

## **CertiVerify – Certificate Verification Platform**

Course: Special Topics 2

Student Name: Adam Eyad Saleh

Student ID: 120204353

Supervisor: Dr.Raed Rashid

Semester: Fall 2025

Date: 27/10/2025

### **Abstract**

This project presents a blockchain-based certificate verification system named CertiVerify. The main objective of this platform is to provide a secure and transparent way to issue, verify, and revoke academic or professional certificates. By leveraging blockchain technology and smart contracts, the system ensures the authenticity of digital credentials and prevents forgery. The project was developed through multiple phases, including design, frontend, backend, and smart contract implementation using the Remix IDE.

### **Introduction**

Certificate forgery and manipulation have become a growing issue in both educational and professional sectors. Traditional systems that depend on centralized databases are vulnerable to tampering and lack transparency. CertiVerify was developed to solve this issue by using blockchain technology to record certificates immutably. The platform allows official institutions to issue certificates, and third parties can verify them instantly. This system enhances trust, reduces administrative overhead, and provides a transparent mechanism for document verification.

### **System Objectives**

- To provide a decentralized solution for certificate issuance and verification.
- To allow official entities to issue certificates through smart contracts.
- To verify authenticity of certificates using blockchain records.
- To revoke certificates when necessary and ensure transparency of all actions.
- To build a user-friendly interface for both issuers and verifiers.

### **System Features**

- User registration and login.
- Issuer registration and management.
- Certificate issuance and record storage on the blockchain.
- Real-time verification of certificates.
- Revocation and re-verification process.
- Admin dashboard to manage users and issuers.
- Frontend developed with HTML, CSS, and JavaScript.
- Backend implemented in PHP for RESTful API communication.

- Smart contract written in Solidity using Remix IDE.

## System Design

The system architecture is divided into three main layers:

1. Frontend Layer: Implements the user interface using HTML, CSS, and JavaScript. It includes pages for login, registration, certificate verification, and administration.
2. Backend Layer: Developed using PHP to handle requests and database operations. It stores issuer details, verification logs, and integrates with the blockchain.
3. Blockchain Layer (Smart Contract): Implemented using Solidity in Remix IDE. It manages the lifecycle of certificates through functions such as `issue()`, `verify()`, and `revoke()`.

Diagrams included in submission: Use Case Diagram, ER Diagram, System Flowchart (Frontend ↔ Backend ↔ Blockchain).

## Implementation Details

Phase I – Platform Sketching: Initial wireframes were designed to visualize system pages and user flows.

Phase II – Frontend Development: HTML, CSS, and JavaScript were used to build static pages for the platform.

Phase III – Backend Development: The backend was implemented using PHP to manage API endpoints for authentication and verification.

Phase IV – Smart Contract: A Solidity smart contract named `CertificateRegistry.sol` was developed and deployed on Remix IDE.

## Testing and Results

The system was tested on the Remix Ethereum Virtual Machine (EVM) using version 0.8.20. Compilation, deployment, issuance, and revocation all worked successfully.

Test Case	Function	Expected Output	Result
1	Compile Contract	No errors	Passed
2	Deploy Contract	Contract deployed successfully	Passed
3	Issue Certificate	New record created	Passed
4	Verify Certificate	Status = Valid	Passed
5	Revoke Certificate	Status = Revoked	Passed
6	Verify Revoked	Valid = False	Passed

## Conclusion

The CertiVerify project successfully demonstrates how blockchain technology can revolutionize document authentication. Through smart contracts, the system ensures transparency, security, and immutability of academic records. Future enhancements may include integration with IPFS for decentralized file storage and linking the PHP backend with the live Ethereum testnet.

## References

1. Ethereum Foundation – Solidity Documentation
2. Remix IDE Official Documentation
3. W3Schools – PHP, HTML, and JavaScript References
4. Blockchain Council – Smart Contract Fundamentals