CapStone_Stability - Evidence Blockchain App

A cross-platform desktop application (Windows & MacOS) for securely submitting image evidence to the blockchain.

Project Overview

According to the previous email, the main objectives of this project are to:

- Ensure integrity and non-repudiation of digital evidence.
- Provide a simple workflow for capturing and registering evidence.
- Use blockchain for immutable timestamping and record storage.
- Offer a public verification tool that confirms image authenticity without exposing the original image.

Besides, the application should be a desktop app compatible with both Windows and MacOS. The requirements and core features include:

- Capture a snapshot of the entire screen or a selected region.
- Generate a SHA-256 hash of the captured image.
- Submit the hash, timestamp, and optional metadata to the blockchain smart contract.
- Optionally store a local record containing the snapshot, hash, and transaction ID.
- Allow users to view previously submitted evidence records.

General Design

Languages & Frameworks

- Solidity - Smart Contract Development
- JavaScript - Application Logic
- *Uncertain - UI Framework (e.g., Electron, React)*

Workflow

1. Upload Image

Click the upload area or drag and drop an image, and the SHA-256 hash will be automatically generated.

2. Add Metadata (Optional)

Enter any additional information about the evidence, such as:

- Description
- Location
- Case number
- Date of capture
- and more.

They will all be encrypted and stored alongside the hash on the blockchain.

3. Submit to Blockchain

After configure the blockchain settings (RPC URL, contract address, wallet address and private key), users can record the evidence on-chain simply in one click.

Alternatively, click "Save Locally Only" to store the evidence without blockchain submission.

4. View Records

Switch to the "View Records" tab to:

- Browse all submitted evidence
- View hash, timestamp, and transaction details
- Copy hashes for verification
- Mark a record, e.g., "Reviewed", "Important", "Deprecated", etc.

Smart Contract

The application is designed to work with the `EvidenceRegistry` smart contract. See `src/contracts/EvidenceRegistry.sol` for the contract source code.

Project Structure (Current Design)

```
├── src/
│   ├── main/
│   │   ├── main.js          # Electron main process
│   │   └── preload.js       # Preload script for IPC
│   ├── renderer/
│   │   ├── index.html       # Main UI
│   │   ├── styles.css       # Styling
│   │   └── renderer.js      # Renderer process logic
│   └── contracts/
│       └── EvidenceRegistry.sol  # Smart contract
├── __tests__/
│   └── hash.test.js         # Unit tests
├── package.json
└── README.md
```

Current Progress

- [F] Still in planning phase
- [D] In development
- [C] Completed but not yet tested
- [T] Tested and verified

[C] Smart contract Implementation.
[T] Basic hash generation and image handling.
[D] Blockchain interaction and transaction submission.
[D] Desktop application UI design and implementation.

[F] History record viewing and management.
[F] Desktop packing and distribution.