|  |
| --- |
| Software Requirements Specifications  **AI and Blockchain-Based Certificate Verification System**  **Project Code:**  BCV-AI-(25-26)  Internal Advisor:  Mr. Muhammad Fahad  Project Manager:  Dr. Muhammad Ilyas  Project Team:   1. Nouman Riaz BSCS51F21R003 (Team Lead)   Submission Date:  October 20, 2025 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Project Manager’s Signature |

**Document Information**

| Category | Information |
| --- | --- |
| Customer | University of Sargodha |
| Project | AI and Blockchain-Based Certificate Verification System |
| Document | Requirement Specifications |
| Document Version | 1.0 |
| Identifier | BCV-AI-(25-26) |
| Status | Draft |
| Author(s) | Muhammad Nouman Riaz |
| Approver(s) | Dr. Muhammad Ilyas |
| Issue Date | October. 20, 2025 |
| Document Location | Department of Computer Science, UOS |
| Distribution | 1. Project Advisor – Mr. Muhammad Fahad  2. Project Manager – Dr. Muhammad Ilyas |

**Definition of Terms, Acronyms and Abbreviations**

*This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.*

| Term | Description |
| --- | --- |
| AI | Artificial Intelligence – used to extract and process text automatically from certificates. |
| OCR | Optical Character Recognition – technology that converts text from images or PDFs into machine-readable text. |
| SHA-256 | Secure Hash Algorithm 256-bit – used to generate a unique digital fingerprint for each certificate. |
| RS | Requirements Specifications – this document outlining the complete system requirements. |
| ETH | Ethereum – a blockchain platform used for storing and verifying certificate hashes. |
| JWT | JSON Web Token – used for secure user authentication and session handling. |
| HTTPS | Hypertext Transfer Protocol Secure – ensures encrypted communication between client and server. |
| API | Application Programming Interface – allows communication between different software components. |
| DM | Project Manager – person responsible for supervising the development process. |
| DB | Database – where user credentials, logs, and verification records are stored (except blockchain data). |

**Table of Contents**

[1. Introduction 5](#_Toc51490198)

[1.1 Purpose of Document 5](#_Toc51490199)

[1.2 Project Overview 5](#_Toc51490200)

[1.3 Scope 5](#_Toc51490201)

[2. Overall System Description 5](#_Toc51490202)

[2.1 User characteristics 5](#_Toc51490203)

[2.2 Operating environment 5](#_Toc51490204)

[2.3 System constraints 5](#_Toc51490205)

[3. External Interface Requirements 6](#_Toc51490206)

[3.1 Hardware Interfaces 6](#_Toc51490207)

[3.2 Software Interfaces 6](#_Toc51490208)

[3.3 Communications Interfaces 6](#_Toc51490209)

[4. Functional Requirements 6](#_Toc51490210)

[5. Non-functional Requirements 6](#_Toc51490211)

[5.1 Performance Requirements 6](#_Toc51490212)

[5.2 Safety Requirements 6](#_Toc51490213)

[5.3 Security Requirements 6](#_Toc51490214)

[5.4 User Documentation 7](#_Toc51490218)

[6. Assumptions and Dependencies 7](#_Toc51490219)

[7.](#_Toc51490220) **[R](#_Toc51490220)**[eferences 7](#_Toc51490220)

1. Introduction

* 1. Purpose of Document

The purpose of this document is to define all the system requirements, functionalities, and features of the **AI and Blockchain-Based Certificate Verification System**. It explains what the system will do, how it will perform its tasks, what problems it aims to solve, and how it interacts with users and external platforms.

This document is intended for **project stakeholders, academic supervisors, developers, and testers**. It will serve as a roadmap for the design, development, and validation phases of the project. Each requirement mentioned here will help ensure that the system is built according to user needs and performs certificate verification accurately and efficiently.

* 1. Project Overview

The **AI and Blockchain-Based Certificate Verification System** aims to create a secure, tamper-proof platform for verifying academic and professional certificates. Fake certificates have become a serious problem in Pakistan, with employers and institutions often struggling to confirm authenticity through slow manual processes[1] . This project combines **Artificial Intelligence (AI)** and **Blockchain** technology to make the verification process fast, transparent, and trustworthy[2].

Here’s how the system works:

* Institutions or universities will upload and register certificates in the system.
* The system will extract text from uploaded certificates using OCR (Optical Character Recognition) and generate a unique SHA-256 hash for each one [4].
* This hash will then be stored on the Ethereum blockchain, which ensures the record cannot be altered or forged.
* When a user or employer uploads a certificate to verify, the system re-generates the hash and checks it against the blockchain record.
* If the hashes match, the certificate is confirmed as authentic. If not, it’s flagged as fake or modified.

The goal of this system is to reduce human error, prevent fraud, and create a centralized digital verification solution that can be used across educational and professional sectors.

* 1. Scope

The system being developed, AI and Blockchain-Based Certificate Verification System, is designed to help organizations and individuals verify academic and professional documents securely.

**In Scope (What the system will do):**

* Allow registered institutions to upload and register certificates.
* Extract certificate data using OCR and generate a hash value.
* Store the certificate hash securely on the blockchain.
* Enable users or employers to upload certificates for instant verification.
* Display verification results (“Verified” or “Fake/Modified”).
* Maintain a simple web-based interface accessible from browsers.

**Out of Scope (What the system will not do):**

* Integration with national databases such as HEC or NADRA.
* Real cryptocurrency transactions (system will use Ethereum testnet only).
* Mobile application development (web version only for this phase).
* Editing or deletion of blockchain data once recorded.
* The system will be developed as a web-based application using Python Flask for the backend, Tesseract OCR for text extraction, and Solidity smart contracts for blockchain operations. It will be hosted on a cloud-based test environment to ensure scalability and real-time access.

By combining AI automation and blockchain’s immutability, the system will offer a secure and efficient way to verify certificates without relying on manual validation or third-party services.

1. Overall System Description

The AI and Blockchain-Based Certificate Verification System is a secure, web-based platform designed to check the authenticity of academic and professional certificates. The system will use Artificial Intelligence (AI) and Blockchain together to prevent fraud and make the verification process automatic and transparent.

AI will handle text extraction through Optical Character Recognition (OCR), while Blockchain will record unique certificate hashes to ensure that once data is stored, it can never be changed or deleted [3].

This combination will help institutions, employers, and individuals easily verify whether a certificate is real or fake.

The system will be most useful in Pakistan’s academic and professional sectors, where fake or altered certificates are a common issue. It will reduce the time and cost required for manual verification, providing a trustworthy digital alternative.

**2.1 User Characteristics**

The system is designed for three main types of users. Each group plays a different role in how the system operates.

**2.1.1 Primary Users (Institutions / Universities)**

* These users will register and upload verified certificates to the system.
* Each uploaded certificate will be processed through OCR and stored securely on the blockchain.
* They are responsible for maintaining the authenticity of uploaded data.
* Institutions include universities, training centers, and boards that issue certificates.

**2.1.2 Secondary Users (Employers / Verifiers)**

* These users will upload certificates submitted by job applicants or trainees.
* The system will automatically check and verify whether the uploaded certificate matches the one registered by the issuing institution.
* Employers or verification agencies can use this system to save time and avoid fake documentation.

**2.1.3 General Users (Individuals / Students)**

* These users can verify their own certificates to confirm that the record exists on the blockchain.
* They can view verification results but cannot modify or upload new records.
* Basic computer skills and internet access are sufficient to use this system.

**2.2 Operating Environment**

The system will run entirely on the web, accessible through desktop and mobile browsers. It does not require special hardware, only a stable internet connection.

**2.2.1 Hardware**

**Servers:** The backend will be hosted on a cloud server or a local test environment (e.g., AWS or PythonAnywhere).

**User Devices:** Any computer or smartphone with an internet connection can use the system. Minimum 2 GB RAM and a web browser are recommended.

**Storage:** Server storage will handle uploaded certificates and temporary OCR files.

**2.2.2 Software**

**Operating Systems:** Works on Windows, Linux, and macOS.

**Frontend:** HTML, CSS, and JavaScript for user interface.

**Backend:** Python Flask framework for handling requests and blockchain interaction.

**Blockchain:** Ethereum Test Network using Solidity smart contracts.

**OCR:** Tesseract OCR for text extraction from images or PDFs.

**Database:** SQLite or MongoDB for storing user and verification data.

**Browser Compatibility:** Chrome, Firefox, Edge, and Safari.

**2.2.3 Network**

* The system requires a stable internet connection for communication with the blockchain.
* Works on standard Wi-Fi or mobile data (3G/4G/5G).
* Blockchain confirmation times depend on network speed and testnet performance.

**2.2.4 Security and Privacy**

* All data transmission will occur through HTTPS to ensure encryption.
* Uploaded certificates and user credentials will be securely stored in the backend database.
* Only authorized institutions can register certificates; unauthorized uploads are blocked.
* Hashes on the blockchain will not contain personal data—only encoded identifiers for privacy protection.

**2.2.5 Language Support**

* The primary language will be English.
* Future versions may include Urdu support for wider accessibility.

**2.3 System Constraints**

During design, development, and deployment, the system will face certain constraints and boundaries that define how it operates.

**2.3.1 Software Constraints**

* The system will only work through web browsers (no mobile app in this phase).
* Requires internet connection for all blockchain-related tasks.
* Compatible with modern browsers only (Chrome, Firefox, Edge, Safari).
* OCR accuracy depends on image clarity; poor-quality scans may reduce precision.

**2.3.2 Hardware Constraints**

* The hosting server must have sufficient CPU, RAM, and storage for blockchain transactions and OCR processing.
* Users need devices capable of uploading images or PDF files.
* Performance may slow down on very low-end devices or weak internet connections.

**2.3.3 Cultural Constraints**

* The interface is initially available in English only.
* The system assumes familiarity with digital document handling and web interfaces.

**2.3.4 Legal Constraints**

* The project will follow Pakistan’s data protection and privacy standards.
* Certificates will be added only by verified institutions to prevent unauthorized data entry.
* No personal information will be stored on the blockchain itself, ensuring compliance with privacy laws.

**2.3.5 User Constraints**

* Users must know how to browse the web and upload files.
* Each institution or employer will need verified credentials to access the system.
* Users must have a reliable internet connection for verification requests.

**2.3.6 Environmental Constraints**

* The system depends on internet-based blockchain services (Ethereum Testnet).
* Power or network outages may temporarily delay verification operations.
* The system will be tested in a controlled environment to ensure stable performance.

**2.3.7 Off-the-Shelf Component Constraints**

**Tesseract OCR:** Accuracy depends on font clarity and image resolution.

**Ethereum Testnet:** May experience delays or limited access.

**Web3.py Library:** Requires proper version compatibility with the blockchain network.

**SHA-256 Algorithm:** Provides strong encryption but cannot prevent poor data input quality.

**Cloud Hosting Services:** Limited free-tier storage and API call restrictions.

1. External Interface Requirements

This section explains the hardware, software, and communication interfaces that the system will use to function properly.

The AI and Blockchain-Based Certificate Verification System will interact with user devices, online services, and the Ethereum test network to perform certificate verification smoothly.

**3.1 Hardware Interfaces**

**3.1.1 User Devices**

* The system will work on standard desktop computers, laptops, and smartphones.
* Users will upload scanned images or PDFs of their certificates from these devices.
* Any modern device with an internet connection and a web browser will support the system.

**3.1.2 Scanner or Mobile Camera**

* Users or institutions can use mobile phone cameras or document scanners to take pictures of certificates.
* The uploaded image must be clear enough for OCR to detect and read text correctly.
* Supported formats include .jpg, .png, and .pdf.

**3.1.3 Server Hardware**

* The system backend will run on a cloud server or a local hosted environment with enough storage and processing power to handle file uploads and blockchain operations.
* Minimum requirements: 2 GB RAM, stable internet connection, and at least 10 GB of storage for certificate records and logs.

**3.2 Software Interfaces**

**3.2.1 Tesseract OCR Engine**

* The system will use Tesseract OCR for extracting text from uploaded certificates.
* Version 5.0 or above will be integrated into the backend for accuracy.
* It converts certificate images into machine-readable text, which is then processed for hashing and blockchain storage.

**3.2.2 Python Flask Framework**

* The backend web framework will be Flask, used to manage routing, handle user uploads, and communicate between the frontend and blockchain modules.
* Flask will also handle the API endpoints for certificate registration and verification.

**3.2.3 Blockchain Smart Contracts**

* The blockchain part will use Solidity for writing smart contracts on the Ethereum test network (Goerli or Sepolia).
* These contracts store and verify certificate hashes.
* The system will use the web3.py library to interact with the blockchain.

**3.2.4 Database**

* The local backend will store basic user details and verification logs using SQLite or MongoDB.
* The blockchain itself will store only hashed certificate data, not the full certificate content, to ensure privacy.

**3.2.5 Operating Systems and Browsers**

* The system will work on Windows, Linux, and macOS through standard browsers like Google Chrome, Firefox, Edge, and Safari.
* The user interface is responsive, meaning it automatically adjusts to different screen sizes and devices.

**3.2.6 Hashing and Security Libraries**

* SHA-256 hashing algorithm will be used to create a unique fingerprint for each certificate.
* SSL/TLS encryption will protect communication between users and the server.
* JSON Web Tokens (JWT) will secure user authentication and session management.

**3.3 Communications Interfaces**

**3.3.1 Network Communication**

* The system will use HTTPS protocol for all data transfer to ensure security.
* Communication between the frontend and backend will occur through RESTful APIs.
* Blockchain transactions will be sent through web3.py using Ethereum’s JSON-RPC protocol.
* Every interaction with the blockchain will be verified before the result is shown to the user.

**3.3.2 Data Transfer**

* Uploaded certificate files (PDFs or images) will be sent from the browser to the server over secure HTTPS.
* After OCR processing, only text data and hashes are sent to the blockchain.
* The system will limit file uploads to a reasonable size (e.g., up to 5 MB per file) to maintain performance.

**3.3.3 Synchronization and Response**

The blockchain verification process depends on transaction confirmation time.

Once confirmed, the user will see a clear status:

“Certificate Verified”

“Certificate Not Found / Fake”

**3.3.4 Data Transfer Rates**

* Normal certificate verification takes only a few seconds depending on blockchain response time.
* The system will automatically retry the request if a transaction fails due to network delay.
* Upload progress bars will show real-time status while sending files.

1. Functional Requirements

This section describes the main functional requirements of the AI and Blockchain-Based Certificate Verification System.

Each function defines a specific task or feature that the system must perform to ensure smooth and secure verification of certificates.

**4.1 User Management**

* Users (institutions, verifiers, and individuals) will be able to register and log in through a secure web interface.
* The system will allow institutions to create verified accounts for certificate uploads.
* Employers or verifiers can register to perform certificate validation checks.
* The system will store user credentials securely using encryption and hashed passwords.
* A password recovery and reset feature will be available through email or a system-generated link.
* Each user will have a role-based dashboard according to their permissions (Institution, Verifier, or General User).

**4.1 Certificate Upload and Registration (For Institutions)**

* Institutions will upload certificates in image or PDF format through the system’s web portal.
* The system will automatically extract text from the uploaded certificate using AI-based OCR (Tesseract).
* The extracted information will include details such as student name, ID, course title, and issue date.
* The system will generate a unique SHA-256 hash from the extracted data.
* This hash will be sent to the blockchain and permanently stored using a Solidity smart contract.
* After successful storage, the institution will receive a blockchain transaction ID as proof of registration.
* Institutions can view a list of uploaded and verified certificates on their dashboard.

**4.3 Certificate Verification (For Employers / Verifiers)**

* Verifiers can upload a certificate to check its authenticity.
* The system will extract text and generate a hash for the uploaded certificate.
* The new hash will be compared with existing hashes stored on the blockchain.
* If the hash matches, the system will display a “Verified” result.
* If no match is found, it will show “Fake or Modified Certificate.”
* Verification results will include metadata like issue date, institution name, and blockchain transaction ID.
* OCR and AI Data Processing
* The system will use an AI-based OCR engine to automatically extract text from certificates.
* OCR will detect and convert printed text into digital data.
* The system will clean and format extracted data before generating a hash.
* The AI component will validate text consistency (for example, checking if names and dates are properly recognized).
* The OCR engine will support English text; future versions may include Urdu support.

**4.4 Blockchain Integration**

* The system will connect to the Ethereum test network (Goerli or Sepolia) [5].
* It will use Solidity smart contracts to store and verify certificate hashes.
* Smart contracts will handle the certificate registration and verification logic automatically.
* Each transaction will have a unique Transaction ID (TxID) visible to users for transparency.
* The blockchain will ensure that once a certificate record is added, it cannot be modified or deleted.
* Institutions can re-upload corrected certificates if necessary; this will create a new blockchain record with a different hash.
* Verification Result Display
* After verification, the system will display results in a clear and user-friendly format.
* Verified certificates will show a green status bar with “Certificate Verified.”
* Fake or altered certificates will show a red warning with “Certificate Not Found.”
* Users will be able to download a verification report as a PDF summary if needed.

**4.4 Admin and Monitoring Features**

* The admin panel will allow the project manager or system administrator to monitor uploaded and verified certificates.
* The admin can approve or block institution accounts if misuse is detected.
* The system will generate analytics such as number of certificates verified, failed verifications, and upload activity.
* Logs of all transactions will be maintained for transparency and security auditing.
* Error Handling and Notifications
* If OCR fails to read the certificate, the system will notify the user to upload a clearer image.
* If blockchain connectivity is interrupted, the system will retry the transaction automatically.
* Users will receive real-time error messages such as “Invalid File Type” or “Upload Failed.”
* All system messages will be displayed in simple, easy-to-understand English.

**4.5 Data Storage and Privacy**

* The blockchain will only store hashed certificate data, not personal or sensitive information.
* The backend database will store basic information such as user accounts and verification logs.
* Uploaded certificates will be temporarily stored on the server and deleted after hashing.
* No editable information will be stored on blockchain; all data will be immutable once recorded.
* All connections between frontend, backend, and blockchain will use secure encryption (HTTPS + SSL/TLS).

**4.6 Multi-Platform Accessibility**

* The system will work on all operating systems including Windows, Linux, and macOS.
* It will run entirely in web browsers without needing any installation.
* The frontend design will be responsive for both desktop and mobile browsers.
* Users only need internet access; all AI and blockchain processing happens on the server side.

1. Non-functional Requirements

This section defines the performance, reliability, security, usability, and compatibility standards that the system must meet. These requirements ensure that the AI and Blockchain-Based Certificate Verification System runs efficiently, safely, and consistently under different conditions.

**5.1 System Performance**

* The system should complete certificate verification within seconds of upload.
* OCR and hashing will be optimized to run quickly and smoothly.
* Blockchain confirmation time may vary, but the user will see a live progress bar.
* Web pages should load in less than five seconds on a normal connection.
* The backend should handle multiple requests at once without crashing or losing data.

**5.2 Reliability and Stability**

* The system should be up and running at least 95% of the time.
* Daily backups will be scheduled to protect user and verification data.
* In case of service interruption, the system will automatically retry operations.
* It will be tested for stable performance with multiple users at once.

**5.3 Data Safety**

* Uploaded certificates will be deleted automatically after verification.
* Dangerous file formats won’t be accepted.
* All uploaded data will be scanned for viruses or tampering.
* Only administrators can access sensitive functions.
* Blockchain entries can’t be changed or corrupted once confirmed.

**5.4 Security and Privacy**

* All communication will use secure HTTPS connections with SSL/TLS encryption.
* User passwords will be hashed with SHA-256 and never stored as plain text.
* JSON Web Tokens (JWT) will handle secure logins and sessions.
* Only verified institutions will have permission to upload certificates.
* No personal data will be written to the blockchain—only hash values.
* Input fields will be validated to block common web attacks like SQL injection or XSS.

**5.5 User Experience**

* The interface will be clean and easy to follow for non-technical users.
* The verification process will include simple on-screen steps and progress updates.
* All notifications and error messages will be written in plain, clear language.
* The design will work in both dark and light mode for better readability.

**5.6 Expansion Capability**

* The system will be modular so that AI, OCR, or blockchain parts can be updated separately.
* As more institutions join, the backend can scale by adding more server resources.
* Database caching and indexing will improve performance with large datasets.
* Future upgrades can move from test blockchain networks to Ethereum mainnet without major redesign.

**5.7 Accuracy and Validation**

* OCR results will target at least 90% accuracy for English text.
* AI will recheck extracted information before generating hashes.
* All main details (name, ID, and course) must match before verification is marked as “authentic.”

**5.8 Legal and Ethical Compliance**

* The system will follow Pakistani data protection and privacy standards.
* Institutions must be verified before uploading official records.
* Smart contracts will follow Ethereum’s security guidelines.
* The system will include a disclaimer noting that results depend on genuine data from institutions.

**5.9 Platform Compatibility**

* Compatible with all major browsers (Chrome, Firefox, Safari, Edge).
* Works across Windows, macOS, and Linux operating systems.
* Mobile-friendly design for Android and iOS browsers.

1. Assumptions and Dependencies

This section lists the assumptions made during development and the external dependencies that the project relies on for proper functioning.

**6.1 Assumptions**

**Internet Connectivity:**

Users have access to a stable internet connection since blockchain verification and certificate uploads require an online network.

**User Device Availability:**

Users and organizations will use modern devices with up-to-date browsers that support HTTPS and blockchain-based web apps.

**Institution Participation:**

Only verified educational institutions or organizations will register and issue certificates through the system.

**User Knowledge:**

Users are expected to understand how to upload files and use simple web applications.

**AI Model Accuracy:**

The AI-powered OCR and data extraction models will be accurate enough to read and interpret text from scanned certificates.

**Blockchain Access:**

The Ethereum test network or blockchain node will remain accessible during operation and testing.

**Legal Compliance:**

Institutions will provide legitimate data and comply with national privacy and educational data laws.

**6.2 Dependencies**

**Third-Party APIs:**

The system depends on external APIs such as OCR services for text extraction, and blockchain APIs (like Infura or Alchemy) for Ethereum transactions.

**Blockchain Network:**

The project relies on the Ethereum testnet (e.g., Goerli or Sepolia) for storing and verifying hashes. Any downtime or congestion could affect verification speed.

**Cloud Infrastructure:**

Hosting servers (e.g., AWS or Google Cloud) are required for system deployment, database storage, and backup operations.

**Development Tools:**

The project depends on frameworks like Flask for the backend, React for the frontend, and Solidity for smart contract integration.

**Financial Resources:**

Some components, such as blockchain gas fees or cloud hosting, may need small ongoing costs for deployment and testing.

**User Adoption:**

The system’s success depends on educational institutions and employers adopting it as a trusted verification tool.

1. References

| Ref. No. | Document Title | Date of Release/ Publication | Document Source |
| --- | --- | --- | --- |
| [1] | Educational Blockchain: A Secure Degree Attestation and Verification System by A. Ayub Khan et al., | 2021 | |  | | --- | |  |   https://www.mdpi.com/2076-3417/11/22/10917? |
| [2] | Bitcoin: A Peer-to-Peer Electronic Cash System — Nakamoto, S. | 2008 | **https://bitcoin.org/bitcoin.pdf** |
| [3] | Gaikwad, H., D’Souza, N., Gupta, R., & Tripathy, A. K. (2021, July). A blockchain-based verification system for academic certificates [Paper presentation]. 2021 International Conference on System, Computation, Automation and Networking (ICSCAN). | 2021 | https://doi.org/10.1109/ICSCAN53069.2021.9526377 |
| [4] | US Secure Hash Algorithm 1 (SHA-1) — Eastlake, D., & Jones, P., IETF RFC 3174 | 2001 | ttps://www.rfc-editor.org/rfc/rfc3174 |
| [5] | A Next-Generation Smart Contract and Decentralized Application Platform (Ethereum White Paper) — Buterin, V., Ethereum Foundation | 2014 | https://ethereum.org/en/whitepaper/ |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |