National College of Ireland

Configuration Manual

LedgerLock

Forensic Evidence Verification Framework Application

MSc Research Project (Thesis)

Cybersecurity

Sajid Hussain Jatoi

Student ID: X23265736

School of Computing

National College of Ireland

Supervisor: Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	SAJID HUSSAIN JATOI
Student ID:	X23265736
Programme:	MSc Cybersecurity **Year:** 2025
Module:	**MSc (Research) Practicum/Internship (Thesis)**
Supervisor:	ROSS SPELMAN
Submission Due Date:	11 DECEMBER 2025
Project Title:	Configuration Manual
Page Count:	**9**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sajid Hussain Jatoi

Date: 11 December 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

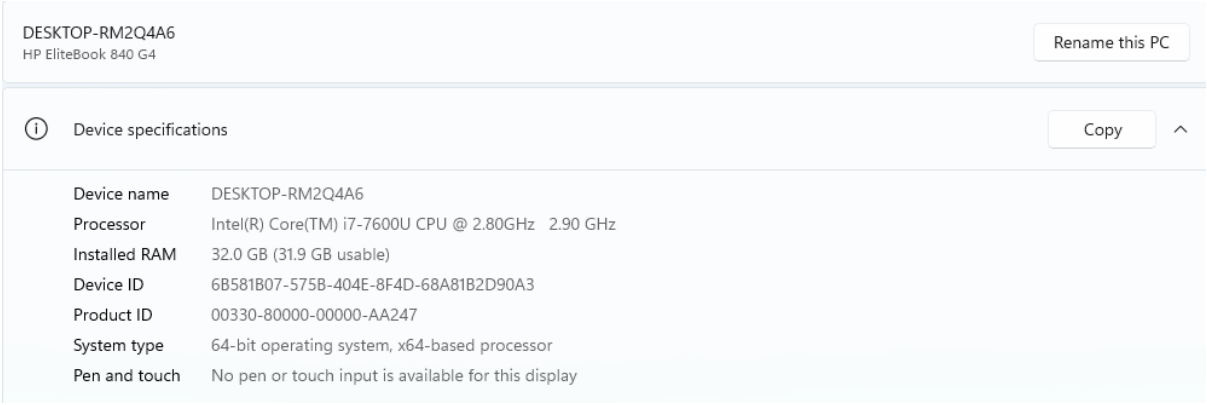
Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

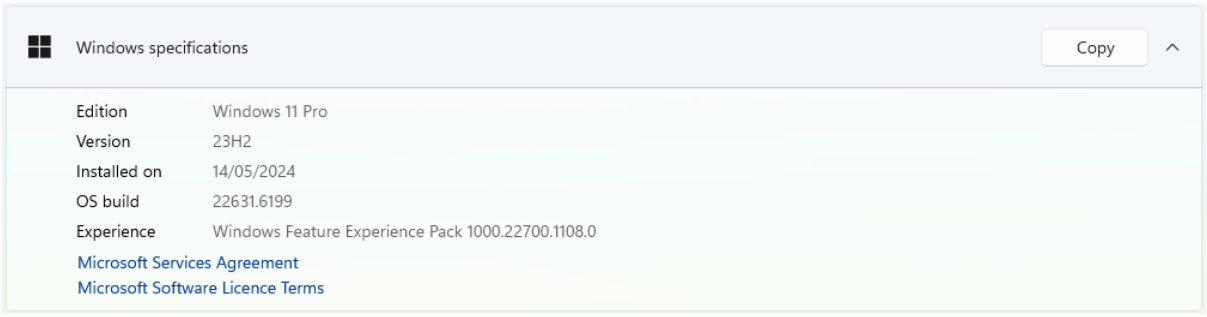
Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

1. Introduction

This configuration manual contains all the data and information of the system used, requirements in this research. This manual outlines the detail description of **data collection, code snippets of data preprocessing, model implementation and evaluations**.





2. System Requirements

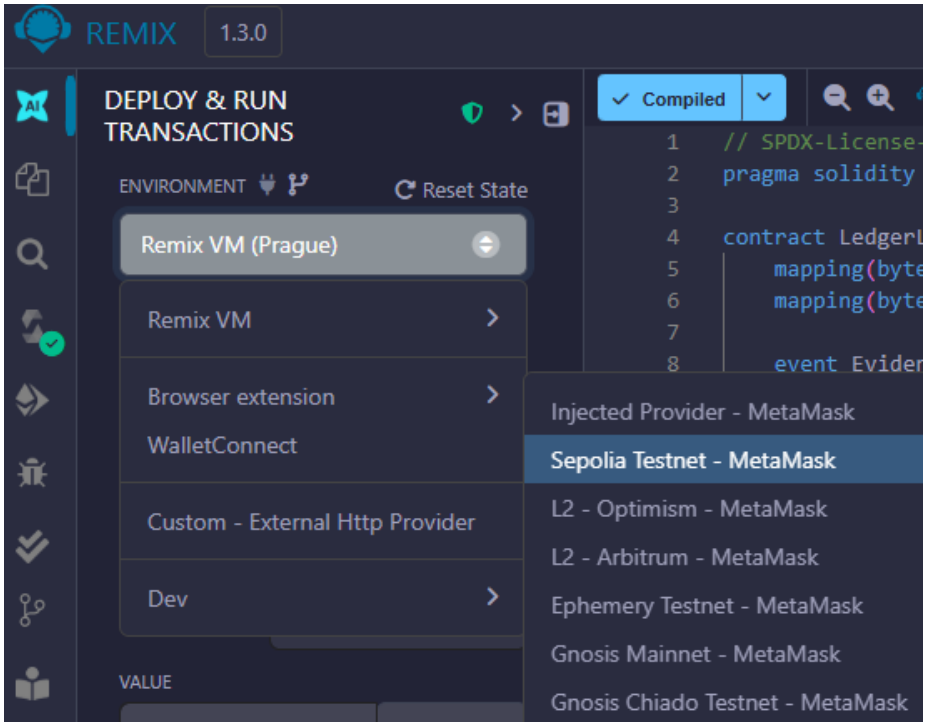
- **Operating System:** Windows 11
- **Programming Language:** Python 3.11 with dependencies (web3, fpdf, tkinter, pyinstaller, requests).
- **RAM:** Minimum 4 GB
- **Disk Space:** 500 MB free for logs, reports, and offline queue storage.
- **Network:** Internet connection required for online anchoring and verification; offline mode supported as well for differed synchronization
- **Blockchain Integration:** Smart contract deployment on Remix IDE, MetaMask wallet browser plugin and wallet
- **Sepolia testnet network:** Infura API key and address

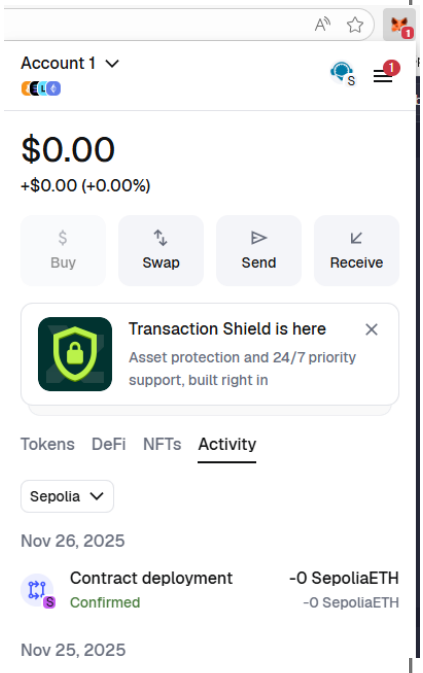
3. Implementation

3.1 Smart Contract Deployment

Smart contract was deployed using Online Remix IDE with MetaMask Wallet using browser extension method

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract LedgerLock {
    mapping(bytes32 => string) public evidence;
    mapping(bytes32 => string) public txByHash;

    event EvidenceRegistered(bytes32 indexed hash, string description, string tx_id);

    function registerEvidence(bytes32 hash, string memory description, string memory tx_id) public {    infinite gas
        require(bytes(evidence[hash]).length == 0, "Evidence already exists");
        evidence[hash] = description;
        txByHash[hash] = tx_id;
        emit EvidenceRegistered(hash, description, tx_id);
    }

    function getEvidence(bytes32 hash) public view returns (string memory) {    infinite gas
        return evidence[hash];
    }

    function getTxId(bytes32 hash) public view returns (string memory) {    infinite gas
        return txByHash[hash];
    }
}
```





3.2 Anchoring Evidence to the blockchain

Application Binary Interface (ABI) interacting with the deployed smart contract and use as inputs metadata mapping using Web3.py load and call the below functions in snippets.

```
# ----------------------Blockchain Setup-----------------------------------------------

INFURA_URL = "https://sepolia.infura.io/v3/de22498fc7104273a9f6a65fc385a621"
CONTRACT_ADDRESS = "0xce1c98c4B7A85Ca2B6d10cB19dB5D44d3080097E"
PRIVATE_KEY = "0x7b3d21685a9147dae57fcb996761b538271bee0d31e98dad29099bad6b210dde"
CHAIN_ID = 11155111

ABI = [
    {
        "inputs": [
            {"internalType": "bytes32", "name": "hash", "type": "bytes32"},
            {"internalType": "string", "name": "description", "type": "string"},
            {"internalType": "string", "name": "tx_id", "type": "string"}
        ],
        "name": "registerEvidence",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [{"internalType": "bytes32", "name": "hash", "type": "bytes32"}],
        "name": "getEvidence",
        "outputs": [{"internalType": "string", "name": "", "type": "string"}],
        "stateMutability": "view",
        "type": "function"
    }
]

# ----------------------Initialize Web3 safely-----------------------------------------
try:
    w3 = Web3(Web3.HTTPProvider(INFURA_URL))
    account = w3.eth.account.from_key(PRIVATE_KEY)
    contract = w3.eth.contract(address=CONTRACT_ADDRESS, abi=ABI)
except Exception as e:
    messagebox.showerror("Blockchain Error", f"Web3 initialization failed: {e}")
    w3, account, contract = None, None, None
```

- CHAIN_ID – a unique identifier assigned to each block used to prevent the replay attacks, ensure contracts are assigned for the correct chain block. This 11155111 is used by Sepolia tesnet network.

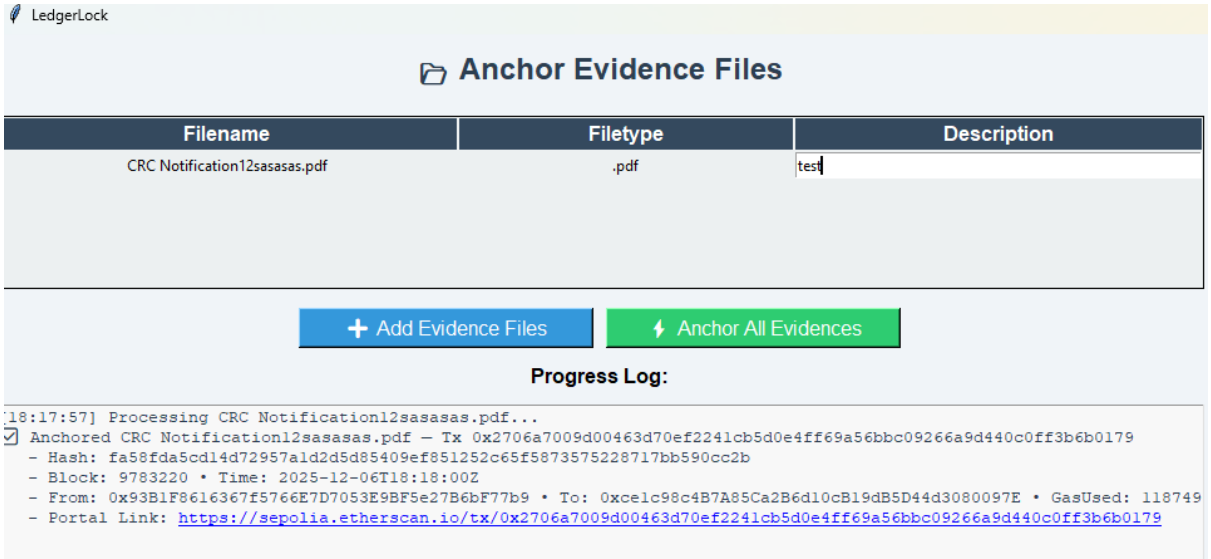
4. Application User Interface

Launch LedgerLock application and use the functionality accordingly

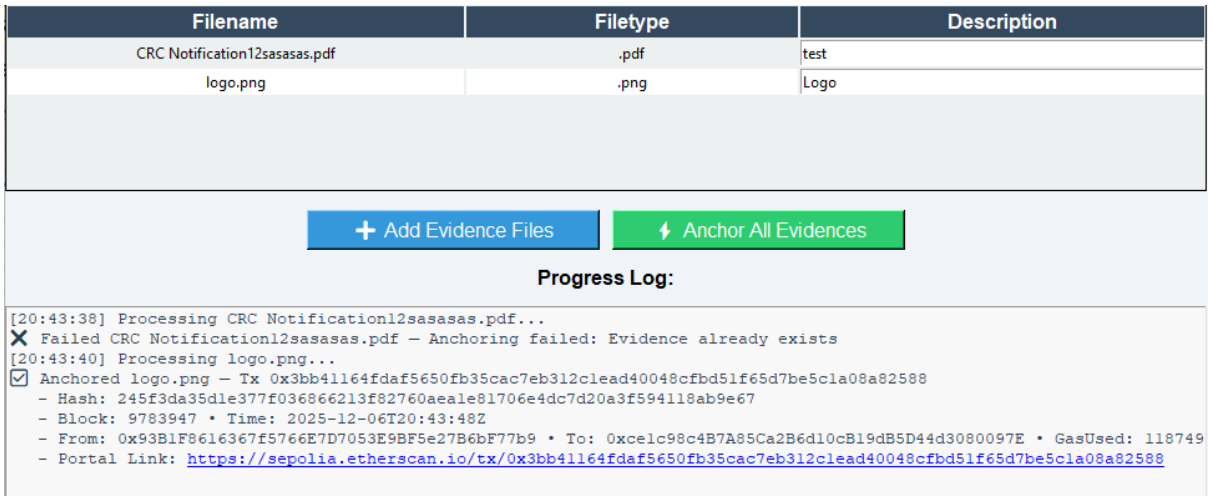
4.1 Main Dashboard

4.2 Anchor Digital Records

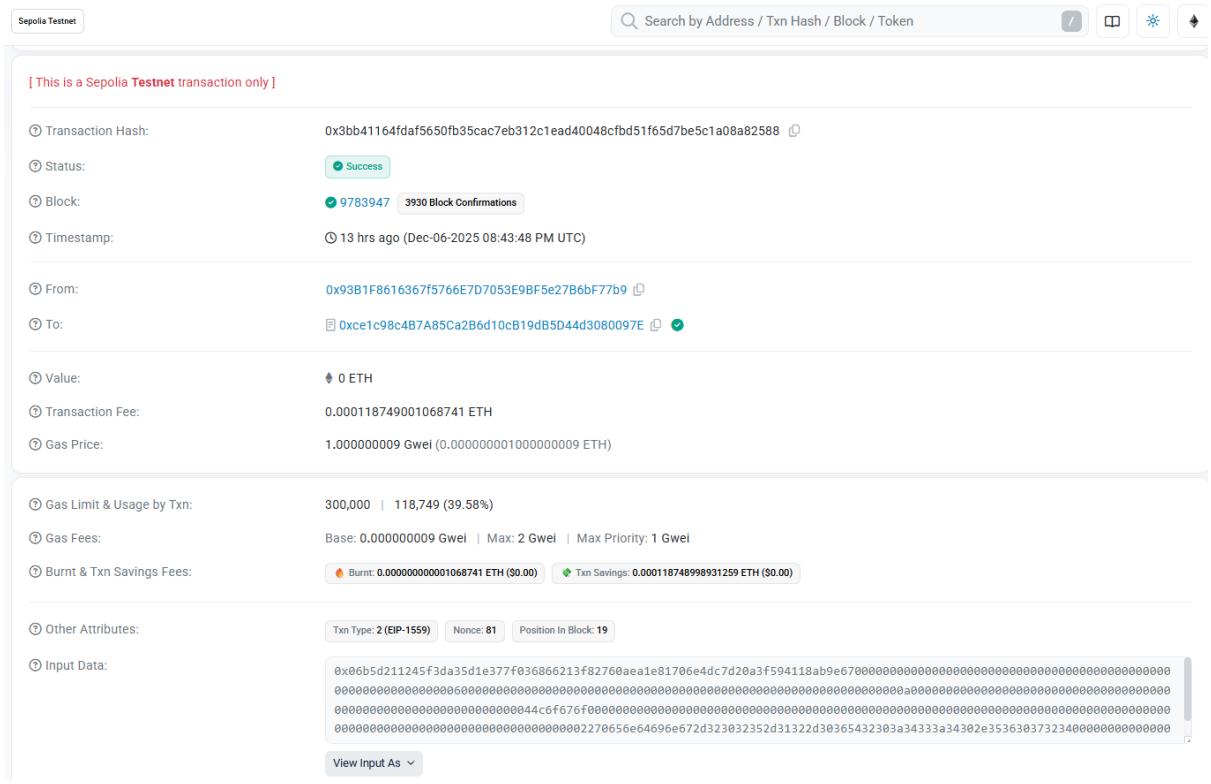
Add Single / Batch of records – provide description against each record and once ready, click on the Anchor All Evidence. It will provide all interactive process logs in logs section for the user to update information real-time. It will first check if evidence exist already by checking the Keccak-256 hash code if doesn't it will provide all the meta-data and store this same for audit-trail on Sepolia Blockchain and export report.



If evidence already exist it will show an error and proceed for the next one, but it's designed to notify to the analyst in process logs and exported information as well.

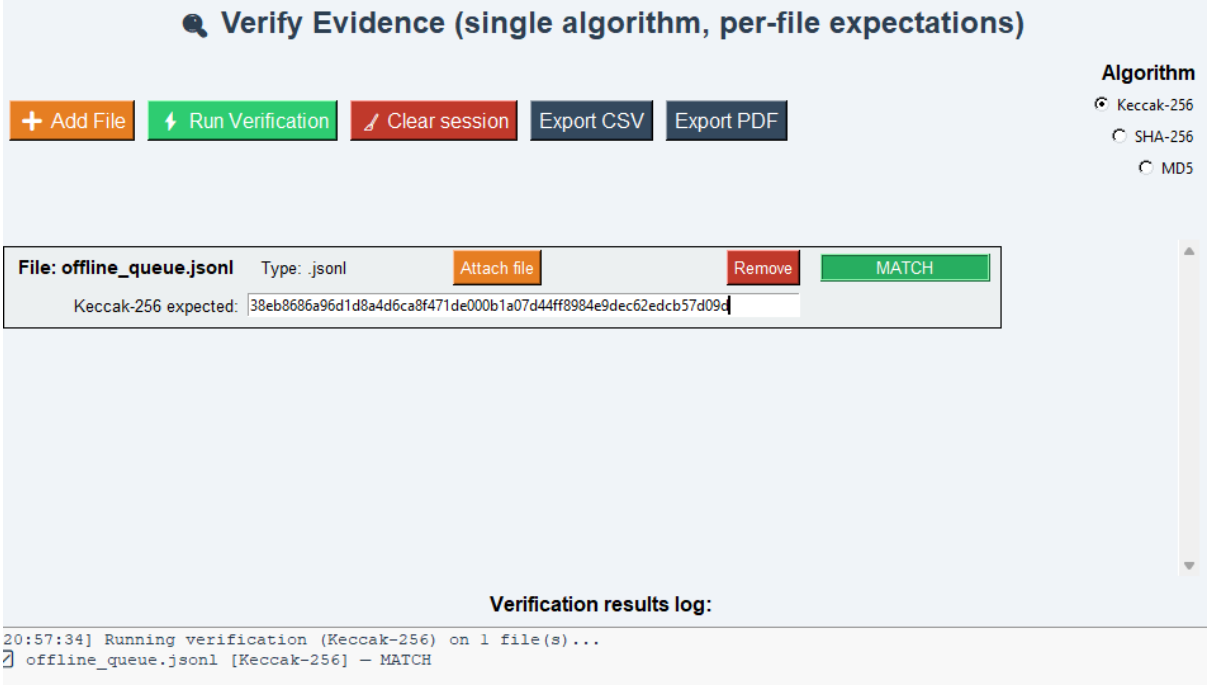


Record confirmation of Sepolia testnet blockchain

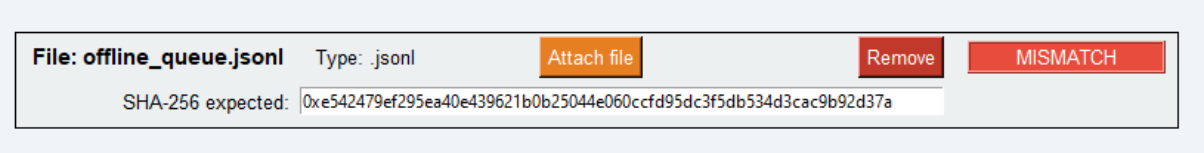


4.3 Verification of digital records

This functionality demonstrates that if the file and expected hash code matches than it proves the authenticity of the evidence and if it doesn't, than it will show that document hash either not related with the document or document is tampered.

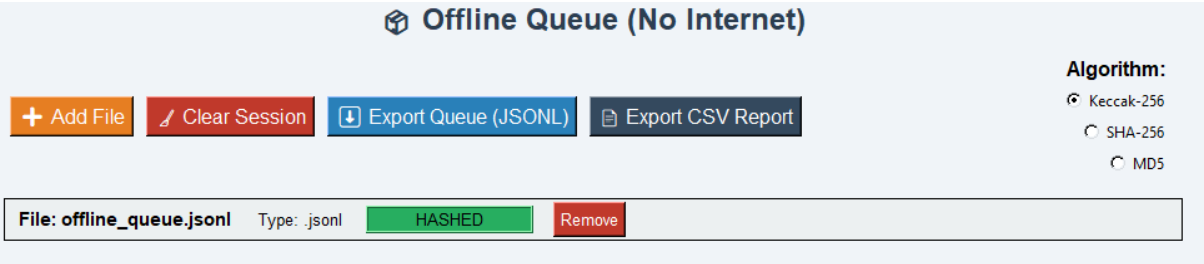


If doesn't match system will show the message as Mismatch



4.4 Work Offline

Application allows to compute hashes as per choice and provide to export the JSON or CSV to upload later to anchor the evidence.

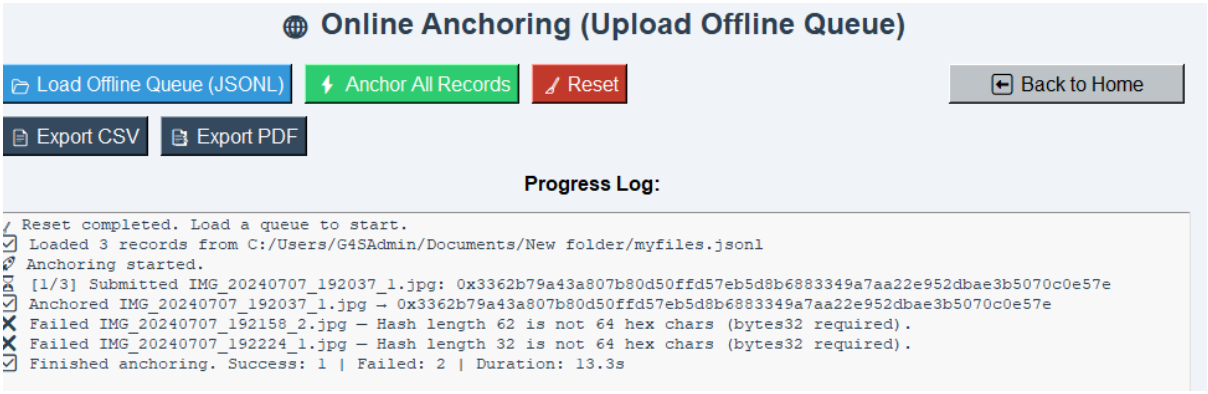


Data recorded in the JSON file - This contain exactly what is required to be uploaded and store information on the blockchain portal.


```
{"evidence_id": "81763731-9bd5-46b4-ae0b-e5a003bee800", "filename": "offline_queue.jsonl", "filetype": ".jsonl", "hash_algorithm": "Keccak-256",
"hash_value": "38eb8686a96d1d8a4d6ca8f471de000b1a07d44ff8984e9dec62edcb57d09d", "collected_at": "2025-12-06T20:56:39.525448"}
```

4.5 Upload Files

This will upload the same format file that was exported during the work offline. While anchoring the data to store on blockchain it will check if compute hash and evidence already exist or not if does, this will show that how many uploaded as below.



4.6 Reports

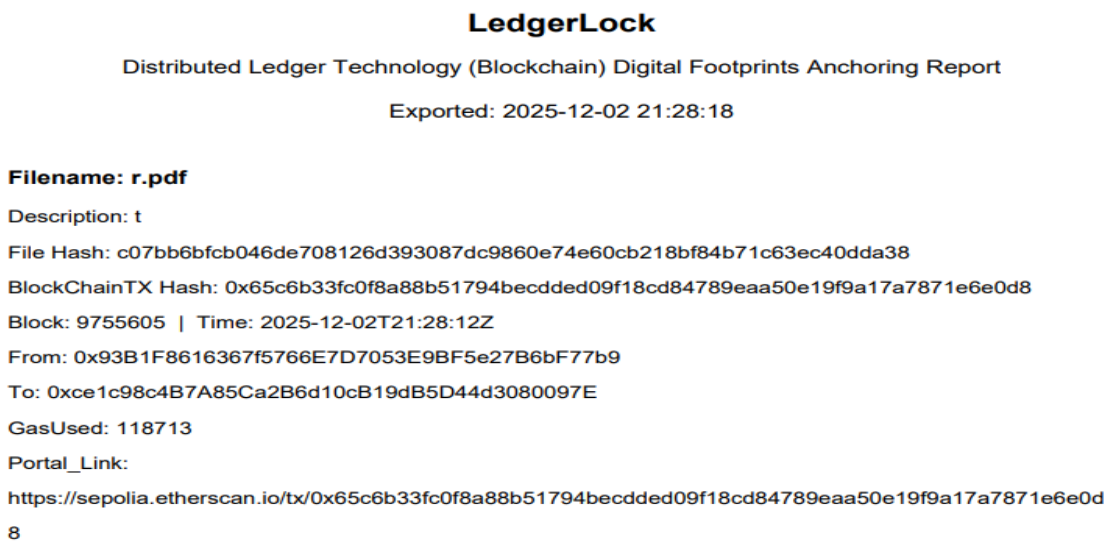
LedgerLock application provide two options currently to export repots in cvs or pdf format. This is available in each section to export and show data accordingly.

Report contains headers such as (Filename, Description given while anchoring the evidence, File unique hash code Keccak-256, Blockchain Transaction hash, Block used, Timestamp, from which account To the address, Gas Used as Transaction fees, Portal Link – URL to click and verify that on the Etherscan Portal).

- CSV – Format

filename	hashHex	txHash	blockNum	timestamp	gasUsed	from	to	portalLink	status	reason
1.pdf	9b92d37a	0xdb7a32	9783298	2025-12-0	96295	0x93B1F8	0xce1c98c4B7A85Ca2B6d10cB19dB5D44d3080097E	https://sepolia.etherscan.io/tx/0xdb7a3233661a546f9695c5693624d157bb4eb672cf3a7c2a747c9d3a0e42e580	SUCCESS	
CRC Notification12sasasas.pdf	0xe252b3c	0xd425617	9783299	2025-12-0	141365	0x93B1F8	0xce1c98c4B7A85Ca2B6d10cB19dB5D44d3080097E	https://sepolia.etherscan.io/tx/0xd42561750d8dbcd23f4dcf017ea8ab1d4bc0bcc6fddb4fb7879412d3dd0be644	SUCCESS	
r.pdf									FAILED	Hash length 32 is not 64 hex cha

- PDF Format



LedgerLock

Distributed Ledger Technology (Blockchain) Digital Footprints Anchoring Report

Exported: 2025-12-02 21:28:18

Filename: r.pdf

Description: t

File Hash: c07bb6bfcb046de708126d393087dc9860e74e60cb218bf84b71c63ec40dda38

BlockChainTX Hash: 0x65c6b33fc0f8a88b51794becdded09f18cd84789eaa50e19f9a17a7871e6e0d8

Block: 9755605 | Time: 2025-12-02T21:28:12Z

From: 0x93B1F8616367f5766E7D7053E9BF5e27B6bF77b9

To: 0xce1c98c4B7A85Ca2B6d10cB19dB5D44d3080097E

GasUsed: 118713

Portal_Link:

https://sepolia.etherscan.io/tx/0x65c6b33fc0f8a88b51794becdded09f18cd84789eaa50e19f9a17a7871e6e0d8