Product Requirements Document (PRD)

BlockVerify: Blockchain-Based Document Verification System

- Document Version: 2.2
- Created by: Miqdad O. Adams.
- Target Release: TBD
- Status: In Progress

1. Executive Summary

This document outlines the product requirements for BlockVerify, a secure, blockchain-based system for document verification and attestation. The product is designed to replace traditional manual verification methods, providing a decentralized and tamper-proof solution for organizations like the University of Lagos to verify the authenticity and provenance of academic and official documents. By leveraging blockchain's immutability and transparency, BlockVerify aims to significantly reduce fraud, enhance data integrity, and streamline administrative overhead for both issuers and verifiers.

2. Goals & Objectives

Primary Goal: To create a secure, efficient, and transparent platform that empowers institutions to issue verifiable digital documents and allows third parties to instantly confirm their authenticity.
Key Objectives:

- Reduce Fraud: Minimize the risk of fraudulent documents by securing document hashes on an immutable blockchain.
- Improve Efficiency: Streamline the verification process, reducing the time and resources required for manual checks.
- Enhance Trust: Provide a single source of truth for document authenticity, verifiable by anyone with an internet connection.
- Simplify User Experience: Create an intuitive platform for both document issuers and verifiers, regardless of their technical knowledge.

3. Target Audience & Personas

- Document Issuers (e.g., University Administration): Responsible for uploading and officially attesting to a document.
    - Needs: A secure and simple way to issue verifiable documents in bulk, manage a history of all issued documents, and maintain an audit trail.
- Verifiers (e.g., Employers, Government Agencies): Responsible for checking the authenticity of a document presented by a student or applicant.
    - Needs: A fast, reliable, and publicly verifiable method to confirm a document's legitimacy without relying on the issuing institution.

- Document Owners (e.g., Students): The end-user who receives the verifiable document.
    - Needs: A secure and easily accessible way to store their documents and share them with third parties for verification.

4. User Flows

Flow 1: Document Issuance (for Issuers)

1. Issuer Login: The Issuer logs into the platform's secure dashboard.
2. Document Upload: The Issuer uploads one or more documents (e.g., student transcripts, certificates).
3. Hash Generation: The system automatically generates a cryptographic hash (SHA-256) of each document.
4. Data Input: The Issuer provides relevant metadata (e.g., student ID, date of issue, document type).
5. Blockchain Attestation: The Issuer reviews the information and confirms the attestation. The system then writes the document's hash and metadata to the blockchain.
6. Confirmation & Sharing: The system displays a unique verification link and/or a QR code for each document. The Issuer can then share this with the Document Owner.

Flow 2: Document Verification (for Verifiers)

1. Access Verification Portal: The Verifier navigates to the public-facing verification portal.
2. Verification Method: The Verifier can either:
    - Upload a document to be verified.
    - Enter a unique verification link or ID.
    - Scan a QR code from a physical document.
3. Hash Comparison: The system computes the hash of the uploaded document and compares it to the hash stored on the blockchain.
4. Verification Result: The system displays the verification result.
    - Success: A "Verified" status with the original issue date, issuer, and a link to the blockchain transaction.
    - Failure: A "Not Verified" status with an explanation (e.g., document has been tampered with).

Flow 3: User Dashboard (for Document Owners)

1. User Login: The Document Owner logs in to their personal dashboard.
2. View Documents: The user sees a list of all documents issued to them.
3. Share/Download: For each document, the user can:
    - Copy the unique verification link.
    - Download the QR code.
    - Download a signed, watermarked version of the document.

5. High-Level Requirements

- User Management: Secure login and role-based access control for all user types.
- Document Management: Functionality for uploading, hashing, and storing document metadata.
- Blockchain Integration: Seamless integration with a chosen blockchain network (e.g., Polygon) to write and retrieve document hashes.
- Verification Interface: A publicly accessible, user-friendly interface for document verification.
- Notifications: System to notify users of successful issuance and verification.
- Dashboard: A central dashboard for Issuers to manage documents and Verifiers to view attestation history.

6. Detailed Requirements (Sub-PRDs)

Sub-PRD 1: Document Upload & Issuance Dashboard

This module is for the Document Issuer.
Features:

- Dashboard Overview: Displays key metrics like total documents issued, verification success rates, and recent activity.
- Bulk Document Upload: Allows for the upload of multiple documents at once via drag-and-drop.
- Document List: A sortable and searchable table of all issued documents. Each entry includes:
  - Document Name
  - Recipient Name
  - Issue Date
  - Verification Status
  - Blockchain Transaction ID
- Single Document Attestation Flow: A step-by-step wizard for attesting a single document.
- Search & Filter: Advanced search functionality by recipient, document type, and date range.
- Attestation History: A detailed log of all blockchain transactions related to document issuance.

User Stories:

- As an Issuer, I want to upload multiple student transcripts at once, so I can save time on administrative tasks.
- As an Issuer, I want to see a clear list of all the documents I have issued, so I can easily track what has been attested.
- As an Issuer, I want to search for a specific document by student ID or name, so I can quickly find the information I need.

Sub-PRD 2: Document Verification Portal

This module is the core public-facing component.
Features:

- QR Code Scanner: A web-based interface that uses the user's camera to scan a QR code.
- Document Upload: An area for Verifiers to upload a digital file for hashing.
- Verification ID Search: A search bar for users to enter a unique verification ID.
- Verification Results Page:
  - Displays a clear "Verified" or "Not Verified" status.
  - If verified, shows the original document metadata (issuer, date, recipient).
  - Provides a direct link to the blockchain explorer to view the transaction.
  - If not verified, it provides a clear error message (e.g., "Hash Mismatch," "Invalid ID").

User Stories:

- As a Verifier, I want to scan a QR code on a physical document, so I can instantly check its authenticity without typing anything.
- As a Verifier, I want to upload a PDF file, so I can verify it against the blockchain record.
- As a Verifier, I want to see the original date of issue and the name of the issuing institution, so I can trust the document's provenance.

Sub-PRD 3: User Dashboard

This module is the personal hub for the Document Owner.
Features:

- My Documents: A list of all documents issued to the user.
- Document Details: When a user clicks on a document, they see:
  - Document Name and Type
  - Issuer
  - Issue Date
  - A unique verification link
  - A downloadable QR code
- Share Functionality: A simple "share" button that allows users to easily share the verification link via email, social media, or other apps.

User Stories:

- As a Student, I want to see all my academic certificates in one place, so I can easily access them.
- As a Student, I want to easily share a unique link with a potential employer, so they can verify my credentials.
- As a Student, I want to download a QR code for my certificate, so I can print it on a physical copy for easy scanning.

Sub-PRD 4: Public Landing Page

This module serves as the marketing and entry point for all users.
Features:

- Hero Section: A prominent section with a clear headline ("Secure Your Credentials with BlockVerify"), a sub-headline, and a call-to-action (CTA) for each user type (e.g., "Verify a Document," "For Issuers," "For Students").
- What is BlockVerify?: A clear, concise explanation of the product's value proposition using simple language and visuals.
- How It Works (3 Steps): A step-by-step visual guide explaining the process for both document issuance and verification.
- Why Use BlockVerify?: A section highlighting key benefits for each user type, such as "Fraud Prevention for Verifiers," "Efficiency for Issuers," and "Digital Ownership for Students."
- Testimonials: Social proof in the form of quotes or logos from pilot users or partner institutions.
- FAQ Section: Answers to common questions about blockchain, security, and the verification process.
- Contact Form: A simple form for business inquiries.
- Responsive Design: Ensures the page is fully functional and visually appealing on all devices (desktop, tablet, mobile).

User Stories:

- As a prospective user, I want to land on the homepage and immediately understand what the product does and who it's for, so I can decide if it's relevant to me.
- As an Issuer, I want to find a clear path on the landing page that explains how my institution can get started with the service.
- As a Verifier, I want to quickly find the verification tool on the landing page, so I can check a document without any friction.

7. Technical Requirements

- Frontend: The web application will be built using a modern JavaScript framework like React.js, leveraging a state management library for efficiency. It will integrate with Web3 libraries like ethers.js or web3.js to connect with user wallets (e.g., MetaMask, WalletConnect).
- Backend: A Node.js backend will manage the application logic, user authentication (JWT), and communication with the blockchain. A PostgreSQL database will be used to store non-sensitive metadata (e.g., user information, document titles).
- Blockchain: The system will be built on a public blockchain network like Polygon for its low transaction fees and high throughput.
- Smart Contracts: Core logic will be implemented in a Solidity smart contract that stores the immutable cryptographic hash and metadata of each document. Security will be a top priority, with smart contracts built using best practices and verified with OpenZeppelin libraries.
- File Storage: Documents will not be stored on the blockchain. Instead, they will be stored securely on a decentralized file storage system like IPFS, with the hash of the IPFS pointer stored on-chain.

8. Risks & Assumptions

- Risk: Users may not be familiar with blockchain technology or wallet management.
    - Mitigation: The UI/UX will be designed to be as simple as possible, abstracting away technical complexities. We will provide clear onboarding and help documentation.
- Risk: Network congestion could lead to delayed transactions and a poor user experience.
    - Mitigation: By using a high-throughput network like Polygon, we minimize this risk. We will also monitor network status and implement appropriate retry logic.
- Assumption: The target users (students, employers) have internet access to use the platform.
- Assumption: The chosen blockchain network remains stable and secure for the foreseeable future.

9. Out of Scope

The following features and functionalities are explicitly not part of this release's scope to ensure we deliver the core value proposition efficiently.

- A dedicated mobile application (iOS/Android). The mobile experience will be a responsive web app.
- Support for multiple languages. The initial version will be in English.
- Integration with legacy university systems for data import/export.
- Custom report generation for Issuers. The initial version will only show a static overview on the dashboard.
- Monetization or payment gateways for verification fees. Verification will be free.

10. Phases & Roadmap

Phase 1: Minimum Viable Product (MVP)

- Timeline: TBD
- Focus: Core functionality for document attestation and verification.
- Milestones:
    - Complete system design and architecture.
    - Develop smart contracts and backend APIs.
    - Build the core Issuer dashboard and public Verification Portal.
    - Release to a pilot group for user testing.

Phase 2: Post-Launch Enhancements

- Timeline: TBD
- Focus: Improving user experience, scalability, and adding key features based on feedback.
- Milestones:
    - Build the User Dashboard for Document Owners.
    - Implement user notifications (email/push).
    - Optimize the QR code scanner for various devices.

    - Conduct a public launch to a wider audience.

Phase 3: Scaling & Analytics

- Timeline: TBD
- Focus: Adding advanced features to serve the enterprise market and enhance data insights.
- Milestones:
    - Integrate advanced analytics and reporting for Issuers.
    - Develop a full-fledged API for third-party developers.
    - Support for multiple document types and custom metadata fields.

11. Success Metrics

- Number of Documents Attested: The total number of documents issued by institutions.
- Verification Volume: The number of documents verified on the platform per month.
- User Adoption: Percentage of targeted institutions and students using the platform.
- Verification Speed: The average time it takes for a verification request to be completed.
- Feedback & Satisfaction: Qualitative feedback from users and a Net Promoter Score (NPS).