**Blockchain Based Data Integrity System**

Project Report

**Created By**: **Aditya Patil**

## **ACKNOWLEDGEMENT**

I would like to express my heartfelt gratitude for the opportunity to explore and develop this project as part of my academic journey in the field of Cyber Security and Blockchain Technology.

This project, \*\*Blockchain-Based Data Integrity System\*\*, represents a hands-on implementation of concepts such as decentralized storage, smart contracts, and cryptographic security. It helped me understand how blockchain can ensure data integrity and transparency, especially when combined with tools like IPFS and AES encryption.

The process of designing, developing, and testing this system has significantly deepened my technical knowledge and problem-solving skills. It has also given me a broader perspective on how decentralized technologies can solve real-world challenges related to trust and tamper-proofing.

I am thankful for the learning resources, documentation, and open-source tools that enabled me to complete this project successfully. Every piece of feedback, every bug fixed, and every line of code contributed to my growth.

This repository is a result of that effort, and I hope it serves as a useful reference for others interested in similar domains.

* Aditya Patil

# **INDEX**

|  |  |
| --- | --- |
| **Title** | Page No |
| 1. Introduction to System | 1 |
| 2. Need for a System | 2 |
| 2.1 Problem Statement |  |
| 2.2 Objective of the System |  |
| 2.3 Scope of the System |  |
| 3. Requirement Analysis | 3 |
| 3.1 Feasibility Study |  |
| 3.2 Fact Finding Techniques |  |
| 3.3 Hardware & Software Specifications |  |
| 4. System Analysis | 6 |
| 4.1 UML Diagrams |  |
| 5. System Design | 9 |
| 5.1 User Interface |  |
| 6. Limitations & Conclusion | 13 |
| 7. Future Enhancements | 15 |
| 8. Bibliography | 16 |

## **1. INTRODUCTION TO SYSTEM:**

In today’s digital world, where storing and sharing data is common, ensuring the safety and correctness of information has become a big challenge. As we rely more on digital systems, problems like data corruption, tampering, and unauthorized changes can lead to loss of trust and efficiency. These issues call for a strong system that can guarantee data is secure and has not been altered.

Blockchain technology is a powerful solution to these problems. It works as a decentralized, tamper-proof ledger that does not depend on a single authority, making it more secure and reliable. By using cryptographic hashing, blockchain creates a unique fingerprint for each piece of data. If anyone tries to change the data, the fingerprint changes, making tampering easy to detect. Its distributed system also provides backups, reducing the risks linked to centralized storage.

This project uses blockchain to build a data integrity system. It encrypts sensitive data before uploading it to decentralized storage systems like IPFS (InterPlanetary File System). The blockchain stores only the hash (fingerprint) of the data, ensuring its authenticity while keeping the actual content private. Users can easily check if their data is unchanged by comparing a re-generated hash with the one stored on the blockchain.

This system is especially useful for applications that need a high level of trust, such as securing academic certificates, protecting medical records, and tracking goods in supply chains. By creating a tamper-proof and clear record of data changes, this project shows how blockchain can improve data security and reliability in a simple and effective way.

## **2. NEED FOR A SYSTEM:**

### **2.1 Problem Statement:**

### Ensuring data integrity is crucial in today's digital landscape, as compromised data can lead to financial losses, reputational harm, and legal issues. Challenges such as malicious attacks, human errors, technical failures, and storage vulnerabilities threaten the accuracy and reliability of data. Traditional data storage methods often lack robust security measures, making data susceptible to unauthorized access and tampering. These issues highlight the need for a system that ensures data integrity and security throughout its lifecycle.

### **2.2 Objective of the System:**

* A tamper-proof mechanism to verify the authenticity of data.
* Secure storage of data hashes on the blockchain to ensure immutability.
* A user-friendly interface for uploading data, encrypting it, and verifying its integrity.
* Enhanced privacy by encrypting data before uploading it to a decentralized storage solution like IPFS.

### **2.3 Scope of the System:**

* **Data Integrity and Security**: Use blockchain to ensure data integrity by encrypting and securely storing it on decentralized platforms like IPFS.
* **Decentralized Storage**: Leverage IPFS for secure, accessible data storage, reducing risks of centralized systems.
* **Tamper-Proof Verification**: Record data hashes on blockchain for users to verify authenticity and integrity.
* **Future Scalability and Integration**: Ensure scalability for real-world deployment and integrate with private blockchains, IoT, and AI for enhanced security.

## **3. REQUIRMENT ANALYSIS:**

### **3.1 Feasibility Study**

#### **3.1.1 Technical Feasibility**

1. **AES Encryption:** AES is a widely adopted symmetric encryption standard known for its efficiency and security, making it suitable for protecting sensitive data before storage or transmission.​
2. **IPFS Integration:** IPFS is a peer-to-peer protocol that enables content-addressed, distributed file storage. It allows files to be stored across a network of nodes, ensuring data availability and integrity. Studies have demonstrated the effectiveness of combining IPFS with blockchain for secure and efficient data storage.
3. **Blockchain Implementation:** Utilizing blockchain technology, particularly platforms like Ethereum, facilitates the creation of immutable records. Smart contracts can be employed to manage metadata associated with files stored on IPFS, ensuring data integrity and traceability. Research indicates that integrating blockchain with IPFS effectively addresses data integrity challenges.

#### **3.1.2 Operational Feasibility:**

The integration of AES encryption, IPFS, and blockchain is operationally feasible, with proven effectiveness in enhancing data security, availability, and integrity. AES encryption ensures secure data protection, while IPFS provides decentralized storage, increasing data resilience. Blockchain adds immutability and transparency by recording metadata, ensuring data integrity.

Existing implementations, such as decentralized storage systems using blockchain and IPFS, demonstrate successful integration with low operational complexity.

Overall, the system can be deployed efficiently with appropriate planning and management of resources.

#### **3.1.3 Economic Feasibility:**

1. The system uses cost-effective platforms like **Polygon**, which significantly reduces blockchain transaction (gas) fees.
2. **Open-source tools** such as IPFS, cryptography libraries, and Web3.js minimize development costs.
3. **Infrastructure costs** are low due to the use of decentralized and cloud-based services.
4. Being an **academic prototype**, it avoids expensive enterprise-level deployment and remains budget-friendly.

### **3.2 Fact Finding Techniques:**

**Literature Review:** Extensive research was conducted through articles, whitepapers, and journals related to blockchain, IPFS, and data integrity to understand current challenges and existing solutions.

**Online Resources and Tutorials:** Technical blogs, developer documentation, GitHub repositories, and video tutorials were explored to understand implementation techniques and best practices.

**Expert Consultation:** Discussions with faculty members, blockchain developers, and cybersecurity professionals provided valuable insights into system design and feasibility.

**Prototyping:** A basic prototype was developed and tested to validate core concepts like file encryption, IPFS upload, and CID storage on the Polygon blockchain.

### **3.3 Hardware & Software Specifications:**

**Hardware Requirements:**

1. RAM: 4GB or More
2. Processor: i3 (or equivalent)
3. Storage: 20GB or more

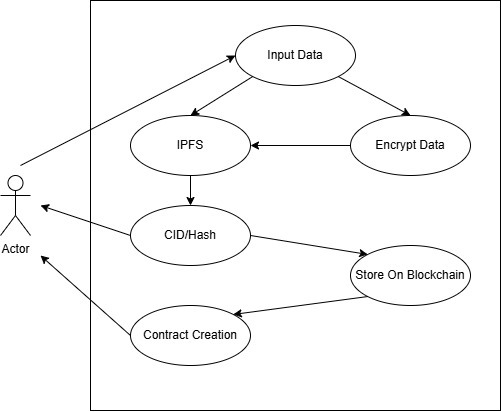
**Software Requirements:**

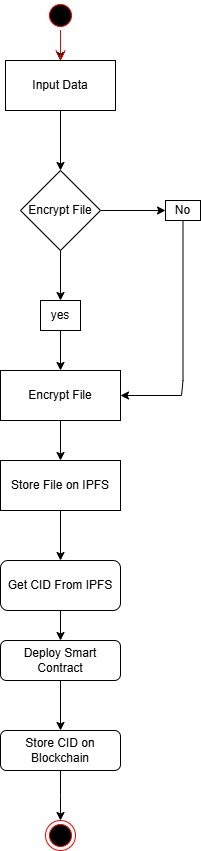
1. Blockchain Platform: Ethereum or Polygon for testnet deployment.
2. Development Tools: remix IDE, MetaMask.
3. Language: Solidity (contract creation)
4. Decentralized Storage: IPFS for file storage

**4. System Analysis:**

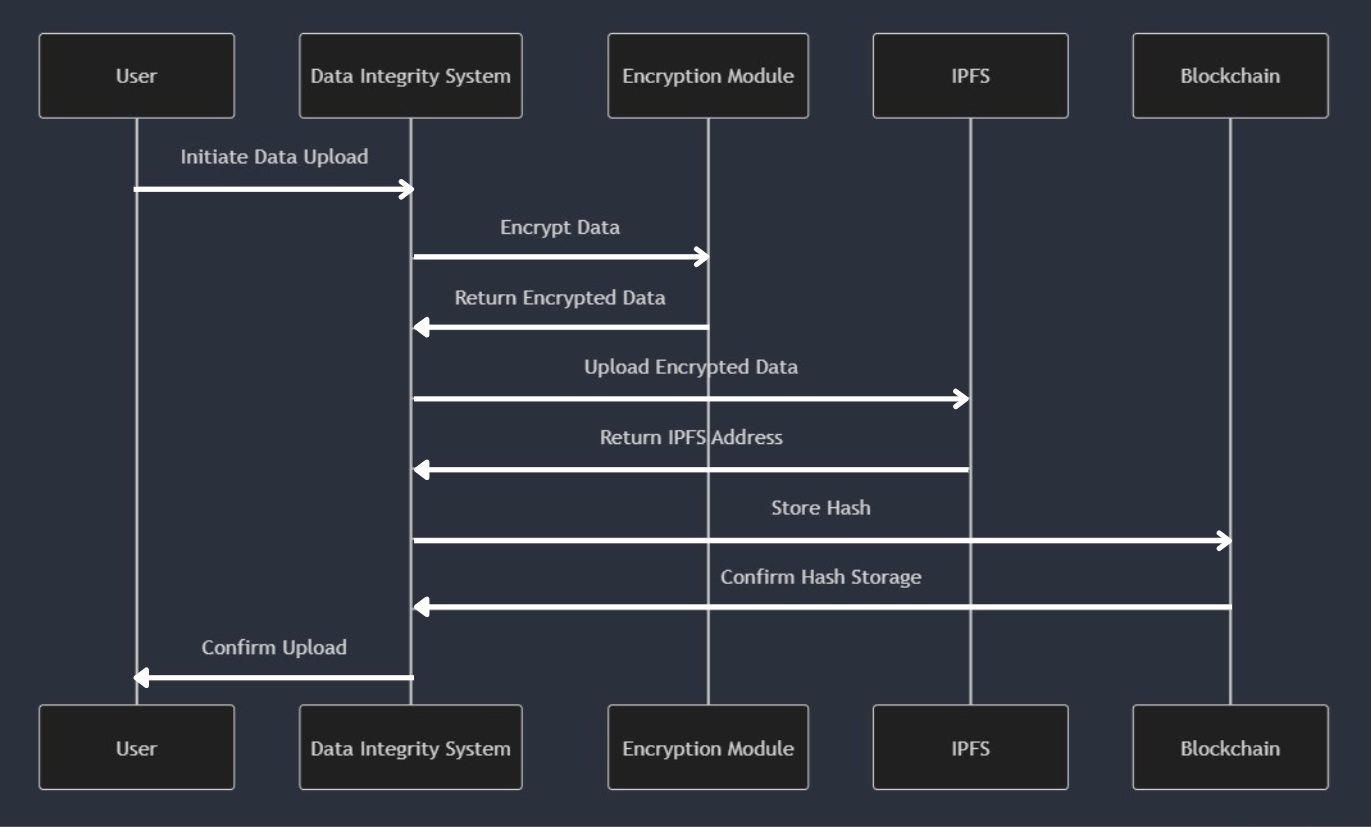
**4.1 UML Diagrams:**

**4.1.1 Use Case Diagram:**



**4.1.2 Activity Diagram:** 

**4.1.3 Sequence Diagram:**

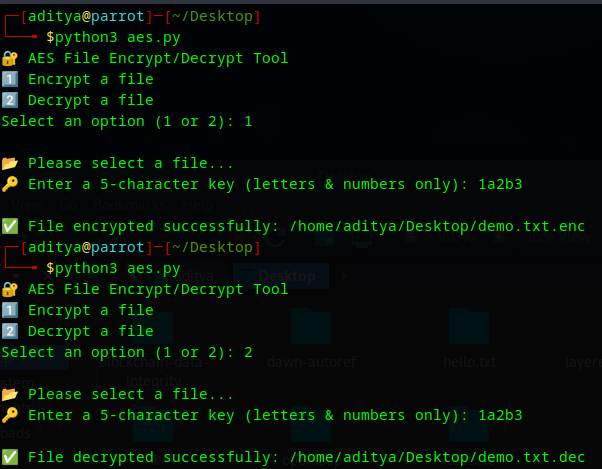
****

**5. System Design:**

**5.1 User interface:**

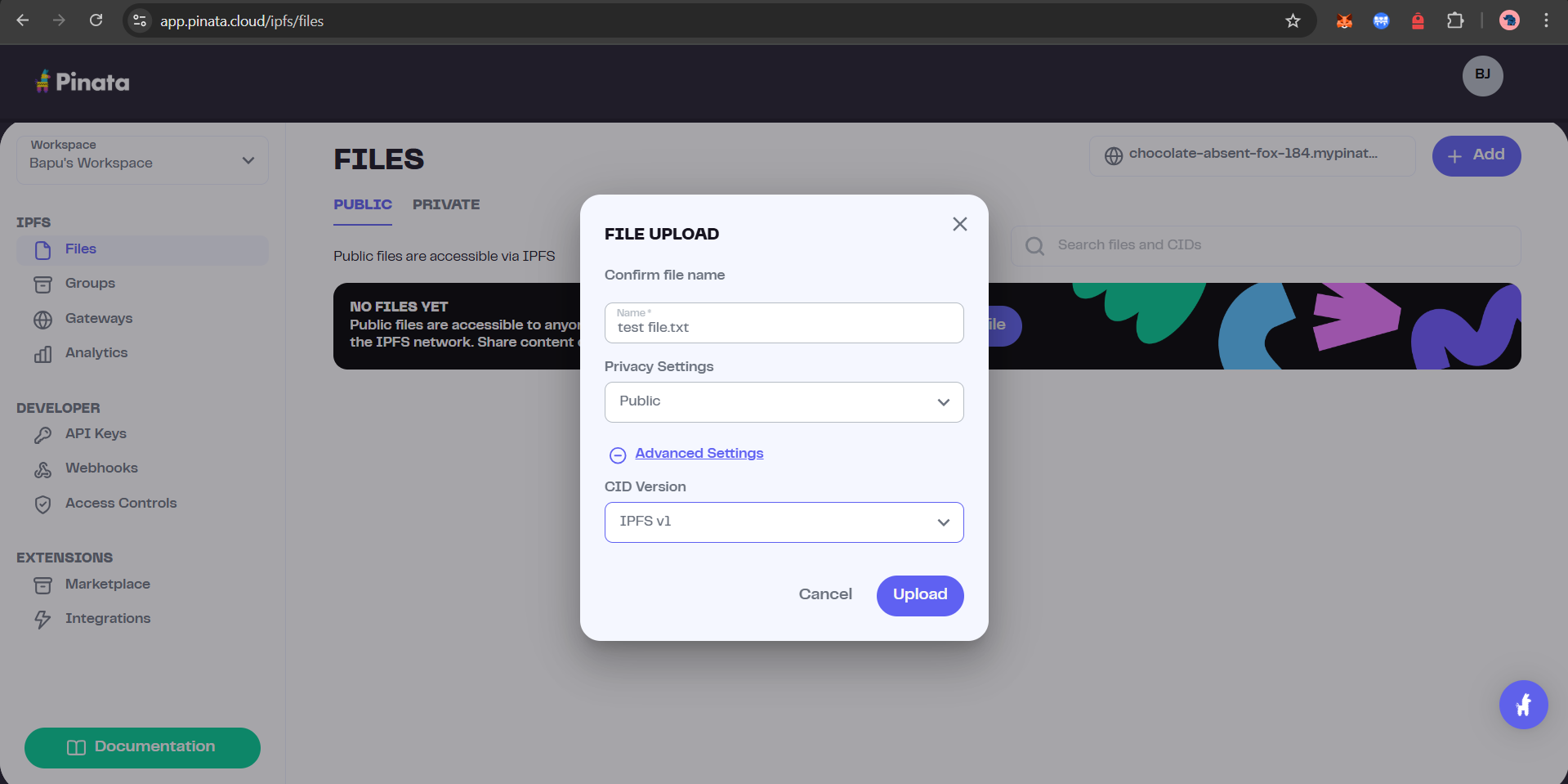
**1. Upload Screen**

User selects a file to upload. File is encrypted using AES encryption.

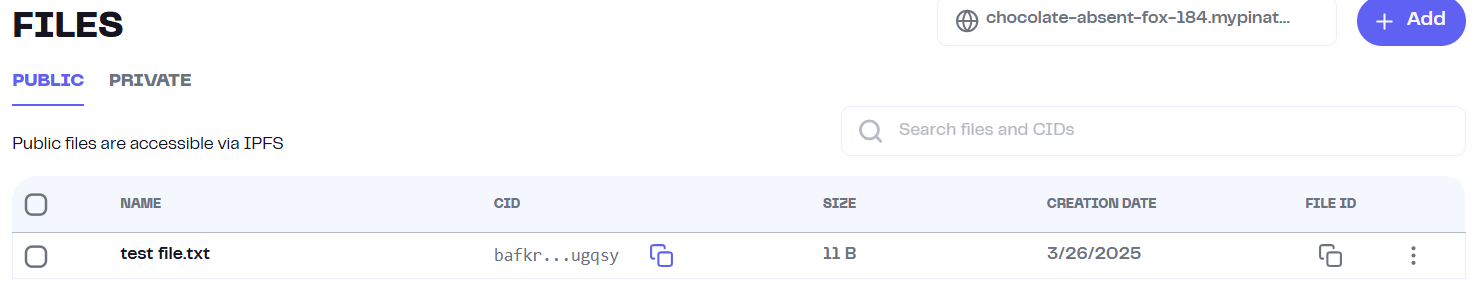


**2. IPFS Storage Screen**

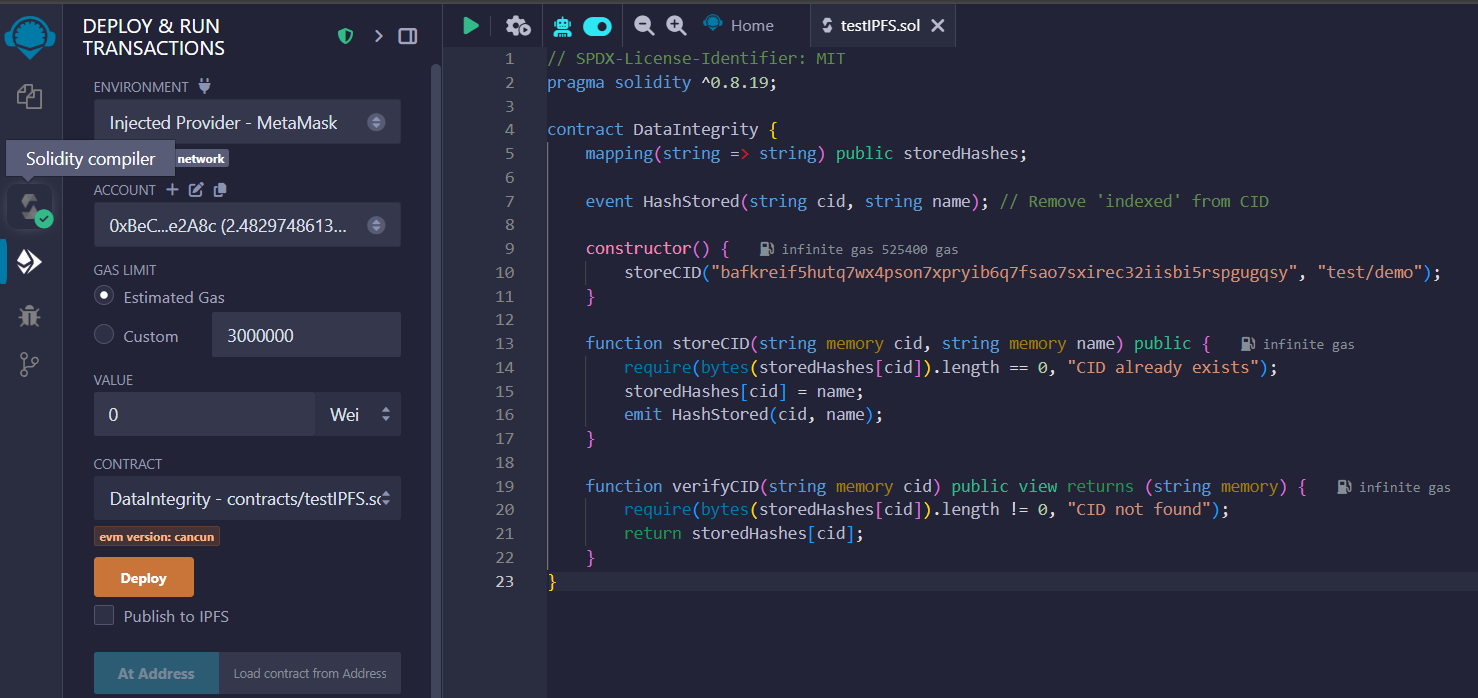
**The file/encrypted file is uploaded to IPFS.**



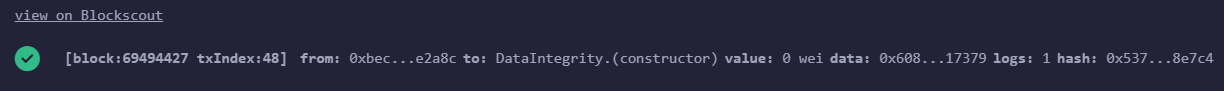
**IPFS generates a unique CID for the file.**  
  
**eg**. bafkreif5hutq7wx4pson7xpryib6q7fsao7sxirec32iisbi5rspgugqsy

****

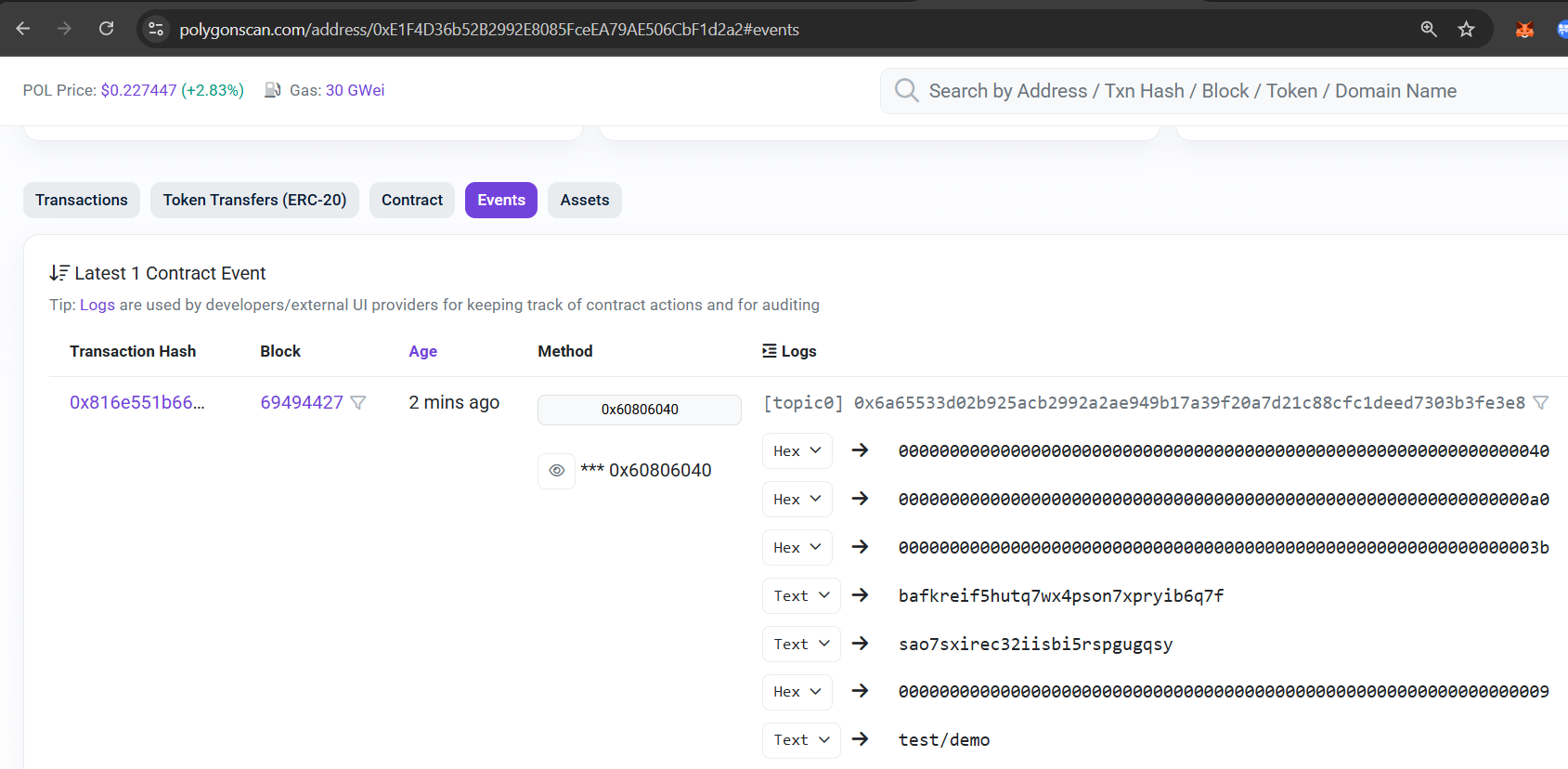
**3. Contract Creation & Deployment Screen**

**A smart contract is created or reused to store the CID on the Polygon blockchain.**  


**After successful deployment, the system displays a confirmation message:  
“Smart Contract Deployed Successfully” or “CID Stored on Blockchain”**

**The Transaction Hash is shown.**

**The CID is also visible on a Polygon blockchain explorer (e.g., Polygonscan) for public verification and transparency.**

****

**6. Limitations & Conclusion:**

**6.1 Limitations:**

**Dependency on Internet:**

The system requires a stable internet connection to interact with IPFS and the Polygon blockchain, which may limit its use in offline environments.

**Encryption Key Management:**

The current system does not include a secure method for distributing or managing encryption keys, which is essential for real-world deployment.

**Scalability Concerns:**

Although suitable for academic and small-scale use, scaling the system for enterprise or high-volume data operations would require significant enhancements.

**Blockchain Storage Cost:**

Even on low-cost networks like Polygon, recording data (CIDs) on-chain may incur minor gas fees, which could add up in larger systems.

**6.2 Conclusion:**

This project demonstrates a secure and efficient method for maintaining data integrity using AES encryption, IPFS for decentralized storage, and the Polygon blockchain for verification. The encrypted file is uploaded to IPFS, and its CID is stored on the blockchain, ensuring tamper-proof, transparent, and verifiable records.

The system serves as a proof-of-concept that can be applied in sectors like education, healthcare, and digital documentation. Its decentralized design removes reliance on central servers and provides strong data protection.

This approach enhances user confidence by preventing unauthorized data changes and offering a permanent audit trail. Integration of blockchain ensures that any retrieval or verification of data can be trusted and validated independently. As the demand for secure digital transactions and record-keeping grows, such systems become increasingly relevant. The use of cost-effective platforms like Polygon makes this solution scalable and sustainable. With future improvements like user authentication and mobile integration, the system can evolve into a full-fledged security solution.

Overall, the project not only validates the concept but also lays a strong foundation for real-world adoption of decentralized integrity verification systems.

**7. Future Enhancements:**

**Secure Key Management System:**

Integrate a secure and user-friendly key management module to handle encryption keys safely, possibly using asymmetric encryption or decentralized identity (DID) systems.

**Multi-User Role Access:**

Add support for user roles (e.g., admin, verifier, uploader) to allow better access control and accountability in real-world use cases.

**Integration with QR Codes:**

Generate QR codes for each stored CID to allow quick access and verification of data via mobile devices.

**Private/Consortium Blockchain Support:**

Enable compatibility with private or permissioned blockchains for organizations that require restricted access and higher data privacy.

**Automated Integrity Checks:**

Implement periodic integrity checks to automatically verify if the IPFS-stored file still matches the original hash on the blockchain.

**AI-Based Anomaly Detection:**

Use machine learning to detect unusual patterns or tampering attempts based on historical blockchain and file usage data.

**8. Bibliography:**

1. Python Software Foundation.(2024) Cryptography Library Documentation.  
   <https://cryptography.io/en/latest/>
2. Pinata. (2024). Pinata Cloud Documentation.  
   <https://docs.pinata.cloud/>
3. Ethereum Foundation. (2023). Introduction to Smart Contracts.

<https://ethereum.org/en/developers/docs/smart-contracts/>

1. Polygon Technology. (2023). Polygon PoS Documentation.

<https://docs.polygon.technology/docs/pos/getting-started/>

1. ValGenesis. (2022). Blockchain and Data Integrity.

<https://www.valgenesis.com/blog/blockchain-and-data-integrity>

1. MDPI Electronics. (2023). Blockchain-Based Secure Storage with IPFS.

<https://www.mdpi.com/2079-9292/12/7/1545>

1. Solidity Team. (2024). Solidity Documentation.  
   <https://docs.soliditylang.org/>
2. IBM. (2022). Building a Data Integrity Strategy.

<https://www.ibm.com/think/insights/data-integrity-strategy/>