Lab Report - Non-Fungible Token Manager using unique_ptr

Student Name: Ahmed Abdelghany

Lab Name: Manage Non-Fungible Tokens with unique_ptr in C++

Overview

This lab involved designing a non-fungible token (NFT) system using smart pointers in C++. It included minting NFTs with pseudo-unique hashes, managing digital wallets, and transferring tokens securely using std::unique_ptr. A benchmark was also performed to compare the performance of different integer types (int, long int, and bigint).

Key Functionalities

- Mint Token: Each token is created with a unique hash based on the asset name and timestamp.
- Transfer Token: Moves token ownership from one wallet to another using std::unique_ptr.
- Benchmark Test: Compared computation times for int, long int, and bigint.

Timing Comparison Chart

Data Type	Time (µs)
int	235
long int	225
bigint	2491

**Note: Results from 100,000 iterations using chrono benchmarking.*

Why int Fails for NFT IDs

Using simple integers risks:

- ID collisions
- Predictable values
- No cryptographic integrity

This is why secure hashing (or bigint for large IDs) is preferred in real-world tokenization.