

# **CS801 - COMPUTING LAB**

## **A REPORT ON THE PROJECT ENTITLED HEALTH RECORD MANAGEMENT SYSTEM USING BLOCKCHAIN**



**Group Members:**

Gaurav Kansal  
242IS012

Jeet Nilesh Desai  
242IS010

**I SEMESTER M-TECH CSE-IS**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA  
SURATHKAL**

**2024 – 2025**

# Contents

<b>1</b>	<b>Declaration</b>	<b>2</b>
<b>2</b>	<b>Description</b>	<b>3</b>
<b>3</b>	<b>Motivation</b>	<b>4</b>
<b>4</b>	<b>Problem Statement and Objective</b>	<b>5</b>
<b>5</b>	<b>Literature Survey</b>	<b>6</b>
<b>6</b>	<b>Novelty</b>	<b>8</b>
<b>7</b>	<b>Solution Design</b>	<b>9</b>
7.1	System Architecture . . . . .	9
7.2	Workflow . . . . .	10
7.3	Technical Stack . . . . .	10
<b>8</b>	<b>Implementation</b>	<b>11</b>
8.1	Frontend . . . . .	11
8.1.1	Doctor Signup . . . . .	11
8.1.2	User Signup . . . . .	11
8.1.3	Login Page . . . . .	13
8.1.4	7.1.4 Captacha Generation . . . . .	13
8.2	Backend . . . . .	14
8.2.1	Database Connection . . . . .	14
8.2.2	FastAPI APIs . . . . .	14
8.3	Blockchain Integration . . . . .	15
8.3.1	MetaMask . . . . .	15
8.3.2	IPFS Storage . . . . .	16
8.3.3	Smart Contracts . . . . .	17
8.3.4	Sending File Over Blockchain . . . . .	17
<b>9</b>	<b>Results</b>	<b>20</b>
<b>10</b>	<b>Future Scope</b>	<b>21</b>
<b>11</b>	<b>Conclusion</b>	<b>23</b>
<b>12</b>	<b>References</b>	<b>24</b>

# 1 Declaration

We hereby declare that the work presented in this Project Report titled **Health Record Management System Using Blockchain**, submitted to the **National Institute of Technology Karnataka**, is in partial fulfilment of the requirements for the award of the degree of **M.Tech** for the course **Computing Lab**. The contents of this Project Report, in full or in parts, have not been submitted to, and will not be submitted by us to, any other Institute or University in India or abroad for the award of any degree.

Gaurav Kansal  
242IS012

Jeet Nilesh Desai  
242IS010

## 2 Description

Managing patient records in healthcare is complex due to fragmented systems and security concerns. These systems store critical data, including patient demographics, medical history, diagnoses, treatments, and administrative details like appointment scheduling. Traditional record-keeping often results in isolated databases, inefficient information sharing, and data vulnerabilities, which can delay care and compromise outcomes.

Our solution introduces a blockchain-based healthcare record management system to address these challenges. Blockchain provides a secure, decentralized, and tamper-resistant platform, ensuring data integrity and privacy. A key feature of this system is enabling patients to control and share their medical records seamlessly across different healthcare providers.

With blockchain, a patient can securely store medical records provided by one doctor and grant access to another doctor when needed. This process is straightforward and fully transparent, ensuring that the receiving doctor can view the necessary records for effective diagnosis and treatment. Patients maintain control over what information is shared and with whom, promoting trust and efficiency in healthcare delivery.

This capability enhances continuity of care by ensuring that all relevant medical information is accessible to authorized providers, reducing redundancies like repeated tests and streamlining the patient experience. Furthermore, blockchain's immutable ledger ensures a complete and auditable history of all shared records, fostering accountability and security.

By empowering patients to manage their medical data, this system improves interoperability, strengthens data privacy, and ensures timely access to accurate information, ultimately enhancing the quality of healthcare.

### 3 Motivation

The fragmented nature of existing electronic health record (EHR) systems poses significant challenges, as patient information is often scattered across multiple platforms, making comprehensive and accurate data access difficult. This lack of integration not only hampers continuity of care but also exposes critical vulnerabilities. For example, the 2017 WannaCry ransomware attack, which paralyzed the UK's NHS by encrypting patient records and demanding ransom, underscored the fragility of traditional systems. The incident disrupted emergency services, delayed surgeries, and led to canceled appointments, highlighting the dire need for more secure and resilient healthcare data management.

Such events motivate the adoption of innovative technologies like blockchain. Its decentralized nature ensures that patient records are tamper-proof and immutable, providing unparalleled data security and integrity. By eliminating single points of failure and enabling secure, seamless data sharing, blockchain offers a robust solution to the vulnerabilities of traditional systems. This motivation drives the development of blockchain-based healthcare record systems to protect sensitive medical information, enhance interoperability, and ultimately improve patient outcomes.

## 4 Problem Statement and Objective

Current healthcare record systems are fragmented and lack seamless integration, resulting in isolated data storage and limited communication between platforms. This disjointed structure makes it challenging to access patient information across systems, leading to inefficiencies in care delivery and increased risks of data breaches and unauthorized access. Patients often face difficulties in sharing their medical records with multiple doctors, requiring manual effort and risking document loss, which impacts the continuity and quality of care.

The primary objective is to develop a blockchain-based healthcare record management system that ensures secure, decentralized, and tamper-proof storage of medical records. The system will digitalize all healthcare documents, transitioning them from fragmented and paper-based formats into a unified digital platform. By incorporating smart contracts, the solution will automate access controls, allowing patients to seamlessly share their medical records with multiple doctors while maintaining privacy and control over their data. This approach aims to enhance patient privacy, eliminate the risks of document loss.

## 5 Literature Survey

Blockchain technology is recognized for its potential to enhance security, privacy, and interoperability in healthcare. Below is a summary of key studies and real-world implementations.

### 1. Blockchain Technology in Healthcare: A Systematic Review (2017)

**Summary:** This review explores blockchain's ability to ensure secure data sharing and patient privacy by providing transparent and tamper-proof records. It discusses the benefits of using blockchain in healthcare to improve trust among stakeholders and ensure data integrity. The paper also outlines challenges in implementing blockchain at scale in healthcare systems and potential solutions.

### 2. Blockchain in Healthcare: A Survey (2020)

**Summary:** The paper reviews several blockchain frameworks for managing Electronic Health Records (EHR), addressing scalability, privacy, and integration challenges. It highlights the benefits of decentralized systems in healthcare, particularly in improving interoperability between disparate healthcare systems. The survey also identifies critical research gaps, such as regulatory compliance and user adoption, that must be addressed before widespread implementation.

### 3. Blockchain-Based EHR Management System for Healthcare 4.0 (2021)

**Summary:** This paper proposes a blockchain-based EHR system with smart contracts for access control, which enhances security and interoperability within the healthcare ecosystem. The authors suggest using blockchain to reduce administrative burdens and improve patient care by ensuring that data is always up-to-date and securely shared between authorized users. The system design leverages both blockchain and smart contracts to automate access and minimize human errors.

## Real-Life Implementations

- **Estonia's e-Health System:** Uses blockchain for decentralized patient data management nationwide. The system enables citizens to access their health data securely and ensures transparency, as all transactions are logged on the blockchain, preventing unauthorized access.
- **Opolo Health:** Manages patient records securely and ensures interoperability across healthcare providers. The platform allows seamless sharing of medical data between different healthcare providers while ensuring patient consent is always obtained, maintaining privacy.

- **MIT MedRec:** Allows patients to control access to their EHRs with secure data sharing. By leveraging blockchain, MedRec provides patients with a single, unified view of their health records, enabling them to grant or revoke access to healthcare providers as needed.
- **Solve.Care:** Uses blockchain for secure and transparent sharing of medical records. The platform's use of smart contracts automates administrative processes, ensuring that healthcare data is accessed and shared only with the patient's consent, reducing delays and errors.
- **Proposed Solution for Indian Government:** The Indian government, through the Ministry of Health and Family Welfare, is exploring the integration of blockchain technology into the National Digital Health Mission (NDHM). The initiative aims to implement a decentralized and secure healthcare system, enabling the secure storage and sharing of electronic health records (EHRs) across healthcare providers nationwide. The blockchain platform will use smart contracts to ensure automated data access, allowing patients to control who can view or modify their records. Privacy-preserving techniques like encryption and zero-knowledge proofs (ZKPs) will protect patient data. This project aims to reduce fraud, enhance data integrity, and improve healthcare access, particularly in underserved areas.



## 6 Novelty

The proposed blockchain-based healthcare record management system introduces a novel approach to addressing longstanding challenges in the healthcare industry, such as data fragmentation, security breaches, and lack of interoperability. Unlike traditional centralized systems, this project leverages blockchain technology and IPFS (InterPlanetary File System) to ensure secure, decentralized, and immutable storage of patient records.

The integration of smart contracts adds a layer of automation and transparency, allowing patients to maintain complete control over their medical records. This unique feature enables secure sharing of reports across multiple healthcare providers without compromising privacy. The system provides a scalable and cost-effective solution for storing and verifying health records, addressing current issues while improving healthcare data management efficiency.

## 7 Solution Design

### 7.1 System Architecture

The system leverages blockchain and the InterPlanetary File System (IPFS) for secure and decentralized data storage, along with a web-based application to manage user interactions. The solution ensures secure, immutable, and tamper-proof handling of healthcare records, with strict access controls enforced via smart contracts.

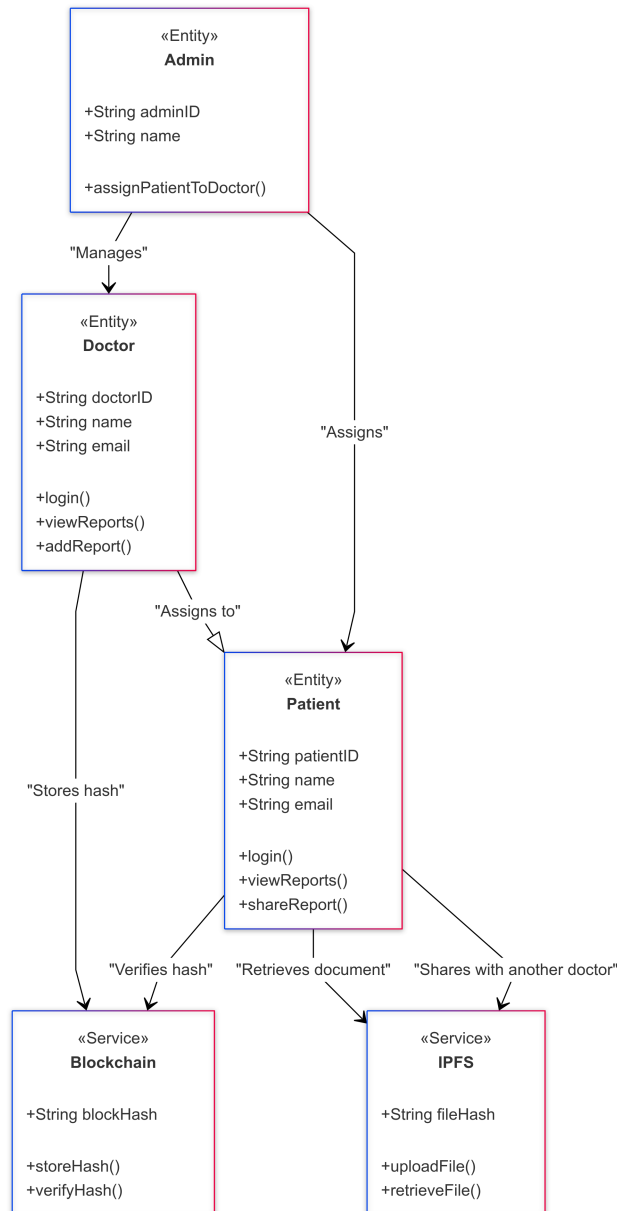


Figure 1: System Design

## 7.2 Workflow

1. **Doctor Registration and Verification:** Doctors register using their government-registered ID, which is cross-verified with the backend dataset. Upon successful verification, they are granted access to the system with permissions to view and update assigned patient records.
2. **Patient Registration and Doctor Assignment:** Patients self-register on the platform. The admin assigns the patient to a doctor within the hospital. The patient dashboard shows a history of all assigned doctors.
3. **Storing and Accessing Records:**
  - **Uploading Records:** Doctors add new medical reports or prescriptions. The report is encrypted and uploaded to IPFS. The IPFS hash is stored on the blockchain along with patient and doctor details.
  - **Accessing Records:** Patients or authorized doctors request access to a report. The system validates permissions using smart contracts, fetches the IPFS hash from the blockchain, and retrieves the file from IPFS.
4. **Report Sharing Between Doctors:** Patients authorize a new doctor to view specific reports. The system updates the smart contract with the new doctor's permissions. The authorized doctor retrieves the file via the IPFS hash after permission validation.
5. **Admin Role:** Admins manage user accounts (Doctors and Patients) but do not have access to medical records. This separation ensures patient privacy and aligns with regulatory compliance.

## 7.3 Technical Stack

- **Frontend:** React.js, Javascript, HTML, CSS
- **Backend:** FastAPI, Python
- **Blockchain:** Ethereum, Metamask
- **Storage:** IPFS, PostgreSQL
- **Smart Contracts:** Solidity
- **Services:** Pinata, Twillo

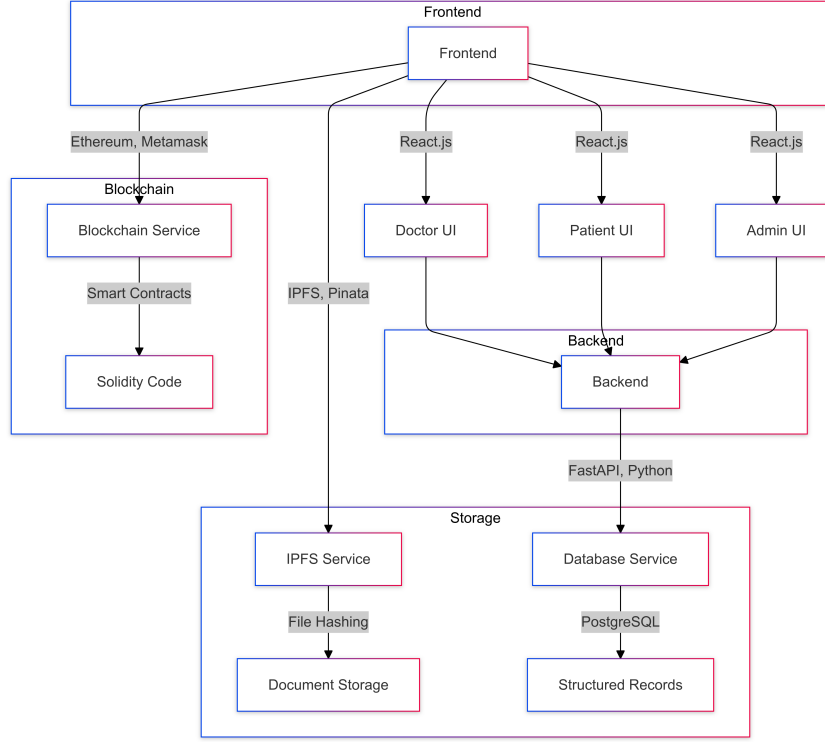


Figure 2: Tech-Stack Distribution

## 8 Implementation

### 8.1 Frontend

#### 8.1.1 Doctor Signup

The Doctor Signup process ensures secure registration and identity verification using OTP (One-Time Password). Initially, the doctor's government ID is checked against the system's database. If a match is found in the `registration_government_ids` table, the associated phone number is retrieved.

An OTP is generated using the `generate_otp()` function, creating a 6-digit code. This OTP is sent via Twilio's SMS service to the doctor's phone number, remaining valid for 5 minutes. The `send_otp_via_twilio()` function handles the sending process.

The doctor then submits the received OTP for verification. The system checks if the OTP is valid and hasn't expired. If verified successfully, the doctor's identity is confirmed. Otherwise, an error message prompts the user to try again.

This workflow ensures that only verified doctors can proceed, offering secure authentication within the platform.

To ensure the security of user passwords, we utilize bcrypt, a robust password hashing algorithm. Each password is hashed with a unique salt, which prevents the use of precomputed hash tables such as rainbow tables.

#### 8.1.2 User Signup

The user signup process involves validating form data and submitting it to a backend API. Initially, the form state holds user input values, updated via the `handleChange`

function. Upon form submission, the `handleSubmit` function validates required fields (name, address, phone, email, username, password) and checks for password matching.

If validation passes, the form data is sent to the backend via a POST request. On success, a confirmation message is displayed, and the form is cleared. If validation fails or the API request encounters an error, appropriate messages are shown to the user.

This logic ensures that only valid data is submitted and users are informed of the results.

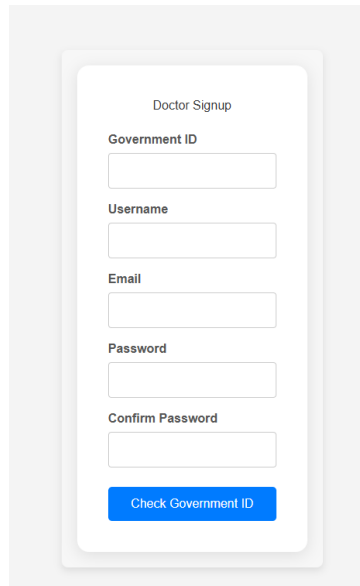
A screenshot of a 'Doctor Signup' form. The form is titled 'Doctor Signup' and contains five input fields: 'Government ID', 'Username', 'Email', 'Password', and 'Confirm Password'. Below the input fields is a blue button labeled 'Check Government ID'.

Figure 3: Doctor Signup

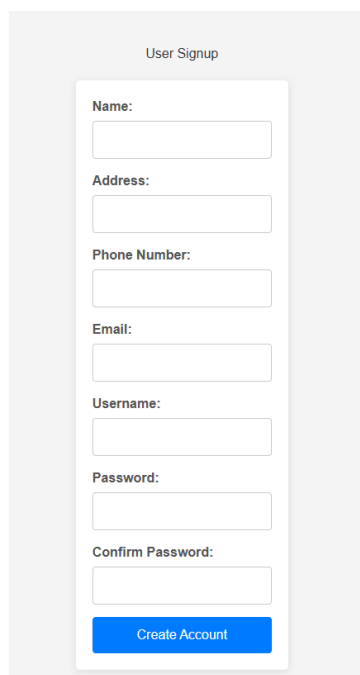
A screenshot of a 'User Signup' form. The form is titled 'User Signup' and contains seven input fields: 'Name:', 'Address:', 'Phone Number:', 'Email:', 'Username:', 'Password:', and 'Confirm Password:'. Below the input fields is a blue button labeled 'Create Account'.

Figure 4: User Signup

### 8.1.3 Login Page

The **App** component provides a role-based login system with CAPTCHA validation. It handles login and signup processes for three roles: **Doctor**, **User**, and **Admin**.

#### Login Flow:

1. **Role Selection:** Users select their role (*Doctor*, *User*, or *Admin*) from a dropdown menu.
2. **Form Submission:** Upon submitting the login form, the entered CAPTCHA is validated. If it doesn't match, an alert is triggered.
3. **API Request:** Based on the selected role, a POST request is made to the corresponding backend endpoint:
  - `/doctor/login` for Doctors
  - `/login/` for Users
  - `/admin/login` for Admins
4. **Success/Failure:** Upon successful login, users are redirected to their respective pages (`/doctortitle`, `/userpage`, or `/Admin`). If the login fails, an error message is displayed.

This logic enables smooth navigation between login and signup forms, with secure CAPTCHA validation and role-specific routing.

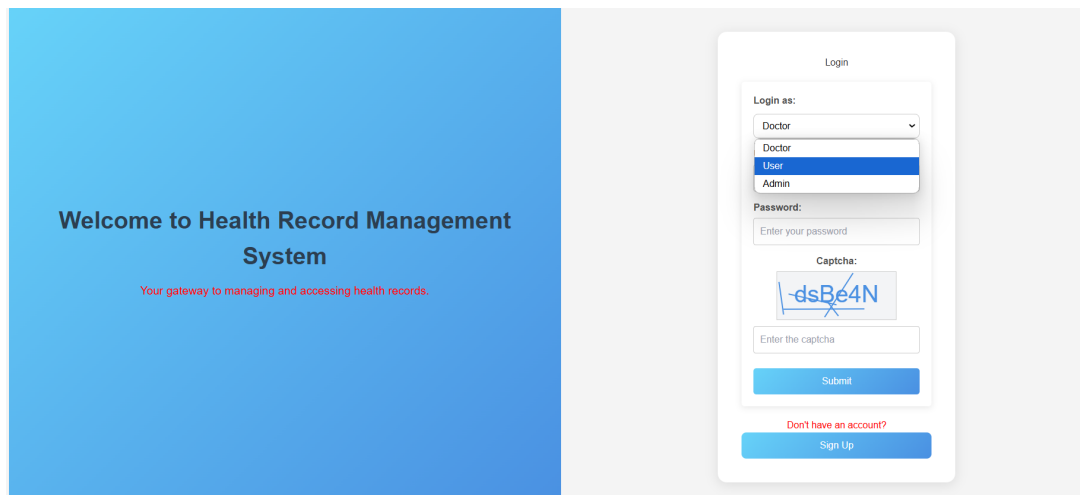


Figure 5: Login Page

### 8.1.4 7.1.4 Captacha Generation

The captcha logic consists of two parts: generation and rendering.

A random string of 6 characters is generated from the set:

$$\text{Characters} = \{A-Z, a-z, 0-9, @, \#, +, *, !\},$$

using the formula:

$$C_i = \text{Characters}[\lfloor \text{Random}() \times |\text{Characters}| \rfloor], \quad i \in \{1, \dots, 6\}.$$

The generated string is displayed on an HTML canvas with a light gray background, blue text centered on the canvas, and random lines drawn to increase complexity.

## 8.2 Backend

### 8.2.1 Database Connection

In a typical FastAPI application, database connections are established using SQLAlchemy, which is an Object-Relational Mapping (ORM) library. The general process of establishing a database connection is as follows:

1. **Engine Creation**: The first step is to create an engine using SQLAlchemy's `create_engine` function. This engine provides the necessary connection to the database. The connection string (URL) specifies the database type, username, password, and host information.

```
engine = create_engine("postgresql://username:password@localhost/dbname")
```

2. **Session Creation**: Once the engine is set up, the next step is to create a session factory using `sessionmaker`. The session object allows interaction with the database to execute queries and transactions.

```
SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)
```

3. **Dependency Injection**: FastAPI uses dependency injection to provide the database session to the endpoint functions. The session is created per request and is closed after the request is processed. This ensures that the database connection is managed efficiently.

```
def get_db(): { db = SessionLocal() try: yield db finally: db.close() }
```

4. **Table Creation**: To ensure that the necessary database tables are created, the `Base.metadata.create_all()` function is called. This creates tables defined by the ORM models if they do not exist in the database.

```
Base.metadata.create_all(bind=engine)
```

By following these steps, a FastAPI application can easily connect to a database and perform operations like inserting, updating, and retrieving records.

### 8.2.2 FastAPI APIs

FastAPI is a modern web framework for building APIs with Python, offering high performance, automatic data validation, and interactive API documentation. Endpoints in FastAPI are defined using decorators such as `@app.get`, `@app.post`, and others.

**Basic Endpoint Example:**

```
@app.post("/signup/")
async def signup(request: SignupRequest):
    return {"message": "User signed up successfully!"}
```

FastAPI supports path parameters and request bodies for data exchange:

```
@app.get("/users/{user_id}")
async def get_user(user_id: int):
    return {"user_id": user_id}
```

It automatically generates API documentation at `/docs` and `/redoc`, providing an interactive interface for testing the API. FastAPI's ease of use and high speed make it a popular choice for modern web development.

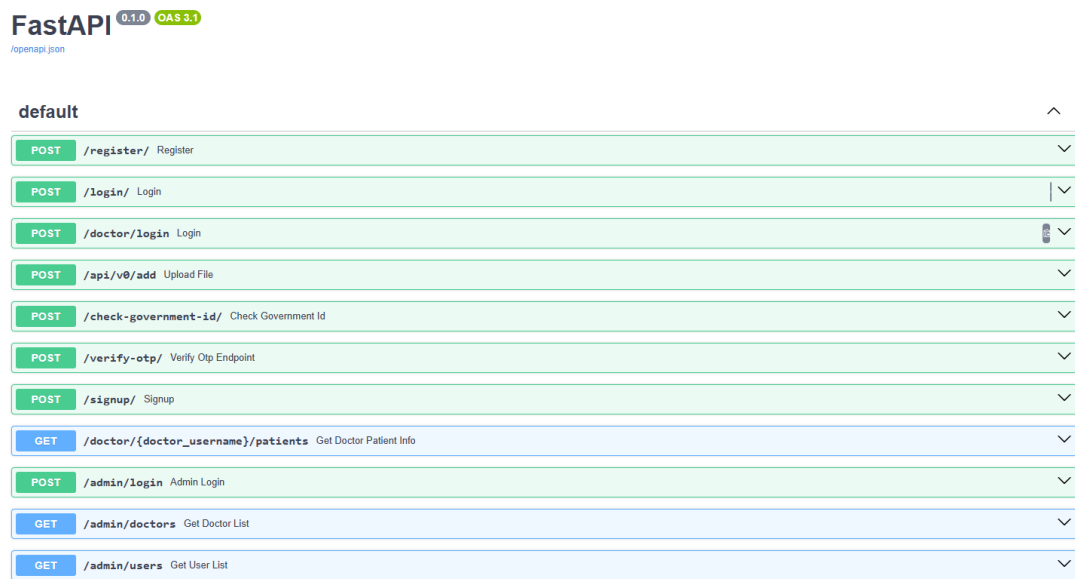


Figure 6: APIs

## 8.3 Blockchain Integration

### 8.3.1 MetaMask

MetaMask is a cryptocurrency wallet and gateway to blockchain applications, specifically for Ethereum and compatible blockchains like Sepolia. It allows users to manage digital assets such as Ether (ETH) and interact with the blockchain directly.

#### How MetaMask Works:

- **Wallet Management:** MetaMask securely stores private keys and generates Ethereum addresses for sending and receiving tokens on the blockchain.
- **Transaction Signing:** MetaMask signs transactions using the private key stored locally, ensuring security without exposing the key.
- **Network Switching:** It supports multiple blockchain networks, including Sepolia, enabling users to easily switch between them.

MetaMask simplifies interactions with the blockchain, particularly for managing assets and performing transactions on the Sepolia testnet.



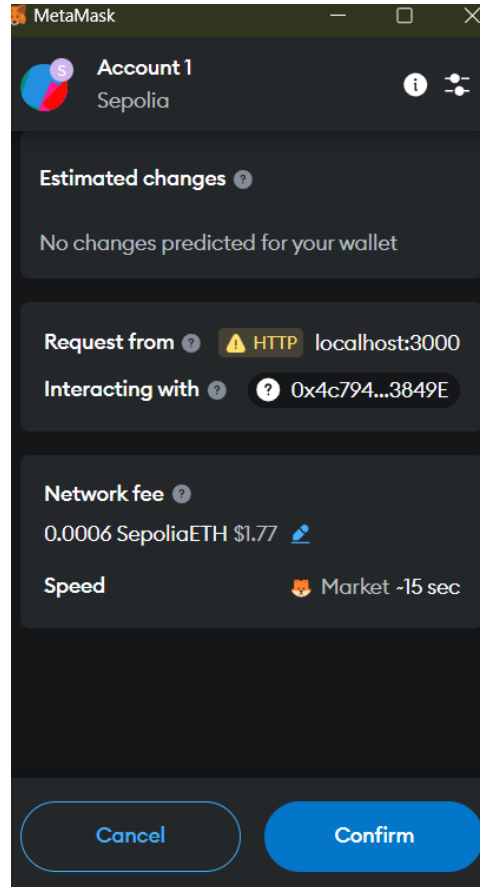


Figure 7: Metamask Wallet

### 8.3.2 IPFS Storage

In this project, we use the InterPlanetary File System (IPFS) to store patient files securely and in a decentralized manner. IPFS allows large files to be stored off-chain while maintaining their immutability and availability.

To upload files to IPFS, we use Pinata, a service that simplifies file pinning and provides reliable storage. Pinata is integrated into the React application, where a doctor can upload patient records.

#### **File Upload Process:**

- The doctor selects a file to upload via the UI.
- The file is sent to the Pinata API via a POST request using a secure API key.
- Once uploaded, Pinata returns an IPFS hash, which serves as a unique identifier for the file.
- The IPFS hash is then used to generate a URL to access the file.
- This URL is stored on the blockchain to associate the file with the patient's record.

By using IPFS and Pinata, we ensure that patient files are stored in a decentralized, secure, and accessible manner while reducing the load on the Ethereum blockchain itself.

### 8.3.3 Smart Contracts

The `PatientRecord` smart contract is designed to store and manage patient medical records on the Ethereum blockchain. By leveraging blockchain technology, it ensures the immutability and transparency of patient data, while utilizing IPFS for off-chain storage.

#### Key Functions:

- **uploadFile:** Allows doctors to upload patient records, associating them with a unique `patientId`, `doctorGovId`, and the IPFS URL of the file.
- **getPatientFiles:** Retrieves all records for a specific `patientId`.

The contract uses an `onlyOwner` modifier to restrict certain functions to the contract owner. The `FileUploaded` event is emitted each time a record is uploaded, providing transparency.

**Contract Address:** After deployment, the contract is assigned a unique address, which is used to interact with it from a decentralized application (dApp).

**ABI:** The Application Binary Interface (ABI) is a JSON file generated during compilation. It defines the contract's functions and events, allowing interaction with the smart contract through a front-end application.

The `PatientRecord` contract provides a secure and decentralized way to manage and access medical records on the blockchain.

### 8.3.4 Sending File Over Blockchain

In this project, we utilize the Ethereum blockchain to store references to patient files securely. When a doctor uploads a file, the following process occurs:

- The file is first uploaded to IPFS using Pinata, where it receives a unique IPFS hash.
- After successfully uploading the file to IPFS, the file's IPFS URL is sent to the Ethereum blockchain via a smart contract.
- The smart contract's `uploadFile` function is called, which records the IPFS URL, the patient's ID, and the doctor's government ID on the blockchain. This transaction is then confirmed after it is mined.
- MetaMask is used as the interface to sign the transaction, ensuring that the operation is secure and authorized.
- The blockchain stores the metadata (IPFS URL, doctor's ID, timestamp) associated with the patient record in a decentralized manner, making it immutable and traceable.

This method ensures the secure, transparent, and immutable storage of patient data while keeping the large files off-chain on IPFS.

The screenshot displays a web interface for a doctor's profile. At the top, a header bar contains a user icon, the text "Welcome, Dr. Gaurav Kansal", and a red "Logout" button. Below the header, the page is divided into sections. The "Doctor Information" section shows the doctor's name, government ID, and email. The "Patient List" section is active, displaying a form for "Patient Information". This form includes fields for patient ID, name, email, and phone, along with an "Access Granted" status. Below the form, there is a file upload section with a "Choose File" button, a file name "T155G24Ap... Form (2).pdf", and an "Upload File" button. A progress bar indicates "I am uploading your prescription." and an "update" button is at the bottom. The "Patient Files" section lists two files with their upload timestamps.

Doctor Information

Welcome, Dr. Gaurav Kansal

Logout

Name: Dr. Gaurav Kansal  
Government ID: Doc-2001-2501  
Email: gauravkansal4462@gmail.com

Patient List

Patient Information

id: 4  
Name: vedant  
Email: clashingjeet@gmail.com  
Phone: 9922056488

Access Granted

Choose File T155G24Ap... Form (2).pdf Upload File

I am uploading your prescription.

update

Patient Files

File (Uploaded on 11/17/2024, 5:07:12 PM)  
File (Uploaded on 11/17/2024, 5:20:36 PM)

Figure 8: Doctor Page

## Fetching File From Blockchain

In this system, both patients and doctors can retrieve files stored on the blockchain. The retrieval process is designed to ensure secure access control and maintain privacy.

### • Patient-Side Retrieval:

- The patient can request to view their files by interacting with the blockchain through a web interface or dApp.
- Using the `getPatientFiles` function of the smart contract, the patient's files are fetched from the blockchain, including the IPFS URL, doctor's government ID, and the timestamp of the file upload.
- The patient's identity and access permissions are validated to ensure that they can view the files.
- The IPFS URL for each file is returned, allowing the patient to access the files directly from IPFS.

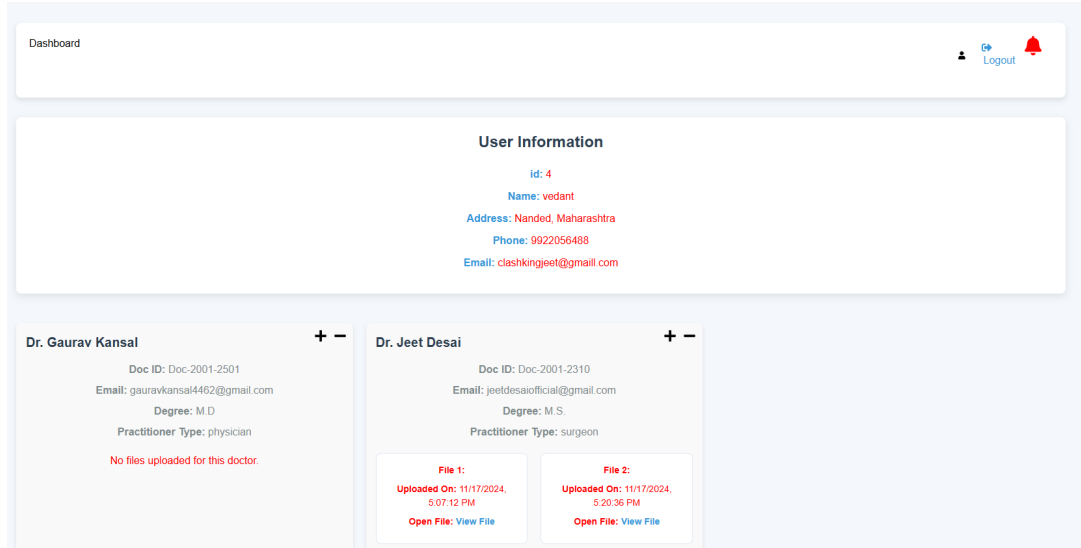


Figure 9: User Page

- **Doctor-Side Retrieval:**

- The doctor can only retrieve files if they have been granted access by the patient.
- Access is verified using the **check-access** API, which ensures that the doctor has the necessary permissions to view a particular patient's files.
- Once access is granted, the doctor can use the **getPatientFiles** function to retrieve the patient's file metadata (IPFS URL, timestamp, doctor's ID) from the blockchain.
- The doctor can then access and download the file directly from IPFS by clicking the link provided in the retrieved data.

- **Security and Access Control:**

- The blockchain ensures that the file metadata is tamper-proof, while access control is managed off-chain through a secure API that checks for the doctor's permission to access the patient's data.
- Files themselves remain off-chain on IPFS, with the blockchain simply storing the IPFS URL and related metadata.

This approach allows for secure, transparent, and permissioned access to sensitive medical files, with both patients and doctors able to retrieve data as needed, while maintaining the privacy and integrity of the data.

Transactions

Token Transfers (ERC-20)

Latest 18 from a total of 18 transactions

Download Page Data

Transaction Hash	Method	Block	Age	From	To	Amount	Txn Fee
<a href="#">0xe2d4568f488...</a>	Transfer	<a href="#">7111098</a>	10 hrs ago	<a href="#">0x993a0f36...63293adD9</a>	<a href="#">0x7c8dC6EE...bd9eA314a</a>	0.05 ETH	0.0000324
<a href="#">0xbc8bd8e111...</a>	<a href="#">0x4da20693</a>	<a href="#">7096262</a>	2 days ago	<a href="#">0x7c8dC6EE...bd9eA314a</a>	<a href="#">0x4c794147...4C013849E</a>	0 ETH	0.00040263
<a href="#">0xb5df6fb0672...</a>	<a href="#">0x4da20693</a>	<a href="#">7095426</a>	2 days ago	<a href="#">0x7c8dC6EE...bd9eA314a</a>	<a href="#">0x4c794147...4C013849E</a>	0 ETH	0.00035725
<a href="#">0x27ec9c53db...</a>	<a href="#">0x4da20693</a>	<a href="#">7095345</a>	2 days ago	<a href="#">0x7c8dC6EE...bd9eA314a</a>	<a href="#">0x4c794147...4C013849E</a>	0 ETH	0.00032199
<a href="#">0x57a81b90f47...</a>	<a href="#">0x4da20693</a>	<a href="#">7095281</a>	2 days ago	<a href="#">0x7c8dC6EE...bd9eA314a</a>	<a href="#">0x4c794147...4C013849E</a>	0 ETH	0.00036634
<a href="#">0xf178fe69d82...</a>	<a href="#">0x0806040</a>	<a href="#">7095087</a>	2 days ago	<a href="#">0x7c8dC6EE...bd9eA314a</a>	<a href="#">Contract Creation</a>	0 ETH	0.00155901
<a href="#">0x45cc4a8b00...</a>	<a href="#">0xfa4d14ce</a>	<a href="#">7041886</a>	10 days ago	<a href="#">0x7c8dC6EE...bd9eA314a</a>	<a href="#">0x0eB4B3b3...09A642a0B</a>	0 ETH	0.00055619
<a href="#">0xb4b575a878...</a>	<a href="#">0xfa4d14ce</a>	<a href="#">7040722</a>	11 days ago	<a href="#">0x7c8dC6EE...bd9eA314a</a>	<a href="#">0x0eB4B3b3...09A642a0B</a>	0 ETH	0.00058325
<a href="#">0xa3cb7c3091...</a>	<a href="#">0xfa4d14ce</a>	<a href="#">7037169</a>	11 days ago	<a href="#">0x7c8dC6EE...bd9eA314a</a>	<a href="#">0x0eB4B3b3...09A642a0B</a>	0 ETH	0.00075105

Figure 10: Blockchain

## 9 Results

The implementation of the blockchain-based healthcare record system achieved the following outcomes:

- **Secure File Upload to Blockchain:**

- Medical records and files are securely uploaded to the blockchain by doctors, ensuring immutability and transparency.
- Files are stored off-chain in IPFS, while the blockchain stores the metadata (doctor's ID, IPFS URL, timestamp).
- Each file is uploaded using the smart contract's `uploadFile` function, where the doctor's government ID and the IPFS URL are recorded on the blockchain.

- **Decentralized File Storage via IPFS:**

- The use of IPFS for file storage allows for efficient decentralized storage of patient records.
- Pinata API was utilized to interact with IPFS, ensuring the files are pinned and available for retrieval at any time.
- The IPFS links stored on the blockchain provide easy access to the medical files, ensuring they are both secure and accessible.

- **Access Control and Retrieval:**

- Files can be retrieved by both the patient and the doctor, but with access control.
- Patients have the ability to grant or deny access to their files, while doctors can only retrieve the files once access is granted.
- Both parties interact with the blockchain via the smart contract's `getPatientFiles` function, which returns the IPFS link for each file.

- **Error Handling and User Experience:**

- The system provides clear feedback on errors, such as invalid file uploads or access permission issues.
- The doctor interface allows for easy file uploads to IPFS and subsequent uploads to the blockchain, providing a seamless user experience.
- Clear notifications and alerts are used to inform users of the success or failure of file uploads, remark updates, and blockchain transactions.

- **Security and Privacy:**

- All file metadata is immutable and stored on the blockchain, ensuring transparency and integrity.
- Access to files is restricted to authorized individuals, and both doctors and patients have full control over who can view the medical records.
- MetaMask is used to interact with the blockchain, providing secure signing of transactions without exposing private keys.

The result of this implementation demonstrates a robust, secure, and efficient system for managing patient records using blockchain and IPFS. It ensures that patient data remains private, access-controlled, and tamper-proof, while enabling easy access for authorized individuals.

## 10 Future Scope

1. **AI-based Recommendation of Nearby Doctors:** An AI system can analyze patient preferences, ratings, geographical locations, and available specialties to recommend doctors nearby. This can be achieved by integrating the platform with a rating system, patient reviews, and location data, and using machine learning algorithms to make recommendations. The AI could also notify patients about free camps or health drives hosted by local medical practitioners or institutions, using real-time data from healthcare providers and organizations. By adding location-based services and machine learning algorithms that assess ratings, specialties, and availability, the system can provide automated suggestions. Additionally, a notification system can be set up for upcoming health events and free camps.
2. **Data Analysis without Disclosing Patient's Identity:** A data analysis engine powered by blockchain can be used to analyze disease trends, demographics, and treatment outcomes without compromising patient identity. Using zero-knowledge proofs or privacy-preserving techniques, researchers and healthcare professionals can access valuable insights while the patient's data remains private. Since all data is stored in a decentralized and secure manner, the integrity of the data can be maintained without exposing any personally identifiable information (PII). Blockchain platforms can implement privacy-preserving techniques such as differential privacy or zero-knowledge proofs to aggregate anonymized data. By analyzing the aggregated data, patterns related to diseases, treatments, and outcomes can be extracted without revealing sensitive information.

3. **Integration with Pharmacies, Medicals, and Insurance Companies:** The platform can be extended to interact with pharmacies, medical supply stores, and insurance companies. This integration can automate prescription fulfillment, insurance claims processing, and help pharmaceutical companies in tracking the dispensation of medicines. Pharmacies could receive prescriptions directly from doctors, and insurance companies could verify patient records to approve claims. Blockchain ensures secure data transmission and transparency throughout the process. By developing APIs that allow seamless interaction between healthcare providers, pharmacies, and insurance companies, this platform can automate the process of prescription management and insurance claim handling. Smart contracts on the blockchain could ensure the transparency of all transactions between parties.
4. **Telemedicine and Virtual Consultations:** Virtual consultations can be integrated into the platform, enabling patients to consult with doctors remotely through secure video calls and messaging. The platform can store and share medical records through blockchain, ensuring that doctors have access to the latest patient data during virtual consultations. The system can incorporate real-time video consultation tools, ensuring that each session is recorded, auditable, and associated with the patient's blockchain record for transparency and accountability. By adding a secure video communication feature that integrates with the patient's medical records, patients can access remote consultations. These sessions can be recorded, and the blockchain will store timestamps and proof of consultation, maintaining a secure, tamper-proof record.
5. **Blockchain for Clinical Trials and Research:** Blockchain can offer a transparent, immutable record-keeping system for clinical trials. By storing research data on a blockchain, researchers can ensure that trial data is tamper-proof, easily auditable, and compliant with regulations. This would help in managing patient consent, drug trial information, and treatment outcomes. Patients' data can be anonymized, allowing researchers to access aggregated data for analysis while ensuring patient privacy. Clinical trials can be registered on a blockchain-based platform where trial data is stored securely. Smart contracts could manage patient consent and track trial progress, ensuring transparency. Additionally, research organizations could access anonymized datasets via secure APIs, allowing them to analyze treatment outcomes and drug effectiveness.

## 11 Conclusion

In conclusion, the integration of blockchain technology in healthcare offers a secure, transparent, and efficient solution for managing patient records, file sharing, and doctor-patient interactions. By utilizing blockchain for storing patient files and medical histories, the system ensures data integrity and privacy, as well as providing a decentralized way to manage sensitive information. The use of IPFS for storing and sharing medical files further enhances the scalability and security of the system.

Through the implementation of smart contracts, doctors can easily upload, access, and manage patient data, while patients have control over their own medical information and can grant or revoke access as needed. Additionally, the future scope of this system includes the potential for AI-based recommendations, secure data analysis, and integration with pharmacies and insurance companies, which would further enhance the healthcare ecosystem.

Ultimately, this blockchain-based healthcare platform has the potential to revolutionize how medical records are managed, improving the efficiency and security of healthcare services, while providing patients with more control over their data. Further development and integration of emerging technologies could enhance the overall healthcare experience for both patients and providers.



## 12 References

- <https://ethereum.org/en/whitepaper/>
- MetaMask Official Website, <https://metamask.io/>
- Pinata IPFS Pinning Service, <https://pinata.cloud/>
- <https://ethereum.org/en/smart-contracts/>
- IPFS Official Website, <https://ipfs.io/>
- <https://soliditylang.org/>
- Introduction to Blockchain Technology - GeeksforGeeks, <https://www.geeksforgeeks.org/blockchain-technology/>
- How Blockchain is Revolutionizing Healthcare - GeeksforGeeks, <https://www.geeksforgeeks.org/how-blockchain-is-revolutionizing-healthcare/>
- <https://en.wikipedia.org/wiki/MetaMask>
- [https://en.wikipedia.org/wiki/InterPlanetary\\_File\\_System](https://en.wikipedia.org/wiki/InterPlanetary_File_System)
- Introduction to Ethereum Smart Contracts - GeeksforGeeks, <https://www.geeksforgeeks.org/introduction-to-ethereum-smart-contracts/>
- <https://blockchainhealth.com/>
- <https://theblockchainacademy.com/>
- <https://blockchainhealthsolutions.com/>