**BLOCKCHAIN MINI PROJECT**

**TITLE: CERTIFICATE VERIFICATION**

**Team Members:**

**Achyuth K (PES1UG22CS023)**

**Vivekananda K(PES1UG22CS708)**

**Vinod M(PES1UG22CS699)**

**Introduction:**

**In today's digital world, the verification of certificates (academic, professional, or training) is often cumbersome and prone to fraud. Traditional systems rely on centralized databases, which can be hacked, manipulated, or require lengthy manual verification processes.**

**This project proposes a decentralized certificate verification system built on Ethereum blockchain using smart contracts. The system ensures:  
✅ Tamper-proof certificates  
✅ Instant verification  
✅ No centralized authority  
✅ Transparent and auditable records**

**By storing certificate hashes on-chain, we eliminate fraud while maintaining privacy.**

**Problem Statement**

**Current Challenges**

* **Fraudulent Certificates: Fake degrees/diplomas are easily created.**
* **Manual Verification: Employers/institutions must contact issuing authorities.**
* **Centralized Databases: Vulnerable to hacking/data manipulation.**
* **No Ownership: Users don’t control their credentials.**

**Solution**

**A blockchain-based system where:**

* **Institutions issue certificates as immutable transactions.**
* **Anyone can verify authenticity in seconds.**
* **No third-party intermediaries.**

**Blockchain Implementation Details**

**Tech Stack**

| **Component** | **Technology Used** |
| --- | --- |
| **Blockchain** | **Ethereum** |
| **Smart Contracts** | **Solidity** |
| **Frontend** | **React.js + ethers.js** |
| **Wallet Integration** | **MetaMask** |
| **Local Blockchain** | **Ganache** |

**Smart Contract Functions**

**contract CertificateVerification {**

**mapping(bytes32 => bool) public certificates;**

**event CertificateIssued(bytes32 indexed certificateHash, address indexed issuer);**

**function issueCertificate(bytes32 \_hash) public {**

**require(!certificates[\_hash], "Certificate already exists");**

**certificates[\_hash] = true;**

**emit CertificateIssued(\_hash, msg.sender);**

**}**

**function verifyCertificate(bytes32 \_hash) public view returns (bool) {**

**return certificates[\_hash];**

**}**

**}**

**Key Features:**

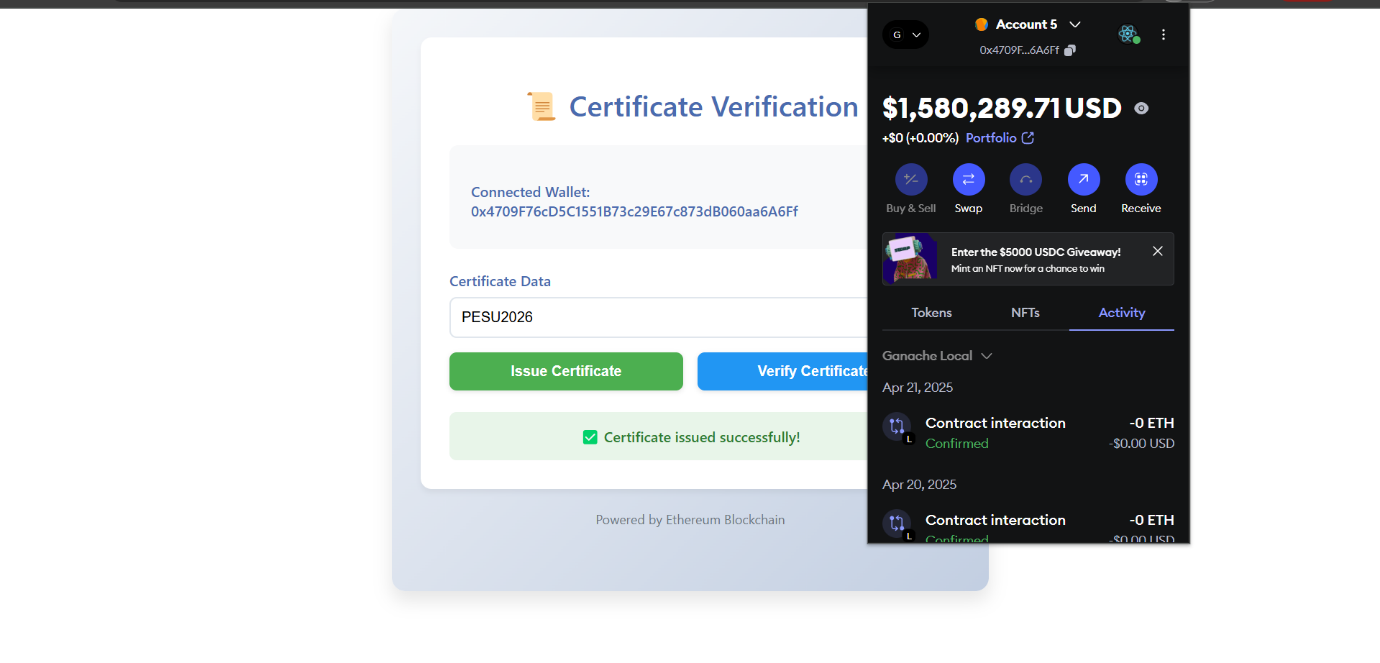
* **Stores hashes of certificates (not raw data).**
* **Emits events for transparency.**
* **Gas-efficient verification.**

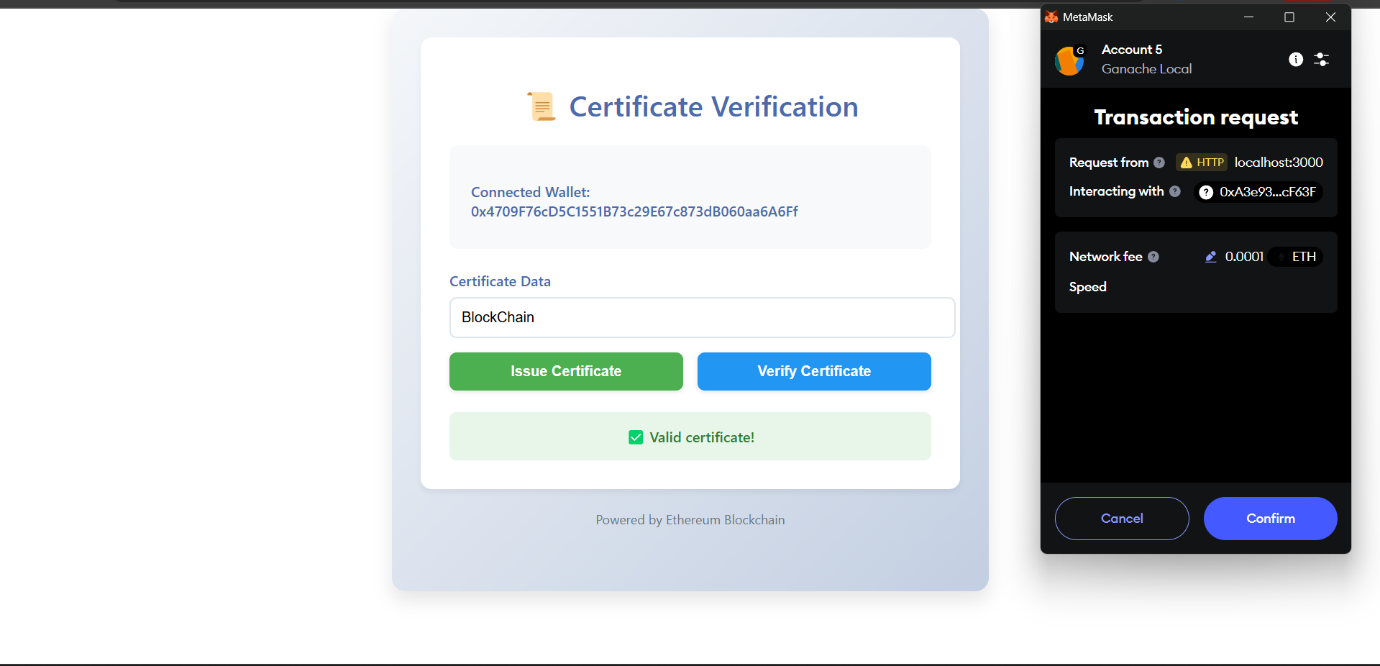
**Workflow**

1. **Issuance:**
   * **Institution converts certificate data → bytes32 hash.**
   * **Calls issueCertificate() to store hash on-chain.**
2. **Verification:**
   * **User provides certificate data.**
   * **Frontend computes hash → checks verifyCertificate().**

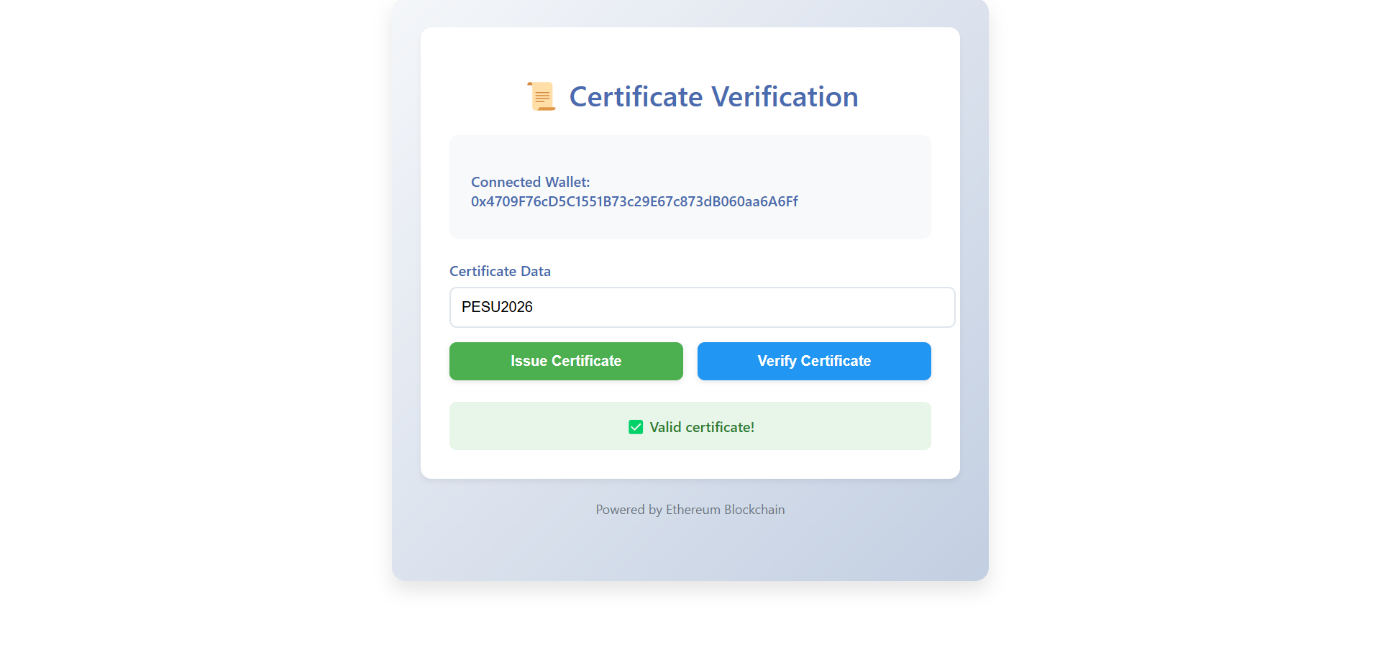
**Photos of working application:**

**Issuing a certificate**





**Checking if our certificate is valid or not**



**Future Enhancements**

**Short-Term**

* **IPFS Integration: Store full certificate data off-chain (IPFS) with on-chain hashes.**
* **Batch Issuance: Allow institutions to issue multiple certificates in one transaction.**
* **QR Code Verification: Generate QR codes for easy mobile verification.**

**Long-Term**

* **Cross-Chain Compatibility: Support Polygon, Binance Smart Chain for lower fees.**
* **Decentralized Identity (DID): Link certificates to self-sovereign identities.**
* **API for Institutions: REST API for universities/companies to automate issuance.**

**Conclusion**

**This project demonstrates how blockchain can revolutionize certificate verification by providing a secure, transparent, and decentralized solution. Future work will focus on scalability and real-world adoption.**