

Roll No.:.....

Amrita Vishwa Vidyapeetham, Coimbatore Campus

B.Tech Lab Examinations #2 – August 2025

Fifth & Seventh Semester

Electrical and Electronics, Electrical and Computer, Communication and Computer, Electronics and Communication, Computer Science and Engineering (Cyber Security)

24CYS336 Blockchain Technology

Duration: One Hour 30 Minutes

Maximum: 30 Marks

Course Outcomes

CO	Course Outcomes
CO1	Understand the basic principles of Distributed Ledger Technology
CO2	Able to demonstrate the cryptographic primitives in Blockchain technology
CO3	Understand and evaluate various consensus protocols
CO4	Develop Smart Contracts and create a DApp using Ethereum Blockchain
CO5	Analyze a real-world use case and provide how blockchain could be used to address the challenges faced

Answer all questions.

Decentralized Academic Certificate Storage System using Solidity and IPFS.

Educational institutions often issue digital certificates, but ensuring authenticity and immutability is a major challenge. Your task is to develop and deploy a **Solidity Smart Contract** on the Ethereum test network that interacts with **IPFS** for decentralized certificate storage and retrieval.

- Design and implement a smart contract named **CertificateRegistry** that allows:
 - The **Admin (Deployer)** to register institutions and manage authorized users.
 - **Registered Institutions** to issue certificates by storing metadata (student name, course, grade, and IPFS hash of the certificate file).
 - **Students** to view their certificate details and verify authenticity using the IPFS hash.
- Functional Requirements:**
 - The contract must define three roles:
 - **Admin** – contract deployer.
 - **Institution** – approved by Admin.
 - **Student** – registered by Institution.
 - Use **modifiers** to restrict:
 - Admin-only functions.
 - Institution-only functions.
 - Use a mapping:

```
mapping(address => Certificate) studentCertificates;
```

to store a certificate per student for many such students.

(d) Each certificate should have:

- i. `studentName`
- ii. `course`
- iii. `grade`
- iv. `ipfsHash`
- v. `issuedOn`

(e) Implement the following functions:

- `addInstitution(instAddress, instName)` – Admin adds a new institution.
- `issueCertificate(student, name, course, grade, ipfsHash)` – Institution issues a certificate.
- `getCertificates(student)` – Anyone can view all certificates of a student.
- `verifyCertificate(student, ipfsHash)` – Returns true if the given IPFS hash exists for that student.

(f) Certificates once issued cannot be modified.

(g) Deployment: Deploy the contract using your Metamask account (Admin address).

3. IPFS Integration:

- Upload a certificate file (PDF or image) to IPFS manually to get the CID/IPFS Hash.
- Obtain the IPFS hash and use it in the smart contract during certificate issuance.

4. Test Case Scenarios

(a) **Admin Tests:**

- Admin successfully adds an institution.
- Non-admin tries to add institution – should revert.

(b) **Institution Tests:**

- Authorized institution issues certificate with valid IPFS hash.
- Unauthorized address issues – should revert.

(c) **Student Tests:**

- Student retrieves certificate.
- `VerifyCertificate` returns `true` for valid IPFS hash.
- `VerifyCertificate` returns `false` for invalid hash.

5. Submission Instructions

- Submit your Solidity file (`CertificateRegistry.sol`).
- Include:
 - Deployment screenshot and contract address.
 - Certificate issue transaction hash.
 - IPFS file link (gateway URL).
 - Output of test cases.

* * * * *

Course Outcome / Bloom's Taxonomy Level (BTL) Mark Distribution Table

CO	Marks	BTL	Marks
CO1	0	BTL-1	0
CO2	0	BTL-2	0
CO3	0	BTL-3	0
CO4	30	BTL-4	0
CO5	0	BTL-5	30
-	-	BTL-6	0