VEGA Hackathon – INDEX.HTML

Secure submission & tamperproof integrity system for NTEC hackathons (Problem CB-03). This deck presents the problem, proposed SHA-256 based solution, innovation, feasibility, technical stack, and verification workflow—designed to prevent post-deadline tampering and ensure fair judging.

Problem Statement: Submission Integrity at Risk

NTEC runs multiple national-level hackathons but the current submission process cannot guarantee project authenticity. Teams can modify files after deadlines, swap code in last moments, or present altered work without detection. Disputes over originality and unverified Git history undermine fairness.

Why the Current System Fails

Dependence on File Uploads & Links

Submissions rely on basic uploads and GitHub links that can be changed or replaced without cryptographic proof.

No Cryptographic Validation

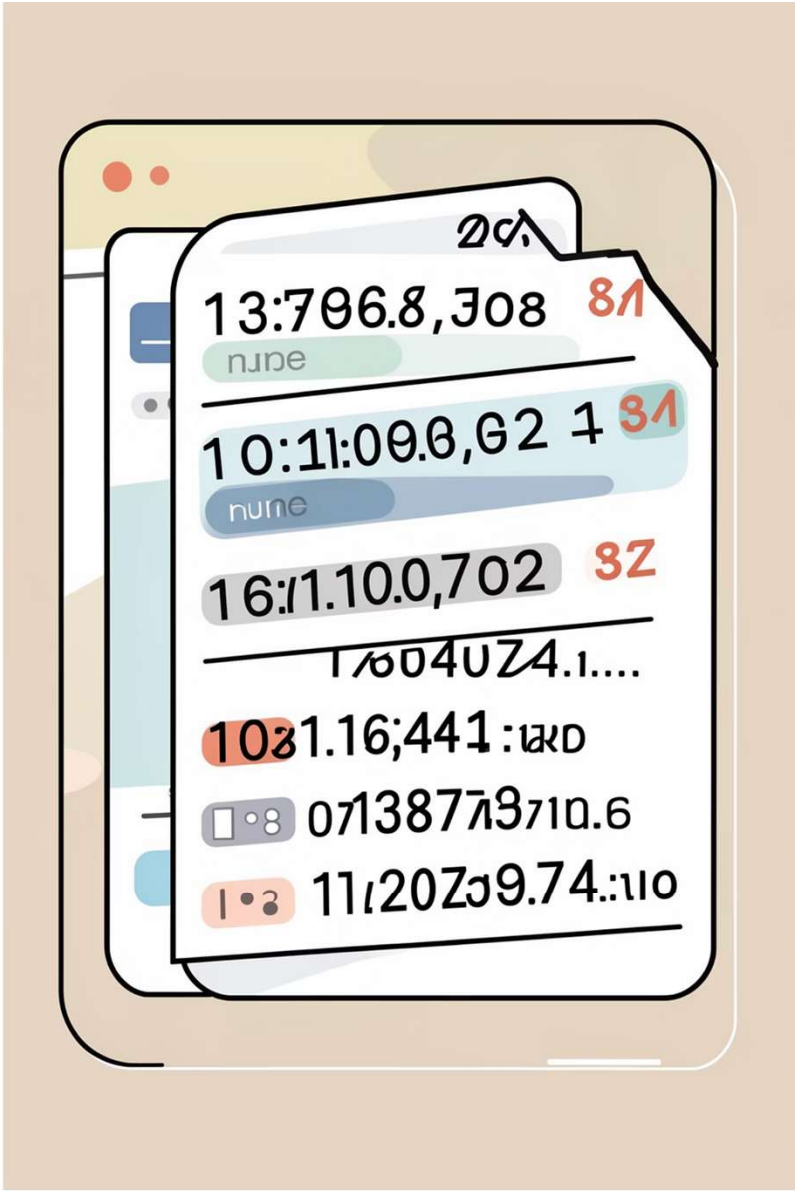
There is no immutable fingerprint or mathematical evidence tying files to a fixed moment in time.

Manual Verification Only

Human checks are slow, error-prone, and inadequate for scaling national events.

No Trusted Timestamps

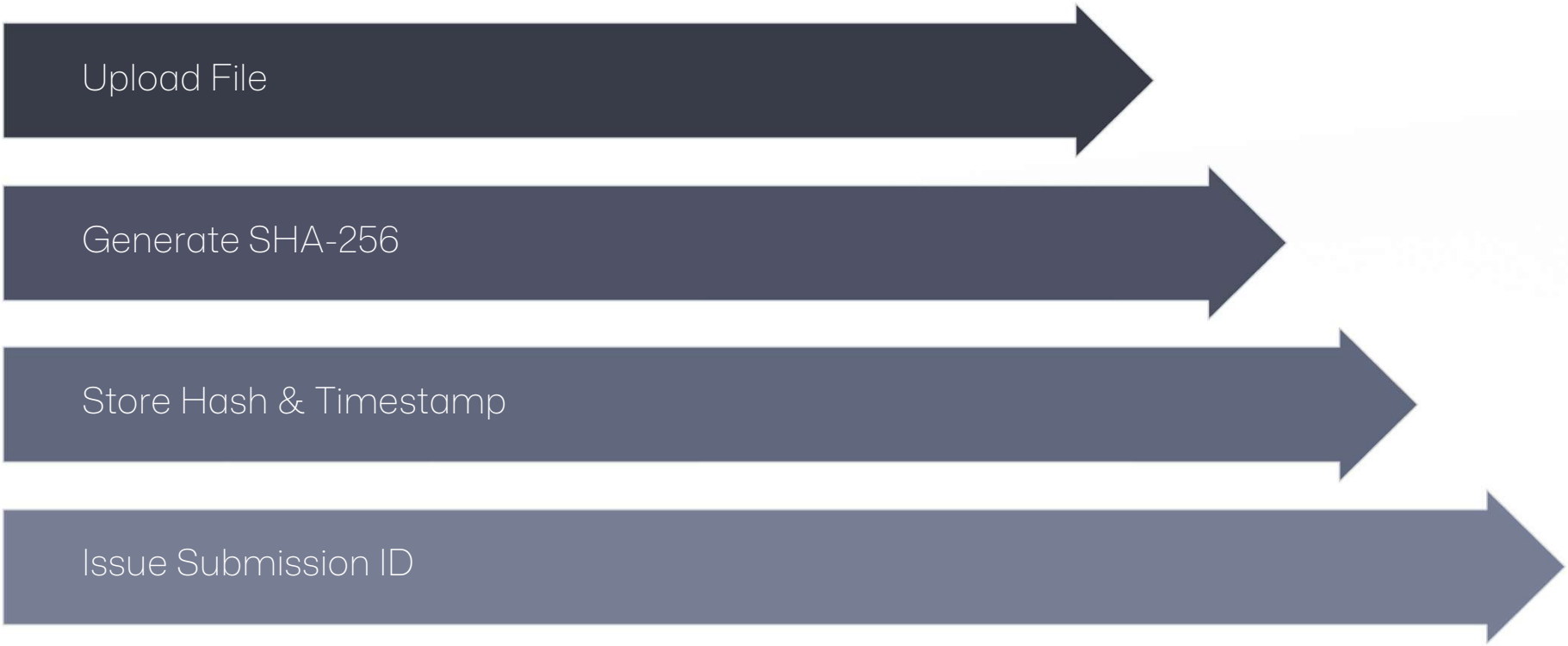
Without reliable timestamps, it's impossible to prove that a file existed unchanged at submission time.



Proposed System – Overview

A secure portal that generates a SHA-256 cryptographic hash for every uploaded file, stores it with a trusted timestamp, and issues a unique Submission ID. Judges verify integrity by recomputing and comparing hashes; any 1-byte change produces a different hash and is instantly detectable.

How It Works – Step-by-Step



This workflow ensures that each uploaded artifact is cryptographically fingerprinted and time-anchored. Verification is deterministic: recompute the SHA-256 hash and compare to stored value; mismatch indicates tampering.

Key Features & Security Properties

Tamper Detection

SHA-256 fingerprints detect any file alteration, even a single-byte change.

Unique Submission ID

Each submission receives an immutable identifier linked to its hash and timestamp for traceability.

Timestamp Validation

Trusted timestamps prove when a hash was recorded, preventing post-deadline replacements.

Role-Based Access

Separate Team and Judge roles with controlled permissions for submitting, reviewing, and verifying artifacts.

Optional Blockchain Anchoring

Anchor hashes to Ethereum (or other chain) for an extra immutable public proof layer when needed.

Innovation & Uniqueness

This approach replaces simple storage with cryptographic fingerprinting, delivering mathematical proof of integrity rather than trust-based verification. It reduces disputes, scales with event size, and optionally leverages blockchain anchoring for public immutability—distinct from manual or metadata-reliant systems.

Feasibility – Practical & Achievable

1

Proven Crypto

Uses standard SHA-256—well-tested, industry-accepted algorithm with negligible collision risk for this use case.

2

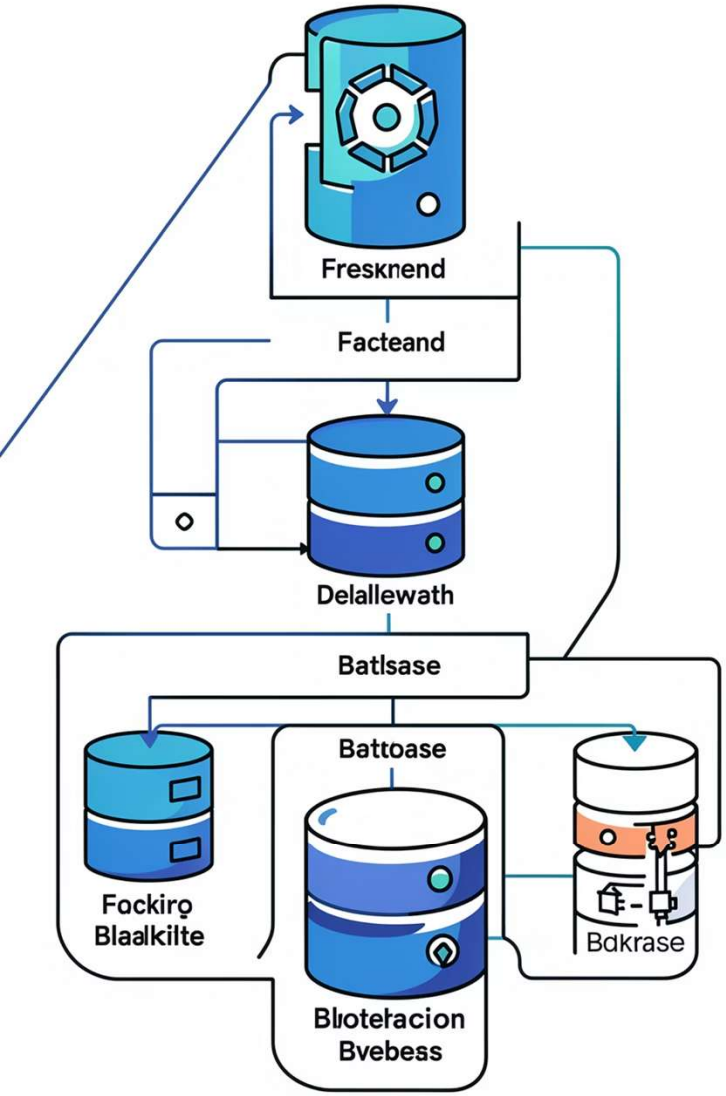
Low Infrastructure Cost

Backend hashing and storage are lightweight; scalable cloud deployment fits hackathon timelines.

3

Mitigations for Challenges

Large files handled via chunked hashing; secure transport with HTTPS and robust access controls; rare collision risk mitigated by industry standards.



Technology Stack Overview

Frontend

React + Vite for fast, responsive user experience and polished submission UI.

Backend

Python + Flask for secure APIs, file handling, and cryptographic hashing services.

Blockchain Layer

Ethereum smart contract optional anchoring to create a public, immutable timestamp for submission hashes.

Core Technologies

SHA-256 hashing, REST API communication, secure database storage, HTTPS transport, role-based access control.

Next Steps & Implementation Roadmap

01

1. Prototype (Week 1–2)

Build upload portal, implement SHA-256 hashing, store hashes with timestamps, and issue Submission IDs for a pilot event.

02

2. Integrate Verification & Roles (Week 3)

Add judge verification UI, role-based access controls, and chunked upload support for large files.

03

3. Optional Blockchain Anchoring (Week 4)

Implement Ethereum anchoring for public immutability and perform security testing, then deploy cloud hosting for scale.

04

Outcome

Deliver a tamperproof, scalable submission system that provides mathematical proof of integrity and restores fairness to NTEC hackathons.