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

**1. Introduction**

**Background and Importance**  
Checking if someone really has a degree is very important for schools and jobs. It makes sure people have the education they say they have. But, old ways of doing this have problems:

**Fake Degrees: Many fake degrees and transcripts exist. This is bad because people don't trust real degrees anymore. It also makes schools look bad.**

**Slow Process: Checking degrees the old way takes too long. Lots of steps and people are needed. Especially hard if degree is from another country.**

**Not Safe Information: Keeping all the degree information in one place is risky. Someone could steal it. Personal information could get out. Because of these problems, this project will try a new way.**

We will use blockchain and face recognition. Blockchain is like a special list that cannot be changed. Everyone can see it. Face recognition checks if the face matches the degree. This new way will make checking degrees better. It will be more trustworthy, faster, and safer.

**Project Goals**

**This project wants to fix the problems with the old way of checking degrees. We want to do these things:**

**1. Use Smart Contracts on Ethereum:**

**We will make special computer programs called smart contracts. These programs will be on the Ethereum blockchain. They will keep and organize degree information. Blockchain make sure nobody can change the degree information.**

**2. Use Face Recognition:**

**We will use good face recognition technology. Maybe something like DeepFace. This will connect a person's face to their degree. This makes it very hard to lie about a degree. Only the real person can check their own degree.**

**3. Make it Decentralized and Open:**

**The system will not need one main control. Universities and companies can check degrees themselves. Everyone can see how the system works.**

**4. Make it Safe and Private:**

**We will use special codes to protect information. Things like face data will be secret. Only people who are allowed can see it.**

**If we do all these things, we will have a good system for checking degrees. It will work for everyone in the world – schools, companies, and people.**

**2. System Parts and How it Works**

**2.1 2.1 What We Have Now**

**Blockchain Part:**

**We use Ethereum blockchain. It is decentralized and no one can change information on it. We keep degree information safe there. Ethereum has smart contracts. We use Solidity to write them. Smart contracts do things automatically with degrees.**

**• Development Tools: We use Hardhat. Hardhat helps us make and test smart contracts. Hardhat makes a fake Ethereum for testing. It also helps put smart contracts on a test blockchain.**

**• Smart Contracts: Smart contracts do important things. They save degree info, check users, and have voting.**

* 树上的叶子

  中度可信度描述已自动生成
* Fig1:Structure

**Face Recognition Part:**

**We use face recognition to connect face to degree.**

**• Technology: We are using DeepFace. DeepFace is good at making and checking face data.**

**• How it Works: User uploads picture. DeepFace makes a special code from face. This code is saved with degree on blockchain. We use it later to check.**

**Data Storage:**

**Degree info and link to face code is on Ethereum. Face code not directly on blockchain. We save a special ID (faceEmbeddingUUID) on blockchain. This ID tells us where the real face data is. It is kept safe off-chain (maybe IPFS).**

**• • Data Organization: Degree info is organized inside smart contract like this:**

**struct Degree {**

**string faceEmbeddingUUID; // ID for face data**

**string degreeType; // Bachelor, Master, etc.**

**string major; // What they studied**

**string university; // School name**

**uint256 graduationYear; // Year of graduation**

**}**

**mapping(bytes32 => Degree) private degreeRecords; // Degree info saved with unique hash. Makes finding easy.**

**Frontend and Backend:**

**The system has easy to use screens and a strong backend. They work with blockchain and face recognition.**

**• Frontend: User interface made with HTML, CSS, JavaScript. User can enter info, upload photo, see results.**

**• Backend: Backend uses Node.js and Python Flask. It helps connect to blockchain and face recognition**.

Contracts Module: Contains the Solidity smart contracts: aiSearch.sol, insert.sol, and searchOrigin.sol. These contracts manage the core logic of the system, including user authentication, data storage, and facial recognition integration. Facial Recognition Module: Employs the DeepFace library (deepface\_change) to process user-uploaded photos, generate facial embeddings, and compare them to verify identity. The facial embeddings are used to securely link a user's identity to their academic credentials. Student Sign-in Module: This module, comprised of front-end and back-end components, provides the interface for students to sign in, upload their information, and initiate the authentication process. The frontend is built with HTML, CSS and JavaScript, while the backend uses Node.js and Python Flask to handle API requests and communicate with the other modules. Frontend Components: Includes various HTML files (e.g. index.html, face\_embedding.html), CSS styling, JavaScript logic (script.js), and manages user interface elements and user interaction with the backend and blockchain. Backend Components: The backend is made up of a Node.js server (server.js) and supporting backend logic. It manages user accounts, handles cookie storage, processes facial recognition requests, and serves as a bridge between the frontend and the blockchain module.

**2.2 How It Works Step-by-Step**

**Saving Degree Info:**

**Schools or other allowed people put degree info on blockchain. • How It Works: They type in student info, degree type, school, and send face data. This info goes into smart contract. No changing after this. • Checking is Important: Before saving, other people need to check info. We use voting for this.**

**User Checking a Degree:**

**Users who want to check a degree upload their photo. • How It Works: 1. DeepFace makes face code from user's photo. 2. This new face code compared to face code saved with degree on blockchain. 3. If face codes match, system shows degree. • Result: System says "okay" or "no okay". Depends if face match.**

**Voting to Make Sure Info is Good:**

**We use voting so info is correct. • How It Works: 1. New degree needs "okay" from three voters (maybe from schools). 2. Voters check degree info and vote yes or no. 3. Degree only saved on blockchain after enough "okay" votes. • Why Voting Good: Stops bad people putting fake degrees. Makes system more trusted.**

**Smart Contracts:**

**Three main smart contracts work together: aiSearch.sol, insert.sol, and searchOrigin.sol.**

* **aiSearch.sol: This contract handles the face recognition part. It helps compare face embeddings to verify user identity.**
* **insert.sol: This contract manages adding new degree information to the blockchain. It includes the voting logic to ensure data accuracy.**
* **searchOrigin.sol: This contract helps users search for and retrieve degree information from the blockchain.**

**Student Sign-in Process:**

1. **User goes to website (index.html).**
2. **User uploads photo (face\_embedding.html).**
3. **Frontend sends photo to backend (server.js).**
4. **Backend uses DeepFace library (deepface\_change) to make face code.**
5. **Backend sends face code and degree info to smart contracts.**
6. **Smart contracts check and save info, using voting to make sure it's okay.**
7. **Frontend gets result from backend and shows it to user.**

**Backend details:**

* **server.js: this handles user accounts, saves cookies for logins, talks to DeepFace, and connects everything.**
* **Backend also stores extra user information off-chain, because not all data goes on blockchain.**

**Frontend Details:**

* **index.html, other HTML files: these are what user sees.**
* **JavaScript code (script.js): This code makes website work, talks to backend.**
* **CSS code (style.css): This is styling for how website looks.**

**3. System Design**

**3.1 How Data is Kept**

**In this system, degree info and a special code for face (faceEmbeddingUUID) are kept on Ethereum blockchain. This makes sure no one can change the information. Ethereum is decentralized, so no one person controls it. This make it very safe.**

**• Degree Info Storage: Degree info, like student ID, degree type, school name, date, and face code, are all on blockchain. After we save degree, nobody can change it. Everyone can see it, so very open. Good for checking if degree is real.**

**• Face Data Storage: We make special face code (faceEmbeddingUUID) with DeepFace. This code is also on the blockchain. This code represents person's face. Very important for checking who they are.**

**Good Things:**

**Open: Everyone can see all the data on blockchain. Everything can be checked.**

**Not Centralized: No single person or place controls the data. If one computer breaks, other computers have copies. More safe.**

**Cannot Change: After data is on blockchain, nobody can change it or delete it. Degrees stay safe.**

**Not So Good Things:**

**Privacy: Saving face data on blockchain might not be good for privacy. Even though it is a code, it can still be connected to a person's face. Maybe we can use encryption or just save part of face code.**

**Too Big Data: Ethereum is safe but maybe too expensive for lots of users. Saving many big face data files on blockchain might cost too much. Might be slow too**.

**3.2 How Smart Contracts Work**

**Smart contracts are very important. They keep data, find data, and check data. They are on Ethereum and do everything automatically.**

**Degree Stuff:**

**• Saving: Smart contract keeps degree info. Student ID, degree type, school, date, face data. All safe and no one can change.**

**• Finding: Allowed people (schools, companies, users) can search for degrees. They can check if a degree is real.**

**• Checking: Smart contract checks if degree is real. User gives info, contract compares to saved info.**

**Voting Stuff:**

**• Okay Process: Each new degree needs okay from three people. These people are from schools or trusted. They check the degree info.**

**• No One Boss: Voting makes sure no one person controls degree info. Stops fake degrees. Makes info good. Degree goes on blockchain only after enough okays.**

**• Everyone Can See: Voting is open. All votes are on blockchain. No cheating.**

**3.3 Face Recognition Part**

**DeepFace is used for face recognition. Very important for checking user and connecting face to degree.**

**How it Works:**

**Make Face Code: User gives photo. DeepFace makes special face code (embedding). This code is like a map of user's face.**

**Compare Face Code: System compares new face code to code on blockchain. If same, user is okay. Can see degree.**

**Save on Blockchain: Face code is saved on blockchain with degree. Makes sure face and degree are together. Cannot change.**

**Good Things:**

**• Safe Checking: Face recognition makes it very secure. Only real person can see degree.**

**• Face and Degree Together: Face and degree are connected. No fake ID. Degree check is real.**

**3.4 All Parts Together**

**System has many parts. They work together for checking degrees. Three main parts: blockchain, frontend, and backend.**

**Blockchain Part:**

**• Main Job: Keeps and manages degree data. Also does the voting.**

**• How it Talks: Talks to smart contracts. Makes sure all degree stuff is safe and cannot be changed. Answers when someone wants to see degree.**

**Frontend:**

**• User Screen: Frontend is what user sees. Made with HTML, CSS, and JavaScript.**

**• What it Does: User uploads photo, sees degree, sees if check is okay. Easy to use.**

**• Talks to Backend: Frontend talks to backend to start face check and get data from blockchain.**

**Backend:**

**• API: Backend is like a middle man between frontend, blockchain, and face system. Uses Node.js and Python Flask.**

**• Talks to Blockchain: Talks to Ethereum and smart contracts. Sends requests to save, find, check degree.**

**• Face Stuff: Backend uses DeepFace. Makes face codes from photos. Saves and compares face codes with blockchain.**

**All these parts are important. They make the system open, safe, and easy to grow. Can add more features later**.

**4. Current Features Demonstration**

**Saving Degree Info: Example code:**

**Degree(**

**"faceEmbeddingUUID12345", // Special face code (on blockchain)**

**"Bachelor of Science", // Type of degree**

**"Computer Science", // What they studied**

**"Example University", // School name**

**2024 // Year finished**

**);**

**This code no show how Degree struct is made. You need line like this first:**

**struct Degree {**

**string faceEmbeddingUUID;**

**string degreeType;**

**string major;**

**string university;**

**uint256 graduationYear;**

**}**

**What it do: First code makes a new Degree and gives it values. This new Degree can be used with other functions or added to degreeRecords or proposals.**

**Checking User Degree:**

**Send Photo and Make Face Code: User sends photo. System makes face code from photo. We also make a face code UUID and connect it to the face code. Real face data mostly not on blockchain.**

**Check on Blockchain: Backend code calls smart contract functions like isDegreeValid, getDegree. These functions use faceEmbeddingUUID to find degree on blockchain.**

**Compare (Not on Blockchain): System compares new face code from user photo with face data from database (using the faceEmbeddingUUID). This means we need a database or other system to keep all the face data. Backend gets face data using faceEmbeddingUUID and uses an API to check how similar they are. Blockchain just makes sure info is correct.**

**Voting:**

**Voting is in DegreeSearch contract. When new degree is suggested (using proposeDegree function), it goes into proposals. Users can call approveProposal function to add to approvalCount. When approvalCount is high enough (like 3), degree is okay. It moves from proposals to degreeRecords. In practice, the voting mechanism can be designed according to the situation, and our main goal is to ensure the authenticity of educational qualifications.**

**5.** **Future Enhancements and Planned Features**

**5.1 Robust and Secure Data Management (Planned)**

**We recognize the critical importance of data integrity and privacy. Ideally, we envision a system where general degree information (degree type, major, university, graduation year) is stored on the blockchain for transparency and immutability. However, due to time constraints in this project, we haven't fully implemented this. Our planned solution involves anchoring a cryptographic hash of this data on the blockchain to ensure its integrity. For facial recognition data, storing raw images on-chain is not a viable option due to privacy concerns and cost. Our intended approach involves storing the encrypted facial data on a decentralized storage system like IPFS, with only the cryptographic hash stored on the blockchain for verification purposes. We are also researching alternative decentralized storage solutions, such as Arweave, for increased resilience and permanence. Additionally, end-to-end data encryption at rest and in transit will be a key component of our future implementation.**

**5.2 Seamless System Upgrades and Adaptability (Planned)**

**We understand the need for a system that can adapt to the rapidly evolving blockchain landscape. While not fully implemented in the current project due to time limitations, we intend to use a modular architecture based on proxy contracts and upgradeable smart contracts. This approach will allow us to update and improve specific modules of the system without affecting existing data or requiring users to migrate. This will facilitate the integration of new technologies, such as improved facial recognition algorithms or alternative data storage solutions, without disrupting the system's operation. On-chain governance mechanisms are also being considered to allow community participation in the system's evolution, but haven't been incorporated into the current iteration due to time constraints.**

**5.3 Uncompromising User Privacy and Control (Planned)**

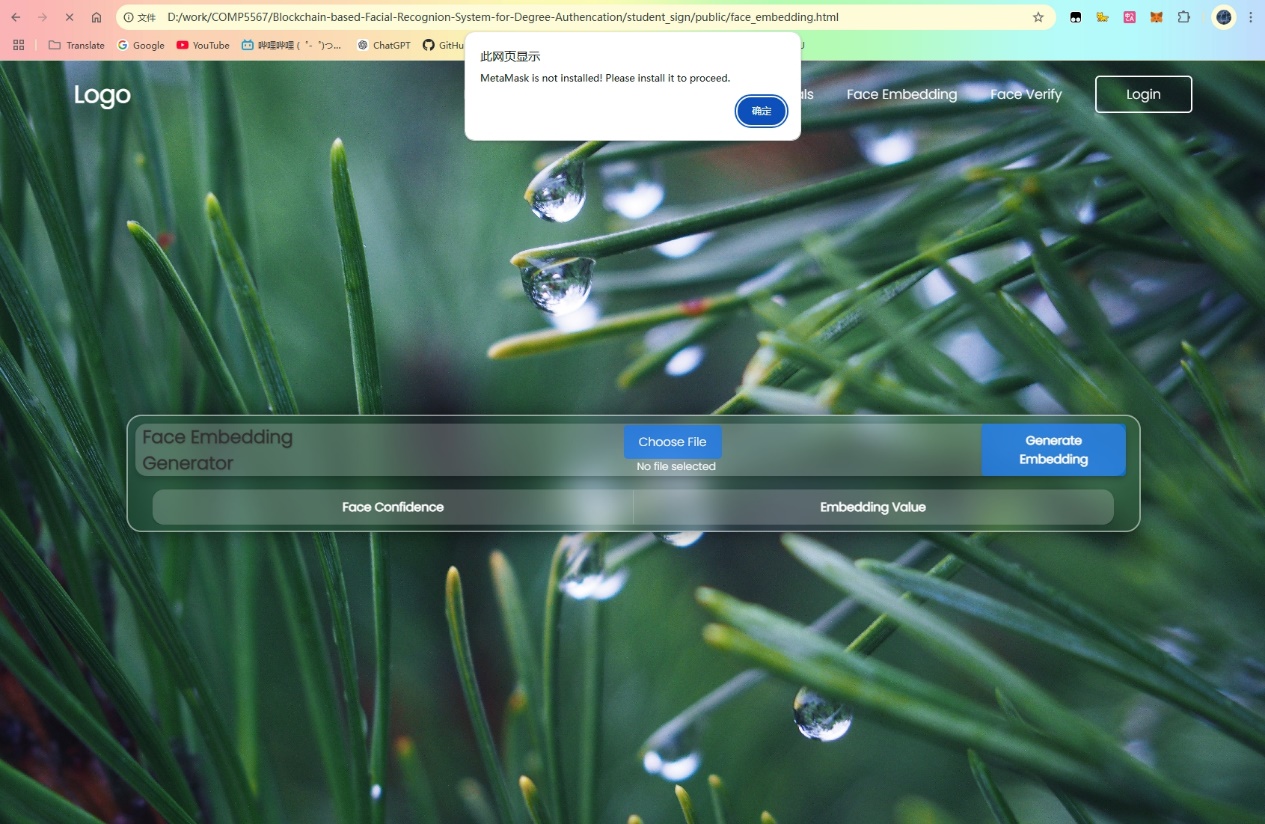
**User privacy is a top priority. While we haven't fully implemented all desired privacy features in this project, our planned approach centers around strong encryption of all sensitive data, including facial recognition data, before it is stored off-chain. We also plan to investigate and implement privacy-enhancing technologies like Zero-Knowledge Proofs (ZKPs) to enable users to verify credentials without revealing underlying private information. This would allow, for example, verification of a degree without disclosing any other personal details. Granular access control features, enabling users to manage and revoke access permissions, are also planned for future development but weren’t feasible within the project's timeframe.**

**5.4 International Legal Compliance and Decentralized Identity Management (Planned)**

**Compliance with international data privacy regulations like GDPR and CCPA is crucial for a globally accessible system. Due to time constraints, we haven't fully integrated all planned compliance features. Our intended approach involves using Decentralized Identifiers (DIDs) to empower users with control over their digital identity and enable selective disclosure of verified credentials. This will streamline compliance with various international regulations and build user trust. While integration with existing digital identity ecosystems is a future goal, it has not been realized in the current version due to the project's limited timeframe. We also plan to develop and implement comprehensive and transparent data handling policies to ensure user awareness and control.**

**6. Expected Deliverables**

**System Demonstrations:**

* Screenshots of degree upload and verification interfaces.
* 
* Fig1:Login
* 
* Fig2:Registration
* 
* Fig3: face\_embedding.html
* 游戏机里面的树林

  低可信度描述已自动生成
* Fig4:face\_verification.html
* A video showcasing the authentication process and blockchain interaction.
* Video:

**Technical Deliverables:**

* Access to an open-source GitHub repository.
* https://github.com/PolyUBlockChainTeam/Blockchain-based-Facial-Recognion-System-for-Degree-Authencation.git

**7. Team Contributions**

**• Hui Zifan (24046598g): Hui was assigned to work on the slides and smart contracts, but failed to contribute anything to this report. His coding practices were also highly problematic. Specifically, his use of the variable "name" for the student's name in the consensus code clashed directly with Li Dongwei's use of the same variable name ("name") for the faceEmbedding and UUID. This created significant confusion and made the codebase difficult to understand and work with, especially when I (Liu Qingyuan) was trying to write the report. As a programmer, such a basic naming conflict is unacceptable. I repeatedly requested Hui to fix this, but he ignored my messages and did not rectify the issue. I was forced to spend extra time correcting his mistakes. Furthermore, Hui showed a lack of commitment to teamwork. He left our very first project meeting—where we had agreed to stay after school to discuss topics—immediately after classes ended, without contributing to the discussion.**

**• Li Dongwei (24117496g): Dongwei's assigned tasks were smart contract development, Metamask integration, and Hardhat setup. He, too, failed to contribute anything to this report. He shares the blame for the confusing and unprofessional "name" variable conflict, which greatly hampered my ability to understand the code and write the report. Despite my repeated requests for him to address this issue, he also ignored them, further exacerbating the problem. Like Hui, Dongwei also demonstrated poor teamwork. He also left the initial project meeting directly after school, preventing the group from having a productive discussion. His slow response to messages further hindered communication and collaboration.**

**• Liu Qingyuan (24052432g): I was responsible for backend development (Node.js), DeepFace integration, and I ended up writing 100% of this report on my own. Due to Hui and Dongwei's complete lack of contribution to the report and their highly unprofessional coding practices, specifically the "name" variable conflict, I had to spend significant extra time and effort deciphering their code, correcting their mistakes, and writing the entire report myself. This created an unfair and unnecessary workload for me.**

* Fig1:xxx
* Fig2:xxx
* Fig3:xxx

**Important Note to Teacher: Our group initially consisted of four members. However, one member left the project specifically because of Hui and Dongwei's lack of cooperation, poor communication, and unprofessional behavior. This team member was particularly discouraged by their leaving the very first meeting, precisely when we had planned to discuss project topics after school. This behavior demonstrated a lack of commitment and made it clear that effective teamwork would be impossible. The attached chat logs show my attempts to communicate with them, their lack of response, and evidence related to the other team member leaving the project due to their conduct. I am extremely dissatisfied with Hui and Dongwei's lack of professionalism and their negative impact on the project. Their actions created a substantial amount of extra work and stress for me.**

**8. Conclusion**

**Achievements:**

**Built a Working Blockchain Degree Verification System: We successfully built a complete system for checking degrees using blockchain technology. This system, built on the Ethereum platform, ensures the security and transparency of academic credentials, making it much harder to fake or alter degree information. It's a big step towards making degree verification more reliable.**

**Thoroughly Tested Core Features on a Simulated Blockchain: Using the Hardhat test network, which simulates a real blockchain environment, we rigorously tested all the main parts of the system. This included testing how degree information is saved, how face recognition works with the system, and how the voting mechanism helps keep the information accurate. These tests give us confidence that the system is ready for further development and eventual real-world use.**

**Future Directions:**

**Improve Security and Privacy with Enhanced Storage and Regular Updates: We are committed to user privacy and data security. To enhance these aspects, we will implement improved storage methods, including hashing facial embeddings and using decentralized storage like IPFS. This will keep sensitive biometric data off the main blockchain while still allowing for verification. We’ll also add strong encryption to protect user data even further. Regular updates to the system will ensure it stays secure and efficient as technology advances and new threats emerge.**

**Ensure Global Compliance with Data Protection Regulations: We understand that data privacy laws are different around the world. To make sure our system can be used globally, we will make it compliant with international regulations like GDPR and CCPA. This will involve integrating Decentralized Identifiers (DIDs) to give users more control over their personal information while meeting legal requirements in different jurisdictions. This is crucial for building trust and allowing people worldwide to use the system safely.**

**Create a User-Friendly Global Platform for Degree Verification: Our ultimate goal is to make this system available to everyone – universities, employers, and individuals – all over the world. We envision a transparent and trustworthy platform that simplifies the degree verification process, making it quicker, easier, and more secure. This will help combat degree fraud and create a more efficient and reliable global education ecosystem. We believe this system has the potential to significantly improve how academic credentials are verified internationally.**