EthInsurance: A Blockchain based alternative approach for Health Insurance Claim

A MAJOR PROJECT REPORT

Submitted By

SIDDHANT SAWALKA [Reg No:RA1811030010050]

ARINDAM LAHIRI [Reg No:RA1811030010067]

Under the guidance of

Ms. D. SAVEETHA

(Assistant Professor, Department of Networking and Communications)

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE ENGINEERING

with specialization in CYBER SECURITY

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR: 603203

MAY 2022

Department of Networking and Communications
SRM Institute of Science & Technology
Own Work* Declaration Form

To be completed by the student for all assessments

Degree/ Course: B.Tech/Computer Science Engineering with specialization with Cyber Security

Student Name : Siddhant Sawalka/Arinda m Lahiri

Registration Number : RA1811030010050/RA1811030010067

Title of Work : EthInsurance: A Blockchain based alternative approach for Health Insurance Claim

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

Clearly references / listed all sources as appropriate

Referenced and put in inverted commas all quoted text (from books, web, etc.)

Given the sources of all pictures, data etc. that are not my own

Not made any use of the report(s) or essay(s) of any other student(s) either past or present

Acknowledged in appropriate places any help that I have received from others (e.g., fellow students, technicians, statisticians, external sources)

Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except were indicated by referring, and that I have followed the good academic practices noted above.

[illegible]	Arindam
RA1811030010050	RA1811030010067
24/04/2022	24/04/2022

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**ETHINSURANCE: A BLOCKCHAIN BASED ALTERNATIVE APPROACH FOR HEALTH INSURANCE CLAIM**" is the bonafide work of "SIDDHANT SAWALKA[RA1811030010050], ARINDAM LAHIRI [RA1811030010067]", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	SIGNATURE
Ms. D. Saveetha GUIDE Assistant Professor Dept. of Networking & Communications	Dr. Annapurani Panaiyappan.K HEAD OF THE DEPARTMENT Dept. of Networking and Communications

Signature of the Internal Examiner	Signature of the External Examiner

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr T.V. Gopal**, for his invaluable support.

We wish to thank **Dr Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department**, Dr K. Annapurani Panaiyappan** Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Panel Head, **Dr. Kayalvizhi Jayavel,** Assistant Professor, and program coordinators **Dr.M.B Mukesh Krishnan,** Associate Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisors**, Dr.M.Anand,** Assistant Professor, Department of Data Science and Business Systems and **Dr.M.Uma**, Associate Professor, Department of Computational Intelligence, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Ms. D.Saveetha,** Assistant Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of my interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Networking and Communications Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

[Siddhant Sawalka]

[Arindam Lahiri]

TABLE OF CONTENTS

ABSTRACT

Health Insurance is something the entire world relies on when it comes to medical treatment. But it is prone to several frauds between the process and removing them is really a challenge when it comes to India. The use of Blockchain technology increases the involvement of the user directly thus making it a clear process in every domain. Blockchain is used in this work to create a claim model that keeps insurance company transactions transparent. Patients, hospitals, and insurance companies will all be in direct touch with one other in an effort to eliminate fraud. In addition, it allows patients, clinicians, and other third parties to safely and efficiently access medical data.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

As long as several parties agree on the legality of a new participant entry, the blockchain is read-only and unchangeable. Health care and insurance firms across the globe are rapidly adopting blockchain technology. Most of the market share is expected to go to blockchain-based data sharing in the healthcare industry over the next few years. Various health insurance challenges, such as interoperability and standardization of medical data, can be addressed with blockchain technology, which might break down data silos in the industry. Due to the advantages of health data being stored universally, access to the whole medical history of patients, and reduced work of humans for processing the data of patients, contract processing is made visible by the capabilities of smart contract and distributed ledger of the blockchain The term "blockchain" In addition, blockchain provides for real-time monitoring and auditing. On the blockchain, all parties involved (doctors, pharmaceutical companies, and insurance companies) are held accountable. Thus, the healthcare industry as a whole, including insurers, policyholders, healthcare providers, and patients, can foster a climate of trust.

Over the last decade, most of the cities throughout the world have started to use technology to protect individual patient data provided by intelligent healthcare systems. Over time, more health records will be added and stored. The patient's physical characteristics such as weight and height are all added. Several parties are responsible for creating, storing, and changing data for effective utilization and care of the patients. Healthcare stakeholders were polled, and 16 percent said they anticipate to use blockchain solutions soon, while 56 percent said they want to use them by 2020. A blockchain is a digital ledger that is safe, private, and unchangeable. The blockchain stores transaction metadata. Smart contracts are used in blockchain technology to implement business logic and to automate a large number of transactions. Health care data is generated every day and stored in a database for future use. In order to avoid data leaks or tampering, this private medical information must be stored securely. Encryption and distributed ledger storage make it difficult to alter or gain unauthorized access to the data in a blockchain. Here, we suggest a health insurance system based on blockchain technology and smart contracts to achieve data secrecy, immutability, and transparency.

1.2 Motivation

Transparency and immutability are the hallmarks of blockchain transactions, making it an ideal solution for small-scale digital transactions. The optimal circumstances for its utilization are those where users on the network do not trust one another and where the data chunks are small. Patients, providers, and suppliers should all be able to prove their identity. Medical situations where blockchain can be particularly useful include handling the dynamic consent of patients to their data being utilized. Immutability, privacy of data and transparency in a blockchain-based health record are all interdependent. The patient is always the proprietor of their own record in an electronic health record thus liable for any changes to the record. The user is the one who has given others permission to read or edit their data. This type of change is transparent because it is automatically detected by the user. Unless the patient has given permission, no data may be deleted or modified, making the records irreversible. Finally, when it comes to privacy of data, each has its own owner, so it is publicly not accessible. Anyone who wants access to the medical records must first obtain permission. It has the potential to improve identification integrity and transparency while combating unequal versioning and maintaining consistent identity. As more information from wearable clinical devices becomes available, this becomes increasingly important. Individual and organizational health-care providers should provide accurate and up-to-date information about their location and service availability so that patients can find them, organizations can understand them, and payers can appropriately reimburse them. The reliability and accuracy of relevant statistics may be improved via a blockchain. It can provide a visible, auditable method for consumers to grant access to their personal fitness data to third parties using their unique credentials and encryption key. This includes allowing fitness professionals, carriers, and other relevant actors access to their clinical statistics and other data for the purposes of direct health-care delivery or allowing research, statistical, or other secondary uses of their data. The fundamental benefit of using blockchain here is that dissipates the necessity of a middleman. The user data contains information on the policies, and the policy provider company can request immediate access to the heath related data and begin the insurance claim procedure. A middleman's role between the provider company and the policy holder requesting treatment-related documentation would be obsolete.

1.3 Scope

Web-based application will be used for the project's primary goal of creating a decentralized network of patients, hospitals, and insurers. A lot of attention is paid to the idea of putting medical records on the blockchain. The goal is to make the insurance claim procedure easier for both the insurance company and the insured. As long as the network is entirely safe and sturdy, as well as completely flawless and smooth, it is never acceptable to compromise security. The goal of this project is to create a dAPP that allows patients to save all of their medical records in one location. During a health insurance claim for a specific treatment, the hospital and insurance company might both request access to those records.

1.4 Objective

The project's goal is to streamline and shorten the time it takes to process insurance claims. Instead of the hospital requesting payment from the insurance company, we are attempting to leverage insurance as a payment method via blockchain. The insurance provider looks into the patient's data to determine the exact amount of the claim. Before sending the final bill to the patient, the hospital settles any outstanding accounts. This reduces the amount of work the patient has to do while yet keeping the process as transparent as possible. Authentication and permission are all that are needed from the patient. As a result, it strives to offer a health insurance claim process that is risk-free, open, and simple.

1.5 Need

In India, the process of obtaining health insurance is just too time-consuming. A central point of storage to the patient's records is not available to the patient. Direct claims have a lower priority because the claim procedure prioritizes compensation. Compensation for numerous persons therefore necessitates the involvement of a middleman. A network that allows patients to access all of their medical records and treatment information in one place is essential. In addition to being more convenient for the insured, filing a claim directly with the hospital and having it settled there is a better way to submit the claim. This means that blockchain might be put to good use in both scenarios.

CHAPTER 2

LITERATURE REVIEW

The goal of this writing survey is to establish the information we need to use blockchain in healthcare and insurance by utilizing our scholarly information and research papers related to the topic. It also appears to provide us with a system for delineating references to the understandings and suspicions, if any exist.

We specifically investigate base papers on how to store health records on blockchain, as well as applications on insurance in healthcare using blockchain.

2.1 Research Findings

Title of Paper	**Journal/ Conference Name**	**Summary**	**Year**	**Author Names**
"Blockchain in Insurance: Exploratory Analysis of Prospects and Threats"	"International Journal of Advanced Computer Science and Applications, Vol. 12, No. 1"	Focuses on opportunities and threats in insurance sector. Uses blockchain to enhance processes like claim submission, processing and also fraud detection. Framework designed for three functionalities in the insurance sector i.e., claim submission, fraud detection and KYC of the client.	2021	"Anokye Acheampong AMPONSAH, Benjamin Asubam WEYORI,Adebayo Felix ADEKOYA"
"Blockchain applications in health care and public health: Increased transparency"	"JMIR MEDICAL INFORMATICS"	The goal is to provide an overview of blockchain solutions aimed at solving healthcare challenges from an industry perspective, with a focus on solutions developed by healthcare and technology companies.	2021	"Velmovitsky, P. E., Bublitz, F. M., Fadrique, L. X., & Morita, P. P"

“Health Insurance Claim Using Blockchain”	“International Research Journal of Engineering and Technology (IRJET)”	A design for insurance claim model in which Blockchain will help our system maintaining transparency between the insurer and the company. This proposed model replaces the agent and offers the direct contact between the insurer, hospital and the company	2021	“Jaya Kuckreja, Pranit Nigde, Pranjal Patil”
“A Study of Block Chain-Based Electronic HealthCare Record System”	“International Journal of Advanced Research in Science & Technology (IJARST)”	The blockchain use in medical systems leads to an automated data collection and verification process, modification and aggregation of data from a variety of immutable and tamper-proof sources, providing secure data with low potential for cybercrime. Increase. Investigate the development of intelligent technology and what is needed. Security requirements to implement in health care.	2020	“Singh, Mr Harjender”
“Implementation of Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Insurance Services”	“International Conference on Biomedical Innovations and Applications (BIA), Varna, Bulgaria”	Idea is to completely digitalize the insurance process. Use cryptography for data protection and authorizing visibility of data. Use smart contracts to create, monitor and automate the process and reduces role of brokers. Also provide a software solution using blockchain to be used in the insurance sector	2020	“Aleksieva, V.; Valchanov, H.; Huliyan, A"
“Blockchain-Based Medical Records Secure Storage and Medical”	“Journal of Medical Systems, Springer Science Business Media, LLC”	This paper showed light on how to use blockchain technology to check for all medical data together. Interoperability is a difficult due to scattered data. The chance of data leakage increases. A blockchain framework to store all health-related data, patient information, past medical history.	2019	“Yi Chen, Shuai Ding, Zheng Xu”

“A systematic review of blockchain”	“Southwestern University of Finance and Economics, Chengdu, China”	Learn how blockchain has evolved over the years, and its different versions. Take advantage of how the blockchain is used and how it translates into the benefits of the business. It gives us ideas in different areas where blockchain technology can be implemented	2019	“Min Xu, Gang Kou”
“Healthcare information exchange (HIE) using hyperledger fabric blockchain”	“Journal of Advanced Research in Dynamical and Control Systems”	The system is inefficient to protect privacy and integrity of health care data from unauthorized access attempted from inside the cloud environment Implementation of a distributed HIE system using Hyperledger fabric blockchain for reliable and trusted exchange of EMR’s which will overcome the limitations of conventional cloud-based ecosystem	2019	“Chechare, T., Murugan, A., Muriganantham, B., & Ganesh Kumar, S”
“A Blockchain-based Architecture Framework for Secure Sharing of Personal Health Data”	“IEEE 20th International Conference on e-Health Networking, Applications and Services”	A blockchain-based scheme for accessing data and monitoring its use. The architecture of the mechanism that controls the data after release. Improved privacy, minimized risk, better data management than current systems	2018	“Sandro Amofa, Emmanuel Boateng Sifah, Kwame O.-B Obour Agyekum, Smahi Abla, Qi Xia, James C. Gee, and Jianbin Gao”
“A Blockchain Framework for Insurance Processes”	IEEE	This document provides a distributed repository to replace the role of mediator who normally reviews all claims-	2018	“Mayank Raikwar, Subhra Mazumdar,

		related documents. It proposed a decentralized system for health insurance claims using blockchain technology		Sushmita Ruj, Sourav Sen Gupta, Anupam Chattopadhyay , and Kwok-Yan Lam”
“A design of blockchain-based architecture for the security of electronic health record (EHR) systems”	“IEEE International Conference on Cloud Computing Technology and Science”	This paper introduces the blockchain-based architecture of the Electronic Health Record System (EHR). This is a blockchain solution to improve the inter-operability of current EHR systems and prevent tampering with And misuse of EHR.	2018	“Yang, GuangLi, Chunlei”
“Integrating blockchain for data sharing and collaboration in mobile healthcare applications”	“IEEE”	It is impossible to overstate the importance of mobile and wearable technology in advancing healthcare and medical research. Sharing personal health information in a secure and simple manner is critical to enhancing the medical industry's interactions and collaboration.	2017	“Liang, X., Zhao, J., Shetty, S., Liu, J., & Li, D”

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

Healthcare systems today are more concerned with how to store electronic patient health data than anything else. One of their primary concerns is ensuring that medical histories are preserved securely so that they may be checked at any time using the blockchain-based ledger. However, when we talk about health records, we must also mention health insurance. In terms of healthcare, it's a critical component. In the healthcare industry, there has been relatively little discussion or study on how to merge blockchain use for record storage with such applications. Healthcare providers and patients file insurance claims on a daily basis with the insurance company. Currently, health care providers and insurance companies work with insurance agents to process insurance claims. Information such as medical records, insurance policy records, and personal details of patients may be leaked by insurers. It takes a long time and a lot of effort to file an insurance claim manually. Traditionally, a distributed database is managed by a single centralized system. These records are subject to alteration. Data breaches and record tampering pose a risk.to record.privacy and authentication when kept in a central server as technology advances and the number of records grows. In the healthcare and insurance industries, data integrity and access control are key concerns. Similarly, insurance companies are coping with the issue of fraudulent claims. False claims might be paid for by providing false information to the insurance company. Finding and preventing fraud has become a major problem for businesses, which not only costs money for the organization, but also increases the amount of time it takes to process and settle payments. The drawbacks of this system are frauds during claim, less transparency to the patient, no privacy to data and takes too long to apply for claim

3.2 Proposed System

We're working on an app that can be used in the healthcare industry to securely store health-care records and provide a transparent and clear procedure for filing health-insurance claims, among other things. In order to implement our proposed solution, we leveraged blockchain

technology. An important first step is for the user to supply their own personal information. There will be "blockchain blocks" in the system that will be used to hold information on a person's insurance policy, medical health report, and medical bill.
A blockchain is a network of nodes that may send and receive data that is stored in a distributed ledger. Contracts that interface with the EVM and communicate with users through the frontend are referred to as the dAPP collectively. End-user nodes connected with patients store and produce records on a regular basis. The primary goal of our proposed strategy is to hide the flaws in the current system. As a result, we built it to be capable of securely storing health records so that the entire medical history could be retrieved in the event of an emergency, all while attempting to speed up the health insurance claims process. According to the idea, health-care professionals and insurance companies would be expected to work together directly. As a result, the method does not require the involvement of a third party. This method also ensures that the CIA trinity is observed and followed. Because the patient maintains his or her own health records, they are private and only accessible to those who have been granted permission by the patient. The information is constantly accessible for reference because it is stored in a decentralized network. It is not possible to modify data that has been saved without the authorization of the owner. Non-repudiation is an important feature since it tells the patient (owner) about who is changing the record and when they are changing it. The dAPP is set up in such a way that any data addition requires the payment of a specific quantity of ETH, also known as a gas cost. Duplicate records have been removed from the dAPP, therefore security is being attempted to be maintained in the same way. In this system, a single patient will have only one record. One-stop access to health information, reduced claim fraud, more transparency in the claims process, easy and smooth claim processing, transparency and integrity of stored data, and the inability to deny or revoke health records in certain circumstances are all advantages of the proposed system.

3.3 System Requirements

3.3.1 Software Requirements

- Smart Contracts using Solidity
- Remix IDE for development and deployment of contracts
- Ganache blockchain for local development
- React for frontend

3.3.2 Hardware Requirements

- Dual core CPU (minimum) / Quad core CPU (recommended)
- 4GB RAM (minimum)/ 8GB RAM(recommended)
- 500GB free space (for full Ethereum blockchain)

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

The patient, the hospital, and the insurance company are all involved in the design of this project's system architecture. Only the patient has access to the blockchain where the patient's health records are maintained. The patient, the hospital, and the insurance company are the three main components of our architecture. The patient has complete control over who has access to their records because to the blockchain, which is linked directly to them. The Consensus Contract, Permission Contract, and Service Contract are the three contracts we use to do any kind of operation on the blockchain. Each module in the framework relies on these connections to function. The diagram below provides an overview of the architecture's basic structure.

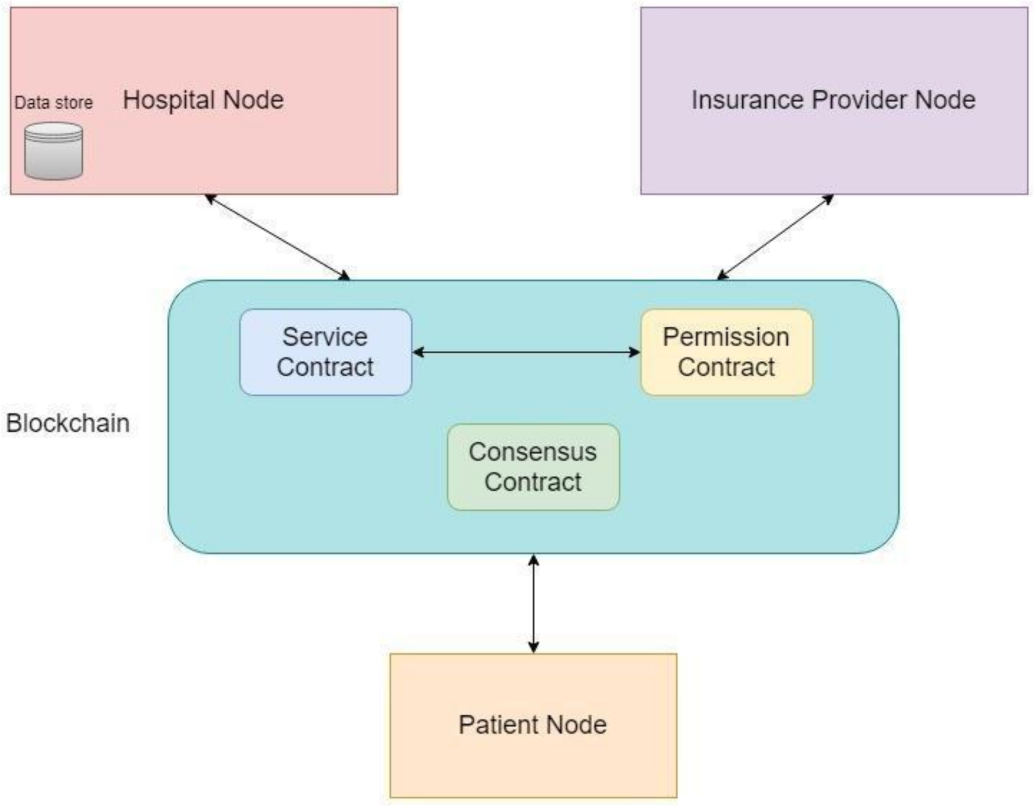


Fig 1: System Architecture

A collaborative procedure between the patient, the hospital (as a health-care provider), and the insurance company is envisioned for this project's system architecture. All three communicate with the blockchain using the dAPP, that stores the patient's health records, but only when the patient gives access.

The following is a representation of the dApp's basic architecture diagram:

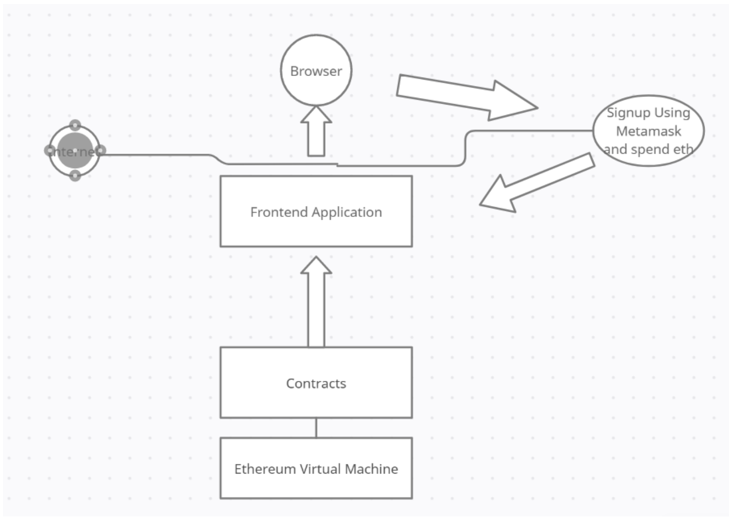
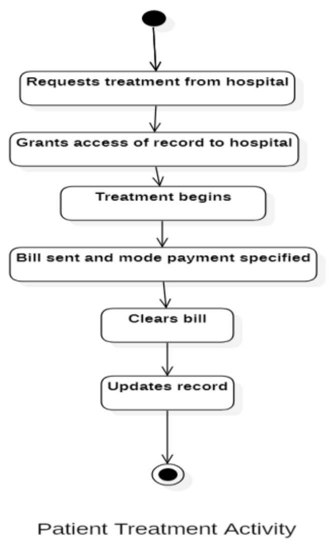


Fig 2: Architecture of dAPP

4.2 Activity Diagrams

4.2.1 Treatment Activity

The treatment of a patient is one of the activities going place between the three modules. Hospitalization is requested by the patients in this area. The hospital then begins therapy after requesting access to a patient's medical records. After the therapy is completed, the patient is emailed a final bill and asked to choose a payment method. If the payment method is an insurance claim. Having interacted with the claimant and successfully processed their claim, the hospital pays the patient's account. After this, the patient will receive their final bill. After the patient pays the bill, the hospital updates the patient's record.



Requests treatment from hospital
Grants access of record to hospital
Treatment begins
Bill sent and mode payment specified
Clears bill
Updates record
Patient Treatment Activity

Figure 3: Activity Diagram of Patient Treatment

4.2.2 Claim Activity

The claim procedure is the other major activity occurring in the modules. As the name suggests, it's all about insurance claims. An insurance claim is requested by the hospital when a patient specifies the method of payment as insurance. The insurance company asks for access to the patient's medical data to verify the insurance coverage. In order to process a claim, the insurance company first checks to see if the policy is valid. A claim amount is calculated based on the information provided, and the hospital is informed of the finalized amount. First, the hospital settles its bills with insurance, and then it sends a final bill to its patients for payment.

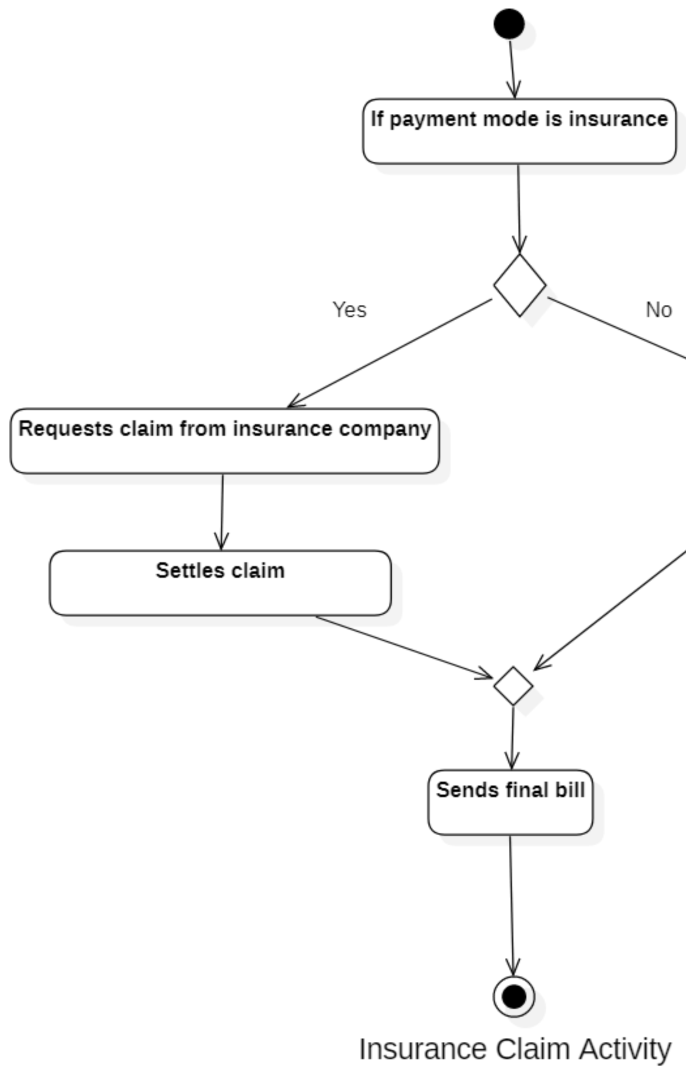

If payment mode is insurance
Yes
No
Requests claim from insurance company
Asks patient to clear bill
Settles claim
Sends final bill
Insurance Claim Activity

Figure 4: Activity Diagram of Insurance Claim

4.2.3 Sequence Diagram

A Sequence Diagram, as it is referred to, is a visual representation of the architecture's modules' interactions with the events that occur sequentially. Patients, the hospital, and the insurance company all communicate with one another and the blockchain in our design.

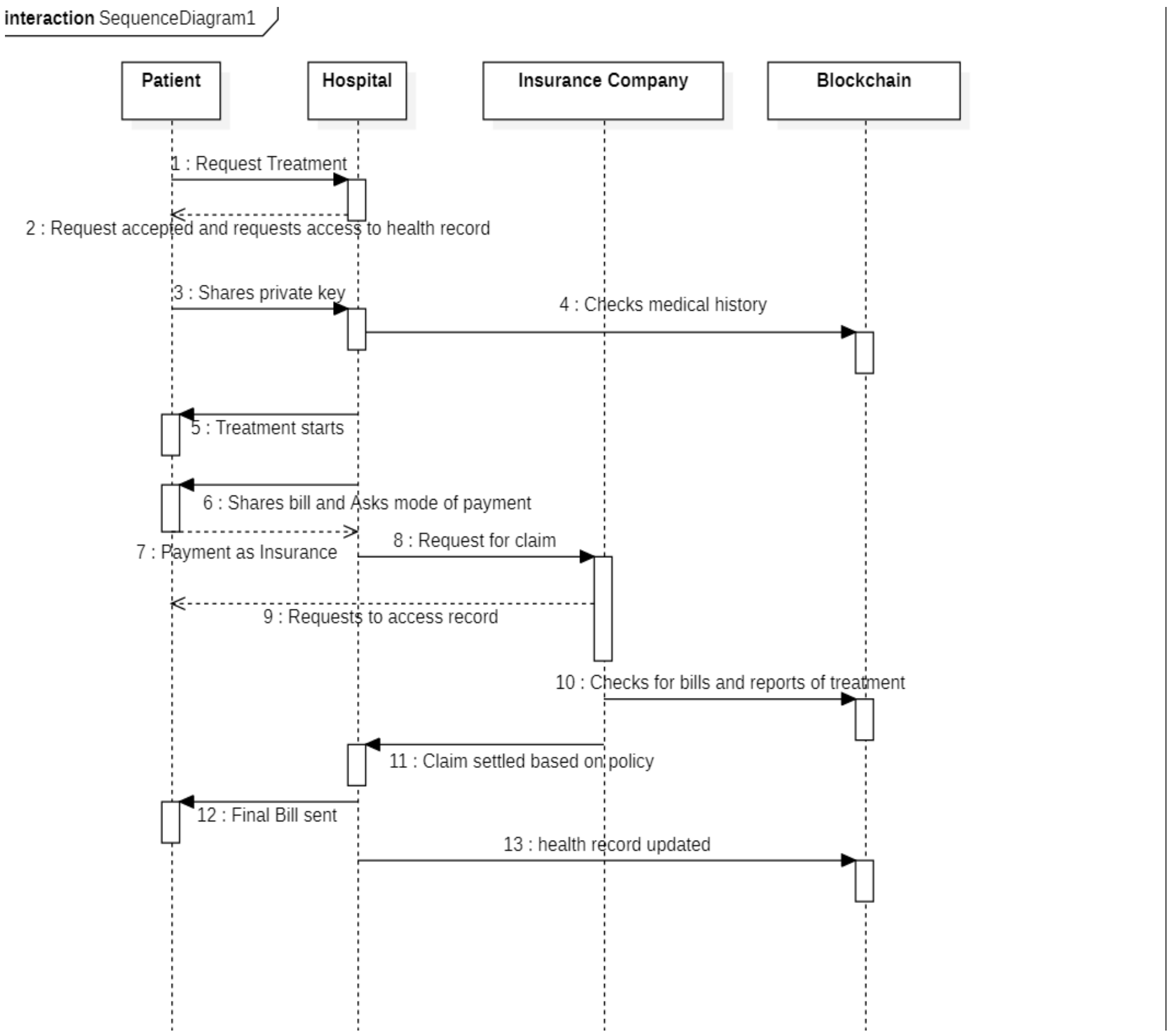

interaction SequenceDiagram1
Patient
Hospital
Insurance Company
Blockchain
1 : Request Treatment
2 : Request accepted and requests access to health record
3 : Shares private key
4 : Checks medical history
5 : Treatment starts
6 : Shares bill and Asks mode of payment
7 : Payment as Insurance
8 : Request for claim
9 : Requests to access record
10 : Checks for bills and reports of treatment
11 : Claim settled based on policy
12 : Final Bill sent
13 : health record updated

Figure 5: Sequence Diagram

CHAPTER 5
MODULES & ANALYSIS

5.1 Modules

5.1.1 Patient

An individual patient's medical history is stored in the blockchain using a patient module. They can update their own records since they have access to the blockchain and can interact with it, allowing them to read and write records. Throughout the duration of their stay at a hospital, the patient maintains regular contact with the facility. They only contact with the insurance company for the purpose of providing access to their records.

5.1.2 Hospital

There are regular interactions with patients and insurance companies in this module, which makes it the busiest one in the framework. It's also in charge of updating patient records as needed. The hospital asks for the patient's consent to record the patient's medical history, as well as to bill the patient's personal information. Patients are informed of the final bill once they ask the insurance company for a claim for a specific patient. In addition, they are in charge of ensuring that patient records are kept up to date following the conclusion of treatment.

5.1.3 Insurance Company

In order to settle a claim, this module verifies the patient's medical records and sends the claim amount to the hospital. To request access to read the record, you must speak with the patient.

5.2 Contracts

In a distributed ledger, contracts can be implemented, and contract execution can be fully automated. As computer code that governs interactions between network nodes and is kept on the blockchain, this notion is applied in numerous blockchain projects, including Ethereum and Hyper-Ledger. Using a public shared ledger on a distributed network, the run process will be immutable if you accept to execute.

5.2.1 Consensus Contract

By regularly approving new nodes' security, the blockchain uses consensus to identify new members of the network. In the event that a node needs to be terminated or overwritten, contracts can cause damage to the chain. All registered users and miners in the consensus agreement are recorded. The registered node's voting rights and Ethereum address are saved. Validating new nodes on the chain is also accomplished through the use of consensus contracts.

5.2.2 Service Contract

This guarantees that users are made aware of their use of data and the transactions that are occurring. Relationships between chain nodes are critical to the chain's overall functionality. A list of existing and future relationships is included in service contracts as a result. For each node, it also includes the Ethereum address. The contract can be used to check on the patient's request for consent.

5.2.3 Permission Contract

The addresses of various nodes that can access and interact with the records stored in the chain are stored here. Each time a record is added, a new one is created with a different id. You can have varying degrees of access to a site. To communicate in the framework, you can do so in two ways: first, when two contracts interact to execute a specific activity, and secondly, when one of the modules communicates with one of the contracts for activities like adding a node or seeking authorization.

CHAPTER 6

COMPONENTS OF dAPP

6.1 Tools / IDE

Remix IDE: In addition to being a desktop and web application, Remix IDE is free and open source. There are several plugins with simple user interfaces and a fast development cycle. Useful for contract development and as a teaching and learning tool for Ethereum. There are plugin-based development tools provided as part of the Remix Project. Subprojects include the Remix Plugin Engine, Remix Libs, and, of course, the Remix-IDE. For building solidity contracts, Remix IDE is a powerful open-source tool that can be used right from the browser. Deployment and deployment testing components are also included.

Visual Studio Code: Visual Studio Code is a small but powerful source code editor for Windows, macOS, and Linux that runs on your desktop. Additionally, it has a robust ecosystem of extensions for other programming languages (including C++/C#/Java/Python/PHP/Go) and runtimes (such as .NET and Unity).

Ganache Blockchain: A personal blockchain called Ganache enables the rapid construction of distributed apps based on the Ethereum and Corda platforms. There are no restrictions on when you can use Ganache; it can be used at any stage in the development cycle to construct, deploy, and test your decentralized applications. It's available in both a user-friendly and command-line interface version. Ganache UI is a desktop application that works with both Corda and Ethereum. Ganache-cli (formerly known as the TestRPC) has been released for Ethereum development.

Metamask: MetaMask is a browser extension for Ethereum enabled distributed applications (Dapps). That way, blockchain-reading dAPP can read from any website that uses the Ethereum web3 API. User approval or rejection of a Dapp's transaction is facilitated using a secure interface provided by the Dapp itself.

6.2 Technology Stack

Solidity: Solidity is an object-oriented, high-level programming language that is used to create smart contracts. Smart contracts are computer programmes that regulate the behaviour of accounts in the Ethereum state of affairs. It is a curly-bracket programming language that is intended for use with the Ethereum Virtual Machine (EVM). C++, Python, and JavaScript have all had an impact on it. In the section on language influences, you can learn more about the languages that Solidity was influenced by and how Solidity was created. Among other things, it is statically typed and enables inheritance, libraries, and sophisticated user-defined types, among other things. You may use this to establish contracts for a variety of purposes, including voting, crowdsourcing, blind auctions, and multi-signature wallets. We are using solidity version 0.8.7 and have written the code in the Remix IDE about which we have mention earlier.

React: When it comes to developing user interfaces, React is a free and open-source front-end JavaScript library that relies on UI components. Next.js, for example, may be used as a foundation for the construction of single-page, mobile, and server-rendered applications, while React can be used as a platform for these types of applications. The main focus of React is with state management and rendering that information to the DOM; therefore, constructing React apps typically indulges the usage of additional libraries for routing and some client-side functions in addition to the use of React.

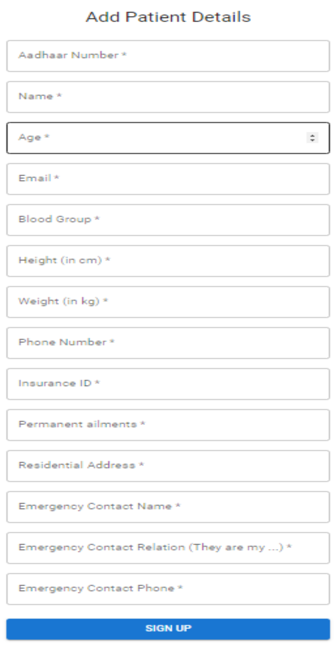
Material UI: Material UI is a front-end framework for React that is free and open-source. Based on Google's Material Design, Material UI is designed to give a high-quality digital experience while designing front-end visuals for websites and other applications.

CHAPTER 7

FUNCTIONALITIES

7.1 Add Patient

The add patient functionality is basically like a patient registration to the dApp. It is a onetime function for every patient who wants to get added to the blockchain. Here the patient has to enter their personal details along with their previous medical history. This helps maintain all patient records in one place making it easier to access. Here in our dApp, patients **Aadhar number** is being used a unique id. If he or she tries to register again with same Aadhar number the dApp returns an error.



Add Patient Details

Aadhaar Number *

Name *

Age *

Email *

Blood Group *

Height (in cm) *

Weight (in kg) *

Phone Number *

Insurance ID *

Permanent ailments *

Residential Address *

Emergency Contact Name *

Emergency Contact Relation (They are my ...) *

Emergency Contact Phone *

SIGN UP

Fig 6: Add Patient

7.2 Get Patient

The get patient details functionality is basically to retrieve and check the details of patient which are already stored on the blockchain. As mentioned earlier, the patient aadhar is the unique identification parameter thus it is being used to retrieve all patient data stored with that particular aadhar as the unique parameter.

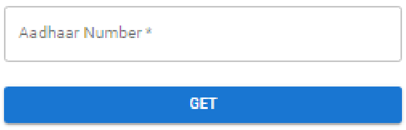


Fig 7: Get Patient Details

7.3 Add Hospital

The add hospital functionality is like a hospital registration to the dApp. It is a one-time function for every hospital for getting added to the blockchain. It is a mandate to get added if any patient record has to be accessed. Here the hospital enters all the necessary details like it address, contact , email etc. This is a step to maintain a level of security by allowing only registered hospitals to ask access for health records.

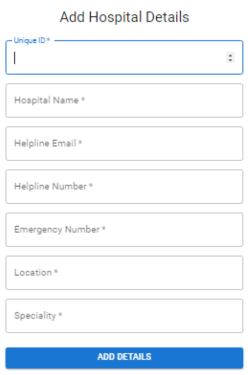


Fig 8 : Add Hospital

7.4 Add Treatment Details

This is a hospital specific functionality. Here the hospital adds the treatments details to the blockchain when a patient gets treated at the hospital. Like mentioned before, the dApp is created to maintain a one-stop access for patient health records thus when the patient is treated the hospital directly adds the details to the blockchain. Here the hospital assigns a unique treatment id for each treatment so as to keep a check on the treatments conducted.

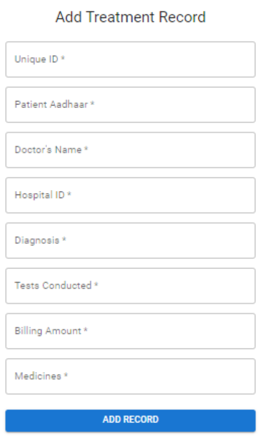


Fig 9: Add Treatment

7.5 Get Patient Treatment Details

The get patient treatment details is basically to retrieve the patient treatment details which they have undergone at the hospital. The treatment details can be retrieved using the treatment id which was the unique number related to that treatment assigned by the hospital.

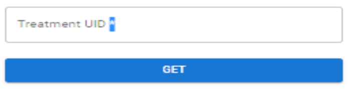


Fig 10: Get Treatment Details

7.6 Add Insurance Provider

The add insurance provider functionality is like a provider registration to the dApp. It is a one-time function for every provider for getting added to the blockchain. It is a mandate to get added if any patient record has to be accessed. Here the provider enters all the necessary details like it address, contact , email etc. This is a step to maintain a level of security by allowing only registered providers to ask access for health records.

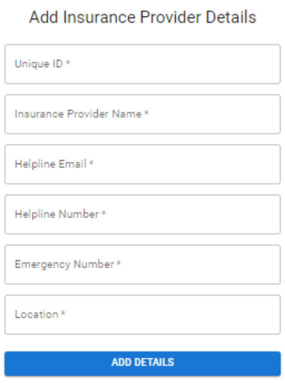


Fig 11: Add Insurance Provider

7.7 Add Insurance Policy

The add insurance policy is the where the policy details are entered. Like all details based on the policy number are added in this

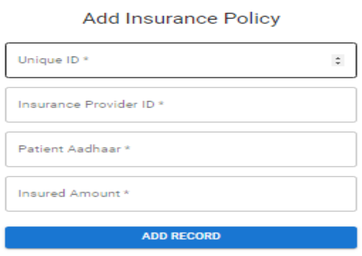


Fig 12: Add Insurance Policy

7.8 Get Policy Details

Here based on the based on the policy details whatever details are stored on the blockchain are retrieved back.

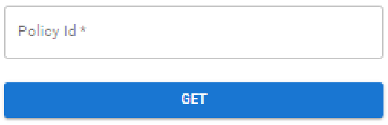


Fig 13: Get Policy Details

7.9 Connect to Metamask

It is like an on-off button on the dApp where the user can directly interact with the Metamask extension before using the functionalities. Once the Metamask is connected to the desired account the button changed to "Disconnect" only after which the user can interact and add data to the blockchain.

Fig 14: Connect to Metamask

CHAPTER 8

CONNECTION WITH DAPP

8.1 Connect to Metamask

- Click on the Connect to Metamask button

Fig 15: Connect to Metamask

- Metamask extension pops up and select account which has to selected.

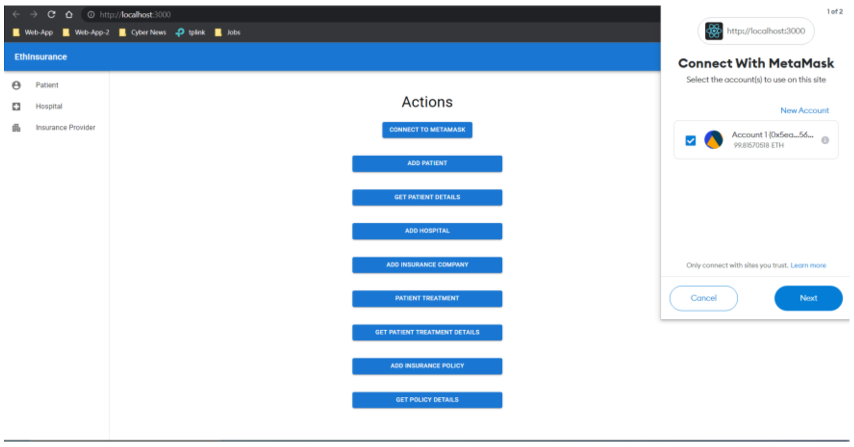
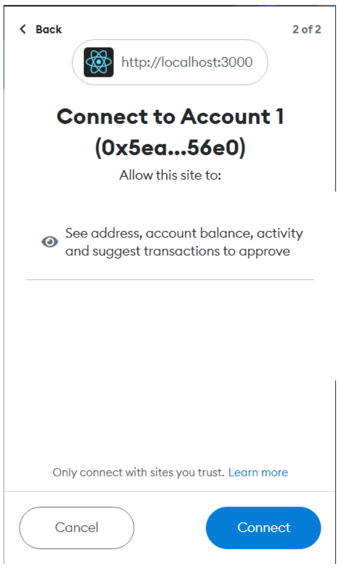


Fig 16: Metamask connection popup

- Click Connect for getting connection


Back

2 of 2

http://localhost:3000

Connect to Account 1 (0x5ea...56e0)

Allow this site to:

See address, account balance, activity and suggest transactions to approve

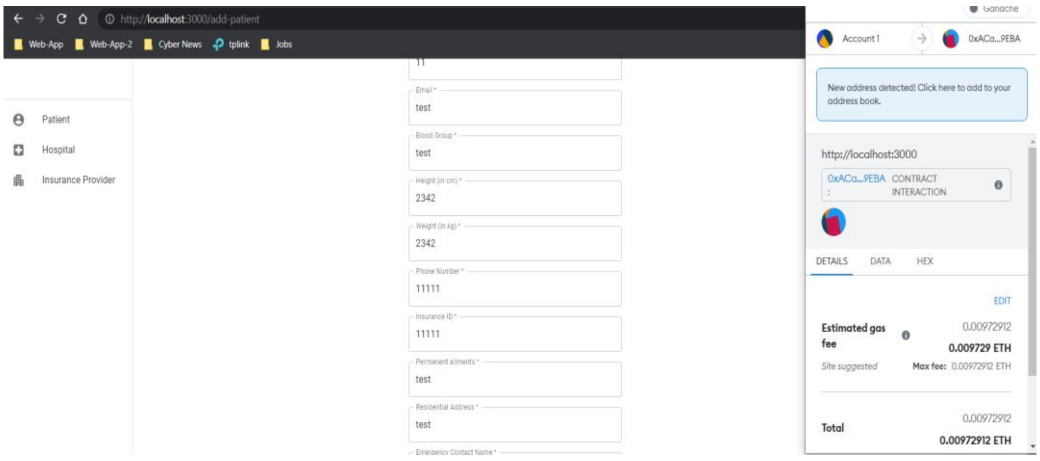
Only connect with sites you trust. Learn more

Cancel | Connect

Fig 17: Connect to account

8.2 Add details on Blockchain

- Click on Add Patient/ Add Hospital / Add Insurance Company / Add Insurance Policy
- Add the information with respect to the fields on that page
- Click on Signup and the Metamask pop ups directly from the extension


http://localhost:3000/add-patient

Web-App | Web-App-2 | Cyber News | tplink | Jobs

Patient
Hospital
Insurance Provider

11

Email *: test

Blood Group *: test

Height (in cm) *: 2342

Weight (in kg) *: 2342

Phone Number *: 11111

Insurance ID *: 11111

Permanent ailments *: test

Residential Address *: test

Emergency Contact Name *

Ganache

Account 1 → 0xACa...9EBA

New address detected! Click here to add to your address book.

http://localhost:3000

0xACa...9EBA : CONTRACT INTERACTION

DETAILS | DATA | HEX

EDIT

Estimated gas fee: 0.00972912 — 0.009729 ETH

Site suggested — Max fee: 0.00972912 ETH

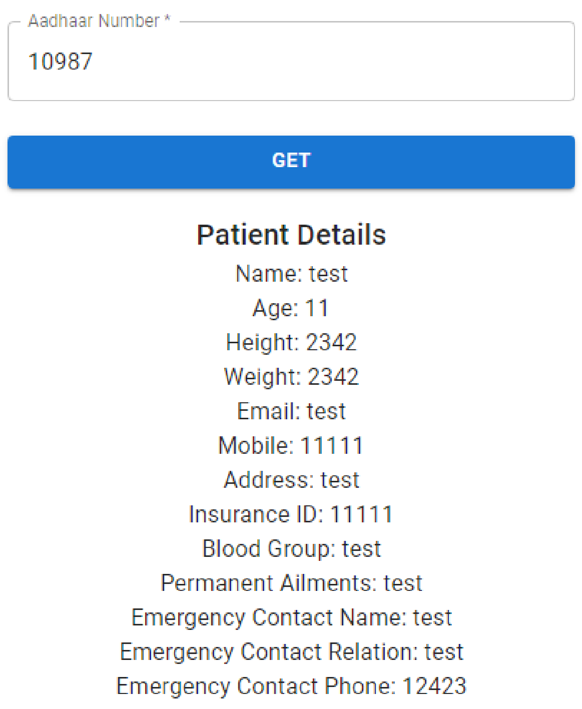
Total: 0.00972912 — 0.00972912 ETH

Fig 18: Gas Fee for adding data to Blockchain

- Check the gas fee and click confirm. The details get added on the blockchain.

8.3 Get Details from Blockchain

- Click on any get function seen on the dAPP
- There is any one particular field which is the unique id for the module
- Enter the field value and the details appear.


Aadhaar Number *

10987

Patient Details

Name: test

Age: 11

Height: 2342

Weight: 2342

Email: test

Mobile: 11111

Address: test

Insurance ID: 11111

Blood Group: test

Permanent Ailments: test

Emergency Contact Name: test

Emergency Contact Relation: test

Emergency Contact Phone: 12423

Fig 19: Get Details from Blockchain

- No gas fee is charged for the get function.

CHAPTER 9

CONCLUSION

Decentralization, permanence, anonymity, and verifiability are just a few of the benefits that blockchain offers over traditional systems. Reliability and safety are guaranteed by the suggested system. A decentralized healthcare system built on the blockchain that removes the associated costs. Before transferring the record over the network, validation and authorization are used to decrease the risk of unauthorized use of the record. To prevent illegal access, each patient is assigned an individual Ethereum address and identifier. The framework's data security is further enhanced by the use of multiple contracts as stated. In-transaction registration and paperwork may be more necessary if you're running numerous contracts on a single node. In order to secure the highest level of security for your health record, this system uses some sensitive health information.

CHAPTER 10

FUTURE IMPROVEMENTS

The proposed framework is an improvement in the model that eliminates the cost of intermediaries in obtaining insurance, and the hospital handles the application process directly with the insurance company, eliminating the hassle of the insurance company's work. However, a front end for such applications is absolutely necessary for a smooth process. Therefore, creating a dApp using a proposed model with a blockchain on the backend is a great opportunity to use this transition for your customers. There are also doubts about adding various security mechanisms to data stored in web applications and block chains. Therefore, it is important to implement security best practices and guidelines. Our ultimate goal is to build a completely secure architecture by always adhering to security controls.

REFERENCES

[1] Sari, P. K., & Yazid, S. (2020). Design of blockchain-based electronic health records for Indonesian context: Narrative review. *2020 International Workshop on Big Data and Information Security, IWBIS 2020*.

[2] Chechare, T., Murugan, A., Muriganantham, B., & Ganesh Kumar, S. (2019). Healthcare information exchange (HIE) using hyper ledger fabric blockchain. *Journal of Advanced Research in Dynamical and Control Systems*, *11*(5 Special Issue).

[3] al Omar, A., Jamil, A. K., Nur, M. S. H., Hasan, M. M., Bosri, R., Bhuiyan, M. Z. A., & Rahman, M. S. (2020). Towards a transparent and privacy-preserving healthcare platform with blockchain for smart cities. *Proceedings - 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2020*.

[4] Akbar, A., & Ali Khan, A. M. (2021). Modernizing the Health Insurance Industry Using Blockchain and Smart Contracts. In *Blockchain for Healthcare Systems*. https://doi.org/10.1201/9781003141471-6

[5] Liang, X., Zhao, J., Shetty, S., Liu, J., & Li, D. (2018). Integrating blockchain for data sharing and collaboration in mobile healthcare applications. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, *2017-October*. https://doi.org/10.1109/PIMRC.2017.8292361

[6] Fauziah, Z., Latifah, H., Omar, X., Khoirunisa, A., & Millah, S. (2020). Application of Blockchain Technology in Smart Contracts: A Systematic Literature Review. *Aptisi Transactions on Technopreneurship (ATT)*, *2*(2). https://doi.org/10.34306/att.v2i2.97

[7] Velmovitsky, P. E., Bublitz, F. M., Fadrique, L. X., & Morita, P. P. (2021). Blockchain applications in health care and public health: Increased transparency. In *JMIR Medical Informatics* (Vol. 9, Issue 6).

[8] Bell, L., Buchanan, W. J., Cameron, J., & Lo, O. (2018). Applications of Blockchain Within Healthcare. *Blockchain in Healthcare Today*, *1*. https://doi.org/10.30953/bhty.v1.8

[9] Tanwar, S., Parekh, K., & Evans, R. (2020). Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *Journal of Information Security and Applications*, *50*.

[10] Saldamli, G., Reddy, V., Bojja, K. S., Gururaja, M. K., Doddaveerappa, Y., & Tawalbeh, L. (2020). Health Care Insurance Fraud Detection Using Blockchain. *2020 7th International Conference on Software Defined Systems, SDS 2020*.

[11] S, M. C. (2008). Secure Health Care Data using Blockchain-A Survey. *International Research Journal of Engineering and Technology*.

[12] Singh, M. H., & Professor, A. (2020). A Study of Block Chain-Based Electronic HealthCare Record System. *International Journal of Advanced Research in Science & Technology (IJARST)*, *5*(5).

[13] Shobana, G., & Suguna, M. (2019). Block Chain Technology towards Identity Management in Health Care Application. *Proceedings of the 3rd International Conference on I-SMAC IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2019*.

[14] Jagtap, S. T., Thakar, C. M., el Imrani, O., Phasinam, K., Garg, S., & Ventayen, R. J. M. (2021). A Framework for Secure Healthcare System Using Blockchain and Smart Contracts. *Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021*.

[15] Kakkar, B., Johri, P., & Kumar, A. (2021). Blockchain Applications in various sectors beyond: Bitcoin. *2021 International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2021*. https://doi.org/10.1109/ICACITE51222.2021.9404580

[16] Grigoreva, E. A., Garifova, L. F., & Polovkina, E. A. (2019). The future of digital technology in russia: Blockchain as one of the priority directions of development. *International Journal on Emerging Technologies*, *10*(2).

[17] Gong, J., & Zhao, L. (2020). Blockchain application in healthcare service mode based on Health Data Bank. *Frontiers of Engineering Management*, *7*(4). https://doi.org/10.1007/s42524-020-0138-9

[18] Shrimali, B., & Patel, H. B. (2021). Blockchain state-of-the-art: architecture, use cases, consensus, challenges and opportunities. In *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2021.08.005

[19] Acheampong AMPONSAH, A., Asubam WEYORI, B., & Adebayo Felix ADEKOYA, P. (n.d.). Blockchain in Insurance: Exploratory Analysis of Prospects and Threats. In (*IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 12, Issue 1). www.ijacsa.thesai.org

[20] Ajayi, O., Abouali, M., & Saadawi, T. (n.d.). *Blockchain Architecture for Secured Inter-Healthcare Electronic Health Records Exchange*.

[21] Yang, G., & Li, C. (2018). A design of blockchain-based architecture for the security of electronic health record (EHR) systems. *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom*, *2018-December*, 261–265. https://doi.org/10.1109/CloudCom2018.2018.00058

[22] Agbo, C. C., Mahmoud, Q. H., & Eklund, J. M. (2019). Blockchain technology in healthcare: A systematic review. In *Healthcare (Switzerland)* (Vol. 7, Issue 2). https://doi.org/10.3390/healthcare7020056

[23] Hölbl, M., Kompara, M., Kamišalić, A., & Zlatolas, L. N. (2018). A systematic review of the use of blockchain in healthcare. *Symmetry*, *10*(10). https://doi.org/10.3390/sym10100470

[24] Zubaydi, H. D., Chong, Y. W., Ko, K., Hanshi, S. M., & Karuppayah, S. (2019). A review on the role of blockchain technology in the healthcare domain. In *Electronics (Switzerland)* (Vol. 8, Issue 6). https://doi.org/10.3390/electronics8060679

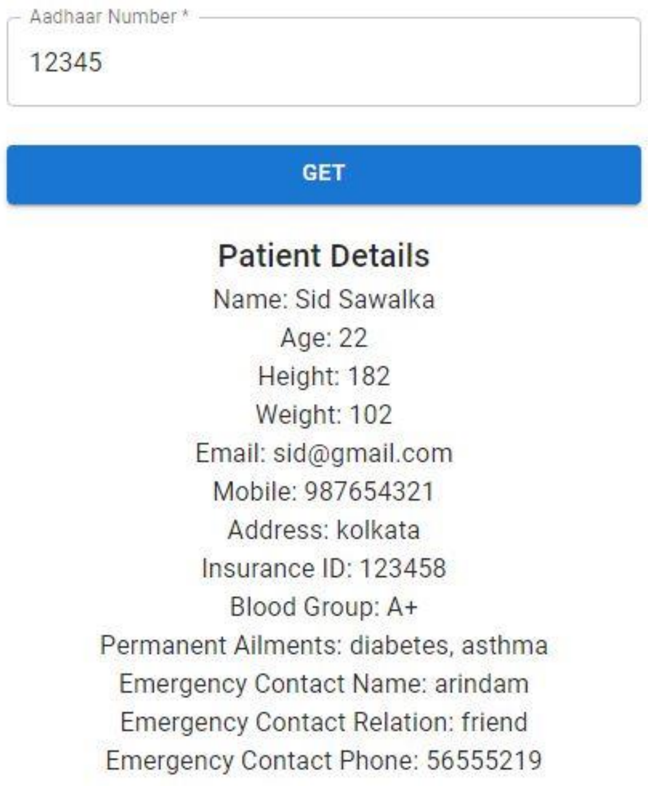
[25] Dorcas, A. D., Afolashade, K., Bonde, L., & Olujimi, A. (2020). Application of Blockchain Technology to HealthCare Sector: A Review. *International Journal of Computer Trends and Technology*, *68*(3).

[26] Sandro Amofa, Emmanuel Boateng Sifah, Kwame O.-B Obour Agyekum, Smahi Abla, Qi Xia, James C. Gee, and Jianbin Gao," A Blockchain-based Architecture Framework for Secure Sharing of Personal Health Data",2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)

[27] Aleksieva, V.; Valchanov, H.; Huliyan, A. Implementation of Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Insurance Services. In Proceedings of the International Conference on Biomedical Innovations and Applications (BIA), Varna, Bulgaria, 24–27 September 2020; IEEE: New York, NY, USA, 2020.

APPENDIX

SAMPLE OUTPUTS from dAPP:

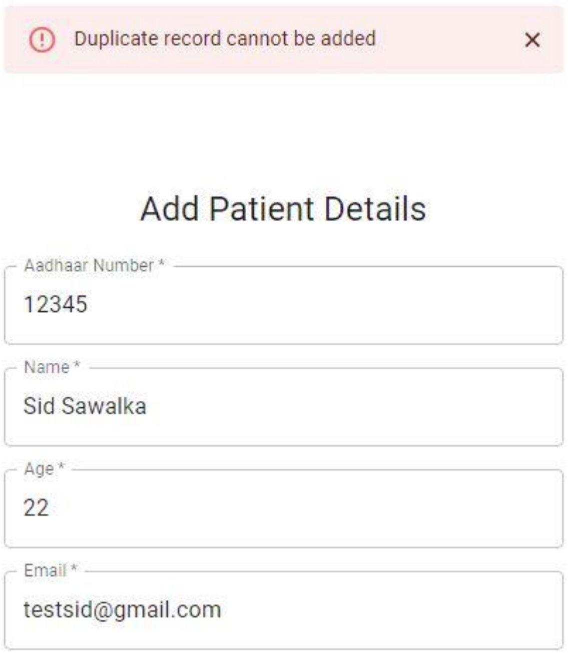


12345

Patient Details

Name: Sid Sawalka
Age: 22
Height: 182
Weight: 102
Email: sid@gmail.com
Mobile: 987654321
Address: kolkata
Insurance ID: 123458
Blood Group: A+
Permanent Ailments: diabetes, asthma
Emergency Contact Name: arindam
Emergency Contact Relation: friend
Emergency Contact Phone: 56555219

Fig 20: Get Patient Details


Duplicate record cannot be added

Add Patient Details

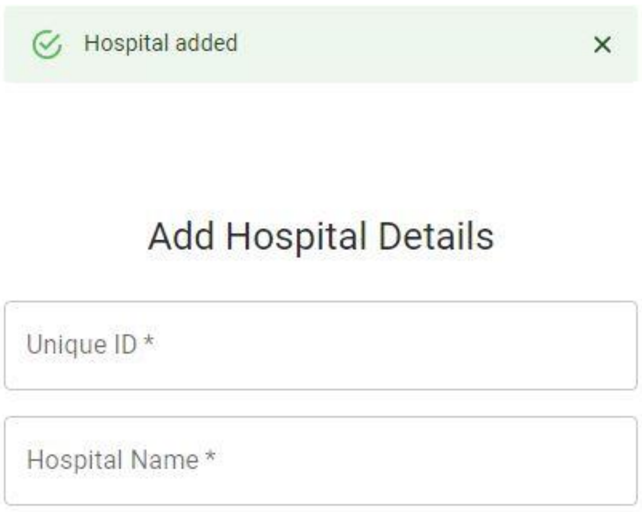
Aadhaar Number * 12345

Name * Sid Sawalka

Age * 22

Email * testsid@gmail.com

Fig 21: Duplicate record validation for Aadhar Number

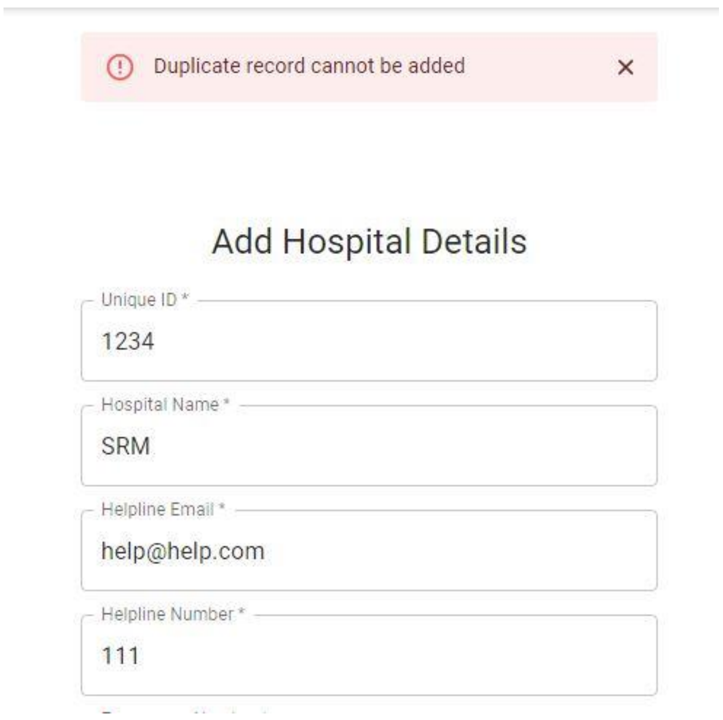

Hospital added

Add Hospital Details

Unique ID *

Hospital Name *

Fig 22: Add Hospital


Duplicate record cannot be added

Add Hospital Details

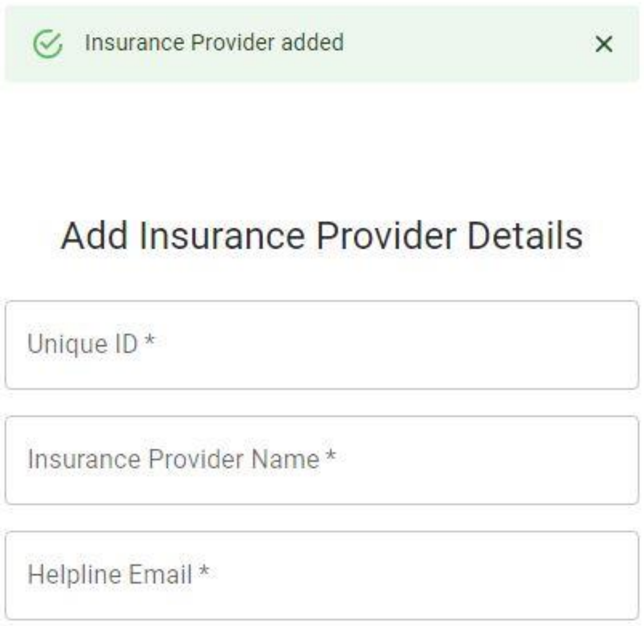
Unique ID *: 1234

Hospital Name *: SRM

Helpline Email *: help@help.com

Helpline Number *: 111

Fig 23: Duplicate record validation for Unique ID


Insurance Provider added

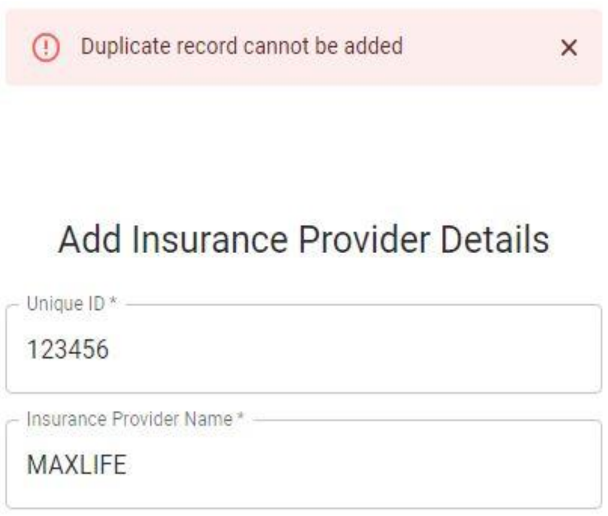
Add Insurance Provider Details

Unique ID *

Insurance Provider Name *

Helpline Email *

Fig 24: Add Insurance Provider

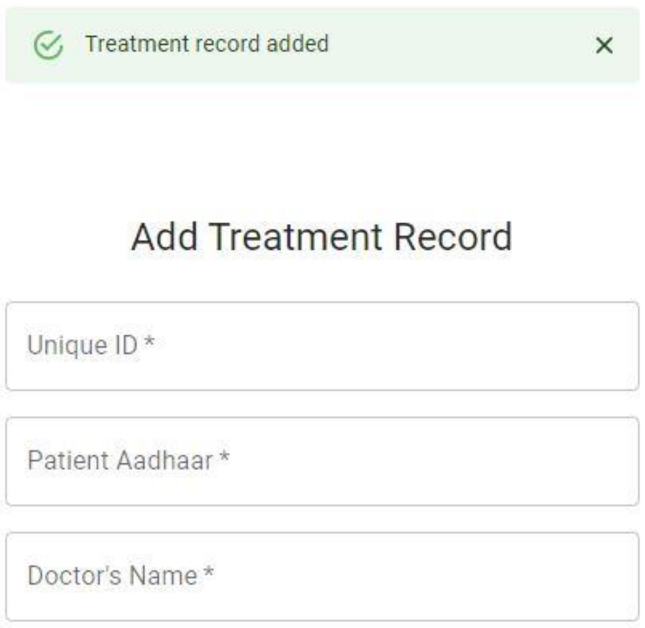

Duplicate record cannot be added

Add Insurance Provider Details

Unique ID *: 123456

Insurance Provider Name *: MAXLIFE

Fig 25: Duplicate record validation for Unique Id


Treatment record added

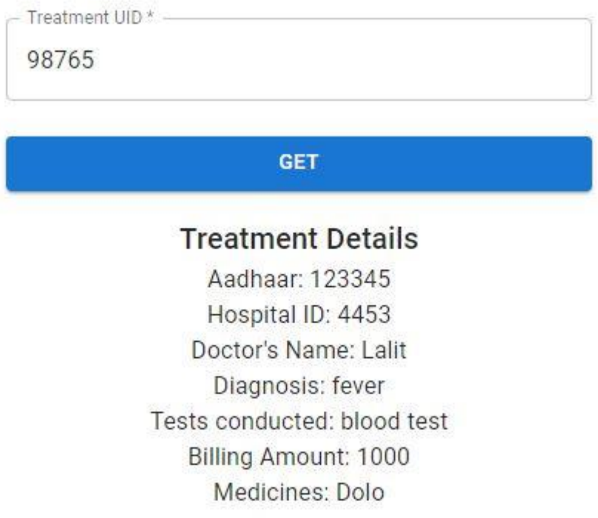
Add Treatment Record

Unique ID *

Patient Aadhaar *

Doctor's Name *

Fig 26: Add Patient Treatment


Treatment UID *

98765

Treatment Details

Aadhaar: 123345

Hospital ID: 4453

Doctor's Name: Lalit

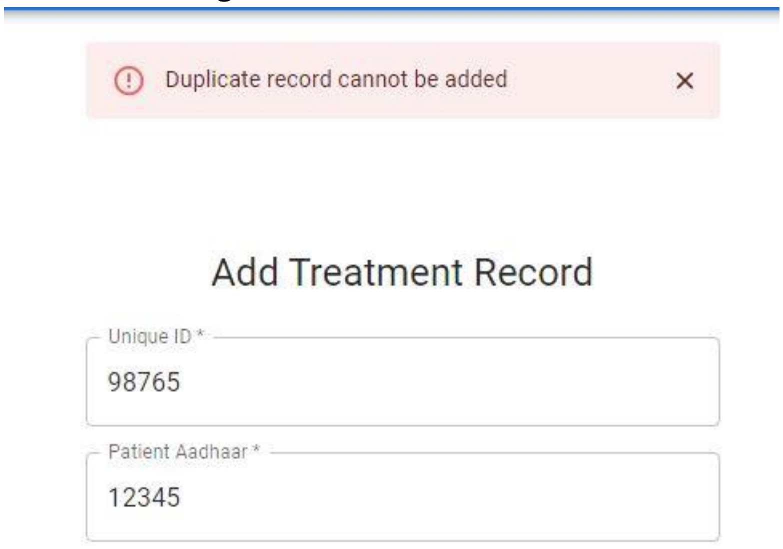
Diagnosis: fever

Tests conducted: blood test

Billing Amount: 1000

Medicines: Dolo

Fig 27: Get Treatment Details


Duplicate record cannot be added

Add Treatment Record

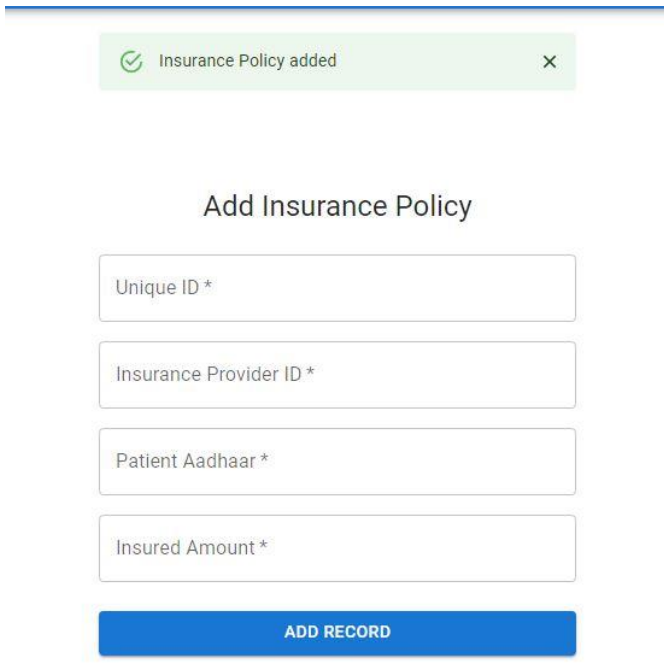
Unique ID *

98765

Patient Aadhaar *

12345

Fig 28: Duplicate record validation for Unique ID


Insurance Policy added

Add Insurance Policy

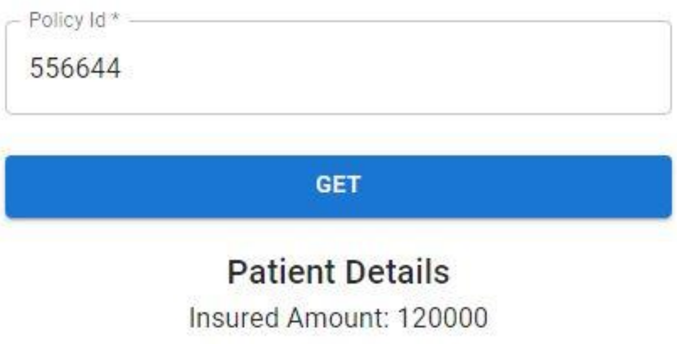
Unique ID *

Insurance Provider ID *

Patient Aadhaar *

Insured Amount *

Fig 29: Add Insurance Policy

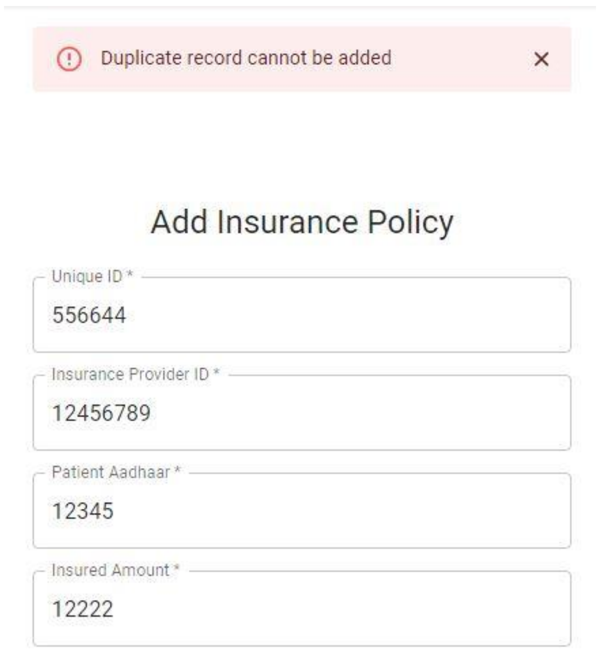

Policy Id *

556644

Patient Details

Insured Amount: 120000

Fig 30: Get Policy



Duplicate record cannot be added

Add Insurance Policy

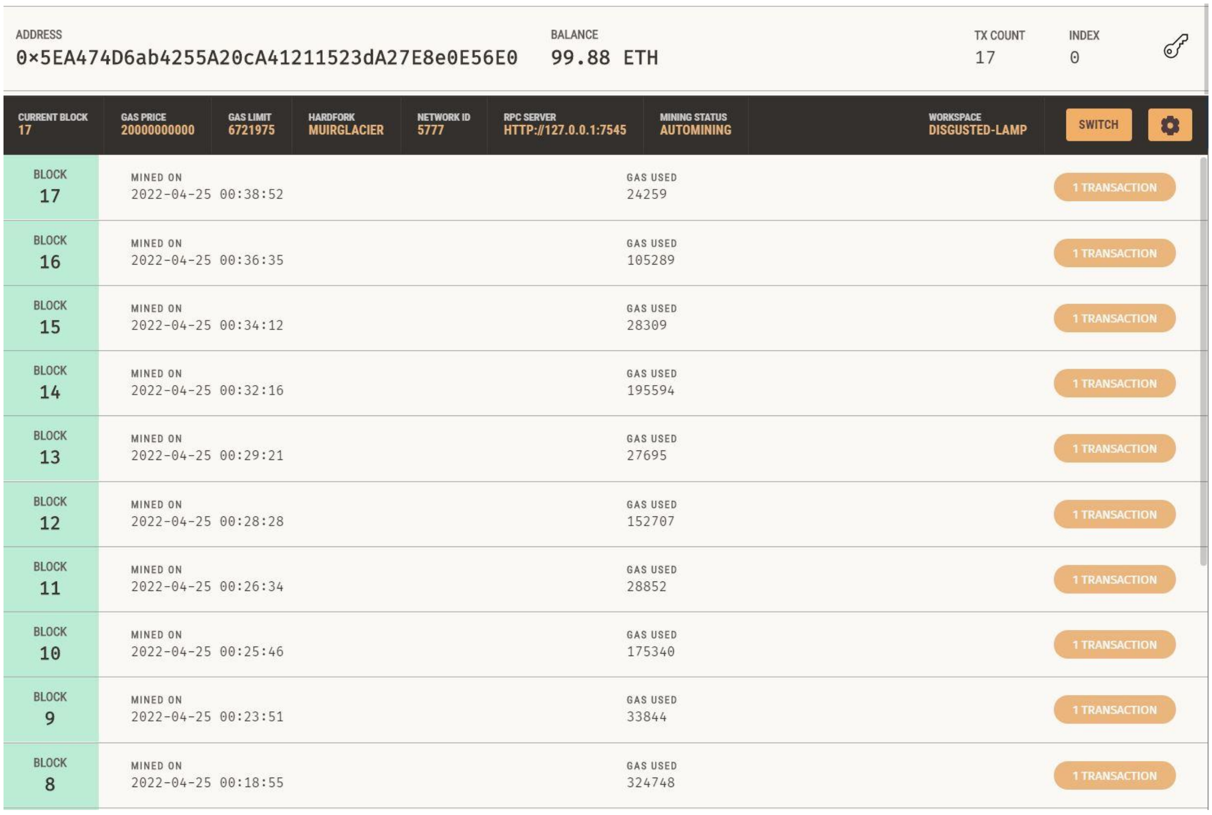
Unique ID * 556644

Insurance Provider ID * 12456789

Patient Aadhaar * 12345

Insured Amount * 12222

Fig 31: Duplicate record validation for Unique Id

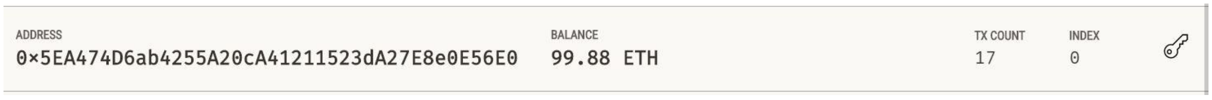


ADDRESS 0x5EA474D6ab4255A20cA41211523dA27E8e0E56E0 BALANCE 99.88 ETH TX COUNT 17 INDEX 0

CURRENT BLOCK 17 GAS PRICE 20000000000 GAS LIMIT 6721975 HARDFORK MUIRGLACIER NETWORK ID 5777 RPC SERVER HTTP://127.0.0.1:7545 MINING STATUS AUTOMINING WORKSPACE DISGUSTED-LAMP

BLOCK	MINED ON	GAS USED	
17	2022-04-25 00:38:52	24259	1 TRANSACTION
16	2022-04-25 00:36:35	105289	1 TRANSACTION
15	2022-04-25 00:34:12	28309	1 TRANSACTION
14	2022-04-25 00:32:16	195594	1 TRANSACTION
13	2022-04-25 00:29:21	27695	1 TRANSACTION
12	2022-04-25 00:28:28	152707	1 TRANSACTION
11	2022-04-25 00:26:34	28852	1 TRANSACTION
10	2022-04-25 00:25:46	175340	1 TRANSACTION
9	2022-04-25 00:23:51	33844	1 TRANSACTION
8	2022-04-25 00:18:55	324748	1 TRANSACTION

Fig 32: Transactions Block shown in Ganache Blockchain



ADDRESS 0x5EA474D6ab4255A20cA41211523dA27E8e0E56E0 BALANCE 99.88 ETH TX COUNT 17 INDEX 0

Fig 33: Wallet in use

SAMPLE CODES :

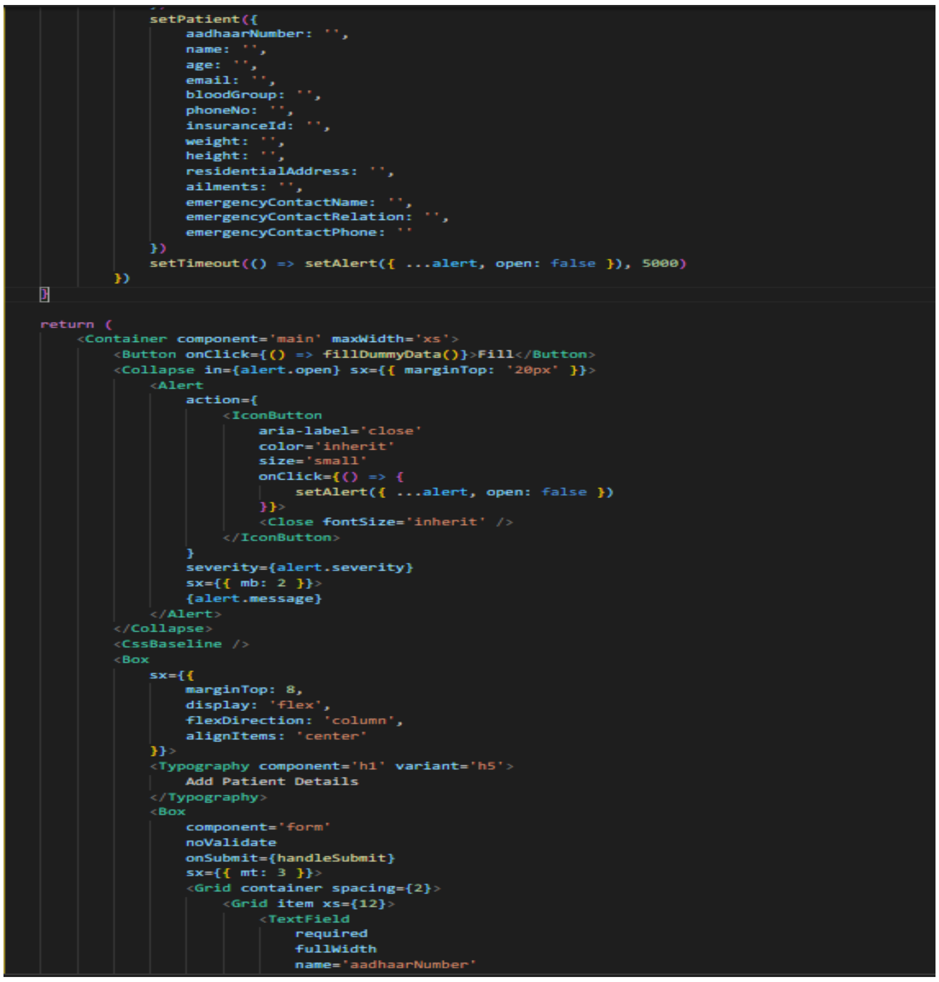
1) SOLIDITY :

```
pragma solidity ^0.8.7;

// SPDX-License-Identifier: MIT
contract Ethinsurance {

    address private owner;
    modifier ownerAccess{
        require(owner== msg.sender);
        _;
    }

    constructor(){
        owner = msg.sender;
    }

    struct patient {
        string email;
        string name;
        uint age;
        uint phoneNo;
        string bloodGroup;
        uint insuranceCompanyId;
        uint height;
        uint weight;
        string residentialAddress;
        string ailments;
        string emergencyContactName;
        string emergencyContactRelation;
        uint emergencyContactPhone;

    }

    struct inputPatient {
        uint aadharno;
        string email;
        string name;
        uint age;
        uint phoneNo;
        string bloodGroup;
        uint insuranceCompanyId;
        uint height;
        uint weight;
        string residentialAddress;
        string ailments;
        string emergencyContactName;
        string emergencyContactRelation;
        uint emergencyContactPhone;
    }
    mapping (uint => patient) patientInfo;
    mapping (uint => bool) patientRecord;
    function addPatientInfo (inputPatient memory inputPatientInfo) public
    {
        require(patientRecord[inputPatientInfo.aadharno] != true, "Record already exists");
        patientRecord[inputPatientInfo.aadharno] = true;
        patientInfo[inputPatientInfo.aadharno].email = inputPatientInfo.email;
        patientInfo[inputPatientInfo.aadharno].name = inputPatientInfo.name;
        patientInfo[inputPatientInfo.aadharno].age = inputPatientInfo.age;
        patientInfo[inputPatientInfo.aadharno].phoneNo = inputPatientInfo.phoneNo;
        patientInfo[inputPatientInfo.aadharno].bloodGroup = inputPatientInfo.bloodGroup;
        patientInfo[inputPatientInfo.aadharno].insuranceCompanyId =
inputPatientInfo.insuranceCompanyId;
        patientInfo[inputPatientInfo.aadharno].height = inputPatientInfo.height;
        patientInfo[inputPatientInfo.aadharno].weight = inputPatientInfo.weight;
        patientInfo[inputPatientInfo.aadharno].residentialAddress = inputPatientInfo.residentialAddress;
        patientInfo[inputPatientInfo.aadharno].ailments = inputPatientInfo.ailments;
        patientInfo[inputPatientInfo.aadharno].emergencyContactName=
inputPatientInfo.emergencyContactName;

patientInfo[inputPatientInfo.aadharno].emergencyContactRelation=inputPatientInfo.emergencyConta
ctRelation;
        patientInfo[inputPatientInfo.aadharno].emergencyContactPhone =
inputPatientInfo.emergencyContactPhone;
    }

    function getPatientInfo(uint aadharno) external view returns (patient memory patientInfoValue)
    {
        return patientInfo[aadharno];
    }

struct ins_comp {
        string name;
        string helplineEmail;
        uint helplineNumber;
        uint emergencyNumber;
        string location;
    }

    struct ins_comp_info {
        uint uid;
        string name;
        string helplineEmail;
        uint helplineNumber;
        uint emergencyNumber;
        string location;
    }

    mapping (uint => ins_comp) CompInfo;
    mapping (uint => bool) providerRecord;
    function addInsuranceProvider(ins_comp_info memory ins_comp_infor) public{
        require(providerRecord[ins_comp_infor.uid] != true, "Record already exists");
        providerRecord[ins_comp_infor.uid] = true;
        CompInfo[ins_comp_infor.uid].name=ins_comp_infor.name;
        CompInfo[ins_comp_infor.uid].helplineNumber=ins_comp_infor.helplineNumber;
        CompInfo[ins_comp_infor.uid].helplineEmail=ins_comp_infor.helplineEmail;
        CompInfo[ins_comp_infor.uid].emergencyNumber=ins_comp_infor.emergencyNumber;
        CompInfo[ins_comp_infor.uid].location=ins_comp_infor.location;
    }

    struct ins_policy {
        uint insuranceProviderId;
        uint patientAadhar;
        uint insuredAmount;
    }

    struct ins_policy_info {
        uint uid;
        uint insuranceProviderId;
        uint patientAadhar;
        uint insuredAmount;
    }

    mapping (uint => ins_policy) PolicyInfo;
    mapping (uint => bool) insuranceRecord;

    function addInsurancePolicy(ins_policy_info memory ins_policy_infor) public{
        require(insuranceRecord[ins_policy_infor.uid] != true, "Record already exists");
        insuranceRecord[ins_policy_infor.uid] = true;
        PolicyInfo[ins_policy_infor.uid].insuranceProviderId=ins_policy_infor.insuranceProviderId;
        PolicyInfo[ins_policy_infor.uid].patientAadhar=ins_policy_infor.patientAadhar;
        PolicyInfo[ins_policy_infor.uid].insuredAmount=ins_policy_infor.insuredAmount;
    }
    function getProviderInfo(uint uid) external view returns (ins_comp memory InsCompValue)
    {
        return CompInfo[uid];
    }
    function getPolicyInfo(uint uid) external view returns (ins_policy memory InsPolicyValue)
    {
        return PolicyInfo[uid];
    }


    struct hospital {
        string name;
        string helplineEmail;
        uint helplineNumber;
        uint emergencyNumber;
        string location;
        string speciality;
    }

    struct hospital_info {
        uint uid;
        string name;
        string helplineEmail;
        uint helplineNumber;
        uint emergencyNumber;
        string location;
        string speciality;
    }
```

```
mapping (uint => hospital) HospitalInfo;
mapping (uint => bool) hospitalRecord;

function addHospital(hospital_info memory hospital_infor) public{
    require(hospitalRecord[hospital_infor.uid] != true, "Record already exists");
    hospitalRecord[hospital_infor.uid] = true;
    HospitalInfo[hospital_infor.uid].name=hospital_infor.name;
    HospitalInfo[hospital_infor.uid].helplineNumber=hospital_infor.helplineNumber;
    HospitalInfo[hospital_infor.uid].helplineEmail=hospital_infor.helplineEmail;
    HospitalInfo[hospital_infor.uid].emergencyNumber=hospital_infor.emergencyNumber;
    HospitalInfo[hospital_infor.uid].location=hospital_infor.location;
    HospitalInfo[hospital_infor.uid].speciality=hospital_infor.speciality;
}

function getHospitaltInfo(uint uid) external view returns (hospital memory HospitalData)
{
    return HospitalInfo[uid];
}

struct treatment {
    uint patientAadhaar;
    string doctorName;
    uint hospitalId;
    string diagnosis;
    string testsConducted;
    uint billingAmount;
    string medicines;
}
struct treatment_info {
    uint uid;
    uint patientAadhaar;
    string doctorName;
    uint hospitalId;
    string diagnosis;
    string testsConducted;
    uint billingAmount;
    string medicines;
}
mapping(uint=>treatment) TreatmentInfo;
mapping (uint => bool) treatmentRecord;

function treatPatient(treatment_info memory treatment_infor) public{
     require(treatmentRecord[treatment_infor.uid] != true, "Record already exists");
    treatmentRecord[treatment_infor.uid] = true;
    TreatmentInfo[treatment_infor.uid].patientAadhaar=treatment_infor.patientAadhaar;
    TreatmentInfo[treatment_infor.uid].doctorName=treatment_infor.doctorName;
    TreatmentInfo[treatment_infor.uid].hospitalId=treatment_infor.hospitalId;
    TreatmentInfo[treatment_infor.uid].diagnosis=treatment_infor.diagnosis;
    TreatmentInfo[treatment_infor.uid].testsConducted=treatment_infor.testsConducted;
    TreatmentInfo[treatment_infor.uid].billingAmount=treatment_infor.billingAmount;
    TreatmentInfo[treatment_infor.uid].medicines=treatment_infor.medicines;
}

function getTreatmentInfo(uint uid) external view returns (treatment memory TreatmentData)
```

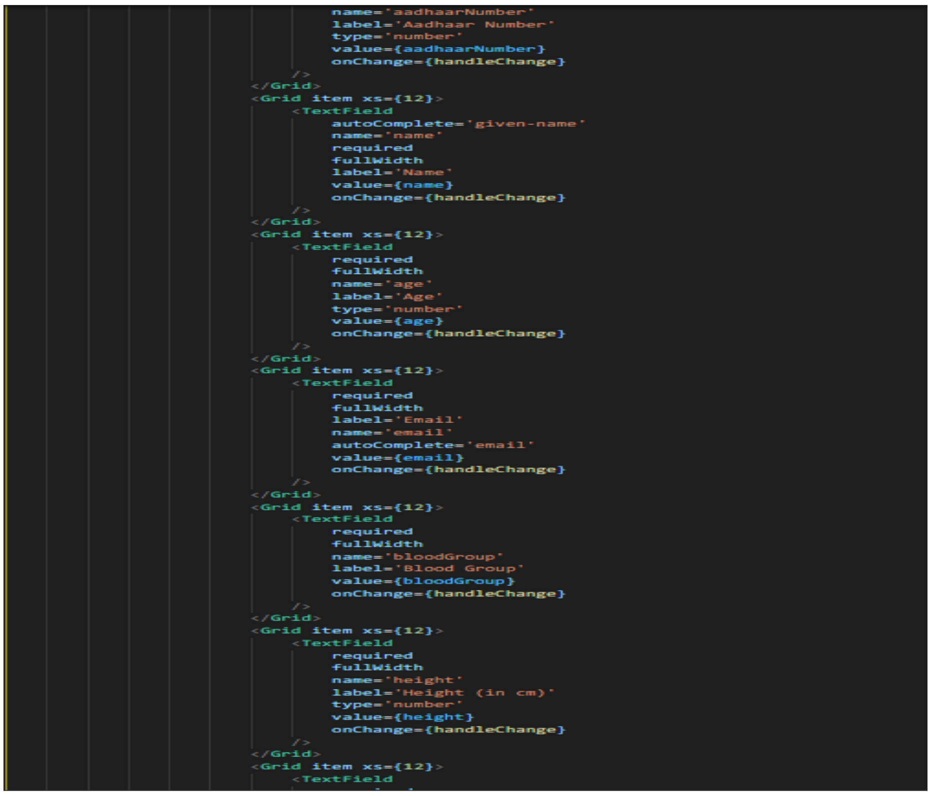
```
    {
        return TreatmentInfo[uid];
    }

}
```

2) GET PATIENT DETAILS CODE :



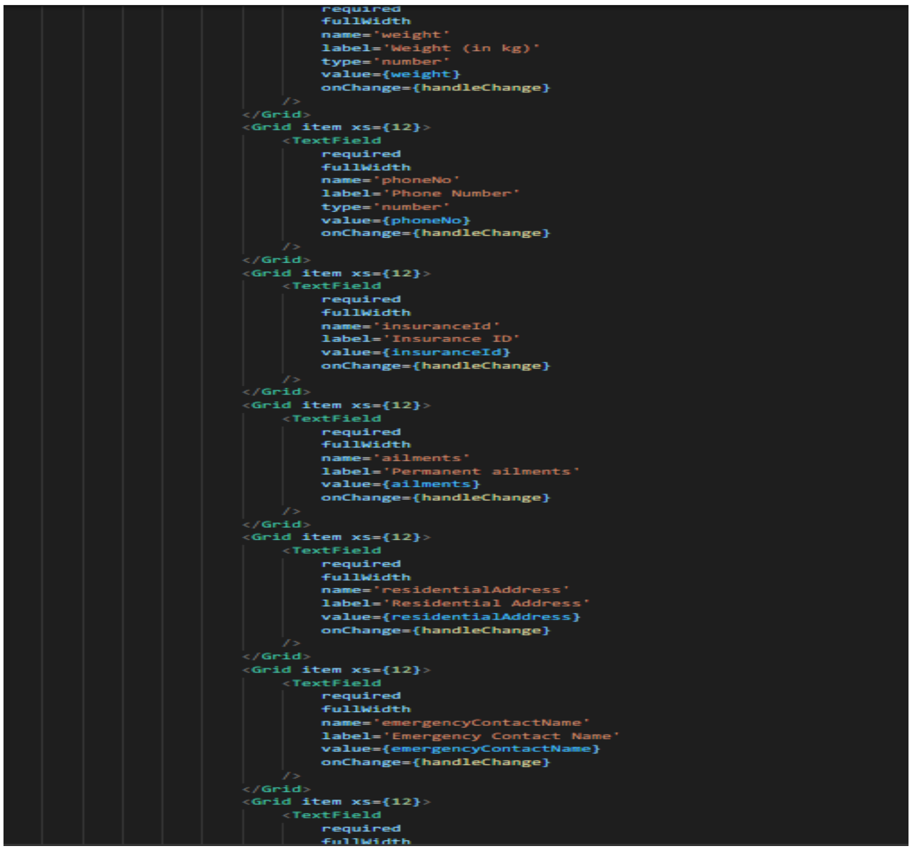
```
                setPatient({
                    aadhaarNumber: '',
                    name: '',
                    age: '',
                    email: '',
                    bloodGroup: '',
                    phoneNo: '',
                    insuranceId: '',
                    weight: '',
                    height: '',
                    residentialAddress: '',
                    ailments: '',
                    emergencyContactName: '',
                    emergencyContactRelation: '',
                    emergencyContactPhone: ''
                })
                setTimeout(() => setAlert({ ...alert, open: false }), 5000)
            })
    }

    return (
        <Container component='main' maxWidth='xs'>
            <Button onClick={() => fillDummyData()}>Fill</Button>
            <Collapse in={alert.open} sx={{ marginTop: '20px' }}>
                <Alert
                    action={
                        <IconButton
                            aria-label='close'
                            color='inherit'
                            size='small'
                            onClick={() => {
                                setAlert({ ...alert, open: false })
                            }}>
                            <Close fontSize='inherit' />
                        </IconButton>
                    }
                    severity={alert.severity}
                    sx={{ mb: 2 }}>
                    {alert.message}
                </Alert>
            </Collapse>
            <CssBaseline />
            <Box
                sx={{
                    marginTop: 8,
                    display: 'flex',
                    flexDirection: 'column',
                    alignItems: 'center'
                }}>
                <Typography component='h1' variant='h5'>
                    Add Patient Details
                </Typography>
                <Box
                    component='form'
                    noValidate
                    onSubmit={handleSubmit}
                    sx={{ mt: 3 }}>
                    <Grid container spacing={2}>
                        <Grid item xs={12}>
                            <TextField
                                required
                                fullWidth
                                name='aadhaarNumber'
```

Fig 34 : Patient SignUp Code



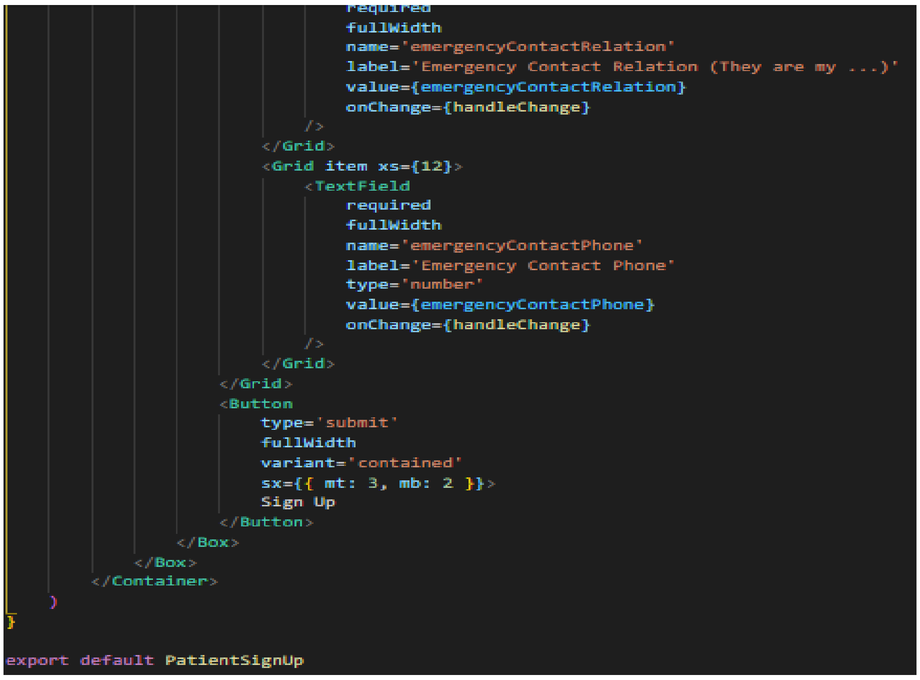
```
                    name='aadhaarNumber'
                    label='Aadhaar Number'
                    type='number'
                    value={aadhaarNumber}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
                    autoComplete='given-name'
                    name='name'
                    required
                    fullWidth
                    label='Name'
                    value={name}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
                    required
                    fullWidth
                    name='age'
                    label='Age'
                    type='number'
                    value={age}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
                    required
                    fullWidth
                    label='Email'
                    name='email'
                    autoComplete='email'
                    value={email}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
                    required
                    fullWidth
                    name='bloodGroup'
                    label='Blood Group'
                    value={bloodGroup}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
                    required
                    fullWidth
                    name='height'
                    label='Height (in cm)'
                    type='number'
                    value={height}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
```

Fig 35 : Patient SignUp Code



```
                    required
                    fullWidth
                    name='weight'
                    label='Weight (in kg)'
                    type='number'
                    value={weight}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
                    required
                    fullWidth
                    name='phoneNo'
                    label='Phone Number'
                    type='number'
                    value={phoneNo}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
                    required
                    fullWidth
                    name='insuranceId'
                    label='Insurance ID'
                    value={insuranceId}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
                    required
                    fullWidth
                    name='ailments'
                    label='Permanent ailments'
                    value={ailments}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
                    required
                    fullWidth
                    name='residentialAddress'
                    label='Residential Address'
                    value={residentialAddress}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
                    required
                    fullWidth
                    name='emergencyContactName'
                    label='Emergency Contact Name'
                    value={emergencyContactName}
                    onChange={handleChange}
                />
            </Grid>
            <Grid item xs={12}>
                <TextField
                    required
                    fullWidth
```

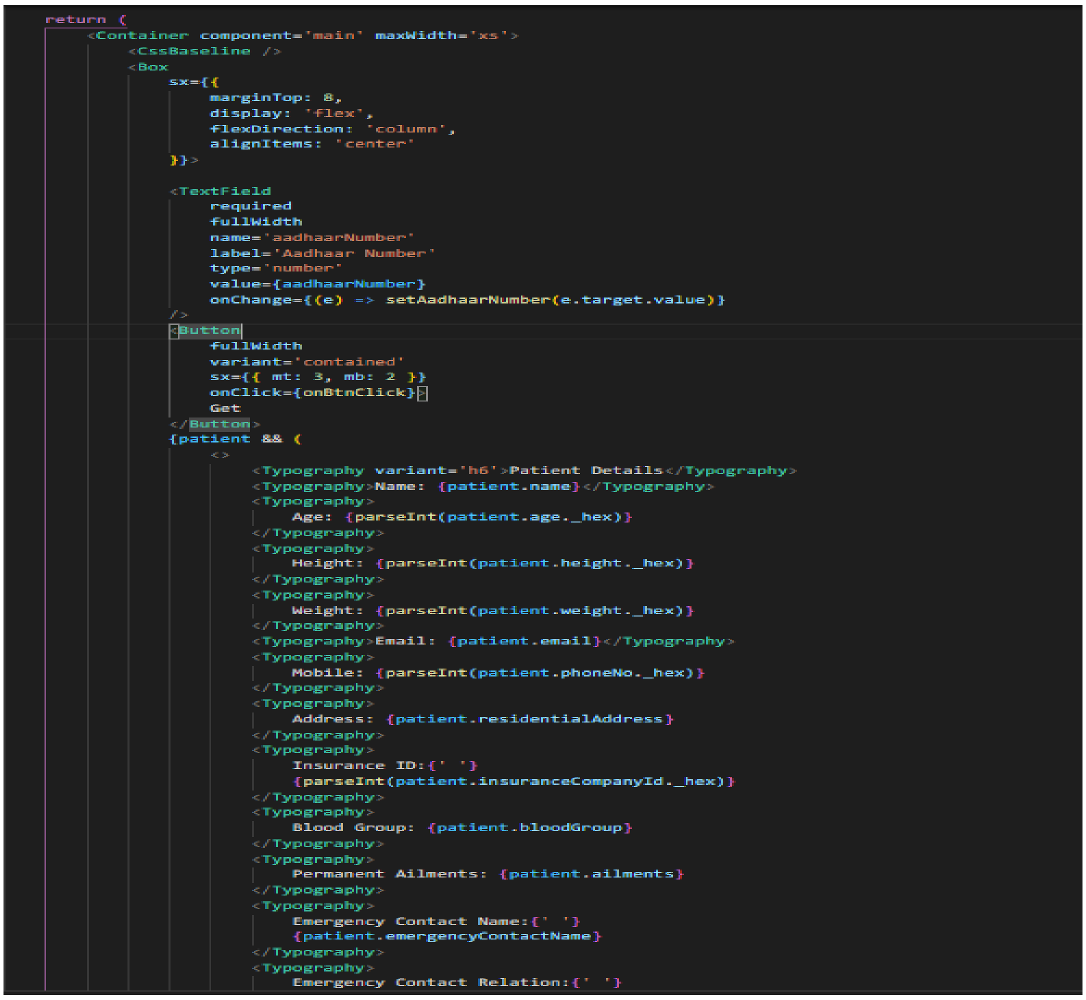
Fig 36 : Patient SignUp Code



```
                                required
                                fullWidth
                                name='emergencyContactRelation'
                                label='Emergency Contact Relation (They are my ...)'
                                value={emergencyContactRelation}
                                onChange={handleChange}
                            />
                        </Grid>
                        <Grid item xs={12}>
                            <TextField
                                required
                                fullWidth
                                name='emergencyContactPhone'
                                label='Emergency Contact Phone'
                                type='number'
                                value={emergencyContactPhone}
                                onChange={handleChange}
                            />
                        </Grid>
                    </Grid>
                    <Button
                        type='submit'
                        fullWidth
                        variant='contained'
                        sx={{ mt: 3, mb: 2 }}>
                        Sign Up
                    </Button>
                </Box>
            </Box>
        </Container>
    )
}

export default PatientSignUp
```

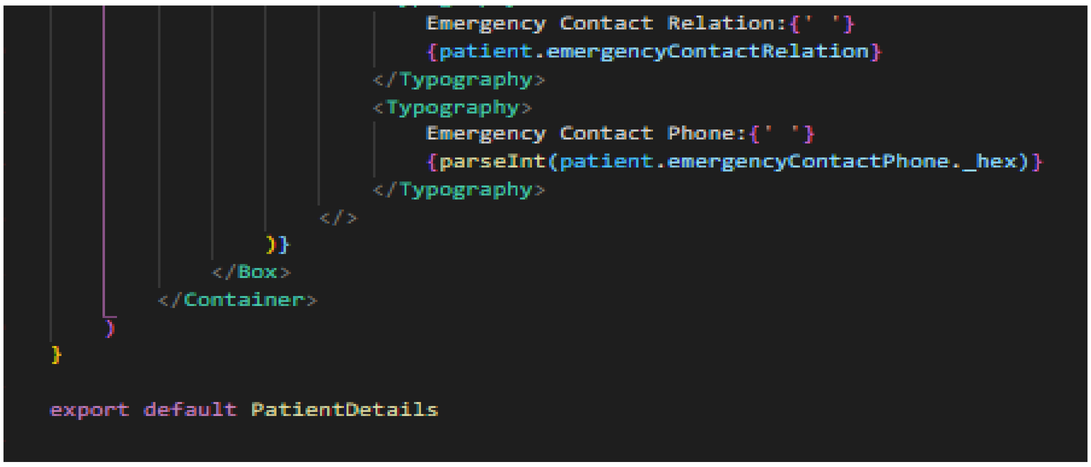
Fig 37 : Patient SignUp Code

3) Patient Treatment Details Code:



```
    return (
        <Container component='main' maxWidth='xs'>
            <CssBaseline />
            <Box
                sx={{
                    marginTop: 8,
                    display: 'flex',
                    flexDirection: 'column',
                    alignItems: 'center'
                }}>

                <TextField
                    required
                    fullWidth
                    name='aadhaarNumber'
                    label='Aadhaar Number'
                    type='number'
                    value={aadhaarNumber}
                    onChange={(e) => setAadhaarNumber(e.target.value)}
                />
                <Button
                    fullWidth
                    variant='contained'
                    sx={{ mt: 3, mb: 2 }}
                    onClick={onBtnClick}>
                    Get
                </Button>
                {patient && (
                    <>
                        <Typography variant='h6'>Patient Details</Typography>
                        <Typography>Name: {patient.name}</Typography>
                        <Typography>
                            Age: {parseInt(patient.age._hex)}
                        </Typography>
                        <Typography>
                            Height: {parseInt(patient.height._hex)}
                        </Typography>
                        <Typography>
                            Weight: {parseInt(patient.weight._hex)}
                        </Typography>
                        <Typography>Email: {patient.email}</Typography>
                        <Typography>
                            Mobile: {parseInt(patient.phoneNo._hex)}
                        </Typography>
                        <Typography>
                            Address: {patient.residentialAddress}
                        </Typography>
                        <Typography>
                            Insurance ID:{' '}
                            {parseInt(patient.insuranceCompanyId._hex)}
                        </Typography>
                        <Typography>
                            Blood Group: {patient.bloodGroup}
                        </Typography>
                        <Typography>
                            Permanent Ailments: {patient.ailments}
                        </Typography>
                        <Typography>
                            Emergency Contact Name:{' '}
                            {patient.emergencyContactName}
                        </Typography>
                        <Typography>
                            Emergency Contact Relation:{' '}
```

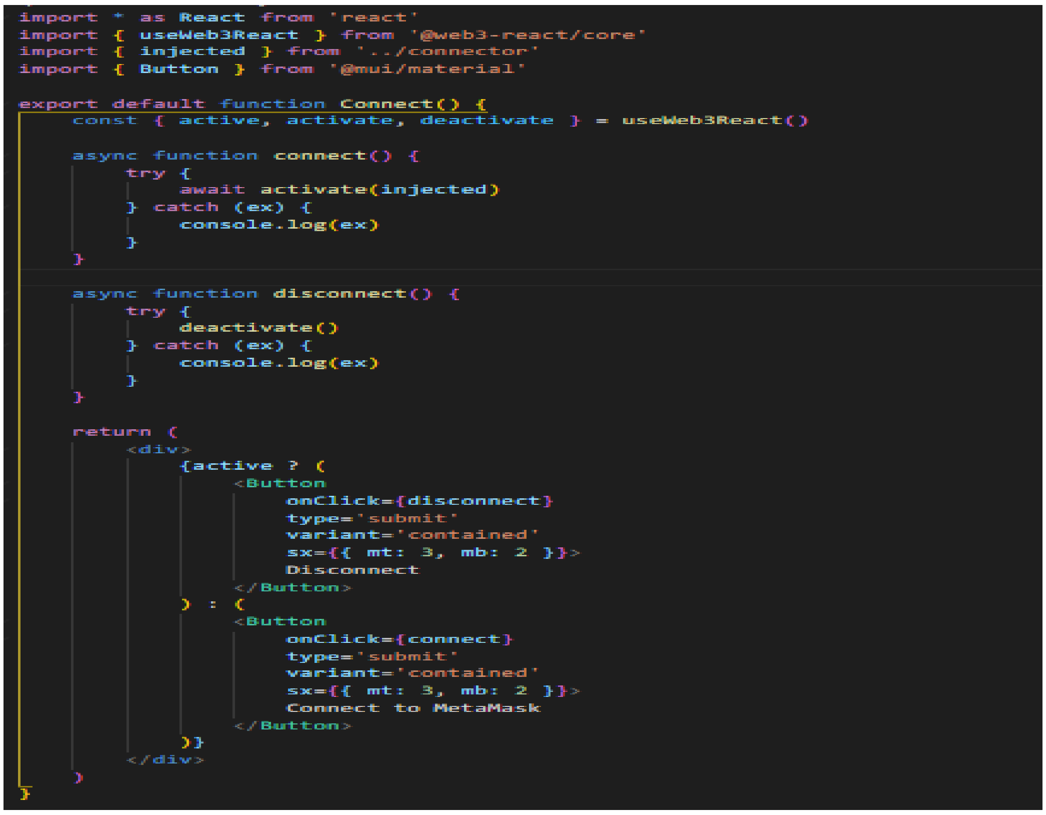
Fig 38 : Get Patient Details Code



```
                                    Emergency Contact Relation:{' '}
                                    {patient.emergencyContactRelation}
                                </Typography>
                                <Typography>
                                    Emergency Contact Phone:{' '}
                                    {parseInt(patient.emergencyContactPhone._hex)}
                                </Typography>
                            </>
                        )}
                    </Box>
                </Container>
            )
        }

        export default PatientDetails
```

Fig 39 : Get Patient Details Code

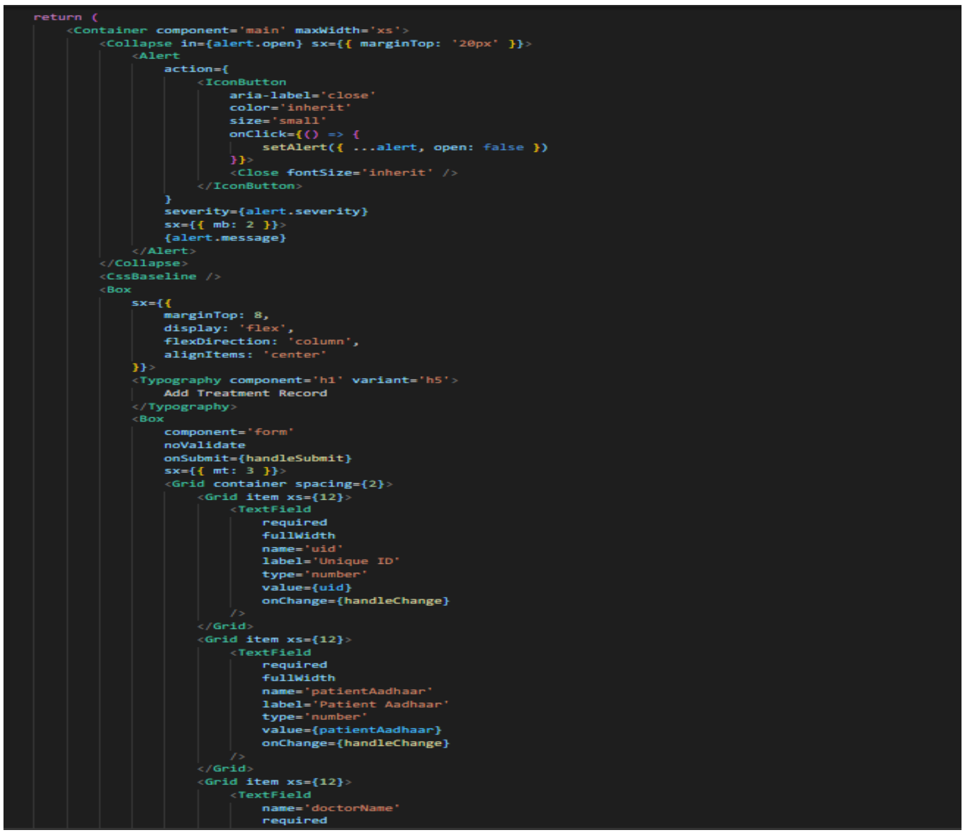
4) Connect to Metamask from dAPP Code:



```
import * as React from 'react'
import { useWeb3React } from '@web3-react/core'
import { injected } from '../connector'
import { Button } from '@mui/material'

export default function Connect() {
    const { active, activate, deactivate } = useWeb3React()

    async function connect() {
        try {
            await activate(injected)
        } catch (ex) {
            console.log(ex)
        }
    }

    async function disconnect() {
        try {
            deactivate()
        } catch (ex) {
            console.log(ex)
        }
    }

    return (
        <div>
            {active ? (
                <Button
                    onClick={disconnect}
                    type='submit'
                    variant='contained'
                    sx={{ mt: 3, mb: 2 }}>
                    Disconnect
                </Button>
            ) : (
                <Button
                    onClick={connect}
                    type='submit'
                    variant='contained'
                    sx={{ mt: 3, mb: 2 }}>
                    Connect to MetaMask
                </Button>
            )}
        </div>
    )
}
```

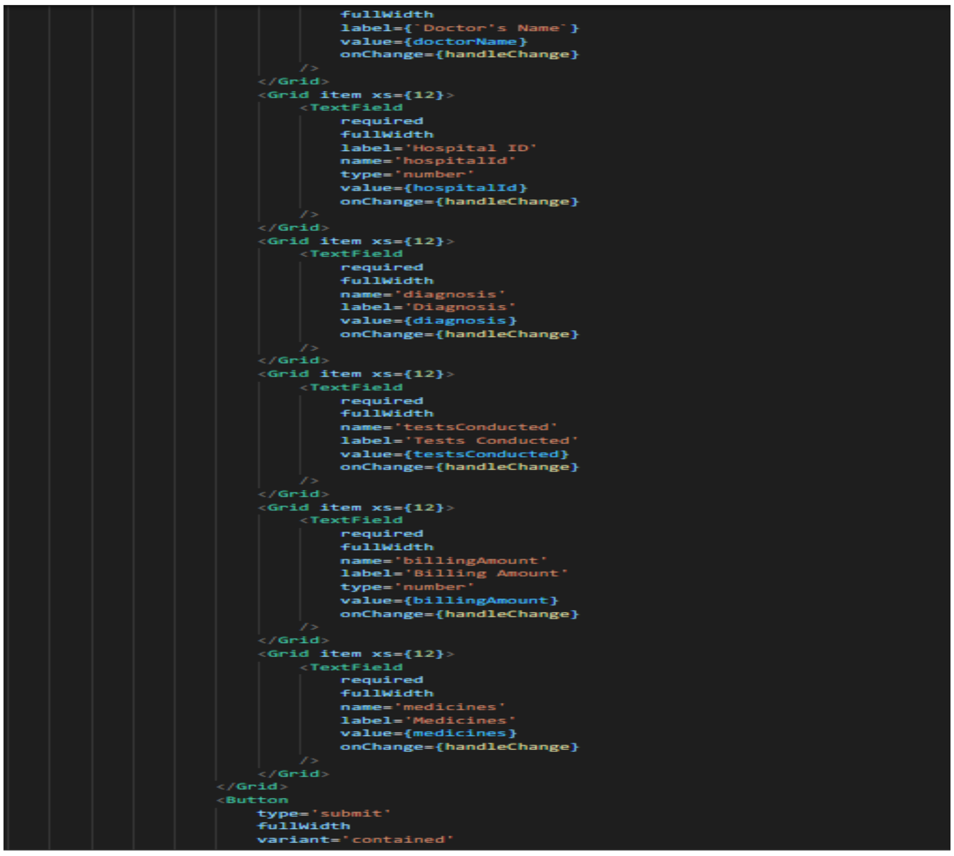
Fig 40 : Connect Metamask

5) Patient Treatment Details:



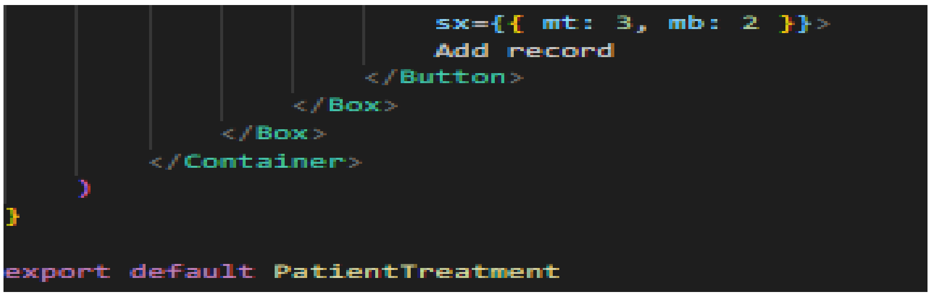
```
return (
    <Container component='main' maxWidth='xs'>
        <Collapse in={alert.open} sx={{ marginTop: '20px' }}>
            <Alert
                action={
                    <IconButton
                        aria-label='close'
                        color='inherit'
                        size='small'
                        onClick={() => {
                            setAlert({ ...alert, open: false })
                        }}>
                        <Close fontSize='inherit' />
                    </IconButton>
                }
                severity={alert.severity}
                sx={{ mb: 2 }}>
                {alert.message}
            </Alert>
        </Collapse>
        <CssBaseline />
        <Box
            sx={{
                marginTop: 8,
                display: 'flex',
                flexDirection: 'column',
                alignItems: 'center'
            }}>
            <Typography component='h1' variant='h5'>
                Add Treatment Record
            </Typography>
            <Box
                component='form'
                noValidate
                onSubmit={handleSubmit}
                sx={{ mt: 3 }}>
                <Grid container spacing={2}>
                    <Grid item xs={12}>
                        <TextField
                            required
                            fullWidth
                            name='uid'
                            label='Unique ID'
                            type='number'
                            value={uid}
                            onChange={handleChange}
                        />
                    </Grid>
                    <Grid item xs={12}>
                        <TextField
                            required
                            fullWidth
                            name='patientAadhaar'
                            label='Patient Aadhaar'
                            type='number'
                            value={patientAadhaar}
                            onChange={handleChange}
                        />
                    </Grid>
                    <Grid item xs={12}>
                        <TextField
                            name='doctorName'
                            required
```

Fig 41: Patient Treatment Details



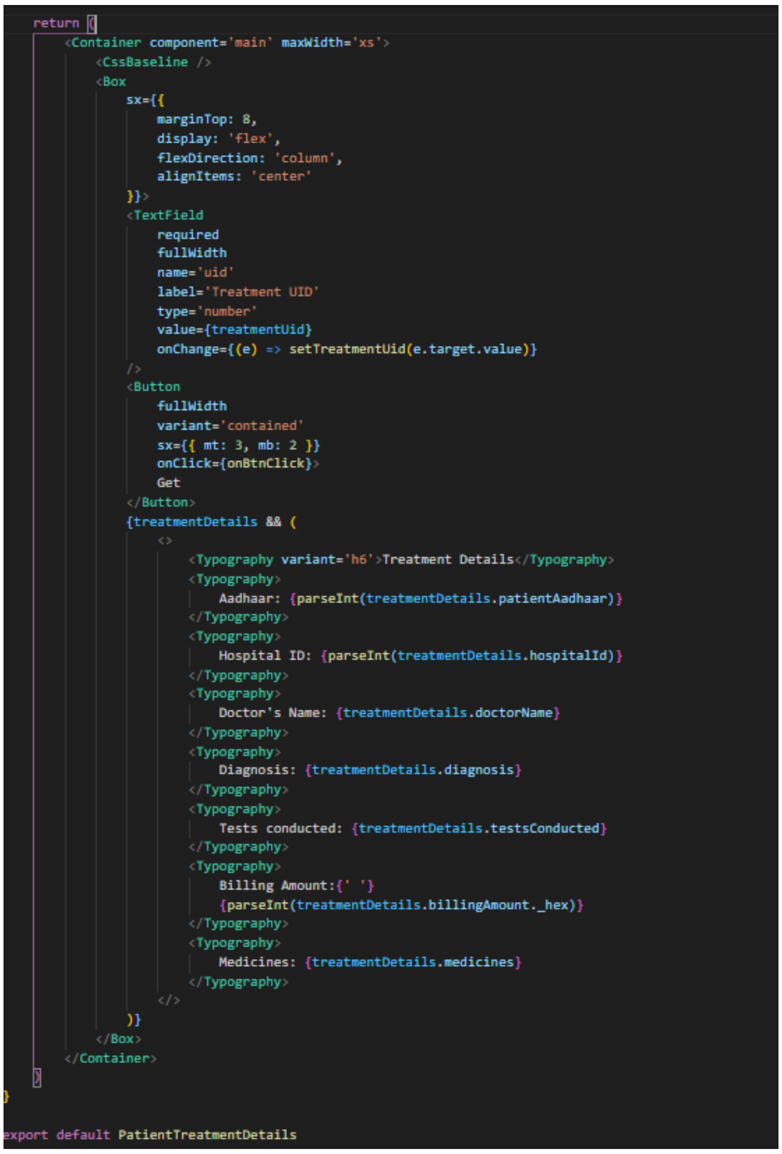
```
                            fullWidth
                            label={`Doctor's Name`}
                            value={doctorName}
                            onChange={handleChange}
                        />
                    </Grid>
                    <Grid item xs={12}>
                        <TextField
                            required
                            fullWidth
                            label='Hospital ID'
                            name='hospitalId'
                            type='number'
                            value={hospitalId}
                            onChange={handleChange}
                        />
                    </Grid>
                    <Grid item xs={12}>
                        <TextField
                            required
                            fullWidth
                            name='diagnosis'
                            label='Diagnosis'
                            value={diagnosis}
                            onChange={handleChange}
                        />
                    </Grid>
                    <Grid item xs={12}>
                        <TextField
                            required
                            fullWidth
                            name='testsConducted'
                            label='Tests Conducted'
                            value={testsConducted}
                            onChange={handleChange}
                        />
                    </Grid>
                    <Grid item xs={12}>
                        <TextField
                            required
                            fullWidth
                            name='billingAmount'
                            label='Billing Amount'
                            type='number'
                            value={billingAmount}
                            onChange={handleChange}
                        />
                    </Grid>
                    <Grid item xs={12}>
                        <TextField
                            required
                            fullWidth
                            name='medicines'
                            label='Medicines'
                            value={medicines}
                            onChange={handleChange}
                        />
                    </Grid>
                </Grid>
                <Button
                    type='submit'
                    fullWidth
                    variant='contained'
```

Fig 42: Patient Treatment Details



```
                        sx={{ mt: 3, mb: 2 }}>
                        Add record
                    </Button>
                </Box>
            </Box>
        </Container>
    )
}

export default PatientTreatment
```

Fig 43: Patient Treatment Details



```
    return (
        <Container component='main' maxWidth='xs'>
            <CssBaseline />
            <Box
                sx={{
                    marginTop: 8,
                    display: 'flex',
                    flexDirection: 'column',
                    alignItems: 'center'
                }}>
                <TextField
                    required
                    fullWidth
                    name='uid'
                    label='Treatment UID'
                    type='number'
                    value={treatmentUid}
                    onChange={(e) => setTreatmentUid(e.target.value)}
                />
                <Button
                    fullWidth
                    variant='contained'
                    sx={{ mt: 3, mb: 2 }}
                    onClick={onBtnClick}>
                    Get
                </Button>
                {treatmentDetails && (
                    <>
                        <Typography variant='h6'>Treatment Details</Typography>
                        <Typography>
                            Aadhaar: {parseInt(treatmentDetails.patientAadhaar)}
                        </Typography>
                        <Typography>
                            Hospital ID: {parseInt(treatmentDetails.hospitalId)}
                        </Typography>
                        <Typography>
                            Doctor's Name: {treatmentDetails.doctorName}
                        </Typography>
                        <Typography>
                            Diagnosis: {treatmentDetails.diagnosis}
                        </Typography>
                        <Typography>
                            Tests conducted: {treatmentDetails.testsConducted}
                        </Typography>
                        <Typography>
                            Billing Amount:{' '}
                            {parseInt(treatmentDetails.billingAmount._hex)}
                        </Typography>
                        <Typography>
                            Medicines: {treatmentDetails.medicines}
                        </Typography>
                    </>
                )}
            </Box>
        </Container>
    )
}

export default PatientTreatmentDetails
```

Fig 44: Get Patient Treatment Details

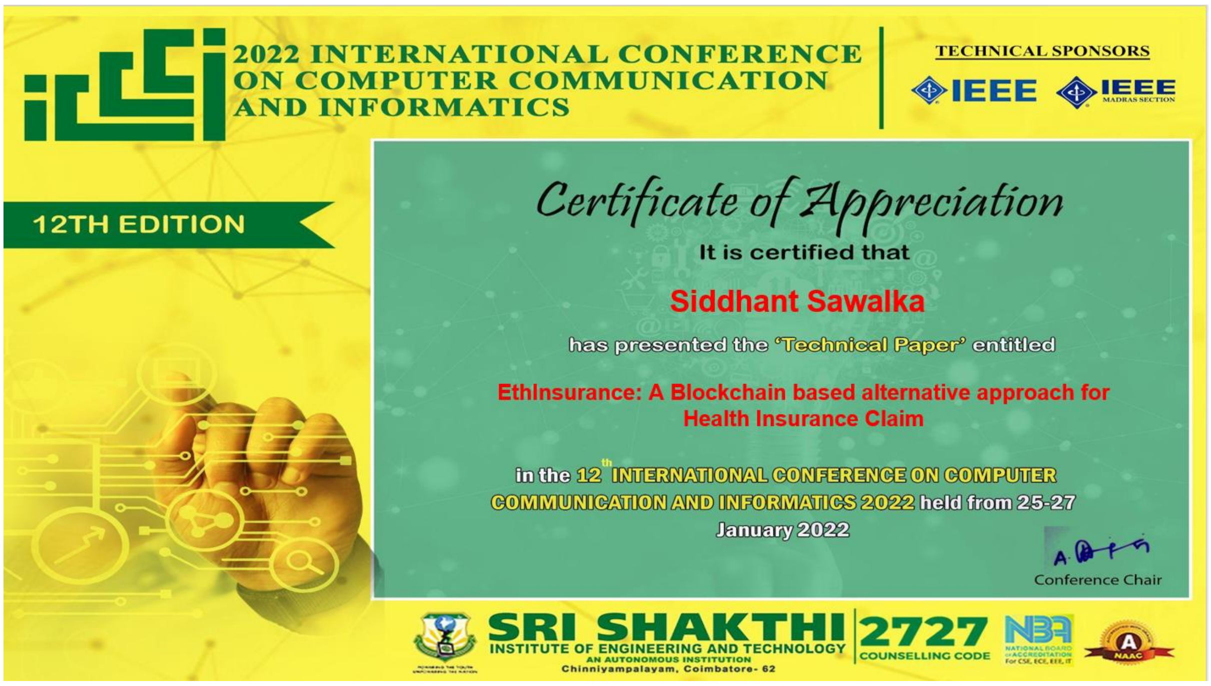
PAPER PUBLICATION STATUS

Conference Name: 2022 International Conference on Computer Communication and Informatics

Name of Paper: EthInsurance: A Blockchain based alternative approach for Health Insurance Claim

Status : Published and Indexed in scopus

Certificates:



iCCCI 2022 INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATION AND INFORMATICS

TECHNICAL SPONSORS IEEE IEEE MADRAS SECTION

12TH EDITION

Certificate of Appreciation

It is certified that

Siddhant Sawalka

has presented the 'Technical Paper' entitled

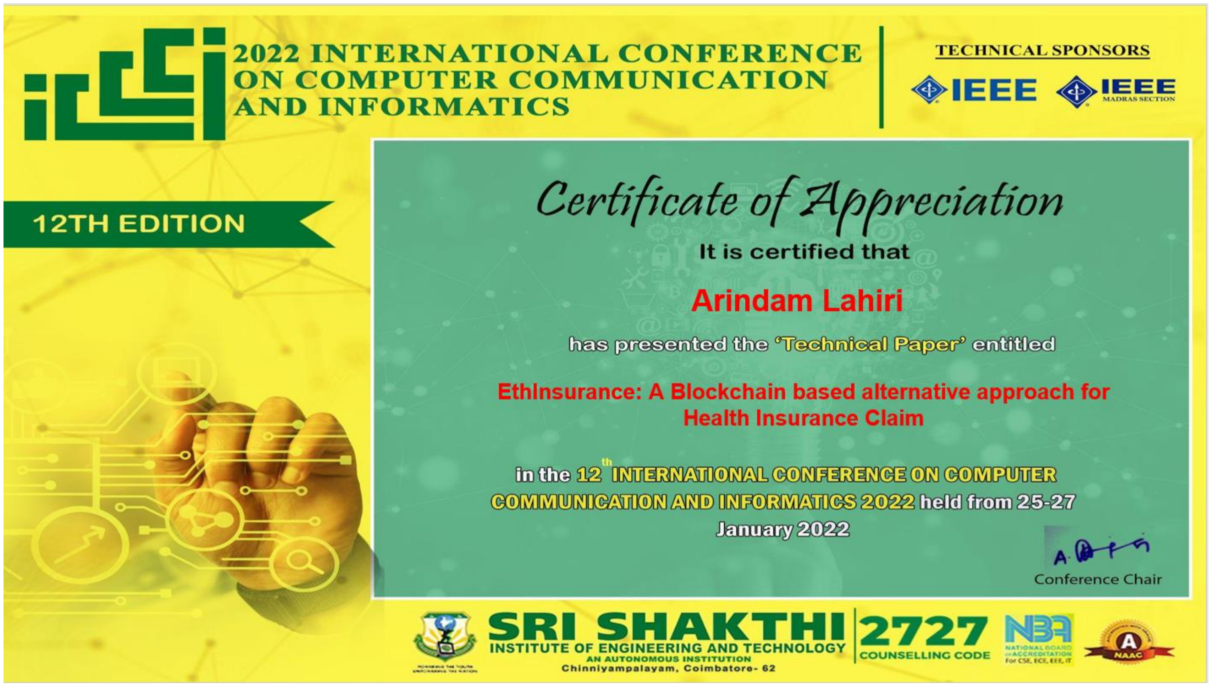
EthInsurance: A Blockchain based alternative approach for Health Insurance Claim

in the 12th INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATION AND INFORMATICS 2022 held from 25-27 January 2022

[illegible]

Conference Chair

SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY AN AUTONOMOUS INSTITUTION Chinniyampalayam, Coimbatore- 62 2727 COUNSELLING CODE NBA NAAC A



iCCCI 2022 INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATION AND INFORMATICS

TECHNICAL SPONSORS IEEE IEEE MADRAS SECTION

12TH EDITION

Certificate of Appreciation

It is certified that

Arindam Lahiri

has presented the 'Technical Paper' entitled

EthInsurance: A Blockchain based alternative approach for Health Insurance Claim

in the 12th INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATION AND INFORMATICS 2022 held from 25-27 January 2022

[illegible]

Conference Chair

SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY AN AUTONOMOUS INSTITUTION Chinniyampalayam, Coimbatore- 62 2727 COUNSELLING CODE NBA NAAC A