Medical Records using Blockchain

A MINI-PROJECT REPORT

Submitted by

Tushar [RA2011050010060]
Anirudh[RA2011050010064]

Studying
B. Tech

Under the Guidance of

Dr. SV. Shri Bharathi
Assistant Professor, Department of DSBS

DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR- 603 203

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this B.Tech mini-project report titled Crypto Trading Platform is the bonafide work of **Tushar and Anirudh** 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 part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Dr. SV. SHRI BHARATHI	Dr. M. Lakshmi
SUPERVISOR	**PROFESSOR & HOD**
Assistant Professor	Department of DSBS
Department of DSBS	

Signature of Internal Examiner **Signature of External Examiner**

ABSTRACT

Years of heavy regulation and bureaucratic inefficiency have slowed innovation for electronic medical records (EMRs). We now face a critical need for such innovation, as personalization and data science prompt patients to engage in the details of their healthcare and restore agency over their medical data. In this paper, we propose MedRec: a novel, decentralized record management system to handle EMRs, using blockchain technology. Our system gives patients a comprehensive, immutable log and easy access to their medical information across providers and treatment sites. Leveraging unique blockchain properties, MedRec manages authentication, confidentiality, accountability and data sharing– crucial considerations when handling sensitive information. A modular design integrates with providers' existing, local data storage solutions, facilitating interoperability and making our system convenient and adaptable. We incentivize medical stakeholders (researchers, public health authorities, etc.) to participate in the network as blockchain "miners". This provides them with access to aggregate, anonymized data as mining rewards, in return for sustaining and securing the network via Proof of Work. MedRec thus enables the emergence of data economics, supplying big data to empower researchers while engaging patients and providers in the choice to release metadata. The purpose of this short paper is to expose, prior to field tests, a working prototype through which we analyze and discuss our approach.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Electronic Health Record (EHR) systems have been increasingly used as an effective method to share patients' records among different hospitals. However, it is still a challenge to access scattered patient data through multiple EHRs because existing EHRs are regionally limited or belong to affiliated hospitals.
the main barrier to access patient records lies in the difficulty to find provider's addresses. So far, there have been several projects to overcome these problems; however, the solutions they have produced are difficult and involve redesigning or upgrading of existing EHR systems, which would require substantial expenses.

Chapter 2
LITERATURE REVIEW

2.1 EXISTING SYSTEM:

Medical records also prove critical for research. The ONC's report emphasizes that biomedical and public health researchers "require the ability to analyze information from many sources in order to identify public health risks, develop new treatments and cures, and enable precision medicine" [4]. Though some data trickles through to researchers from clinical studies, surveys and teaching hospitals, we note a growing interest among patients, care providers and regulatory bodies to responsibly share more data, and thus enable better care for others .

In this work, we explore a blockchain structure applied to EMRs. We build on this distributed ledger protocol originally associated with Bitcoin . The blockchain uses public key cryptography to create an append-only, immutable, timestamped chain of content. Copies of the blockchain are distributed on each participating "node" in the network. The Proof of Work algorithm used to secure the content from tampering depends on a "trustless" model, where individual nodes must compete to solve computationallyintensive "puzzles" (hashing exercises) before the next block

Chapter 3
SYSTEM ANALYSIS

3.1 PROBLEMSTATEMENT:

records are stored and maintained under the organization. So that, the patient can't able to access these records for further references. When the particular server(database) gets crashed then all the records will be spoiled. To overcome these drawbacks the proposed system is 0developed.
The patient should have right to access his EHRs for managing and sharing them independently Institutions, etc.

3.2 PROPOSED SOLUTION:

The patient should have right to access his EHRs for managing and sharing them independently. The patient can be access his medical report directly and can use the digitalized report with anyone.

By storing the data in the blockchain the user's data is encrypted and stored as blocks in the etherscan. The user stores data by two way authentication process such as getting secret key generated by the Metamask. Electronic Health Record Systems are proprietary that is centralized by design.

.

3.3 SOFTWARE and HARDWARES

1. Software Requirements
Operating System: Windows 11
Tools: JS, mongoDB, Sql

2. Hardware requirements:
Processor: intel Pentium 4
Hard disk: 40GB
RAM: 1TB

Chapter 4

SYSTEM DESIGN AND MPLEMENTATION

FRONT END

HTML:

The Hypertext Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images, and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items.

HTML elements are delineated by tags, written using angle brackets. Tags such as </img>and <input/> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

JAVASCRIPT:

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

Common examples of JavaScript that you might use every day include the search box on Amazon, a news recap video embedded on The New York Times, or refreshing your Twitter feed. Incorporating JavaScript improves the user experience of the web page by converting it from
a static page into an interactive one. To recap, JavaScript adds behaviour to web pages.

BACK END

Php

PHP is a general-purpose scripting language geared toward web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1993 and released in 1995. The PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page,but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on a variety of operating systems and platforms.

SOLIDITY

Solidity is an object-oriented programming language created specifically by the Ethereum Network team for constructing and designing smart contracts on Blockchain platforms.

- It's used to create smart contracts that implement business logic and generate a chain of transaction records in the blockchain system.
- It acts as a tool for creating machine-level code and compiling it on the Ethereum Virtual Machine (EVM).
- It has a lot of similarities with C and C++ and is pretty simple to learn and understand. For example, a “main” in C is equivalent to a “contract” in Solidity.

Like other programming languages, Solidity programming also has variables, functions, classes, arithmetic operations, string manipulation, and many other concepts.

ALGORITHM

Step 1:

Users upload medical records to IPFS.

Upload(Medical Record)→IPFSUpload(Medical Record)→IPFS

Step 2:

IPFS translates medical records into a hash address according to its operational mechanism.

IPFS(Medical Record)⟶translatehashIPFS(Medical Record)⟶translatehash

Step 3:

Send the hash address to the blockchain.

Send(hash)→blockchainSend(hash)→blockchain

Step 4:

Save medical information to the SDB and ledger by running the smart contract RSC.

Run(RSC)→{SDB, ledger}

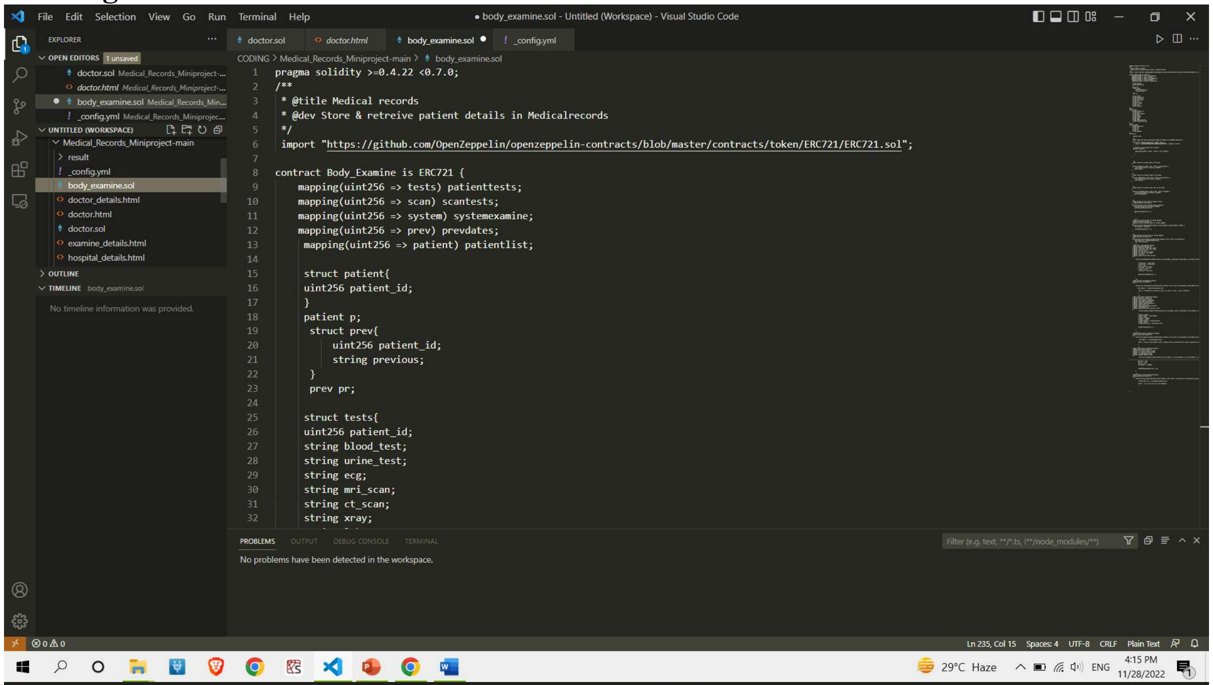
```
Algorithm 1 Smart Contract for Patient Records
 Assign Roles:
 function Define Roles (New Role, New Account )
         add new role and account in
         roles mapping
 end function
 Add Data:
 function Add Patient Record ( contains variables to add data )
      if ( msg.sender = = doctor ) then
                   add data to particular patient‟s record
      else Abort session
 end if end function
 Retrieve Data:
 function View Patient Record ( patient id )
   if ( msg.sender = = doctor || patient) then
      if ( patient id) = = true then
                retrieve data from specified patient ( id )
                return (patient record)
                to the account that requested the retrieve
 operation
else Abort session
 end if
 end if
 end function
 Update Data:
 function Update Patient Record ( contains variables to update data )
 if ( msg.sender = = doctor ) then
if ( id = = patient id && name = = patient name )

then
```

```
                  update data to particular patient’s record
                  return success
        else return fail
        end if
      else Abort session
    end if
end function
```

Delete Data:

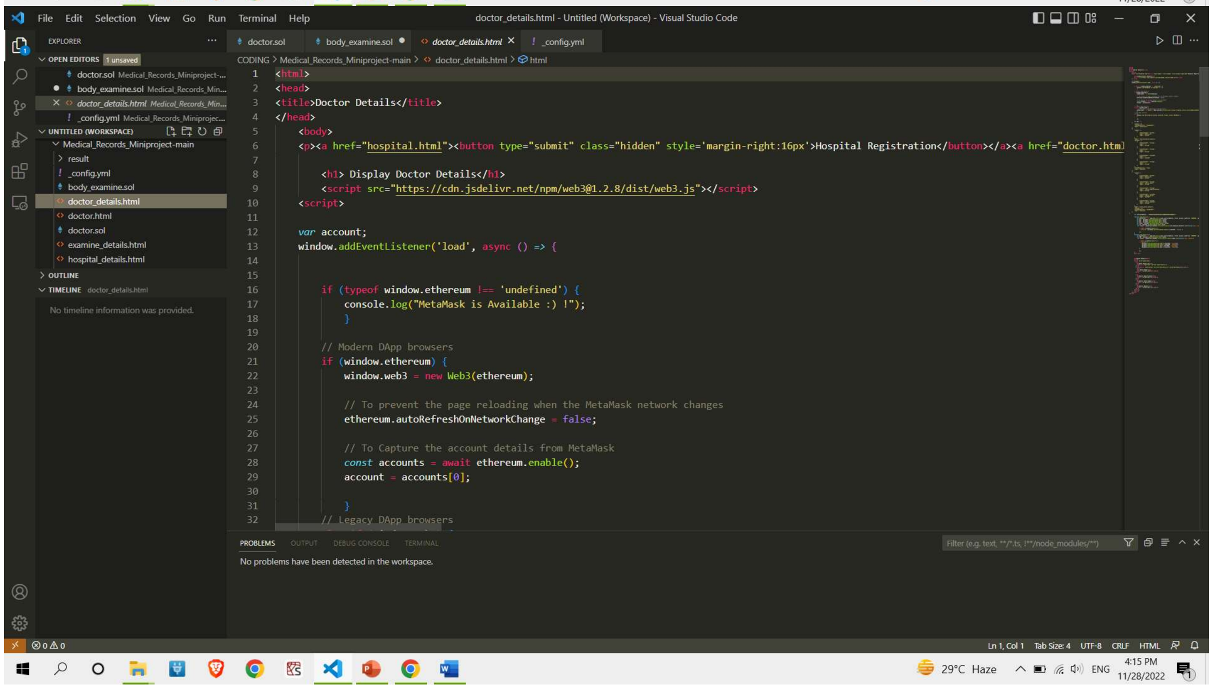
```
function Delete Patient Record ( patient id )
   if(msg.sender = = doctor ) then
      if ( id = = patient id ) then
               delete particular patient‟s record
               return success
      else return fail
      end if
   else Abort session
   end if
end function
```

Coding



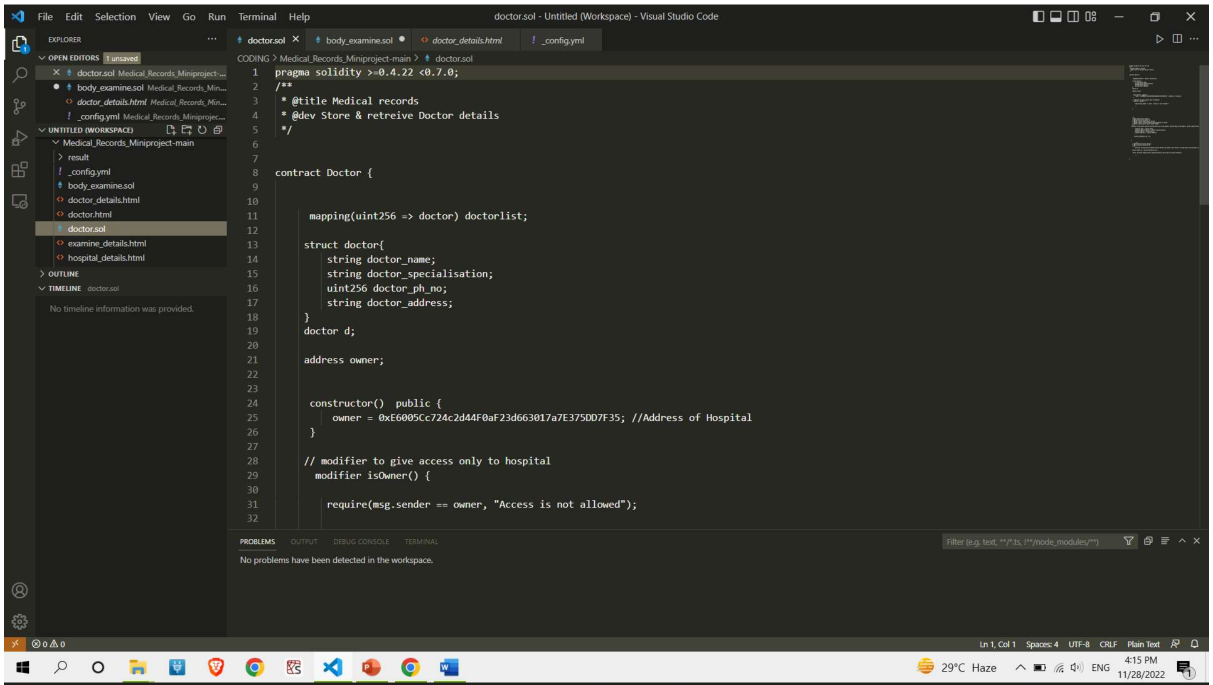
```
pragma solidity >=0.4.22 <0.7.0;
/**
 * @title Medical records
 * @dev Store & retreive patient details in Medicalrecords
 */
 import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC721/ERC721.sol";

contract Body_Examine is ERC721 {
    mapping(uint256 => tests) patienttests;
    mapping(uint256 => scan) scantests;
    mapping(uint256 => system) systemexamine;
    mapping(uint256 => prev) prevdates;
     mapping(uint256 => patient) patientlist;

     struct patient{
     uint256 patient_id;
     }
     patient p;
      struct prev{
          uint256 patient_id;
          string previous;
      }
      prev pr;

     struct tests{
     uint256 patient_id;
     string blood_test;
     string urine_test;
     string ecg;
     string mri_scan;
     string ct_scan;
     string xray;
```

No problems have been detected in the workspace.



```
<html>
<head>
<title>Doctor Details</title>
</head>
    <body>
    <p><a href="hospital.html"><button type="submit" class="hidden" style='margin-right:16px'>Hospital Registration</button></a><a href="doctor.html

        <h1> Display Doctor Details</h1>
        <script src="https://cdn.jsdelivr.net/npm/web3@1.2.8/dist/web3.js"></script>
    <script>

    var account;
    window.addEventListener('load', async () => {


        if (typeof window.ethereum !== 'undefined') {
            console.log("MetaMask is Available :) !");
            }

        // Modern DApp browsers
        if (window.ethereum) {
            window.web3 = new Web3(ethereum);

            // To prevent the page reloading when the MetaMask network changes
            ethereum.autoRefreshOnNetworkChange = false;

            // To Capture the account details from MetaMask
            const accounts = await ethereum.enable();
            account = accounts[0];

            }
        // Legacy DApp browsers
```

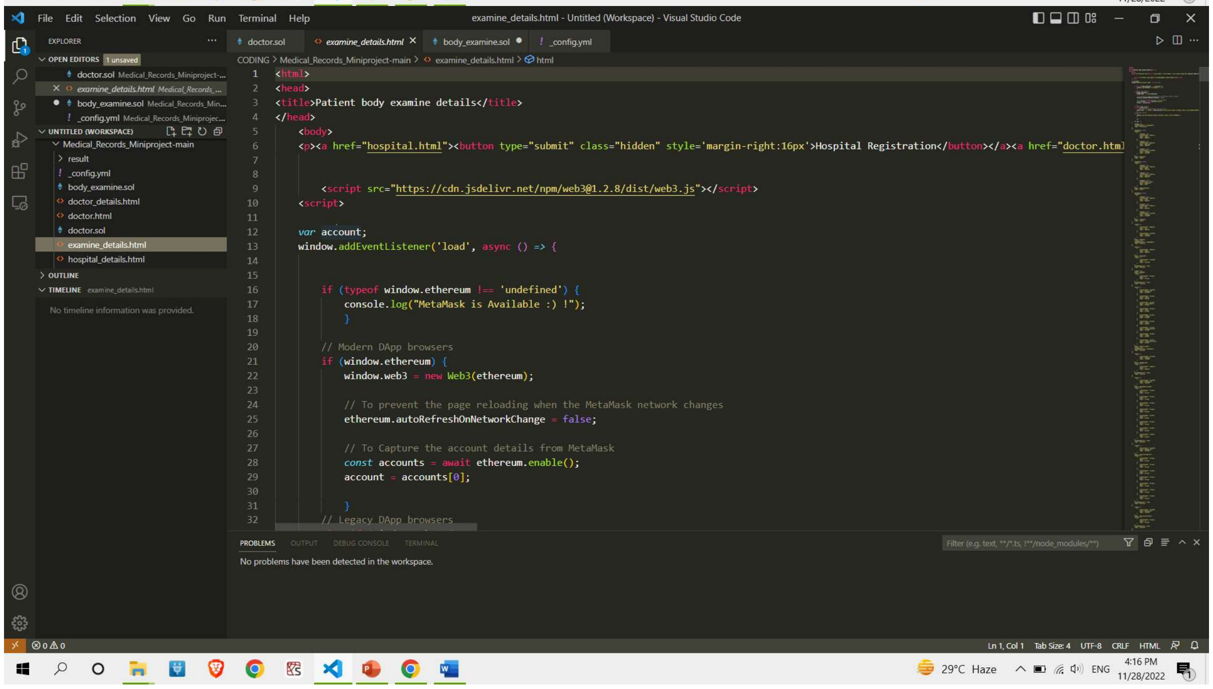
No problems have been detected in the workspace.



doctor.sol - Untitled (Workspace) - Visual Studio Code

```
pragma solidity >=0.4.22 <0.7.0;
/**
 * @title Medical records
 * @dev Store & retreive Doctor details
 */


contract Doctor {


      mapping(uint256 => doctor) doctorlist;

     struct doctor{
         string doctor_name;
         string doctor_specialisation;
         uint256 doctor_ph_no;
         string doctor_address;
     }
     doctor d;

     address owner;


      constructor()  public {
          owner = 0xE6005Cc724c2d44F0aF23d663017a7E375DD7F35; //Address of Hospital
      }

     // modifier to give access only to hospital
       modifier isOwner() {

         require(msg.sender == owner, "Access is not allowed");

```

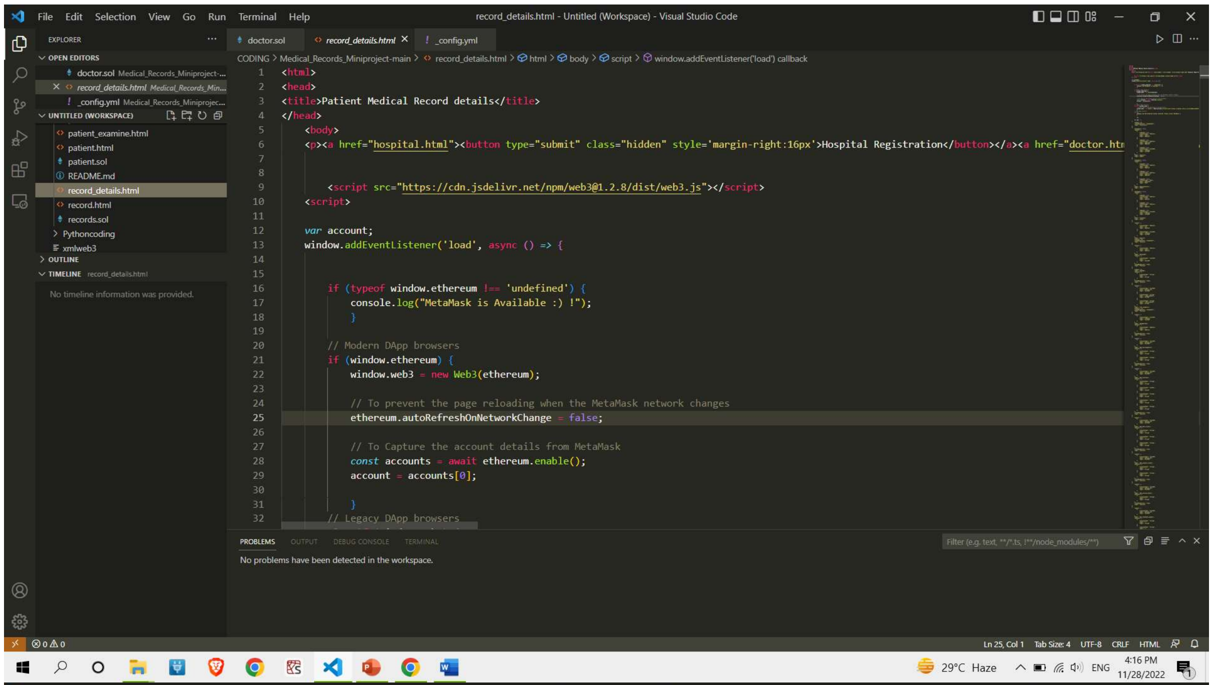
No problems have been detected in the workspace.



examine_details.html - Untitled (Workspace) - Visual Studio Code

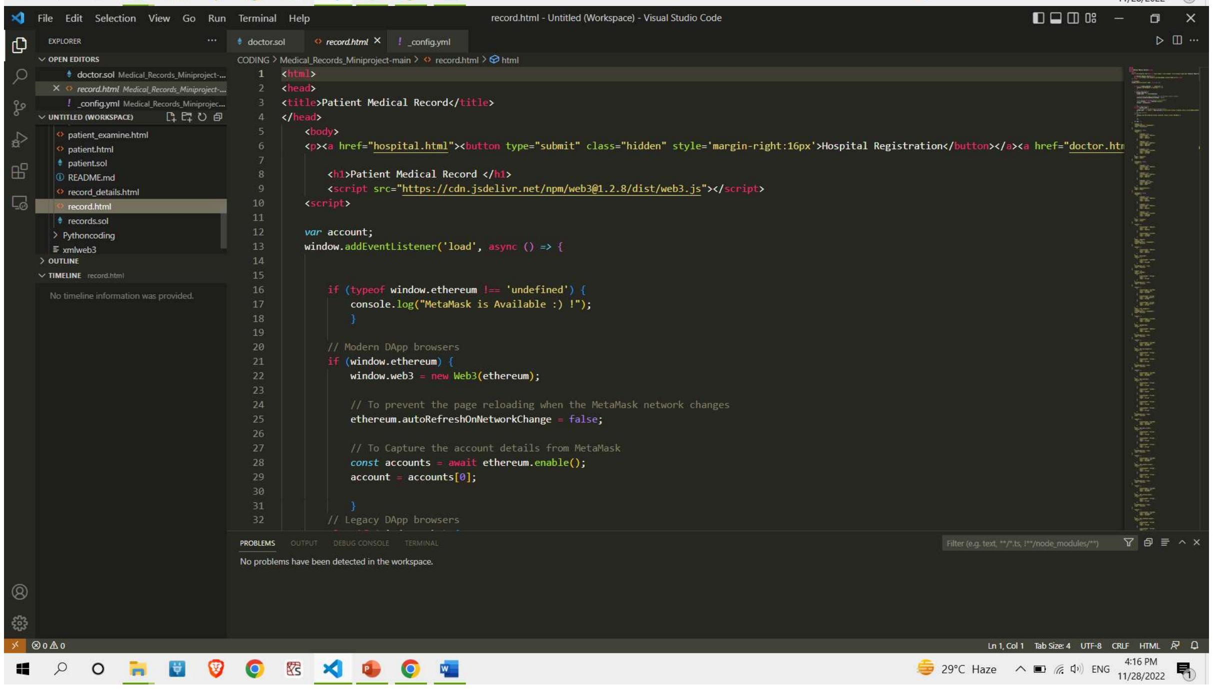
```
<html>
<head>
<title>Patient body examine details</title>
</head>
    <body>
    <p><a href="hospital.html"><button type="submit" class="hidden" style='margin-right:16px'>Hospital Registration</button></a><a href="doctor.html


        <script src="https://cdn.jsdelivr.net/npm/web3@1.2.8/dist/web3.js"></script>
    <script>

    var account;
    window.addEventListener('load', async () => {


        if (typeof window.ethereum !== 'undefined') {
            console.log("MetaMask is Available :) !");
            }

        // Modern DApp browsers
        if (window.ethereum) {
            window.web3 = new Web3(ethereum);

            // To prevent the page reloading when the MetaMask network changes
            ethereum.autoRefreshOnNetworkChange = false;

            // To Capture the account details from MetaMask
            const accounts = await ethereum.enable();
            account = accounts[0];

            }
        // Legacy DApp browsers
```

No problems have been detected in the workspace.


File Edit Selection View Go Run Terminal Help
record_details.html - Untitled (Workspace) - Visual Studio Code
EXPLORER
OPEN EDITORS
doctor.sol Medical_Records_Miniproject-...
record_details.html Medical_Records_Min...
_config.yml Medical_Records_Miniprojec...
UNTITLED (WORKSPACE)
patient_examine.html
patient.html
patient.sol
README.md
record_details.html
record.html
records.sol
Pythoncoding
xmlweb3
OUTLINE
TIMELINE record_details.html
No timeline information was provided.
doctor.sol
record_details.html
_config.yml
CODING > Medical_Records_Miniproject-main > record_details.html > html > body > script > window.addEventListener('load') callback
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 25, Col 1 Tab Size: 4 UTF-8 CRLF HTML
29°C Haze ENG 4:16 PM 11/28/2022

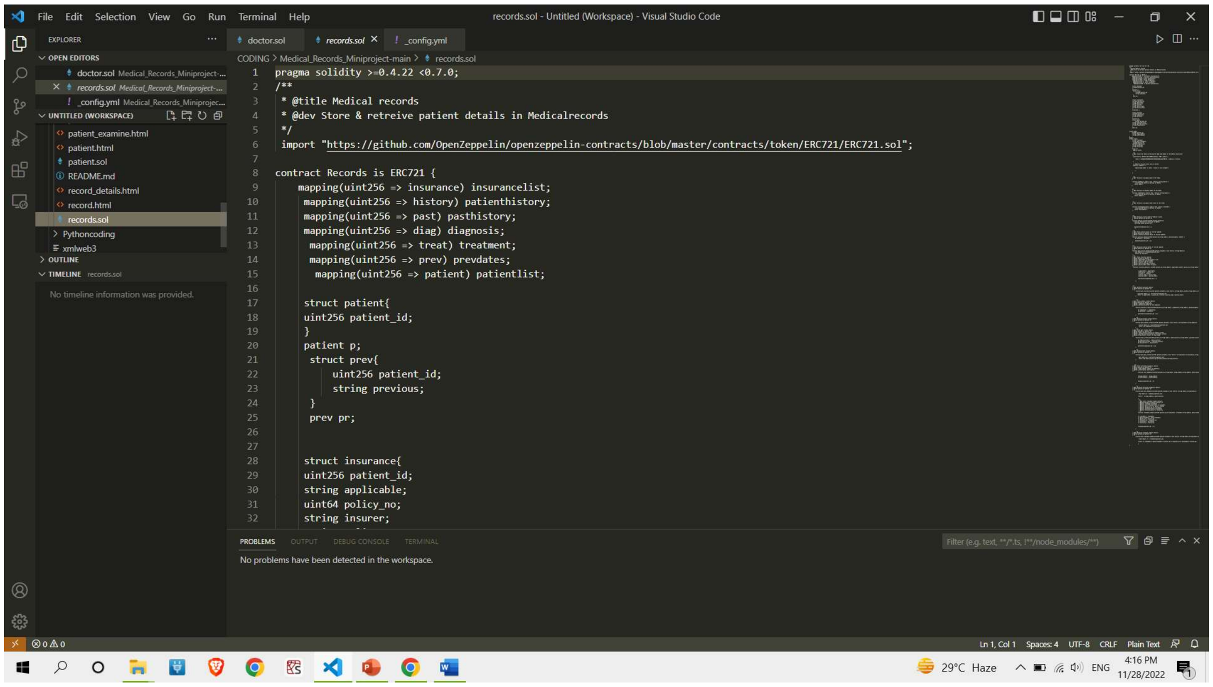
```html
<html>
<head>
<title>Patient Medical Record details</title>
</head>
    <body>
    <p><a href="hospital.html"><button type="submit" class="hidden" style='margin-right:16px'>Hospital Registration</button></a><a href="doctor.htm


        <script src="https://cdn.jsdelivr.net/npm/web3@1.2.8/dist/web3.js"></script>
    <script>

    var account;
    window.addEventListener('load', async () => {


        if (typeof window.ethereum !== 'undefined') {
            console.log("MetaMask is Available :) !");
            }

        // Modern DApp browsers
        if (window.ethereum) {
            window.web3 = new Web3(ethereum);

            // To prevent the page reloading when the MetaMask network changes
            ethereum.autoRefreshOnNetworkChange = false;

            // To Capture the account details from MetaMask
            const accounts = await ethereum.enable();
            account = accounts[0];

            }
        // Legacy DApp browsers
```

No problems have been detected in the workspace.


File Edit Selection View Go Run Terminal Help
record.html - Untitled (Workspace) - Visual Studio Code
EXPLORER
OPEN EDITORS
doctor.sol Medical_Records_Miniproject-...
record.html Medical_Records_Miniproject-...
_config.yml Medical_Records_Miniprojec...
UNTITLED (WORKSPACE)
patient_examine.html
patient.html
patient.sol
README.md
record_details.html
record.html
records.sol
Pythoncoding
xmlweb3
OUTLINE
TIMELINE record.html
No timeline information was provided.
doctor.sol
record.html
_config.yml
CODING > Medical_Records_Miniproject-main > record.html > html
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 1, Col 1 Tab Size: 4 UTF-8 CRLF HTML
29°C Haze ENG 4:16 PM 11/28/2022

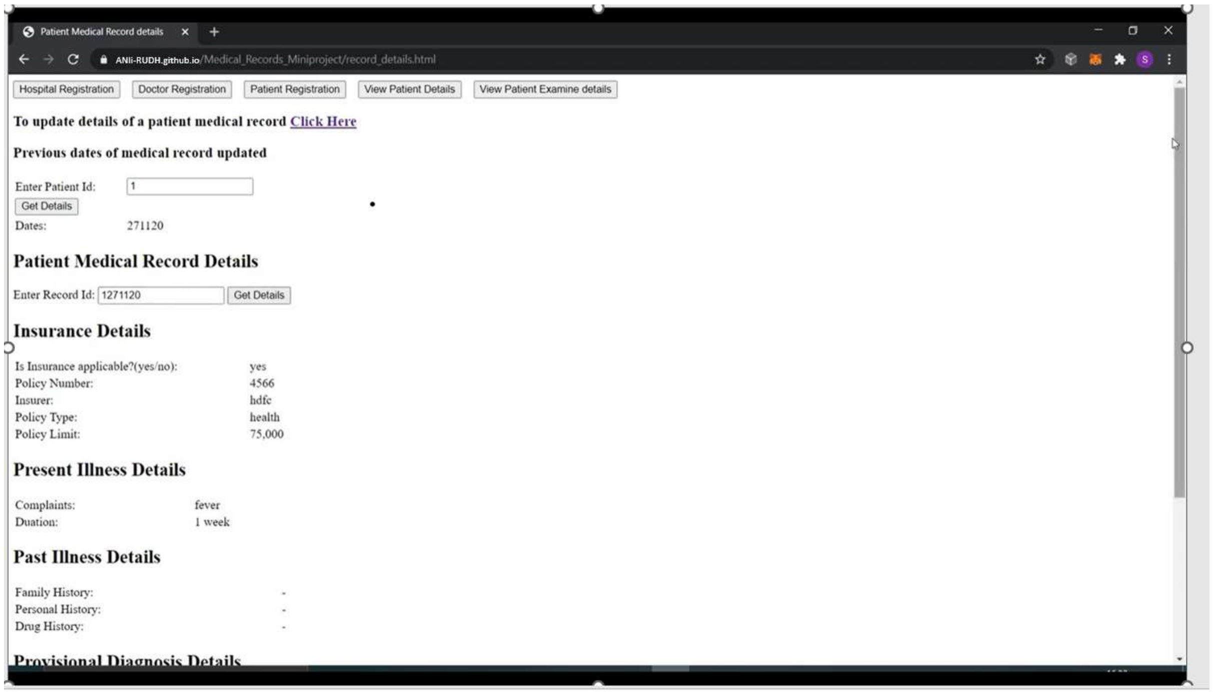
```html
<html>
<head>
<title>Patient Medical Record</title>
</head>
    <body>
    <p><a href="hospital.html"><button type="submit" class="hidden" style='margin-right:16px'>Hospital Registration</button></a><a href="doctor.htm

        <h1>Patient Medical Record </h1>
        <script src="https://cdn.jsdelivr.net/npm/web3@1.2.8/dist/web3.js"></script>
    <script>

    var account;
    window.addEventListener('load', async () => {


        if (typeof window.ethereum !== 'undefined') {
            console.log("MetaMask is Available :) !");
            }

        // Modern DApp browsers
        if (window.ethereum) {
            window.web3 = new Web3(ethereum);

            // To prevent the page reloading when the MetaMask network changes
            ethereum.autoRefreshOnNetworkChange = false;

            // To Capture the account details from MetaMask
            const accounts = await ethereum.enable();
            account = accounts[0];

            }
        // Legacy DApp browsers
```

No problems have been detected in the workspace.


records.sol - Untitled (Workspace) - Visual Studio Code
EXPLORER
OPEN EDITORS
doctor.sol Medical_Records_Miniproject-...
records.sol Medical_Records_Miniproject-...
_config.yml Medical_Records_Miniprojec...
UNTITLED (WORKSPACE)
patient_examine.html
patient.html
patient.sol
README.md
record_details.html
record.html
records.sol
Pythoncoding
xmlweb3
OUTLINE
TIMELINE records.sol
No timeline information was provided.
CODING > Medical_Records_Miniproject-main > records.sol
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
No problems have been detected in the workspace.
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
Plain Text
29°C Haze
ENG
4:16 PM
11/28/2022

```
pragma solidity >=0.4.22 <0.7.0;
/**
 * @title Medical records
 * @dev Store & retreive patient details in Medicalrecords
 */
 import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC721/ERC721.sol";

contract Records is ERC721 {
    mapping(uint256 => insurance) insurancelist;
     mapping(uint256 => history) patienthistory;
     mapping(uint256 => past) pasthistory;
     mapping(uint256 => diag) diagnosis;
      mapping(uint256 => treat) treatment;
      mapping(uint256 => prev) prevdates;
       mapping(uint256 => patient) patientlist;

     struct patient{
     uint256 patient_id;
     }
     patient p;
      struct prev{
          uint256 patient_id;
          string previous;
      }
      prev pr;


     struct insurance{
     uint256 patient_id;
     string applicable;
     uint64 policy_no;
     string insurer;
```

Output



Patient Medical Record details

ANIi-RUDH.github.io/Medical_Records_Miniproject/record_details.html

To update details of a patient medical record Click Here

Previous dates of medical record updated

Enter Patient Id: 1

Get Details

Dates: 271120

Patient Medical Record Details

Enter Record Id: 1271120 Get Details

Insurance Details

Is Insurance applicable?(yes/no):	yes
Policy Number:	4566
Insurer:	hdfc
Policy Type:	health
Policy Limit:	75,000

Present Illness Details

Complaints:	fever
Duation:	1 week

Past Illness Details

Family History:	-
Personal History:	-
Drug History:	-

Provisional Diagnosis Details

CONCLUSIONS

Main feature of the website include flexibility, ease the process of buying and selling of cryptocurrencies and ensuring safety of data of user.

Project could be enhances as per the requirements and the need of the hour.

Online trading is still a new concept in market. In india online trading is still at its beginner stage. Online trading has made it easy to trade in stock and crypto as now people can trade while travelling and sitting at their home. Market is easily accessible to people. Investors are loyal to their local brokers and took their advice before investing and ignoring their own research which leads to huge loss.

Leveraging blockchain technology, MedRec has shown how principles of decentralization might be applied to largescale data management in an EMR system. We demonstrate an innovative approach for handling medical records, providing auditability, interoperability and accessibility via a comprehensive log

Some facts about crypto

1. 92% of exchanges use cold storage.

2. 86% multi signature support.

3. A cryptocurrency exchange from Top 10 earns about $62M per month on average.

4. About 99% of cryptocurrency trade happens on centralized exchanges.

5. There are 1.3 million Bitcoin available on crypto exchanges currently.

6. Bitcoin became legal tender alongside the US Dollar in El Salvador in 2021

REFERENCES

- https://grm.institute/blog/research-study-on-cryptocurrency-exchange-platform/
- https://jfin-swufe.springeropen.com/articles/10.1186/s40854-021-00321-6

[1] Health Information and the Law, "Who owns medial records: 50 state comparison," 2015. [Online]. Available: \url{http://www.healthinfolaw.org}

[2] U.S. Department of HSS, "Hipaa administrative simplification: Regulation text," 2006.

[3] K. D. Mandl et al., "Public standards and patients' control: how to keep electronic medical records accessible but private," BMJ, vol. 322, no. 7281, pp. 283–287, 2001.

[4] The Office of the Nat. Coordinator for Health Information Technology, "Report on health information blocking," U.S. Department of HHS, Tech. Rep., 2015.

[5] U.S. Department of HHS, "Individuals' right under hipaa to access their health information," 2015. [Online]. Available: \url{http://www.hhs.gov}

[6] M. McGinnis et al., Clinical Data as the Basic Staple of Health Learning: Creating and Protecting a Public Good. National Academies Press, 2010.

[7] L. J. Kish and E. J. Topol, "Unpatients– why patients should own their medical data," Nature biotechnology, vol. 33, no. 9, pp. 921–924, 2015.

[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," White Paper, 2008.

[9] G. Zyskind et al., "Decentralizing privacy: Using blockchain to protect personal data," in Security and Privacy Workshops (SPW), 2015 IEEE. IEEE, 2015, pp. 180–184.

[10] Factom, "Healthnautica + factom announce partnership," 2015. [Online]. Available: \url{http://blog.factom.org}

Latest Research Papers

1.https://jfin-swufe.springeropen.com/articles/10.1186/s40854-021-00321-6

2.https://www.alliedmarketresearch.com/crypto-currency-market

3. https://www.coindesk.com/research/

4.https://bitcoin.org/en/bitcoin-paper