Mini Project 3 – BT

Batch: P1 (BT) | Class: BE 1 | Subject: LP-3

Shreya Bhide 41111
Riya Bendigeri 41108

1. Title

- **Title:** Blockchain-Based Application for Securing Health-Related Medical Records
- **Date:** 28th September,2024

---

2. Abstract

This project proposes a Blockchain-based application for securely storing and managing health-related medical records. By leveraging the decentralized and immutable nature of Blockchain, the application ensures that sensitive medical information remains secure and accessible only to authorized personnel. The system enhances data privacy, integrity, and availability while allowing patients to control access to their medical records. We used Ethereum as the Blockchain platform and implemented smart contracts to automate access control.

---

3. Introduction

The handling of medical records is highly sensitive, requiring stringent measures to ensure data privacy and security. Traditional healthcare systems often rely on centralized databases, which are vulnerable to data breaches and unauthorized access. Blockchain technology offers a decentralized solution, providing data immutability, transparency, and enhanced security.

In this project, we explore how Blockchain can be used to create a secure, tamper-proof application for storing and managing health-related medical records. This application allows patients to securely store their records and provide access to healthcare professionals as needed.

Key Points:

- **Problem Statement:** How can Blockchain technology be used to secure health records and ensure privacy while providing authorized access to medical professionals?
- **Solution:** A decentralized application (DApp) using Blockchain to store and manage medical records, where patients control access via smart contracts.

---

4. Objective

The main objectives of this project are:

1. To design and develop a Blockchain-based application for managing health records.
2. To implement smart contracts for access control and ensure the immutability of medical data.
3. To ensure that patients have full control over who can access their medical information.
4. To enhance the security, transparency, and integrity of health data using Blockchain technology.

---

5. Blockchain Overview and Technology Stack

5.1. Blockchain Basics

Blockchain is a distributed ledger technology that ensures data immutability, transparency, and security. Every transaction in a Blockchain is recorded in blocks, which are linked together in a chain. Each block contains a cryptographic hash of the previous block, ensuring the integrity of the entire chain.

5.2. Smart Contracts

Smart contracts are self-executing contracts with the terms directly written into code. In the context of this project, smart contracts will be used to control access to medical records. The contract specifies conditions under which healthcare professionals can access patient records.

5.3. Technology Stack

- **Blockchain Platform:** Ethereum
- **Smart Contract Language:** Solidity
- **Web Interface:** HTML, CSS, JavaScript (React.js)
- **Development Framework:** Truffle
- **Testing Environment:** Ganache (for local Blockchain development)

---

6. Methodology

6.1. System Architecture

The system comprises three key components:

1. **Patient:** The owner of the medical record who can grant or revoke access to healthcare providers.
2. **Healthcare Provider:** A doctor or institution that needs authorized access to a patient's records.
3. **Smart Contract:** A Blockchain contract that automates access control, stores pointers to medical records, and records access logs.

6.2. Steps in Application Development

Step 1: Smart Contract Design

Smart contracts form the backbone of this Blockchain application, providing secure, decentralized access control. The contract stores:

- **Patient Data:** An encrypted hash of the patient's medical record.
- **Access Control:** A mapping of healthcare providers authorized to access patient data.
- **Audit Trail:** A log of access attempts.

Example Solidity Smart Contract Code:

solidity

Copy code

```
pragma solidity ^0.8.0;


contract MedicalRecord {
    address public patient;
    mapping(address => bool) public authorizedDoctors;
    string private medicalDataHash; // Encrypted hash of the medical record


    event AccessGranted(address indexed doctor);
    event AccessRevoked(address indexed doctor);


    constructor(string memory _medicalDataHash) {
        patient = msg.sender;
        medicalDataHash = _medicalDataHash;
    }


    modifier onlyPatient() {
        require(msg.sender == patient, "Not authorized");
```

```
        _;
    }

    function grantAccess(address doctor) public onlyPatient {
        authorizedDoctors[doctor] = true;
        emit AccessGranted(doctor);
    }

    function revokeAccess(address doctor) public onlyPatient {
        authorizedDoctors[doctor] = false;
        emit AccessRevoked(doctor);
    }

    function viewRecord() public view returns (string memory) {
        require(authorizedDoctors[msg.sender], "Access denied");
        return medicalDataHash;
    }
}
```

Step 2: Web Interface

The web interface allows patients and doctors to interact with the smart contract. Patients can upload medical records, grant or revoke access, while doctors can request access to the records.

Frontend Functionality:

- **Patient Dashboard:** Displays patient information and an option to upload a medical record, grant, or revoke access.
- **Doctor Dashboard:** Allows doctors to request access and view records (if authorized).

Step 3: Data Storage

While Blockchain stores the encrypted hash of the medical records, actual records are stored off-chain due to storage limitations of Blockchain. IPFS (InterPlanetary File System) is used for decentralized storage.

6.3. Deployment and Testing

- **Development Environment:** The smart contract was tested locally using Ganache, which simulates an Ethereum Blockchain for development purposes.

- **Deployment:** The smart contract was deployed to the Ethereum test network using the Truffle framework.

Example Code for Deploying the Contract (Truffle):

javascript

Copy code

```
const MedicalRecord = artifacts.require("MedicalRecord");


module.exports = function(deployer) {
  deployer.deploy(MedicalRecord, "QmEncryptedMedicalDataHash");
};
```

---

7. Results and Discussion

7.1. Key Features

- **Decentralized Storage:** Medical records are stored in a decentralized manner using IPFS, while Blockchain ensures the integrity of the records.
- **Immutability:** Once a medical record is added to the Blockchain, it cannot be modified, ensuring data integrity.
- **Access Control:** Patients have full control over who can access their medical records, providing better privacy and security.
- **Audit Trail:** Every access attempt is logged, ensuring transparency and accountability.

7.2. Security and Privacy

The Blockchain-based system ensures that medical records are tamper-proof, and unauthorized access is prevented. The use of encrypted hashes ensures that even if the Blockchain is public, the actual medical data remains private. The decentralized nature of Blockchain also eliminates the need for a central authority, reducing the risk of data breaches.

7.3. Performance

The system was tested for scalability and performance. While Blockchain transactions may take time due to network latency, the overall system is robust for small- to medium-scale deployments. For large-scale systems, Layer 2 solutions like sidechains could be explored to improve scalability.

---

8. Conclusion

This project successfully demonstrates the application of Blockchain technology in managing and securing medical records. By using Ethereum smart contracts, we created a

decentralized application where patients can securely store and control access to their medical records. Blockchain's immutability and decentralized nature ensure that sensitive data is protected from unauthorized access, while smart contracts automate the access control process.

---

9. Future Scope

- **Integration with Healthcare Systems:** The application could be integrated with existing healthcare management systems for broader adoption.
- **Scalability Solutions:** Implementing Layer 2 solutions like Plasma or state channels to improve the performance and scalability of the system.
- **Interoperability:** Developing standards for medical record exchange between different Blockchain networks to ensure interoperability across healthcare systems.