**Report Outline: Blockchain-Based Facial Recognition Degree Authentication System**

**1. Introduction**

**Background and Significance**  
Degree verification has long been a critical process in academia and employment, ensuring that individuals possess the qualifications they claim. However, traditional methods often rely on centralized institutions or manual verification processes, which present several challenges:

1. **Forgery and Fraud:** Fake diplomas and forged academic credentials are pervasive issues, undermining trust in the certification process and damaging institutional reputations.
2. **Inefficiency:** Verifying degrees through conventional methods can be time-consuming, involving multiple steps and significant manual effort, especially in cross-border scenarios.
3. **Privacy Concerns:** Centralized storage of academic records increases the risk of data breaches, leading to the potential exposure of sensitive information.

Given these challenges, this project proposes a novel approach by integrating blockchain and facial recognition technologies. Blockchain provides a decentralized, immutable ledger, ensuring data transparency and security, while facial recognition offers robust identity verification, linking academic credentials directly to their rightful owner. Together, these technologies can revolutionize the degree authentication process, making it more reliable, efficient, and secure.

**Project Objectives**

To address the issues in traditional degree verification systems, the project aims to achieve the following objectives:

1. **Deploy Smart Contracts on Ethereum:**  
   Develop and deploy smart contracts on the Ethereum blockchain to record, store, and manage degree information. By leveraging the immutable nature of blockchain, the project ensures that academic credentials cannot be altered or tampered with.
2. **Integrate Facial Recognition Technology:**  
   Incorporate advanced facial recognition technologies, such as DeepFace, to establish a strong, reliable connection between a user's identity and their academic records. This integration not only enhances the authentication process but also prevents identity fraud by ensuring that only the rightful owner can verify their credentials.
3. **Enable Decentralization and Transparency:**  
   Design the system to function without reliance on a central authority, empowering various stakeholders, such as universities and employers, to verify degrees directly through a transparent and decentralized platform.
4. **Enhance Security and Privacy:**  
   Implement encryption techniques to protect sensitive user data, ensuring that personal information, including facial embeddings, remains secure and accessible only to authorized parties.

By achieving these objectives, the project aims to provide a scalable, global solution for degree authentication, capable of addressing the needs of educational institutions, employers, and individuals alike.

**2. System Features and Implementation**

**2.1 Current Features**

**Blockchain Platform:**  
The system is built on the Ethereum blockchain, leveraging its decentralized and immutable nature to securely store and manage degree-related data. Ethereum's smart contract functionality, developed using the Solidity programming language, allows for automated execution of degree-related processes.

* **Development Framework:** The Hardhat framework is utilized for efficient development and testing of smart contracts. Hardhat provides a local Ethereum environment for debugging and supports the deployment of smart contracts to test networks.
* **Smart Contracts:** These contracts handle key functionalities, including the storage of degree data, user verification, and voting mechanisms.

**Facial Recognition Module:**  
Facial recognition technology is integrated to link user identity with their academic credentials.

* **Technology Used:** The system utilizes **DeepFace**, a robust deep learning framework for facial embedding generation and comparison.
* **Process:** When a user uploads a photo, DeepFace processes it to generate a unique embedding. This embedding acts as a biometric identifier and is stored alongside degree data on the blockchain for future verification.

**Data Storage:**  
Degree information and facial embeddings are stored directly on the Ethereum blockchain to ensure data immutability and transparency.

* **Data Structure:** The degree information is stored in a structured format within the smart contract:

solidity

Copy code

struct Degree {

string studentId; // Unique identifier for the student

string degree; // Degree title (e.g., Bachelor of Science)

string university; // Issuing institution

string issuedDate; // Date of issuance

string faceEmbedding; // Encoded facial feature vector

}

mapping(string => Degree) private degrees;

This structure ensures that each entry is tied to a unique identifier and is easily queryable.

**Frontend and Backend Architecture:**  
The system architecture comprises a user-friendly interface and a robust backend for seamless interaction with the blockchain and facial recognition module.

* **Frontend:** The user interface, built using HTML, CSS, and JavaScript, provides features such as data entry, photo uploads, and viewing verification results.
* **Backend:** The backend is powered by **Node.js** and **Python Flask**, which serve as middleware for blockchain interactions and facial recognition operations.

**2.2 Workflow Description**

**Degree Upload:**  
Educational institutions or authorized users upload degree data to the blockchain.

* **Process:** The uploader inputs the student's degree details and submits their facial embedding. The data is recorded in the smart contract, ensuring its immutability.
* **Authentication:** To prevent malicious data entries, each upload must be verified through a voting mechanism, ensuring credibility.

**User Verification:**  
Users seeking degree verification interact with the system by uploading a photo for authentication.

* **Process:**
  1. The photo is processed by the DeepFace module to generate a facial embedding.
  2. The generated embedding is compared with the stored embedding on the blockchain.
  3. If the embeddings match, the system retrieves and displays the corresponding degree information.
* **Outcome:** The user is either authenticated successfully or denied access based on the match results.

**Voting Mechanism:**  
To maintain data integrity and trust, the system employs a decentralized voting mechanism for approving degree data uploads.

* **Process:**
  1. Each new degree entry requires approval from at least three voters (e.g., institution representatives or other authorized stakeholders).
  2. The voters review the submitted data and cast their votes.
  3. Only after receiving the required number of approvals is the degree data officially stored on the blockchain.
* **Benefits:** This mechanism ensures that data tampering or unauthorized uploads are minimized, promoting transparency and reliability in the system.

This combination of advanced technologies and structured workflows ensures that the degree authentication system is both secure and user-friendly, addressing key challenges in the traditional verification process.

**3. System Architecture**

**3.1 Data Storage Design**

In this system, degree information and facial embeddings are directly stored on the Ethereum blockchain, ensuring data immutability and security. By leveraging the decentralized nature of Ethereum, the system prevents unauthorized alterations or tampering with stored data, ensuring its trustworthiness.

* **Degree Data Storage:** Degree information, including the student's unique ID, degree type, university name, issue date, and facial embedding, is recorded in the blockchain. Each degree record is immutable once stored, providing a transparent, auditable, and verifiable system of academic credentials.
* **Facial Embeddings Storage:** The facial embeddings, which are generated by the facial recognition module (DeepFace), are also stored on the blockchain. These embeddings are unique representations of the user's facial features and are essential for the verification of identity during the authentication process.

**Advantages:**

1. **Transparency:** Since all data is stored on the blockchain, it is visible to all participants, ensuring that all transactions and records are auditable.
2. **Decentralized Storage:** The blockchain eliminates the need for a central authority, reducing the risk of single points of failure or data manipulation. Multiple nodes ensure the data is replicated and secure, enhancing system resilience.
3. **Immutability:** Once the data is stored on the blockchain, it cannot be altered or deleted, ensuring the integrity and reliability of the stored degree information.

**Limitations:**

1. **Privacy Concerns:** Storing facial embeddings directly on the blockchain may expose sensitive personal information. While the embeddings are unique identifiers, they can still be linked to an individual’s biometric data, raising potential privacy risks. Techniques like encryption or storing only hashed versions of facial embeddings can mitigate this concern.
2. **Scalability Issues:** Ethereum’s storage capabilities, while secure, can become costly and inefficient for large-scale systems. As the number of users increases, the cost and complexity of storing large volumes of data (like high-resolution facial embeddings) on-chain may grow.

**3.2 Smart Contract Architecture**

Smart contracts serve as the backbone for data storage, querying, and verification within the system. These contracts are deployed on the Ethereum blockchain and handle all the logic and automation necessary for the system’s operation.

**Degree Management:**

* **Storage:** The smart contract stores degree-related information, such as the student’s ID, degree type, issuing university, issue date, and facial embeddings, ensuring that this information is both secure and immutable.
* **Querying:** The contract allows authorized users (such as educational institutions, employers, or individuals) to query stored degree information, ensuring that users can confirm the validity of academic credentials.
* **Verification:** The smart contract provides a mechanism for verifying the authenticity of degree records. When users request verification, the contract checks if the provided information matches the stored data.

**Voting Mechanism:**

* **Approval Process:** Each degree upload requires the approval of at least three authorized voters. These voters could be educational institution representatives or other trusted authorities who review and verify the submitted data.
* **Decentralized Control:** This voting system ensures that no single entity has sole control over the degree data, preventing fraudulent uploads and enhancing the integrity of the stored records. Only after receiving the necessary approvals is the degree information officially recorded on the blockchain.
* **Transparency:** The voting process is transparent, and all actions are recorded on the blockchain, making the process fully auditable and tamper-proof.

**3.3 Facial Recognition Module**

The facial recognition module, powered by **DeepFace**, plays a critical role in authenticating users and linking their identities to their academic credentials.

**Process:**

1. **Face Embedding Generation:** Users upload a photo, which is processed by the DeepFace algorithm to generate a facial embedding—a unique vector representing the user’s facial features.
2. **Embedding Comparison:** During the verification process, the system compares the newly generated embedding with the one stored on the blockchain. If they match, the user is authenticated and granted access to their degree information.
3. **Blockchain Storage:** The facial embedding is stored on the blockchain as part of the user’s degree record. This ensures that the facial recognition data is securely associated with the user’s academic credentials and is immutable.

**Benefits:**

* **Secure Authentication:** The use of facial recognition technology provides an additional layer of security, ensuring that only the individual who holds the degree can authenticate it.
* **Identity Linkage:** The system uses facial embeddings to tightly link the user's identity with their academic credentials, preventing identity fraud and ensuring the authenticity of degree verification.

**3.4 System Components**

The system is comprised of several interrelated components that work together to provide degree authentication services. These components are divided into three main modules: blockchain, frontend, and backend.

**Blockchain Module:**

* **Primary Function:** The blockchain module is responsible for storing and managing degree data, as well as facilitating the decentralized voting mechanism for degree verification.
* **Interaction:** This module interacts with the smart contracts, ensuring that all transactions related to degree data are recorded securely and immutably. It also handles querying and verification requests from authorized users.

**Frontend:**

* **User Interface:** The frontend, built with HTML, CSS, and JavaScript, serves as the primary interface for users to interact with the system.
* **Features:** It allows users to upload their photos for facial recognition, view degree information, and see the results of the verification process. The frontend also provides an intuitive and user-friendly experience for interacting with the blockchain.
* **Interaction with Backend:** The frontend communicates with the backend to trigger facial recognition processes and retrieve relevant data from the blockchain.

**Backend:**

* **API Services:** The backend, powered by Node.js and Python Flask, serves as the intermediary between the frontend, blockchain, and facial recognition system.
* **Blockchain Interaction:** It handles the communication with Ethereum via smart contracts, ensuring that requests for data storage, querying, and verification are properly executed.
* **Facial Recognition Integration:** The backend manages the DeepFace module, processes user-uploaded photos, generates facial embeddings, and ensures that they are properly stored and matched against the blockchain data.

Each of these components plays a crucial role in the smooth functioning of the system, contributing to its transparency, security, and overall effectiveness. The system’s modular design allows for scalability, flexibility, and easy integration of additional features in the future.

**4. Current Features Demonstration**

**Degree Data Storage:** Example contract:

solidity

Copy code

degrees["studentId123"] = Degree(

"studentId123",

"Bachelor of Science",

"Example University",

"2024-12-01",

"faceEmbedding12345..."

);

**User Verification Process:**

* Users upload photos, generate embeddings, and verify against blockchain data.

**Voting Mechanism:**

* Degree uploads require a minimum of three voter approvals.

**5. Extended Features**

**5.1 Data Storage Optimization**

* Store degree data on-chain, while facial embeddings are hashed and stored in distributed storage like IPFS.

**5.2 Model Upgrade Support**

* Enable upgrades via smart contracts, ensuring compatibility with older data.

**5.3 Privacy and User Authorization**

* Encrypt user data and implement Zero-Knowledge Proof (ZKP) for privacy protection.

**5.4 Internationalization and Compliance**

* Support GDPR and CCPA regulations with decentralized identifiers (DIDs) for global authentication.

**6. Expected Deliverables**

**System Demonstrations:**

* Screenshots of degree upload and verification interfaces.
* A video showcasing the authentication process and blockchain interaction.

**Technical Deliverables:**

* Complete smart contract code.
* Access to an open-source GitHub repository.

**Result Data:**

* Successfully stored degree data and authentication logs.

**7. Team Contributions**

* **Hui Zifan (24046598g): PPT preparation and smart contract development.**
* **Li Dongwei (24117496g): Smart contract development, Metamask integration, and Hardhat-based development.**
* **Liu Qingyuan (24052432g): Backend development (Node.js) and DeepFace integration.**

**8. Conclusion**

**Achievements:**

* Successfully implemented a blockchain-based degree authentication system.
* Validated core functionalities on the Ethereum test network.

**Future Directions:**

* Enhance security through storage optimization, encryption, and model upgrades.
* Expand to support international regulations for a more secure and practical solution.
* Provide a transparent, trustworthy, and efficient global degree verification system.