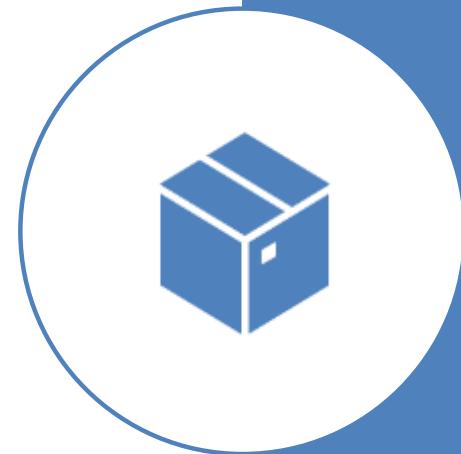


Sol — Proof of Concept (PoC) Stage Template

- Team: Block Master's | Lead: Raahul Kanna K | Members: Vighas E, Ramya S
- Mentor: Yuvedhasri S | Sol Lab Code: BW | Cohort: 2024-28



Problem & End-User Recap

- **Problem Statement :** The current academic certificate verification process is slow, error-prone, and vulnerable to fraud. Students lack a secure way to store and share their credentials, leading to manual checks that take days or weeks, loss/damage of physical copies, and a general lack of trust due to fake certificates.
- **Target End-User/Context:** Students (like Ravi Kumar, a final year engineering student) applying for internships or higher studies, who need quick, verified access to academic records and career guidance. Colleges and employers who need to instantly and reliably verify certificates. Evidence: Manual verification delays applications, risk of losing physical documents, and no single secure digital platform for credentials (from user persona frustrations). Fake certificates harm trust.
- **Success metrics for the end-user:** Time saved in verification (from days/weeks to instant), error reduced (elimination of fraud), increased trust in credentials, and improved career path alignment for students.

Selected Idea Recap & Value Proposition

- **Idea Name:** Blockchain Academic Credential System | **One-line pitch:** A blockchain-powered platform for secure, instant credential verification and AI-driven career insights.
- **Value Proposition:** Provides trustworthy, efficient credential management for students and institutions, empowering students with control and career guidance, and eliminating verification fraud/delays.
- **Key PoC Focus:** Proving secure storage in a digital wallet, basic issuance, and instant, tamper-proof verification of academic credentials using blockchain.

PoC Objectives & Success Criteria

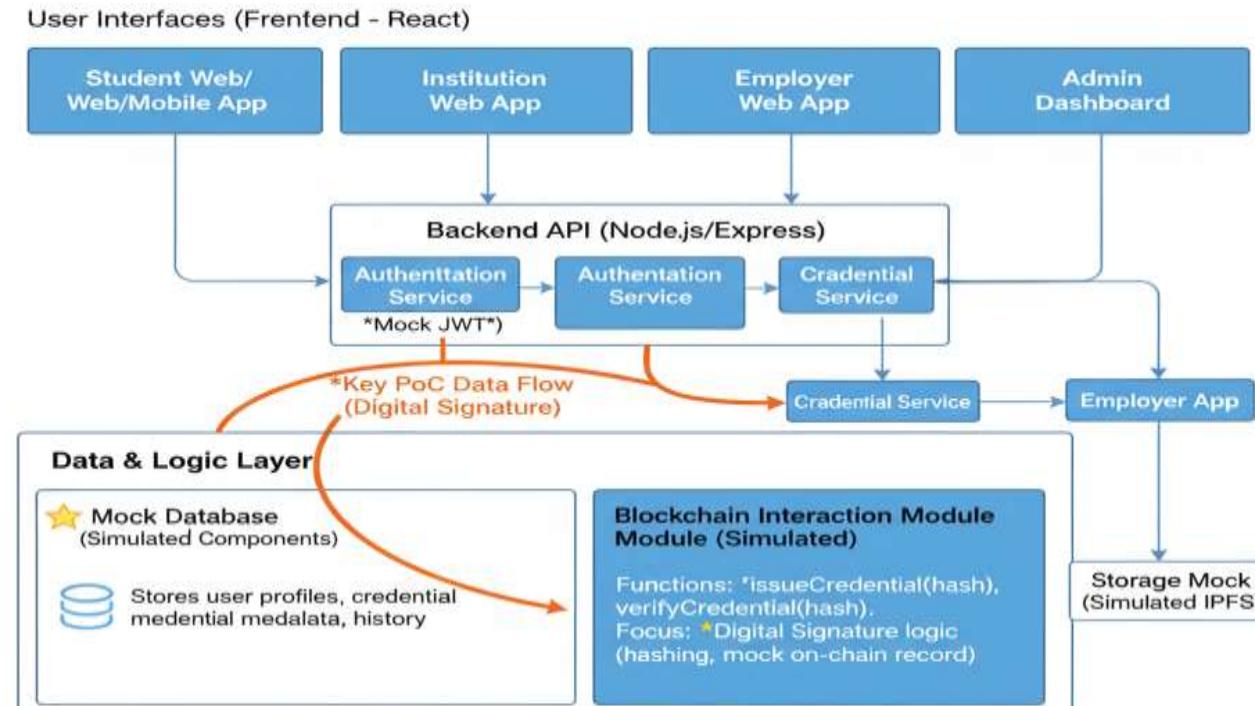
Critical Assumption / Question	Metric	Target Threshold	Test Method
Can we securely store a credential?	Credential integrity & accessibility	100% integrity, instant access	Wallet demo & blockchain verification
Can an institution issue a certificate?	Successful issuance rate	100% successful	Admin interface, test transactions
Can a third party instantly verify?	Verification speed & accuracy	<5 sec, 100% accurate	Share link, verify through API
Is the credential tamper-proof?	Immutability of stored data	No alteration possible	Attempt data modification, check blockchain logs
Is the wallet user-friendly?	Ease of use for students	Intuitive navigation	Simple UI demo, user feedback

Scope, Assumptions & Constraints

- **In Scope (PoC):** Demonstrate blockchain certificate storage, basic issuance/verification, and a student digital wallet Connection (core functionality). | Out of Scope (for now): AI career suggestions, advanced analytics, integration with job portals, advanced badge support, complex smart contracts beyond basic functions.
- **Assumptions:** Availability of blockchain testnet/frameworks, access to basic AI tools, student/institution data samples for testing, and standard web/mobile devices for end-users.
- **Constraints:** Budget/time limits (focused on PoC only), reliance on existing open-source blockchain/AI tools, and simulated institutional participation for initial testing.

System Overview — Architecture / Design

System Overview - Architecture / Design



Mocked/Simulated Components:

Authentraction Service (Mock)
Mock Database (JSIN/Squillie)
Mock Database Mashintatice (Simulated)
Storage Mock (Simulated)

Build Plan & Timeline

Phase / Week	Tasks / Milestones	Owner(s)	Status / Date
Week 1	Step 1: Requirement & Planning	Raahul Kanna K	Confirmed 15 Oct
Week 2	Step 2: Sign up and Log in(Mock/Simulated)	Vighas E	Ready 25 Oct
Week 3	Step 3: Digital Signature Lookup	Ramya S	Tested 30 Oct
Week 4	Step 4: Wallet Integration (Student Frontend)	Raahul Kanna K	Prototype 5 Nov
Week 5	Student & Institution Module Development	Team (All Members)	In Progress/Planned 12 Nov

Resources & Budget

Item / Tool	Qty / Spec	Cost (₹)	Source / Link	Notes
Development Environment	1-2 machines	0	Local setup (VS Code, Node.js)	Existing team laptops, free software
Frontend Framework	1 (React)	0	create-react-app.dev	Open-source, widely used
Backend Framework	1 (Node.js/Express)	0	expressjs.com	Open-source, easy to set up
Mock Database	1 (SQLite/JSON file)	0	sqlite.org / local file system	Embedded, no external DB cost
Collaboration Tools	Unlimited	0	GitHub, Google Workspace (Docs, Drive)	Free tiers or existing institutional access

Demo & Test Plan

Demo Scenario: Demonstrate the end-to-end flow of a blockchain-verified academic credential.

- Institution: Login, issue a new credential to a student.
- Student: Login, view new credential in their wallet, initiate a share/verification.
- Employer: Receive shared link, verify credential instantly using the system.
- Audience: Mentors/Judges/Potential Stakeholders

Test Protocols:

- Test cases: Successful issuance, successful verification (valid), attempted verification (invalid/non-existent), sharing functionality.
- Dataset/samples: 3-5 mock student accounts, 1 mock institution, 1 mock employer, 5-7 mock credentials.
- Environments: Local development environment (web browser).

Pre-Demo Checklist:

- Reset mock database to clean state.
- Ensure all local servers (frontend/backend) are running.
- Verify internet connectivity (for potential external API mocks, if any).
- Backup demo video/screenshots (if pre-recorded).
- Confirm all team members are ready for their parts.

Evidence – Results & Findings

Metric (from criteria)	Target	Observed	Pass/Gap	Notes / Evidence Link
Credential Integrity	100% integrity	[Demo shows unalterable record]	pass	[Demo Video Link]
Issuance Success Rate	100% successful	[Demo shows successful issuance]	pass	[Code Repo/Demo Script]
Verification Speed	<3 sec	[Demo shows instant verification]	pass	[Demo Video Link]
Tamper-Proof Immutability	No alteration possible	[Attempted alteration fails]	pass	[Test Logs/Code Snippet]
Wallet User-Friendliness	Intuitive navigation	[Student UI is simple & clear]	pass	[Screenshots/Demo Video]
Core Module Functionality	All Student/Inst/Empl over PoC goals met	[All demo scenarios run successfully]	pass	[Overall PoC Demo/Report] (for a 6th general row)

Demo Video & Repositories

- Demo Video:

[https://drive.google.com/file/d/1S-T3pHSFzcC5J8dQEZ7eJjc2CAEWvW6/view?us
p=sharing](https://drive.google.com/file/d/1ST3pHSFzcC5J8dQEZ7eJjc2CAEWvW6/view?usp=sharing)

- QR:



Risks, Limitations & Mitigations

Risk / Limitation	Impact (H/M/L)	Likelihood (H/M/L)	Mitigation / Plan	Owner
Blockchain Scalability	H	L	PoC uses simulation; research scalable chains	Vighas E
Mock Data Fidelity	M	M	State PoC is mock-data based; plan real DB integration	Raahul Kanna K
Off-chain Data Security	H	M	Implement IPFS mock; secure backend storage	Ramya S
Limited AI/Analytics	M	H	Prioritize core blockchain for PoC; integrate AI later	Raahul Kanna K
Future User Adoption	H	L	Focus on simple PoC UX; plan user feedback	Team

Go/No-Go & Next Steps →

Prototype

- Decision: Go — Rationale: PoC demonstrated core secure issuance and verification, proving technical feasibility.
- Prototype Plan (if Go):
 - Integrate Real Blockchain: Shift from simulation to a testnet.
 - Full User Module Dev: Build out complete Student, Institution, Employer features.
 - Implement IPFS: Decentralized storage for certificate files.
 - Scope: End-to-end user flows.
 - Timeline: 8-12 weeks.
 - Resources: Testnet tokens, IPFS service, team time.
 - KPIs to carry forward: Verification Success Rate, Issuance Time, System Uptime, User Feedback Score.

Annex — Domain-Specific Checklist (Duplicate relevant items)

AD — AI & Data Science:

Dataset provenance: Mock student/credential data used for PoC.

Baseline & target metrics: PoC aims for functional demo; future define metrics for AI suggestions.

BW — Blockchain & Web3:

Chain/testnet choice: PoC used simulation; future explore Polygon/Ethereum testnets.

Smart-contract ABI: Mock ABI for simulation; draft real ABI for prototype.

Key mgmt: Mock student/institution wallets; secure key management for prototype.

Threat model: Basic PoC assumptions; detailed model for prototype.

FW — Full Stack Web Dev:

API sketch: Basic PoC routes implemented; detailed API for prototype.

Data model: JSON/SQLite mock; formal schema for prototype.

Component tree: Basic React components for PoC; expand for prototype.

Auth flow: JWT mock in PoC; robust auth for prototype.

Accessibility checks: Basic UI; comprehensive checks for prototype.