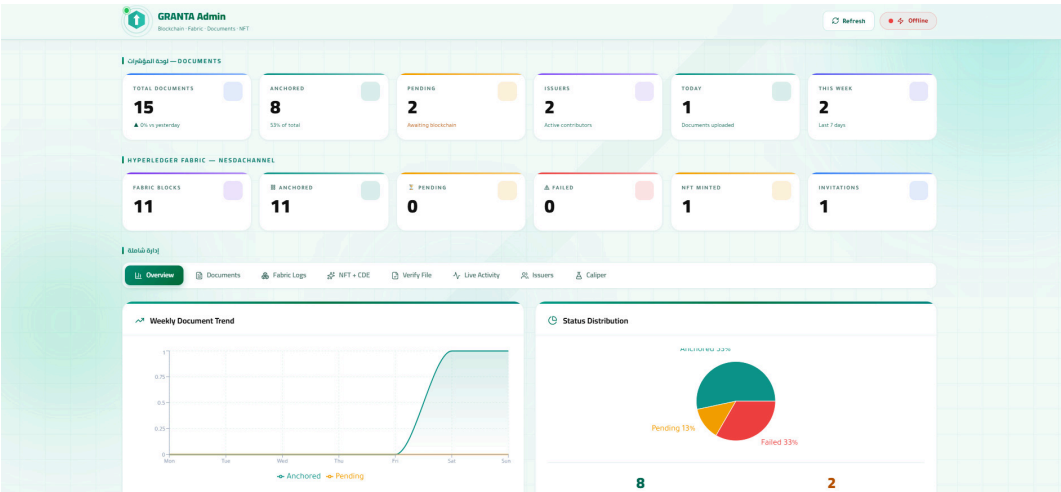
5.3.7 التحقق العمومي من الشهادة



شكل 11.7: صفحة التحقق العمومي من أصالة الشهادة قبل نموذج تسجيل ملف طلب الدعم

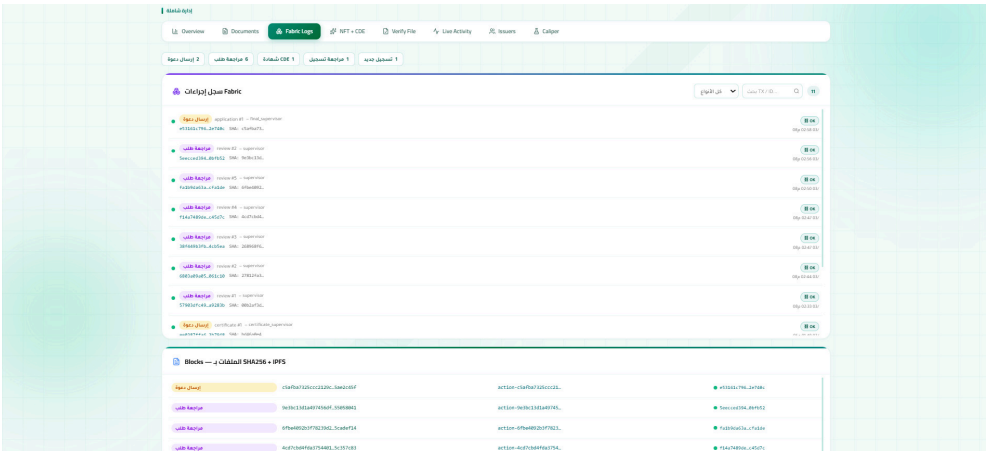
المصدر : من إعداد الطالب

6.3.7 اللوحة الإدارية و المراقبة



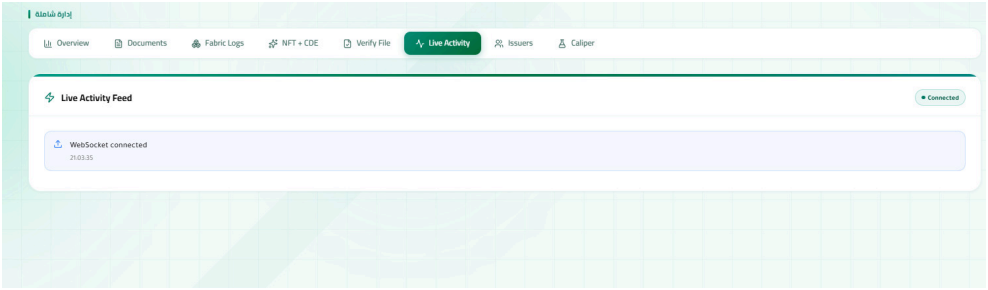
شكل 12.7: اللوحة الإدارية - نظرة عامة

المصدر : من إعداد الطالب



شكل 13.7: مراقبة سجلات Fabric Hyperledger من اللوحة الإدارية

المصدر : من إعداد الطالب



شكل 14.7: مراقبة النشاط اللحظي على المنصة

المصدر : من إعداد الطالب

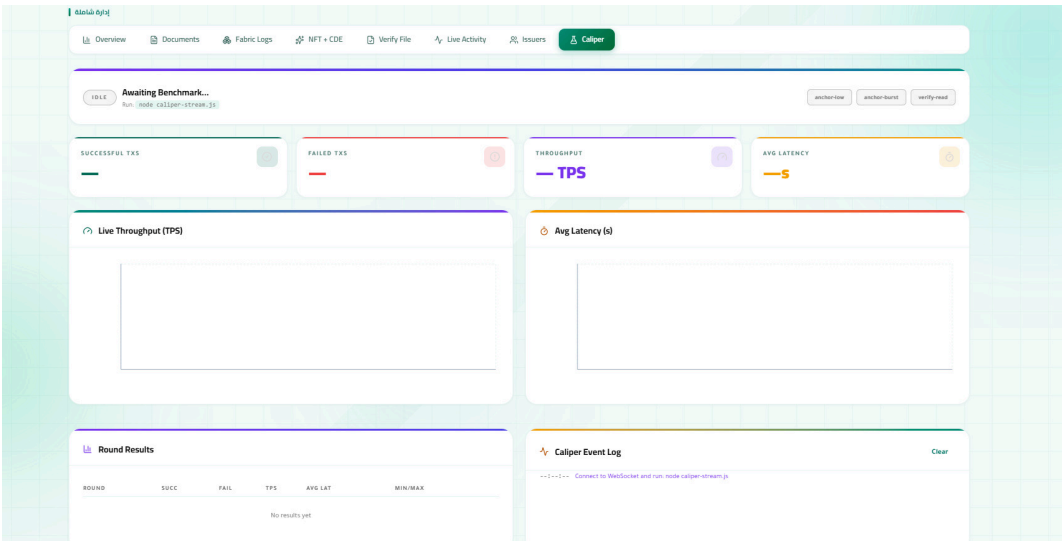
4.7 قياس الأداء (Performance Benchmarking)

تم إجراء حملة قياس أداء صارمة باستخدام Hyperledger Caliper، الأداة المرجعية لاختبار شبكات Hyperledger. النتائج المُحصَّل عليها كانت كالتالي:

المؤشر	القيمة
الإنتاجية القصوى (Throughput)	TPS 228
نسبة فشل المعاملات	0%
متوسط زمن الاستجابة (Latency)	عالي الكفاءة
خوارزمية التوافق	Raft
عدد العقد (Nodes) في الشبكة	+3

جدول 2.7: مؤشرات أداء منصة GRANTA المُقاسة بـ Caliper Hyperledger

المصدر : من إعداد الطالب



شكل 15.7: لوحة قياس الأداء بـ Caliper Hyperledger

المصدر : من إعداد الطالب

تُظهر النتائج أنّ المنصة قادرة على تلبية أعباء العمل المؤسسية لتسجيل الشهادات بشكل موثوق و كفاءة عالية، ما يؤكد جاهزيتها للنشر التشغيلي.

5.7 خارطة طريق التطوير (Roadmap)

المرحلة	المدة	الأهداف
المرحلة 1	0 - 3 أشهر	تشغيل تجريبي مع شريك أولي NESDA) أو حاضنة جامعية) - معالجة 100 ملف.
المرحلة 2	3 - 9 أشهر	توقيع عقود مع CNAC و ANGEM - معالجة 1000+ ملف.
المرحلة 3	9 - 18 شهر	تكامل API مع البنوك للتحقق - دعم تكوينات الجامعات.
المرحلة 4	18 - 36 شهر	توسع على مستوى وطني - دعم لغات إضافية - تكامل مع منصات أوروبية.

جدول 3.7: خارطة طريق تطوير و توسع منصة GRANTA

المصدر : من إعداد الطالب

خاتمة عامة

في خاتمة هذا التقرير، نستعرض أهم النتائج المُحصَّل عليها من مشروع **GRANTA**، الذي قُدِّم في إطار القرار الوزاري 1275 المُتعلق بالمؤسسات الناشئة. لقد تمّ تصميم و تطوير منصة رقمية متكاملة لإدارة و توثيق ملفات دعم رواد الأعمال الشباب في الجزائر، بالاعتماد على تقنيات حديثة تشمل Hyperledger، Fabric، IPFS و الشهادات الرقمية على شكل NFT.

و تمثلت أهمّ المخرجات في:

- **معالجة فجوة حقيقية** في منظومة الدعم الجزائرية المُتمثّلة في غياب آلية موثوقة للتحقق من أصالة الوثائق.
- **تقديم نموذج أعمال متكامل (BMC)** يحدّد مسارات الإيرادات و الشركاء و القيمة المُقترحة.
- **تطوير نموذج أوّلي تشغيلي** كامل تم اختباره عبر Caliper Hyperledger مع إنتاجية تصل إلى 228 TPS و معدل فشل صفري.
- **بناء استراتيجية تسويقية و تنظيمية** متناسقة مع الإطار القانوني الجزائري.
- **تأسيس هيكل مالي** واقعي يأخذ بعين الاعتبار التمويل الذاتي و الدعم الحكومي.

التوصيات المُقترحة:

1. توقيع اتفاقية شراكة أولى مع إحدى الهيئات الداعمة الكبرى (NESDA أو ANGEM) لإطلاق مرحلة تجريبية فعلية.
2. السعي للحصول على شهادة المؤسسة الناشئة لفتح مسار التمويل العمومي و الإعفاءات الضريبية.
3. استكمال تدقيق أمني خارجي للعقود الذكية قبل النشر الواسع.
4. توسيع الفريق ليشمل مسؤول تطوير أعمال (Business Developer) و مسؤول قانوني.
5. العمل على مطابقة المنصة لمعايير حماية البيانات الجزائرية و الأوروبية (GDPR) لاستقطاب شراكات دولية.

يُمثّل مشروع **GRANTA** مساهمة عملية في دفع التحوّل الرقمي للإدارة الجزائرية، و خطوةً نحو بناء منظومة دعم أكثر شفافيةً و موثوقيةً لرواد الأعمال الشباب. و يبقى المشروع مفتوحاً للتطوير و التوسع وفق احتياجات السوق و تطوّر المنظومة التشريعية.

GRANTA نموذج الأعمال التجاري — مشروع

شرائح العملاء

- حاملو أفكار ومشاريع
- مؤسسات مصغرة وناشئة
- خريجو الجامعات ومراكز التكوين
- شباب بطالين
- هيئات داعمة عمومية

علاقات العملاء

- استشارات ادارية وتقنية
- مرافقة وتكوين
- دعم تقني مباشر
- متابعة دورية

القنوات

- وكالات محلية
- منصات التواصل الاجتماعي
- أيام تحسيسية
- شراكات مع الجامعات
- الموقع الإلكتروني
- ملتقيات وفعاليات

القيمة المقترحة

- تمويل
- مرافقة
- محاربة البطالة
- دعم الإدماج المهني
- تسهيل الإجراءات الإدارية
- توثيق رقمي موثوق

الأنشطة الرئيسية

- دراسة المشاريع
- تمويل المشاريع
- تنسيق مع البنوك
- متابعة المشاريع
- التطوير والصيانة
- التسويق والشراكات

الموارد الرئيسية

- تمويل
- موارد بشرية
- مكاتب وبنية تحتية
- مرافقة
- تنظيمات واطار قانوني

الشراكات الرئيسية

- وزارة العمل والتشغيل
- مراكز التكوين المهني
- الجامعات والحاضنات
- بنوك ومؤسسات مالية
- غرف الصناعة التقليدية
- غرف التجارة

مصادر الإيرادات

- تمويل حكومي
- شراكات مع بنوك
- خدمات استشارية
- رسوم الخدمات

هيكل التكاليف

- تكاليف التمويل
- تطوير وصيانة
- تسويق
- أجور الموظفين
- تكاليف التكوين
- مصاريف التسيير

ملخّص نموذج الأعمال التجاري (Business Model Canvas) لمشروع GRANTA