

Blockchain-Based Job Posting Application

Overview

This application implements a blockchain-like structure to store job postings securely. It follows blockchain principles by linking blocks cryptographically, ensuring immutability, and enabling verification of data integrity. Each job posting is stored as a block, containing details such as job title, company, location, and description. Blocks are added sequentially, each referencing the previous block's hash, forming a chain.

Blockchain Principles Implemented

1. **Immutable Data Storage:** Each block contains a SHA-256 hash, preventing unauthorized modifications.
2. **Chaining Blocks:** Each block references the previous block's hash, ensuring an ordered and tamper-resistant structure.
3. **Genesis Block:** The blockchain starts with a predefined "Genesis Block," serving as the foundation of the chain.
4. **Tamper Detection:** A validation function recalculates hashes to detect unauthorized modifications.
5. **Data Transparency:** The blockchain allows for easy retrieval and verification of job postings.

Cryptographic Hashing Mechanism

The application uses **SHA-256** hashing for data integrity. Each block's hash is computed using:

- The block index
- Timestamp
- Previous block's hash
- Job details (title, company, location, description)

Importance of Hashing

- **Ensures Data Integrity:** Any modification in a block alters its hash, invalidating the chain.

- **Prevents Tampering:** Since each block references the previous block's hash, changes to a block require recalculating all subsequent hashes, making unauthorized modifications detectable.
- **Enables Verification:** The `validate_blockchain` function ensures each block's hash matches the stored hash, preventing data corruption.

Challenges and Resolutions

1. Correct Memory Management

- **Challenge:** Properly allocating and freeing memory for blockchain structures to prevent leaks.
- **Resolution:** Implemented `free_blockchain` to traverse and free all allocated memory correctly.

2. Correct Data Types for Variables

- **Challenge:** Ensuring each field in the structure used the correct data type to store relevant data.
- **Resolution:** Used appropriate types such as `unsigned char` for hashes, `char` arrays for text fields, and `int` for indexing.

3. Buffer Sizing and String Handling

- **Challenge:** Ensuring buffers were large enough to store strings while avoiding overflows.
- **Resolution:** Used `strncpy` with size constraints and null-terminated strings to prevent buffer overflows.

Conclusion

This application successfully demonstrates blockchain principles in job posting management. By leveraging cryptographic hashing and block chaining, it ensures data integrity and prevents unauthorized modifications. The challenges faced in memory management, data typing, and buffer handling were resolved through careful structuring and validation techniques, ensuring robustness and security in the implementation.