**SUMMER INTERNSHIP**

**ELECTRonic Medical report using blockchain**

Submitted by:

Samundar Singh

2019272033

MCA(SS)

**PROJECT WIDE VIEW**

**A PROJECT REPORT**

***Submitted by***

**SAMUNDAR SINGH**

**(2019272033)**

*A report for the Technical Seminar and Report Writing of the project*

*submitted to the Faculty of*

**SUMMER INTERNSHIP**

*in partial fulfilment for the award of the*

*degree* ***of***

**MASTER OF COMPUTER APPLICATIONS**



**DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY COLLEGE OF ENGINEERING, GUINDY ANNA UNIVERSITY**

**CHENNAI 60002**

**JANUARY 2022**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Title** | **p.no** |
| 01 | **Introduction**   * Motivation * Proposed system | **01** |
| **02** | **System Requirements Specifications**   * Requirements analysis | **02** |
| **03** | **System design**   * Architecture diagram * Use case Diagram * UI Design | **03-06** |
| **04** | **Implementation and testing**   * Tools/packages used * Pseudo code/ Snapshots * Code Logic | **07-45** |
| **05** | **Results**   * PERFORMANCE ASSESMENT * PERFORMANCE EVALUATION (TRANSACTION) * COMPARISON OF PROPOSED FRAMEWORK WITH RELATED WORK * Security issues | **45-46** |
| **06** | **Conclusion** | **47** |

**INTRODUCTION**

The recent advent in technology is affecting all parts of human life and is changing the way we use and perceive things previously. Just like the changes technology has offered in various other sectors of life, it is also finding new ways for improvement in healthcare sector. The main benefits that advancement in technology is offering are to improve security, user experience and other aspects of healthcare sector. These benefits were offered by Electronic Health Record (EHR) and Electronic Medical Record (EMR) systems. However, they still face some issues regarding the security of medical records, user ownership of data, data integrity etc. The solution to these issues could be the use of a novel technology, i.e., Blockchain. This technology offers to provide a secure, temper-proof platform for storing medical records and other healthcare related information. Before the advent of modern technology, healthcare sector used paper based system to store the medical records, i.e., using handwritten mechanism. This paper-based medical record system was inefficient, insecure, unorganized and was not temper-proof. It also faced the issue of data- duplication and redundancy as all the institutions that patient visited had various copies of patient’s medical records. The healthcare sector faced a trend shift towards EHR systems that were designed to combine paper-based and electronic medical records (EMR). These systems were used to store clinical notes and laboratory results in its multiple components . They were proposed to enhance the safety aspect of the patients by preventing errors and increasing information access. The goal of EHR systems was to solve the problems faced by the paper-based healthcare records and to provide an efficient system that would transform the state of healthcare sector. The EHR systems have been implemented in a number of hospitals around the world due the benefits it provides, mainly the improvement in security and its cost-effectiveness. They are considered a vital part of healthcare sector as it provides much functionality to the healthcare. These functionalities are electronic storage of medical records, patients‟ appointment management, billing and accounts, and lab tests. They are available in many of the EHR system being used in the healthcare sector. The basic focus is to provide secure, temper-proof, and shareable medical records across different platforms. Despite the fact that notion behind usage of EHR systems in the hospitals or healthcare was to improve the quality of healthcare, these systems faced certain problems and didn’t meet the expectations associated with them . A study was conducted in Finland to find the experiences of nursing staff with the EHR, it was concluded that EHR systems faced the problems related to them being unreliable and having a poor state of user-friendliness. The EHR system also faces some other problems which are as follows: Interoperability: It is the way for different information systems to exchange information between them. The information should be exchangeable and must be usable for further purposes.

**PROPOSED SYSTEM:**

Electronic restorative statistics (EMRs) are basic but very responsive to innate information for finding and treat human services, which have to be as often as viable disseminated and shared amongst pals, as an example, medicinal services suppliers, insurance agencies, drug stores, analysts, sufferers families, amongst others. This represents a noteworthy take a look at on preserving a patient's medicinal history splendid. Putting away and sharing records among various elements. In our system we have proposed the following methodology wherein the Patient has about 90% of the access of the system and rest is accessible to the Doctor. Doctor have the access to view the documents only. Patient only has the access to upload documents and modify them. As the Patient has most of the access which helps us to deal with the confidentiality of the documents. The System which we have implemented is totally different from traditionally used medical record system in Hospitals. We have implemented Different types of software to setup our project, among which Ganache is the most vital one which deals with whole of our Ethereum Platform. We have eliminated the problem of storing large documents by using blockchain. Our system can be deployed from remote servers also

**SYSTEM REQUIREMENTS SPECIFICATIONS**

Requirement in the proposed system are

* Ethereum
* Smart Contract
* Frontend Contract
* Truffle Framework
* Chrome Browser

**1 Ethereum**

Ethereum is smart contract platform that is inspired by block chain technology. Its elemental unit is called ether. Ether, similarly to bitcoin is divisible up to 10-18, its smallest subunit is called wei. Due to the fee-by-computation18 policy, Ether (abbr. ETH) is sometimes referred as the fuel of Ethereum. The intention of Ethereum is to merge together enhanced scripting possibilities, meta protocol and time stamped database to allow development of an arbitrary application. The key difference from other block chain protocols is built-in programming language, various types of accounts and unlimited variation of application that can be built on top of it.

**2 Client**

Geth is the client software used to download the blockchain and run the Ethereum node on a local machine. It is a Go language implementation, hence meaning Go-Ethereum for Geth. It enables us to sync with the Testnet or the Mainnet in order to interact with it. Once Geth is initiated, it will connect to the respective blockchain and start downloading the blocks from the peers. While the sync is happening, the block numbers will be displayed in the output which can be used to verify the blocks.

**3.Accounts In Ethereum,**

The state is defined by the objects named “accounts”, those are 20bytes addresses and the state transitions are the transfers between such accounts. Example of Ethereum address: 0xc2b1918bc7a2c398ec6f20b754992d7c10d3e2cb

Account contains four elemental fields:

* The nonce – counter ensuring each transaction is processed only once
* Ethereum balance
* Contract code – optional depending if account is used as contract or as a standard transaction. Contracts specify hash of the bytecode, for standard transaction is used empty string.
* Storage – space for contract bytecode

**SYSTEM**  **DESIGN**

**Architecture of** **Electronic medical record and insurance claim system.**

Verification Acknowledgement

Smart Contracts

Distributive ledger

P

A

T

I

E

N

T

**PUBLIC BLOCKCHAIN NETWORK**

Verifies medical transaction

Medical Transaction sent for verification

Transfer money after verification

Wallet

I

N

S

U

R

E

R

D

O

C

T

O

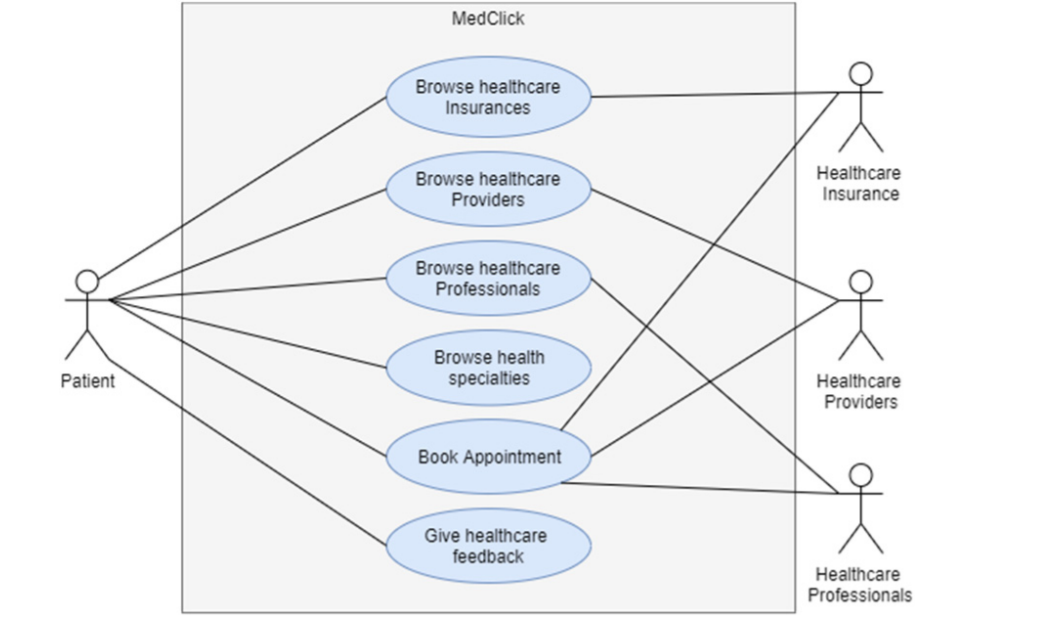
R

BLOCKCHAIN HANDSHAKER

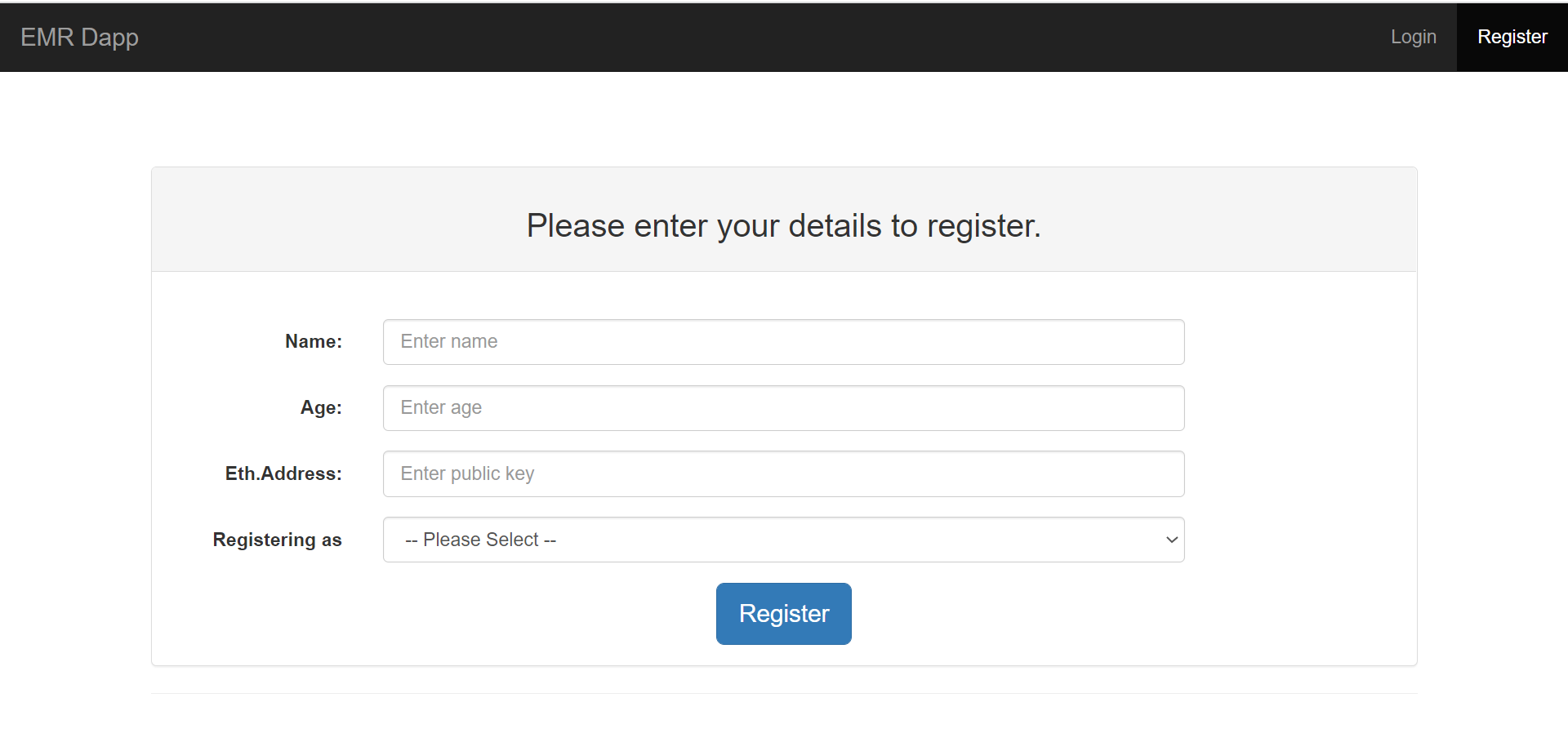
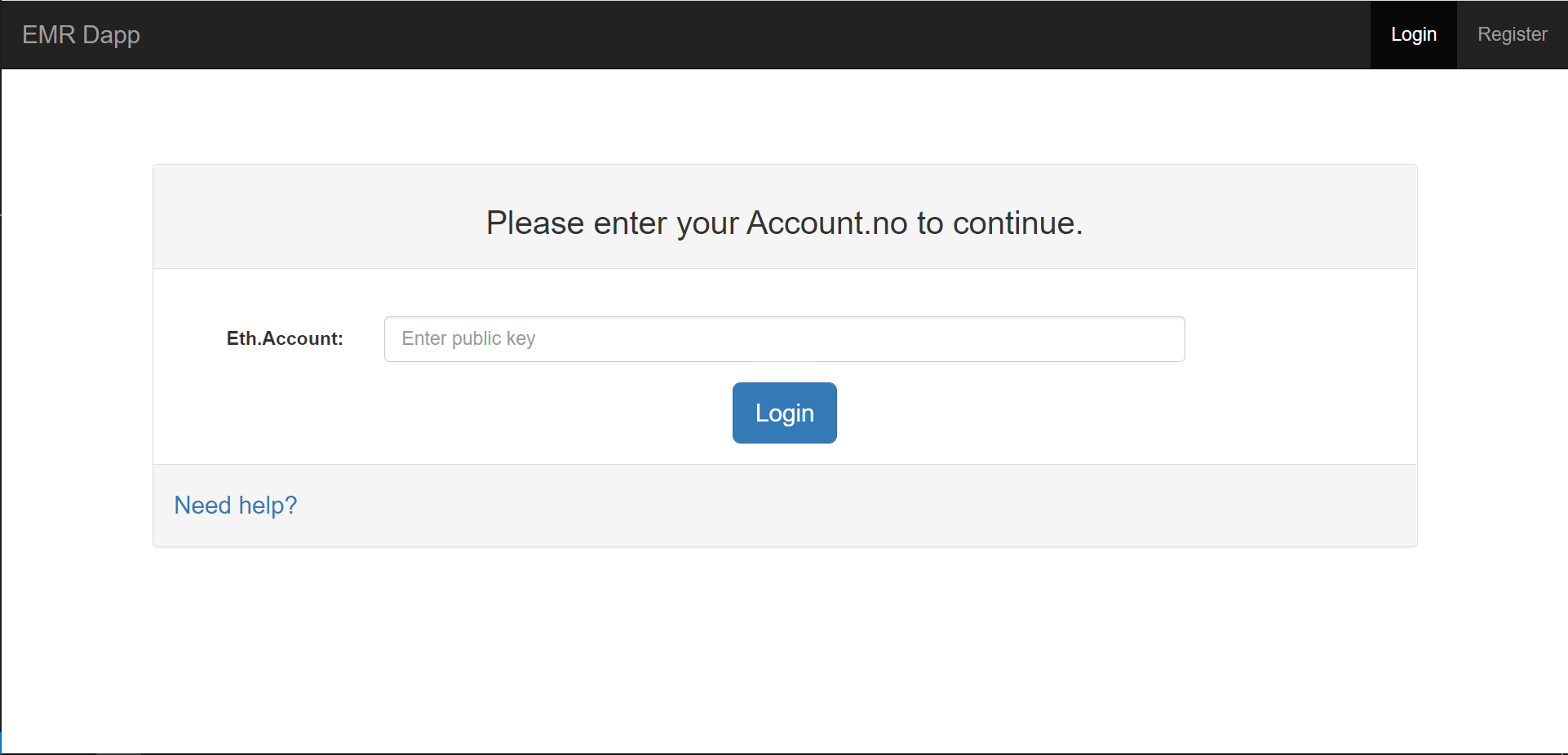
IPFS storage

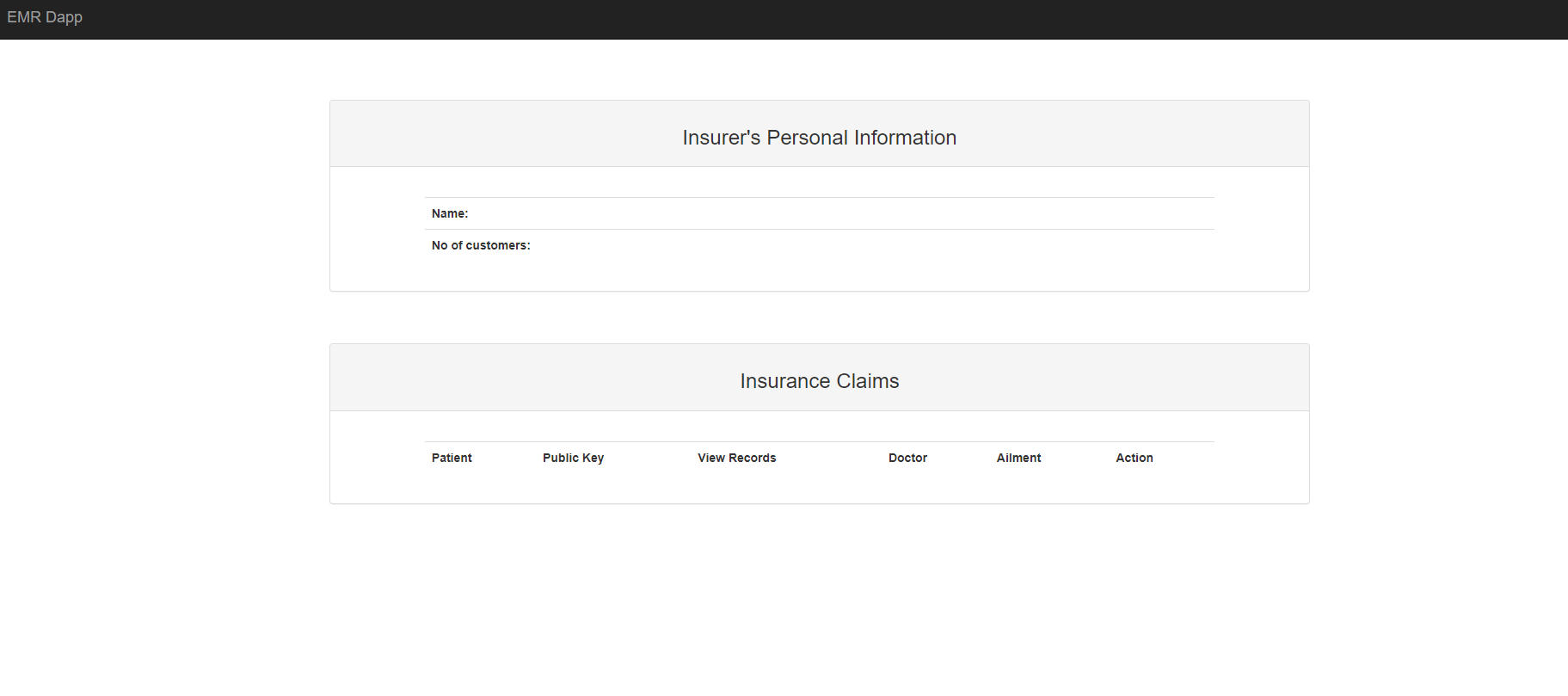
Contain the implementation of Blockchain wrapper

**Use Case Diagram**

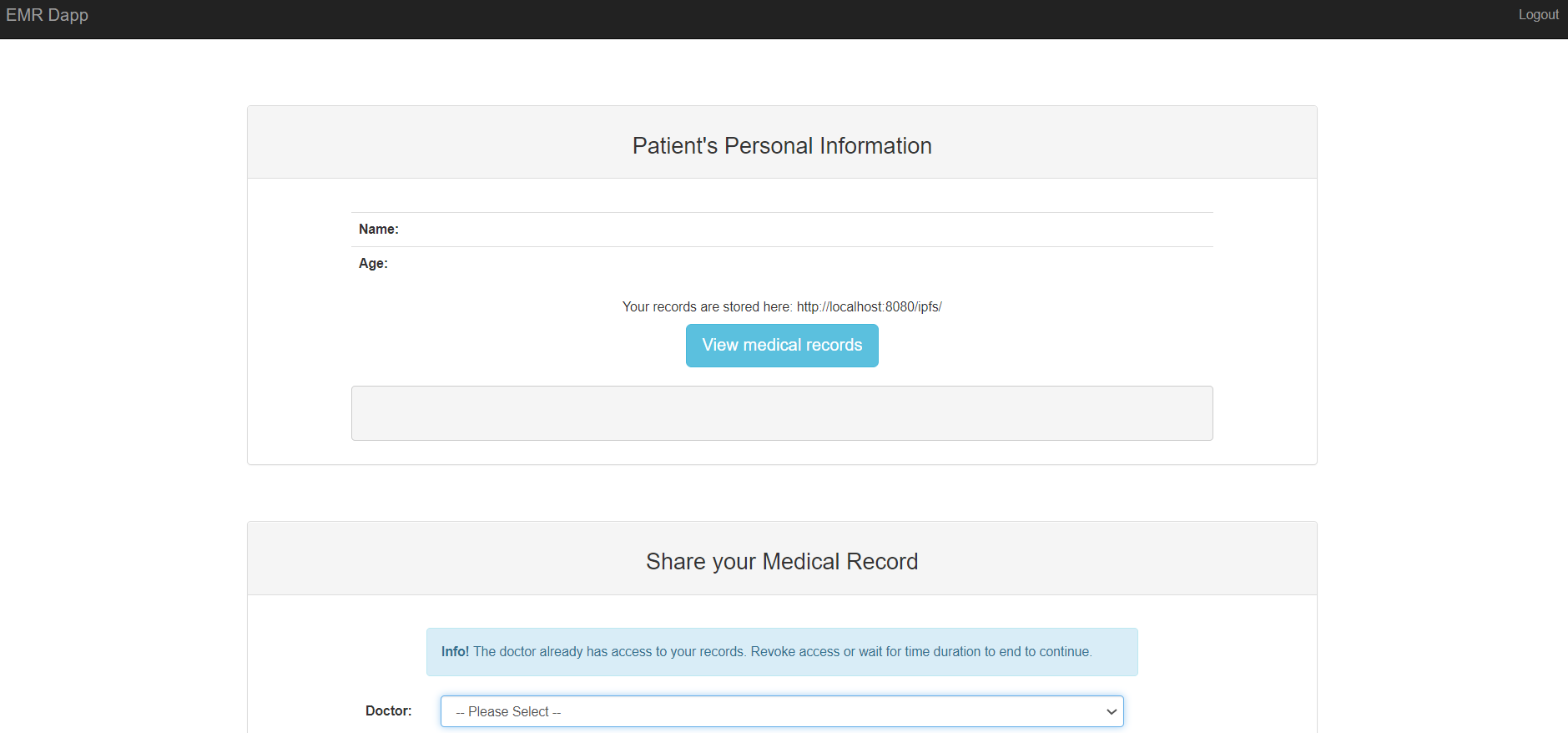


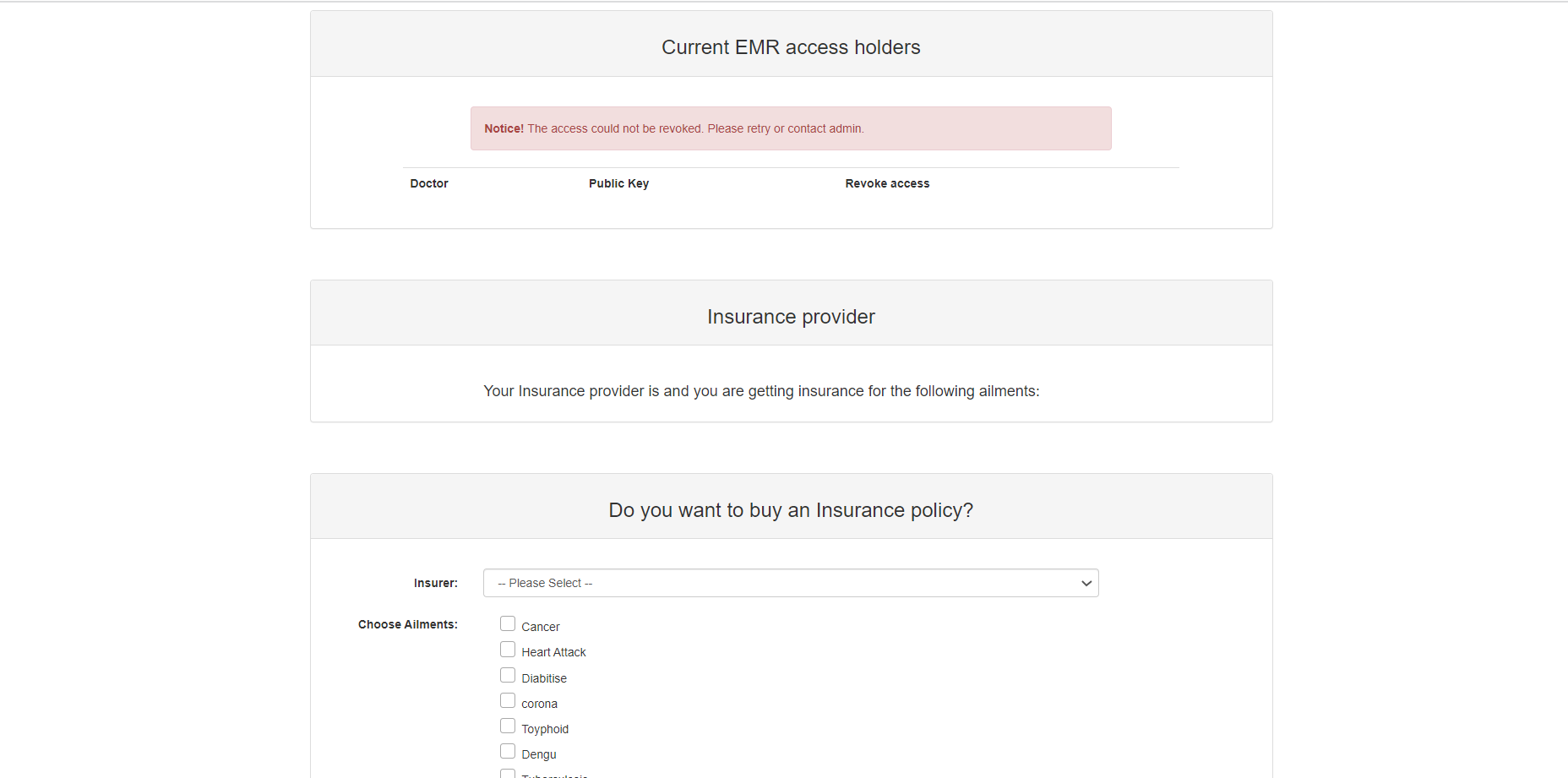
**UI Design:**

1. Register Module:
2. Login Mode
3. Insurer Module



1. Patient Module:





**IMPLEMENTATION AND TESTING**

**Tools Used:**

1. **Ganache -**A instant auto mine feature enabled Blockchain.
2. **Truffle -** Testing framework and asset pipeline for blockchains.
3. **Metamask –** A cryptocoin Wallet.
4. **IPFS-** For storing and sharing data in a distributed file system]

**Code(Doctor):**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <meta name="description" content="">

    <meta name="author" content="">

    <title>EMR Dapp</title>

    <!-- Bootstrap Core CSS -->

    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- <script src="js/bundle.js"></script> -->

    <!-- Custom CSS -->

    <style>

        body {

            max-width: 100vw;

        }

        .panel {

            margin-bottom: 60px;

        }

        .navbar {

            margin-bottom: 70px;

        }

        .panel-heading {

            margin-bottom: 20px;

        }

        .nav-pills>li>a {

            padding: 0;

            padding-right: 10px;

        }

        .nav-pills>li>a:hover {

            background-color: initial;

        }

        .nav-pills>li.active>a {

            color: #23527c;

            background-color: initial;

        }

        .nav-pills>li.active>a:hover,

        .nav-pills>li.active>a:focus {

            color: #23527c;

            background-color: inherit;

        }

    </style>

    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->

    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

    <!--[if lt IE 9]>

        <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>

        <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>

    <![endif]-->

</head>

<body>

    <!-- Navigation -->

    <nav class="navbar navbar-inverse navbar-static-top" role="navigation">

        <div class="container-fluid">

            <!-- Brand and toggle get grouped for better mobile display -->

            <div class="navbar-header">

                <button type="button" class="navbar-toggle" data-toggle="collapse"

                    data-target="#bs-example-navbar-collapse-1">

                    <span class="sr-only">Toggle navigation</span>

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                </button>

                <a class="navbar-brand" href="#">EMR Dapp</a>

            </div>

            <!-- Collect the nav links, forms, and other content for toggling -->

            <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

                <ul class="nav navbar-nav navbar-right">

                    <li>

                        <a href="./index.html">Logout</a>

                    </li>

                </ul>

            </div>

            <!-- /.navbar-collapse -->

        </div>

        <!-- /.container -->

    </nav>

    <div class="container">

        <div class="panel panel-default">

            <div class="panel-heading">

                <h3 class="text-center">Doctor's Personal Information</h3>

            </div>

            <div class="panel-body">

                <div class="row">

                    <div class="col-sm-offset-1 col-sm-10">

                        <table class="table">

                            <tr>

                                <th>Name:</th>

                                <td id="name"></td>

                            </tr>

                            <tr>

                                <th>Age:</th>

                                <td id="age"></td>

                            </tr>

                        </table>

                        <!-- <div class="form-group">

                            <div class="text-center">

                                <button type="submit" class="btn btn-info btn-lg">Update details</button>

                            </div>

                        </div> -->

                    </div>

                </div>

            </div>

        </div>

        <div class="panel panel-default">

            <div class="panel-heading">

                <h3 class="text-center">Accessible EMRs</h3>

            </div>

            <div class="panel-body">

                <div class="row">

                    <div class="alert alert-danger col-sm-8 col-sm-offset-2">

                        <strong>Notice!</strong> Could not access records. Access might have been revoked. Contact admin

                        or patient.

                    </div>

                </div>

                <div class="row">

                    <div class="col-sm-offset-1 col-sm-10">

                        <table id="viewPatient" class="table table-hover">

                            <tr>

                                <th>Patient</th>

                                <th class="publicKeyPatient">Public Key</th>

                                <th>Action</th>

                            </tr>

                        </table>

                    </div>

                </div>

            </div>

        </div>

    </div>

    <script src="js/jquery.js"></script>

    <!-- Bootstrap Core JavaScript -->

    <script src="js/bootstrap.min.js"></script>

    <script src="js/web3.min.js"></script>

    <script src="js/app.js"></script>

    <script src="https://unpkg.com/ipfs-api/dist/index.min.js" crossorigin="anonymous"></script>

    <script>

        var ailmentsDict = {};

        var url\_string = window.location.href;

        var url = new URL(url\_string);

        var key = url.searchParams.get("key");

        var docName = ""

        key = key.toLocaleLowerCase();

        var ipfs = window.IpfsApi('localhost', '5001')

        const Buffer = window.IpfsApi().Buffer;

        displayDoctors()

        async function displayDoctors() {

            var contract = await init()

            ailmentsDict[0] = "Cancer";

            ailmentsDict[1] = "Heart Attack";

            ailmentsDict[2] = "Diabitise";

            ailmentsDict[3] = "Corona"

            ailmentsDict[4] = "Toyphoid";

            ailmentsDict[5] = "dengu";

            ailmentsDict[6] = "Maleria";

            ailmentsDict[7] = "Tuberclosis";

            toggleRecordsButton = 0;

            $(document).ready(function () {

                $(".alert-danger").hide();

                var a = 0;

                var b = 0;

                contract.methods.get\_doctor(key).call({ gas: 1000000 }, function (error, result) {

                    if (!error) {

                        a = result[0];

                        b = result[1];

                        docName = a;

                        $("#name").html(a);

                        $("#age").html(b);

                    }

                    else

                        console.error(error);

                });

                var patientAddressList = 0;

                contract.methods.get\_accessed\_patientlist\_for\_doctor(key).call({ gas: 1000000 }, function (error, result) {

                    if (!error) {

                        patientAddressList = result;

                        console.log(result);

                        patientAddressList.forEach(function (patientAddress, index) {

                            contract.methods.get\_patient(patientAddress).call({ gas: 1000000 }, function (error, result) {

                                var table = document.getElementById("viewPatient");

                                if (!error) {

                                    a = result[0];

                                    // console.log("Hi");

                                    // console.log(i);

                                    var row = table.insertRow(index + 1);

                                    var cell1 = row.insertCell(0);

                                    var cell2 = row.insertCell(1);

                                    var cell3 = row.insertCell(2);

                                    cell2.className = "publicKeyPatient";

                                    cell1.innerHTML = a;

                                    cell2.innerHTML = patientAddress;

                                    cell3.innerHTML = '<input class="btn btn-success" onclick="showRecords(this)" id="viewRecordsButton" type="button" value="View records"></input>';

                                }

                                else

                                    console.error(error);

                            })

                        })

                    }

                    else

                        console.error(error);

                });

            });

        }

        function showRecords(element) {

            var table = document.getElementById("viewPatient");

            var index = element.parentNode.parentNode.rowIndex;

            var patientAddress = table.rows[index].cells[1].innerHTML;

            if (toggleRecordsButton % 2 == 0) {

                var patientRecord = ""

                contract.methods.get\_hash(patientAddress).call({ gas: 1000000 }, function (error, result) {

                    if (!error) {

                        $.get("http://localhost:8080/ipfs/" + result, function (data) {

                            patientRecord = data;

                            console.log(data)

                            content = `<div class="tab-content">

                <div id="view${patientAddress}">

                        <div class="row">

                            <div class="col-sm-12">

                                <pre style="margin: 20px;" id="records${patientAddress}">${patientRecord}</pre>

                            </div>

                        </div>

                        <div class="row">

                            <div class="col-sm-12">

                                <div class="row">

                                    <div class="form-group col-sm-10">

                                        <div class="row">

                                            <div class="col-sm-2"><label for="ailmentsList" class="control-label">Diagnosis:</label></div>

                                            <div class="col-sm-10">

                                                <select class="form-control" id="ailmentsList${patientAddress}" style="width:inherit;" required>

                                                    <option selected disabled>-- Please Select --</option>

                                                    <option value = "0">Cancer</option>

                                                    <option value = "1">Heart Attack</option>

                                                    <option value = "2">Diabitise</option>

                                                    <option value = "3">Corona</option>

                                                    <option value = "4">Toyphoid's</option>

                                                    <option value = "5">dengu</option>

                                                    <option value = "6">Tuberculosis</option>

                                                    <option value = "7">maleria</option>

                                                </select>

                                            </div>

                                        </div>

                                    </div>

                                </div>

                                <div class="row">

                                    <div class="form-group col-sm-10">

                                        <div class="row">

                                            <div class="col-sm-2">

                                                <label class="control-label" for="details">Details:</label>

                                            </div>

                                            <div class="col-sm-10">

                                                <textarea class="form-control" rows="5" id="details" placeholder="Enter details to be added" name = "Details" style="width: inherit" required autofocus></textarea>

                                                <!-- <input type="text" class="form-control" id="details" placeholder="Enter details to be added" name = "Details" style="width: inherit" required autofocus> -->

                                            </div>

                                        </div>

                                    </div>

                                    <div class="form-group col-sm-2">

                                        <button class="btn btn-primary" onclick = "submitDiagnosis(this,`+ index + `)">Submit</button>

                                    </div>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>`

                            var row1 = table.insertRow(index + 1);

                            var cell1 = row1.insertCell(0);

                            cell1.colSpan = 3;

                            cell1.innerHTML = content;

                        })

                    } else {

                        console.log(error);

                    }

                })

                toggleRecordsButton += 1

                element.value = "Hide Records";

                element.className = "btn btn-danger"

            } else {

                row = table.rows[index + 1];

                $(row).hide();

                toggleRecordsButton -= 1;

                element.value = "View Records";

                element.className = "btn btn-success"

            }

        }

        function submitDiagnosis(element, index) {

            var table = document.getElementById("viewPatient");

            var patientAddress = table.rows[index].cells[1].innerHTML;

            console.log(patientAddress);

            // var dropSelect = document.getElementById("ailmentsList"+patientAddress);

            var diagnosis = $("#ailmentsList" + patientAddress).val();

            diagnosis = parseInt(diagnosis);

            var diagnosed = ailmentsDict[diagnosis];

            var comments = document.getElementById("details").value;

            var oldRecords = $("#records" + patientAddress).html();

            /\* var newRecords = {

                'Diagnosed By': docName,

                "Doctor's Public Key" : key,

                "Diagnosis": diagnosed,

                "Comments" : comments

            } \*/

            var newRecords = `

            Diagnosed By : ${docName}

            Doctor's Public Key : ${key}

            Diagnosis : ${diagnosed}

            Comments : ${comments}`

            var updatedRecords = oldRecords + newRecords;

            if (!isNaN(diagnosis)) {

                var buffer = Buffer(updatedRecords);

                ipfs.files.add(buffer, (error, result) => {

                    if (error) {

                        console.log(error)

                    } else {

                        ipfshash = result[0].hash;

                        contract.methods.insurance\_claim(patientAddress, diagnosis, ipfshash).send({ from: key, gas: 1000000 }, function (error, result) {

                            if (!error) {

                                alert("Your diagnosis has been submitted.");

                                // delete content row

                                table.deleteRow(index + 1);

                                // delete main row of corresponding content row

                                table.deleteRow(index);

                            } else {

                                $(".alert-danger").show();

                                console.log(error);

                            }

                        })

                    }

                });

            }

            else {

                alert("Select a diagnosis");

            }

        }

    </scrip

</body

**Code(Patient):**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <meta name="description" content="">

    <meta name="author" content="">

    <title>EMR Dapp</title>

    <!-- Bootstrap Core CSS -->

    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- <script src="js/bundle.js"></script> -->

    <!-- Custom CSS -->

    <style>

        .navbar {

            margin-bottom: 70px;

        }

        .panel-heading {

            margin-bottom: 20px;

        }

        .panel {

            margin-bottom: 60px;

        }

        /\* .publicKeyDoctor{

        display: none;

    } \*/

        .well>h3 {

            margin: 10px auto;

        }

        .checkbox label:after {

            content: '';

            display: table;

            clear: both;

        }

        .checkbox .cr {

            position: relative;

            display: inline-block;

            border: 1px solid #a9a9a9;

            border-radius: .25em;

            width: 1.3em;

            height: 1.3em;

            margin-right: .5em;

        }

        .checkbox .cr .cr-icon {

            position: absolute;

            font-size: .8em;

            line-height: 0;

            top: 50%;

            left: 15%;

        }

        .checkbox label input[type="checkbox"] {

            display: none;

        }

        .checkbox label input[type="checkbox"]+.cr>.cr-icon {

            opacity: 0;

        }

        .checkbox label input[type="checkbox"]:checked+.cr>.cr-icon {

            opacity: 1;

        }

    </style>

    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->

    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

    <!--[if lt IE 9]>

        <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>

        <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>

    <![endif]-->

</head>

<body>

    <!-- Navigation -->

    <nav class="navbar navbar-inverse navbar-static-top" role="navigation">

        <div class="container-fluid">

            <!-- Brand and toggle get grouped for better mobile display -->

            <div class="navbar-header">

                <button type="button" class="navbar-toggle" data-toggle="collapse"

                    data-target="#bs-example-navbar-collapse-1">

                    <span class="sr-only">Toggle navigation</span>

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                </button>

                <a class="navbar-brand" href="#">EMR Dapp</a>

            </div>

            <!-- Collect the nav links, forms, and other content for toggling -->

            <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

                <ul class="nav navbar-nav navbar-right">

                    <li>

                        <a href="./index.html">Logout</a>

                    </li>

                </ul>

            </div>

            <!-- /.navbar-collapse -->

        </div>

        <!-- /.container -->

    </nav>

    <div class="container">

        <div class="panel panel-default">

            <div class="panel-heading">

                <h3 class="text-center">Patient's Personal Information</h3>

            </div>

            <div class="panel-body">

                <div class="row">

                    <div class="col-sm-offset-1 col-sm-10">

                        <table class="table">

                            <tr>

                                <th>Name:</th>

                                <td id="name"></td>

                            </tr>

                            <tr>

                                <th>Age:</th>

                                <td id="age"></td>

                            </tr>

                        </table>

                        <div class="text-center">

                            <h5>Your records are stored here: http://localhost:8080/ipfs/<span id="recordsHash"></span>

                            </h5>

                            <button type="submit" class="btn btn-info btn-lg" onclick="showRecords(this)">View medical

                                records</button>

                        </div>

                        <pre id="records" style="margin-top: 20px;">

                            </pre>

                    </div>

                </div>

            </div>

        </div>

        <div class="panel panel-default">

            <div class="panel-heading">

                <h3 class="text-center">Share your Medical Record</h3>

            </div>

            <div class="panel-body">

                <div class="row">

                    <div class="alert alert-info col-sm-8 col-sm-offset-2">

                        <strong>Info!</strong> The doctor already has access to your records. Revoke access or wait for

                        time duration to end to continue.

                    </div>

                </div>

                <form class="form-horizontal" action="/action\_page.php">

                    <div class="form-group">

                        <label for="permitDoctorList" class="control-label col-sm-2">Doctor:</label>

                        <div class="col-sm-8">

                            <select class="form-control" id="permitDoctorList">

                                <option selected disabled>-- Please Select --</option>

                            </select>

                        </div>

                    </div>

                    <!-- <div class="form-group">

                            <label for="sel2" class="control-label col-sm-2">Time duration:</label>

                            <div class="col-sm-8">

                                <select class="form-control" id="sel2">

                                    <option selected disabled>-- Please Select --</option>

                                    <option>One time access</option>

                                    <option>Three time access</option>

                                    <option>Ten time access</option>

                                    <option>Six hours from first access</option>

                                    <option>Twelve hours from first access</option>

                                    <option>One day from first access</option>

                                    <option>One week from first access</option>

                                    <option>One month from first access</option>

                                </select>

                            </div>

                        </div> -->

                </form>

                <div class="text-center">

                    <button onclick="giveAccess()" class="btn btn-primary btn-lg">Submit</button>

                </div>

            </div>

        </div>

        <div class="panel panel-default">

            <div class="panel-heading">

                <h3 class="text-center">Current EMR access holders</h3>

            </div>

            <div class="panel-body">

                <div class="row">

                    <div class="alert alert-danger col-sm-8 col-sm-offset-2">

                        <strong>Notice!</strong> The access could not be revoked. Please retry or contact admin.

                    </div>

                </div>

                <div class="row">

                    <div class="col-sm-offset-1 col-sm-10">

                        <table id="accessDoc" class="table table-hover">

                            <tr>

                                <th>Doctor</th>

                                <th class="publicKeyDoctor">Public Key</th>

                                <th>Revoke access</th>

                            </tr>

                        </table>

                    </div>

                </div>

            </div>

        </div>

        <div class="panel panel-default" id="insuranceInfo">

            <div class="panel-heading">

                <h3 class="text-center">Insurance provider</h3>

            </div>

            <div class="panel-body">

                <!-- <div class="row">

                        <div class="alert alert-danger col-sm-8 col-sm-offset-2">

                            <strong>Notice!</strong> The access could not be revoked. Please retry or contact admin.

                        </div>

                    </div> -->

                <div class="row">

                    <div class="col-sm-8 col-sm-offset-2">

                        <h4>Your Insurance provider is <strong id="insuranceProvider"></strong> and you are getting

                            insurance for the

                            following ailments:</h4>

                        <ul id="ailmentsCovered">

                        </ul>

                    </div>

                </div>

            </div>

        </div>

        <div class="panel panel-default" id="buyInsurance">

            <div class="panel-heading">

                <h3 class="text-center">Do you want to buy an Insurance policy?</h3>

            </div>

            <div class="panel-body">

                <!-- <div class="row">

                        <div class="alert alert-danger col-sm-8 col-sm-offset-2">

                            <strong>Notice!</strong> The access could not be revoked. Please retry or contact admin.

                        </div>

                    </div> -->

                <form class="form-horizontal" action="/action\_page.php">

                    <div class="form-group">

                        <label for="permitDoctorList" class="control-label col-sm-2">Insurer:</label>

                        <div class="col-sm-8">

                            <select class="form-control" id="insurers">

                                <option selected disabled>-- Please Select --</option>

                            </select>

                        </div>

                    </div>

                    <div class="form-group">

                        <label for="sel2" class="control-label col-sm-2">Choose Ailments:</label>

                        <div class="col-sm-8">

                            <div class="checkbox">

                                <label><input type="checkbox" name="ailment" value="0" onclick="calc()"

                                        autocomplete="off"><span class="cr"><i

                                            class="cr-icon glyphicon glyphicon-ok"></i></span>Cancer</label>

                            </div>

                            <div class="checkbox">

                                <label><input type="checkbox" name="ailment" value="1" onclick="calc()"

                                        autocomplete="off"><span class="cr"><i

                                            class="cr-icon glyphicon glyphicon-ok"></i></span>Heart Attack</label>

                            </div>

                            <div class="checkbox">

                                <label><input type="checkbox" name="ailment" value="2" onclick="calc()"

                                        autocomplete="off"><span class="cr"><i

                                            class="cr-icon glyphicon glyphicon-ok"></i></span>Diabitise</label>

                            </div>

                            <div class="checkbox">

                                <label><input type="checkbox" name="ailment" value="3" onclick="calc()"

                                        autocomplete="off"><span class="cr"><i

                                            class="cr-icon glyphicon glyphicon-ok"></i></span>corona</label>

                            </div>

                            <div class="checkbox">

                                <label><input type="checkbox" name="ailment" value="4" onclick="calc()"

                                        autocomplete="off"><span class="cr"><i

                                            class="cr-icon glyphicon glyphicon-ok"></i></span>Toyphoid</label>

                            </div>

                            <div class="checkbox">

                                <label><input type="checkbox" name="ailment" value="5" onclick="calc()"

                                        autocomplete="off"><span class="cr"><i

                                            class="cr-icon glyphicon glyphicon-ok"></i></span>Dengu</label>

                            </div>

                            <div class="checkbox">

                                <label><input type="checkbox" name="ailment" value="6" onclick="calc()"

                                        autocomplete="off"><span class="cr"><i

                                            class="cr-icon glyphicon glyphicon-ok"></i></span>Tuberculosis</label>

                            </div>

                            <div class="checkbox">

                                <label><input type="checkbox" name="ailment" value="7" onclick="calc()"

                                        autocomplete="off"><span class="cr"><i

                                            class="cr-icon glyphicon glyphicon-ok"></i></span>Maleria</label>

                            </div>

                        </div>

                    </div>

                </form>

                <div class="row">

                    <div class="col-sm-offset-2 col-sm-8 well-sm well">

                        <h3 class="text-center">You will have to pay <strong><span id="emi">0</span> ETH</strong> every

                            year.</h3>

                        <h3 class="text-center">You will get <strong>4 ETH</strong> for each succesful claim.</h3>

                    </div>

                </div>

                <div class="row">

                    <div class="col-sm-offset-4 col-sm-4 text-center">

                        <button onclick="buyInsurance()" class="btn btn-primary btn-lg">Buy</button>

                    </div>

                </div>

            </div>

        </div>

    </div>

    <script src="js/jquery.js"></script>

    <!-- Bootstrap Core JavaScript -->

    <script src="js/bootstrap.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/web3@1.3.6/dist/web3.min.js"></script>

    <script src="js/app.js"></script>

    <script>

        var url\_string = window.location.href;

        var url = new URL(url\_string);

        var key = url.searchParams.get("key");

        key = key.toLocaleLowerCase();

        var recordHash = "";

        var ailmentsDict = {};

        var acc;

        displayPatient()

        async function displayPatient() {

            contract = await init()

            // console.log(key);

            var acc = await window.ethereum.enable();

            acc = acc[0].toLocaleLowerCase();

            console.log(acc)

            ailmentsDict[0] = "Cancer";

            ailmentsDict[1] = "Heart Attack";

            ailmentsDict[2] = "Diabitise";

            ailmentsDict[3] = "Corona"

            ailmentsDict[4] = "Toyphoid";

            ailmentsDict[5] = "dengu";

            ailmentsDict[6] = "Tuberculosis";

            ailmentsDict[7] = "maleria";

            toggleRecordsButton = 0;

            $(document).ready(function () {

                $("#records").hide();

                $(".alert-info").hide();

                $(".alert-danger").hide();

                var a = "";

                var b = 0;

                var ailments = [];

                var insurerName = "";

                $("#buyInsurance").hide();

                $("#insuranceInfo").hide();

                // print patient details and insurer details (if exists). If insurer does not exist show the buy insurance panel

                contract.methods.get\_patient(key).call({from: acc, gas: 1000000 }, function (error, result) {

                    if (!error) {

                        a = result[0];

                        b = result[1];

                        ailments = result[2];

                        insurerAddress = result[3];

                        recordHash = result[4];

                        console.log(result)

                        $("#name").html(a);

                        $("#age").html(b);

                        $("#recordsHash").html(recordHash);

                        if (insurerAddress != 0) {

                            $("#buyInsurance").hide();

                            contract.methods.get\_insurer(insurerAddress).call({ gas: 1000000 }, function (error, result) {

                                insurerName = result[0];

                                var ailmentsInsurance = ""

                                for (var i = 0; i < ailments.length; i++) {

                                    ailmentsInsurance += "<li>" + ailmentsDict[ailments[i]] + "</li>"

                                }

                                $("#ailmentsCovered").html(ailmentsInsurance);

                                $("#insuranceProvider").html(insurerName);

                                $("#insuranceInfo").show();

                            })

                        } else {

                            var InsurerList = 0;

                            contract.methods.get\_insurer\_list().call({ gas: 1000000 }, function (error, result) {

                                if (!error) {

                                    InsurerList = result;

                                    var list = document.getElementById("insurers");

                                    for (var i = 0; i < InsurerList.length; i++) {

                                        contract.methods.get\_insurer(InsurerList[i]).call({ gas: 1000000 }, function (error, result) {

                                            if (!error) {

                                                d = result[0];

                                                var option = document.createElement("option");

                                                option.text = d;

                                                list.add(option);

                                            } else {

                                                console.log(error);

                                            }

                                        })

                                    }

                                }

                            })

                            $("#buyInsurance").show();

                            $("#insuranceInfo").hide();

                        }

                    }

                    else

                        console.error(error);

                });

                // print out the doctors to share emr

                var DoctorList = 0;

                contract.methods.get\_doctor\_list().call({ gas: 1000000 }, function (error, result) {

                    if (!error) {

                        DoctorList = result;

                        for (var i = 0; i < DoctorList.length; i++) {

                            contract.methods.get\_doctor(DoctorList[i]).call({ gas: 1000000 }, function (error, result) {

                                var list = document.getElementById("permitDoctorList");

                                console.log(result)

                                if (!error) {

                                    a = result[0];

                                    b = result[1];

                                    var option = document.createElement("option");

                                    option.text = a;

                                    list.add(option);

                                    // console.log(a);

                                }

                                else

                                    console.error(error);

                            })

                        }

                    }

                    else

                        console.error(error);

                });

                // print out the doctors who have access

                var doctorAddressList = 0;

                contract.methods.get\_accessed\_doctorlist\_for\_patient(key).call({ gas: 1000000 }, function (error, result) {

                    if (!error) {

                        doctorAddressList = result;

                        // console.log(result);

                        doctorAddressList.forEach(function (doctorAddress, index) {

                            contract.methods.get\_doctor(doctorAddress).call({ gas: 1000000 }, function (error, result) {

                                var table = document.getElementById("accessDoc");

                                if (!error) {

                                    a = result[0];

                                    console.log(a);

                                    // console.log("Hi");

                                    // console.log(i);

                                    var row = table.insertRow(index + 1);

                                    var cell1 = row.insertCell(0);

                                    var cell2 = row.insertCell(1);

                                    var cell3 = row.insertCell(2);

                                    cell2.className = "publicKeyDoctor";

                                    cell1.innerHTML = a;

                                    cell2.innerHTML = doctorAddress;

                                    cell3.innerHTML = '<button onclick="revokeAccess(this)" class="btn btn-danger">Revoke access</button>';

                                    console.log(result);

                                }

                                else

                                    console.error(error);

                            })

                        })

                    }

                    else

                        console.error(error);

                });

            });

        }

        function buyInsurance() {

            var list = document.getElementById("insurers");

            insurerName = $("#insurers").val();

            index = list.selectedIndex;

            var ailments = [];

            $.each($('input[name="ailment"]:checked'), function () {

                ailments.push(parseInt($(this).val()));

            });

            console.log(ailments);

            console.log(ailmentsDict);

            number\_of\_ailments = document.querySelectorAll('input[name="ailment"]:checked').length;

            to\_pay = number\_of\_ailments;

            contract.methods.get\_insurer\_list().call({ gas: 1000000 }, function (error, result) {

                InsurerList = result;

                insurerToBuy = InsurerList[index - 1];

                contract.methods.select\_insurer(insurerToBuy, ailments).send({ from: key, gas: 1000000, value: Web3.utils.toWei(String(to\_pay), 'ether') }, function (error) {

                    if (!error) {

                        $("#buyInsurance").hide();

                        $("#insuranceProvider").html(insurerName);

                        var ailmentsInsurance = ""

                        for (var i = 0; i < ailments.length; i++) {

                            ailmentsInsurance += "<li>" + ailmentsDict[ailments[i]] + "</li>"

                        }

                        $("#ailmentsCovered").html(ailmentsInsurance);

                        $("#insuranceInfo").show();

                    } else {

                        console.log(error)

                    }

                })

            })

        }

        function giveAccess() {

            var list = document.getElementById("permitDoctorList");

            index = list.selectedIndex;

            var DoctorList = 0;

            contract.methods.get\_doctor\_list().call({ gas: 1000000 }, function (error, result) {

                if (!error) {

                    // console.log(index);

                    DoctorList = result;

                    doctorToBeAdded = DoctorList[index - 1];

                    console.log(web3);

                    contract.methods.permit\_access(doctorToBeAdded).send({ from: key, gas: 1000000, value: Web3.utils.toWei('2', 'ether') }, function (error) {

                        if (!error) {

                            var table = document.getElementById("accessDoc");

                            noRows = table.rows.length;

                            var row = table.insertRow(noRows);

                            var cell1 = row.insertCell(0);

                            var cell2 = row.insertCell(1);

                            var cell3 = row.insertCell(2);

                            cell2.className = "publicKeyDoctor";

                            cell1.innerHTML = $("#permitDoctorList").val();

                            cell2.innerHTML = doctorToBeAdded;

                            cell3.innerHTML = '<button onclick="revokeAccess(this)" class="btn btn-danger">Revoke access</button>';

                        } else {

                            $(".alert-info").show();

                            console.log(error);

                        }

                    })

                } else

                    console.log(error);

            })

        }

        function showRecords(element) {

            console.log(recordHash)

            if (toggleRecordsButton % 2 == 0) {

                $.get("http://localhost:8080/ipfs/" + recordHash, function (data) {

                    $("#records").html(data);

                    $("#records").show();

                })

                toggleRecordsButton += 1

                element.innerHTML = "Hide Medical Records";

                element.className = "btn btn-info btn-lg";

            } else {

                $("#records").hide();

                toggleRecordsButton -= 1;

                element.innerHTML = "View Medical Records";

                element.className = "btn btn-info btn-lg"

            }

        }

        function calc() {

            number\_of\_ailments = document.querySelectorAll('input[name="ailment"]:checked').length;

            to\_pay = number\_of\_ailments;

            $("#emi").html(to\_pay);

        }

        function revokeAccess(element) {

            rowNo = element.parentNode.parentNode.rowIndex;

            Row = element.parentNode.parentNode;

            var Cells = Row.getElementsByTagName("td");

            var docKey = Row.cells[1].firstChild.nodeValue;

            contract.methods.revoke\_access(docKey).send({ from: key, gas: 1000000 }, function (error) {

                if (!error) {

                    document.getElementById("accessDoc").deleteRow(rowNo);

                } else {

                    $(".alert-danger").show();

                    console.log(error);

                }

            });

        }

    </script>

</body>

</html>

**Code(Insurer):**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <meta name="description" content="">

    <meta name="author" content="">

    <title>EMR Dapp</title>

    <!-- Bootstrap Core CSS -->

    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- <script src="js/bundle.js"></script> -->

    <!-- Custom CSS -->

    <style>

        .panel {

            margin-bottom: 60px;

        }

        .navbar {

            margin-bottom: 70px;

        }

        .panel-heading {

            margin-bottom: 20px;

        }

        .nav-pills>li>a {

            padding: 0;

            padding-right: 10px;

        }

        .nav-pills>li>a:hover {

            background-color: initial;

        }

        .nav-pills>li.active>a {

            color: #23527c;

            background-color: initial;

        }

        .nav-pills>li.active>a:hover,

        .nav-pills>li.active>a:focus {

            color: #23527c;

            background-color: inherit;

        }

    </style>

    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->

    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

    <!--[if lt IE 9]>

        <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>

        <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>

    <![endif]-->

</head>

<body>

    <!-- Navigation -->

    <nav class="navbar navbar-inverse navbar-static-top" role="navigation">

        <div class="container-fluid">

            <!-- Brand and toggle get grouped for better mobile display -->

            <div class="navbar-header">

                <button type="button" class="navbar-toggle" data-toggle="collapse"

                    data-target="#bs-example-navbar-collapse-1">

                    <span class="sr-only">Toggle navigation</span>

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                    <span class="icon-bar"></span>

                </button>

                <a class="navbar-brand" href="#">EMR Dapp</a>

            </div>

            <!-- Collect the nav links, forms, and other content for toggling -->

            <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

                <ul class="nav navbar-nav navbar-right">

                    <li>

                        <a href="./index.html">Logout</a>

                    </li>

                </ul>

            </div>

            <!-- /.navbar-collapse -->

        </div>

        <!-- /.container -->

    </nav>

    <div class="container">

        <div class="panel panel-default">

            <div class="panel-heading">

                <h3 class="text-center">Insurer's Personal Information</h3>

            </div>

            <div class="panel-body">

                <div class="row">

                    <div class="col-sm-offset-1 col-sm-10">

                        <table class="table">

                            <tr>

                                <th>Name:</th>

                                <td id="name"></td>

                            </tr>

                            <tr>

                                <th>No of customers:</th>

                                <td id="noCustomers"></td>

                            </tr>

                        </table>

                        <!-- <div class="form-group">

                            <div class="text-center">

                                <button type="submit" class="btn btn-info btn-lg">Update details</button>

                            </div>

                        </div> -->

                    </div>

                </div>

            </div>

        </div>

        <div class="panel panel-default">

            <div class="panel-heading">

                <h3 class="text-center">Insurance Claims</h3>

            </div>

            <div class="panel-body">

                <!-- <div class="row">

                    <div class="alert alert-danger col-sm-8 col-sm-offset-2">

                        <strong>Notice!</strong> The access could not be revoked. Please retry or contact admin.

                    </div>

                </div> -->

                <div class="row">

                    <div class="col-sm-offset-1 col-sm-10">

                        <table id="claims" class="table table-hover">

                            <tr>

                                <th>Patient</th>

                                <th class="publicKeyPatient">Public Key</th>

                                <th>View Records</th>

                                <th>Doctor</th>

                                <th>Ailment</th>

                                <th>Action</th>

                            </tr>

                        </table>

                    </div>

                </div>

            </div>

        </div>

    </div>

    <script src="js/jquery.js"></script>

    <!-- Bootstrap Core JavaScript -->

    <script src="js/bootstrap.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/web3@1.3.6/dist/web3.min.js"></script>

    <script src="js/app.js"></script>

    <script>

        displayInsurer()

        var url\_string = window.location.href;

            var url = new URL(url\_string);

            var key = url.searchParams.get("key");

            key = key.toLocaleLowerCase();

            var ailmentsDict = {};

        async function displayInsurer() {

            var contract = await init()

            ailmentsDict[0] = "Cancer";

            ailmentsDict[1] = "Heart Attack";

            ailmentsDict[2] = "Diabitise";

            ailmentsDict[3] = "Corona"

            ailmentsDict[4] = "Toyphoid";

            ailmentsDict[5] = "dengu";

            ailmentsDict[6] = "Tuberclosis";

            ailmentsDict[7] = "Maleria";

            toggleRecordsButton = 0;

            $(document).ready(function () {

                // console.log(key);

                // console.log(contractInstance.get\_patient.call(key, {gas: 1000000}));

                $(".alert-danger").hide();

                var a = 0;

                var b = 0;

                var PatientAddress = 0;

                var DocAddr = 0;

                var Diagnosis = 0;

                var PatientName = [];

                var DocName = [];

                var Patient\_Doctor = 0;

                contract.methods.get\_insurer(key).call({ gas: 1000000 }, function (error, result) {

                    if (!error) {

                        a = result[0];

                        b = result[1];

                        PatientAddress = result[2];

                        DocAddr = result[3];

                        Diagnosis = result[4];

                        // Patient\_Doctor = [PatientAddress,DocAddr];

                        console.log(Diagnosis);

                        $("#name").html(a);

                        $("#noCustomers").html(b);

                        var table = document.getElementById("claims");

                        for (var index = 0; index < Diagnosis.length; index++) {

                            var row = table.insertRow(index + 1);

                            var cell1 = row.insertCell(0);

                            var cell2 = row.insertCell(1);

                            var cell3 = row.insertCell(2);

                            // var row = table.rows[index+1];

                            var cell4 = row.insertCell(3);

                            var cell5 = row.insertCell(4);

                            var cell6 = row.insertCell(5);

                            cell5.innerHTML = ailmentsDict[Diagnosis[index]];

                            cell6.innerHTML = '<button onclick="releaseClaim(this)" class="btn btn-success">Release Claim</button>';

                        }

                        PatientAddress.forEach(function (listItem, index) {

                            contract.methods.get\_patient(listItem).call({ gas: 1000000 }, function (error, result) {

                                // var table = document.getElementById("accessDoc");

                                if (!error) {

                                    PatientName.push(result[0]);

                                    var row = table.rows[index + 1];

                                    var cell1 = row.cells[0];

                                    var cell2 = row.cells[1];

                                    var cell3 = row.cells[2];

                                    cell2.className = "publicKeyPatient";

                                    cell1.innerHTML = result[0];

                                    cell2.innerHTML = listItem;

                                    cell3.innerHTML = '<button onclick="viewRecords(this)" class="btn btn-info">View Records</button>';

                                }

                                else

                                    console.error(error);

                            });

                        });

                        DocAddr.forEach(function (listItem, index) {

                            contract.methods.get\_doctor(listItem).call({ gas: 1000000 }, function (error, result) {

                                // var table = document.getElementById("accessDoc");

                                if (!error) {

                                    DocName.push(result[0]);

                                    var row = table.rows[index + 1];

                                    var cell4 = row.cells[3];

                                    cell4.innerHTML = result[0];

                                }

                                else

                                    console.error(error);

                            });

                        });

                    }

                    else

                        console.error(error);

                });

            });

        }

        function releaseClaim(element) {

            rNo = element.parentNode.parentNode.rowIndex;

            var row = document.getElementById("claims").rows[rNo];

            var patientAddr = row.cells[1].innerHTML;

            var viewRecordsButton = row.cells[2];

            console.log(viewRecordsButton);

            contract.methods.accept\_claim(patientAddr).send({ from: key, gas: 1000000, value: Web3.utils.toWei('4', 'ether') }, function (error, result) {

                if (!error) {

                    if (toggleRecordsButton % 2 != 0) {

                        console.log("boo");

                        $(viewRecordsButton).click();

                    }

                    document.getElementById("claims").deleteRow(rNo);

                }

                else {

                    // $(".alert-danger").show();

                    console.log(error);

                }

            });

        }

        function viewRecords(element) {

            var table = document.getElementById("claims");

            var index = element.parentNode.parentNode.rowIndex;

            var patientAddress = table.rows[index].cells[1].innerHTML;

            if (toggleRecordsButton % 2 == 0) {

                patientRecord = "";

                contract.methods.get\_hash(patientAddress).call({ gas: 1000000 }, function (error, result) {

                    if (!error) {

                        $.get("http://localhost:8080/ipfs/" + result, function (data) {

                            patientRecord = data;

                            var row1 = table.insertRow(index + 1);

                            var cell1 = row1.insertCell(0);

                            cell1.colSpan = 6;

                            cell1.innerHTML = "<pre>" + patientRecord + "</pre";

                        })

                    } else {

                        console.log(error);

                    }

                })

                toggleRecordsButton += 1;

                element.innerHTML = "Hide Records";

                element.className = "btn btn-danger";

            } else {

                row = table.rows[index + 1];

                $(row).hide();

                toggleRecordsButton -= 1;

                element.innerHTML = "View Records";

                element.className = "btn btn-info"

            }

        }

    </script>

</body>

</html>

**Code(Logic):**

pragma solidity ^0.5.16;

contract Agent {

struct patient {

string name;

uint age;

address[] doctorAccessList;

uint[] diagnosis;

string record;

}

struct doctor {

string name;

uint age;

address[] patientAccessList;

}

struct insurer {

string name;

uint count\_of\_patient;

address[] PatientWhoClaimed;

address[] DocName;

uint[] diagnosis;

}

uint creditPool;

address[] public patientList;

address[] public doctorList;

address[] public insurerList;

mapping (address => patient) patientInfo;

mapping (address => doctor) doctorInfo;

mapping (address => insurer) insurerInfo;

mapping (address => address) Patient\_Insurer;

// might not be necessary

mapping (address => string) patientRecords;

//Creating new users

function add\_agent(string memory \_name, uint \_age, uint \_designation, string memory \_hash) public {

address addr = msg.sender;

if(\_designation == 0){

patientInfo[addr].name = \_name;

patientInfo[addr].age = \_age;

patientInfo[addr].record = \_hash;

patientList.push(addr)-1;

}

else if (\_designation == 1){

doctorInfo[addr].name = \_name;

doctorInfo[addr].age = \_age;

doctorList.push(addr)-1;

}

else if(\_designation == 2){

insurerInfo[addr].name = \_name;

insurerList.push(addr)-1;

}

else{

revert();

}

}

//Getting user data

function get\_patient(address addr) view public returns (string memory, uint, uint[] memory, address, string memory, address, address){

return (patientInfo[addr].name, patientInfo[addr].age, patientInfo[addr].diagnosis, Patient\_Insurer[addr], patientInfo[addr].record, msg.sender, addr);

}

function get\_doctor(address addr) view public returns (string memory, uint){

return (doctorInfo[addr].name, doctorInfo[addr].age);

}

function get\_patient\_doctor\_name(address paddr, address daddr) view public returns (string memory, string memory){

return (patientInfo[paddr].name,doctorInfo[daddr].name);

}

function get\_insurer(address addr) view public returns (string memory, uint, address[] memory, address[] memory, uint[] memory ){

return (insurerInfo[addr].name, insurerInfo[addr].count\_of\_patient, insurerInfo[addr].PatientWhoClaimed,

insurerInfo[addr].DocName, insurerInfo[addr].diagnosis);

}

function permit\_access(address addr) payable public {

require(msg.value == 2 ether);

creditPool += 2;

doctorInfo[addr].patientAccessList.push(msg.sender)-1;

patientInfo[msg.sender].doctorAccessList.push(addr)-1;

}

function select\_insurer(address payable iaddr, uint[] memory \_diagnosis) payable public {

uint total\_amount = (\_diagnosis.length);

require(msg.value == total\_amount\*(1 ether));

// require(msg.sender.balance >= msg.value);

iaddr.transfer(msg.value);

Patient\_Insurer[msg.sender] = iaddr;

patientInfo[msg.sender].diagnosis = \_diagnosis;

insurerInfo[iaddr].count\_of\_patient++;

}

//must be called by doctor

function insurance\_claim(address paddr, uint \_diagnosis, string memory \_hash) public {

bool patientFound = false;

for(uint i = 0;i<doctorInfo[msg.sender].patientAccessList.length;i++){

if(doctorInfo[msg.sender].patientAccessList[i]==paddr){

msg.sender.transfer(2 ether);

creditPool -= 2;

patientFound = true;

}

}

if(patientFound==true){

set\_hash(paddr, \_hash);

remove\_patient(paddr, msg.sender);

}else {

revert();

}

bool DiagnosisFound = false;

for(uint j = 0; j < patientInfo[paddr].diagnosis.length;j++){

if(patientInfo[paddr].diagnosis[j] == \_diagnosis)DiagnosisFound = true;

}

if(DiagnosisFound){

insurerInfo[Patient\_Insurer[paddr]].PatientWhoClaimed.push(paddr)-1;

insurerInfo[Patient\_Insurer[paddr]].DocName.push(msg.sender)-1;

insurerInfo[Patient\_Insurer[paddr]].diagnosis.push(\_diagnosis)-1;

}

}

//must be called by insurer

function accept\_claim(address payable paddr) public payable {

// require(msg.sender.balance >= msg.value);

require(msg.value == 4 ether);

paddr.transfer(msg.value);

uint index;

remove\_element\_in\_array(insurerInfo[msg.sender].PatientWhoClaimed,paddr);

if(insurerInfo[msg.sender].diagnosis.length == 1){

delete insurerInfo[msg.sender].DocName[index];

}

else {

insurerInfo[msg.sender].DocName[index] = insurerInfo[msg.sender].DocName[insurerInfo[msg.sender].DocName.length - 1];

delete insurerInfo[msg.sender].DocName[insurerInfo[msg.sender].DocName.length - 1];

}

insurerInfo[msg.sender].DocName.length--;

if(insurerInfo[msg.sender].diagnosis.length == 1){

delete insurerInfo[msg.sender].diagnosis[index];

}

else {

insurerInfo[msg.sender].diagnosis[index] = insurerInfo[msg.sender].diagnosis[insurerInfo[msg.sender].diagnosis.length - 1];

delete insurerInfo[msg.sender].diagnosis[insurerInfo[msg.sender].diagnosis.length - 1];

}

insurerInfo[msg.sender].diagnosis.length--;

}

function remove\_element\_in\_array(address[] storage Array, address addr) internal returns(uint)

{

bool check = false;

uint del\_index = 0;

for(uint i = 0; i<Array.length; i++){

if(Array[i] == addr){

check = true;

del\_index = i;

}

}

if(!check) revert();

else{

if(Array.length == 1){

delete Array[del\_index];

}

else {

Array[del\_index] = Array[Array.length - 1];

delete Array[Array.length - 1];

}

Array.length--;

}

}

function remove\_patient(address paddr, address daddr) public {

remove\_element\_in\_array(doctorInfo[daddr].patientAccessList, paddr);

remove\_element\_in\_array(patientInfo[paddr].doctorAccessList, daddr);

}

function get\_accessed\_doctorlist\_for\_patient(address addr) public view returns (address[] memory)

{

address[] storage doctoraddr = patientInfo[addr].doctorAccessList;

return doctoraddr;

}

function get\_accessed\_patientlist\_for\_doctor(address addr) public view returns (address[] memory)

{

return doctorInfo[addr].patientAccessList;

}

function revoke\_access(address daddr) public payable{

remove\_patient(msg.sender,daddr);

msg.sender.transfer(2 ether);

creditPool -= 2;

}

function get\_patient\_list() public view returns(address[] memory){

return patientList;

}

function get\_doctor\_list() public view returns(address[] memory){

return doctorList;

}

function get\_insurer\_list() public view returns(address[] memory){

return insurerList;

}

function get\_hash(address paddr) public view returns(string memory){

return patientInfo[paddr].record;

}

function set\_hash(address paddr, string memory \_hash) internal {

patientInfo[paddr].record = \_hash;

}

}

**RESULTS**

**PERFORMANCE ASSESMENT**:

In order to understand how our proposed framework would perform in real-case scenario of various users performing different functions on the framework we conducted performance evaluation using Apache JMeter version 5.1.1 and Apache Version 2.00. Apache JMeter is a desktop performance testing tool which is used for analysis and testing of applications .

1. Average Execution Time: The execution time increases with the number of transactions being increased. These transactions are performed for the various functions that are included in the smart contract whose algorithm is defined in Section V. When there is only one user using the system the functions Assign Roles, Add Patient Records and View Patient Records would take 18.29 sec, 1 min 48 sec and 50 sec respectively for these functions to be executed. This time would increase when 100 users are using the system simultaneously.
2. Throughput: Algorithm 1 explains various functions that are included in the smart contract of the proposed framework. By using JMeter we simulated number of users from 100 users to 500 users (with period of 10 to 35), who are using the system and performing its various functions. In JMeter the throughput is represented in Data/time i.e. KB/sec units. While conducting the experiments we simulated the number of users as specified above and evaluated the Before assessing the transaction size we also analyze the transaction payload. This assessment is discussed in detail in the following section.

**PERFORMANCE EVALUATION (TRANSACTION):**

Every transaction on Ethereum contains a data payload field. Data payload is included in that transaction which is meant to invoke smart contract functions. This data payload is in the hex-serialized format and has bytes associated with it. Here we would discuss two functions from Algorithm 1 in order to understand the data payload included in the transactions being generated. Data payload is the optional field of a transaction which is only used when there is some form of interaction with contract functions. It has two important parts, Function Selector Function Arguments The function selector are first 4 bytes of Keccak-256 hash, it is used for identification of the smart contract function which is being invoked. The function arguments include various static and dynamic element types which have different rules for encoding them in payload. Let us now understand the payload of Define Roles function from Algorithm 1 to get an understanding as to how the data payload is generated. Firstly we would separate the function selector and arguments. The function selector is actually the function signature which in this case is:

**COMPARISON OF PROPOSED FRAMEWORK WITH RELATED WORK:**

We also discuss some parameters that are present in our framework and are used for comparison with the related work in this domain. While ensuring the presence of these parameters in the framework it is also considered that it would not compromise the security and privacy of the system. For this both security and privacy are discussed in each of the parameters discusses below.

1. **SCALABILITY:** Scalability in simpler terms refers to the ability of an information system to perform it functions well in such situations when the storage volume of the system increases or decreases. In case of blockchain technology scalability is an issue that needs some permanent solution. As data size or volume is increasing on the blockchain. Our proposed system used the off-chain storage mechanism as the patient’s data stored on the blockchain contains the basic information of patient along with the IPFS hash, i.e., the off-chain scaling solution used in our proposed system framework. This solves the scalability issue mentioned as now huge volume of patient medical record is not stored on the blockchain. As, the data size being stored on the blockchain has now decreased the transactions could also be performed faster. As mentioned earlier, IPFS uses cryptographic hash which is stored in the decentralized manner using peer-to-peer network. This also ensures that while solving the scalability problem the security of the framework is not compromised
2. **INTEGRITY** : Integrity of a system is measured by the trustfulness of that system and also that system storing that information is temper-proof and reliable. This blockchain-based system ensures that it does not compromise this feature. The information stored in this system is intact and is not changed by any unauthorized channel. Moreover, information is available to only the associated parties that are doctors and patients. The users of the system and any third party do not have the right to make any changes in the smart contract as they are not having any access to it. This is done by using the access rules which ensure that the private data or medical records of patients are not accessible and remain temper-proof. Moreover, using IPFS for storage of records also ensures the security of the medical records of the patients.
3. **ACCESS CONTROL:** Using the Role-based access mechanism, this framework makes sure that every entity of the system is assigned a role. Any third party who is not authorized to have access to the system would not be able to access the system. This system provides a two core security as firstly blockchain technology in itself is secure and uses certain protocols and mechanism to keep itself secure from third-part intrusions. And secondly our system uses the Role-based access that also only allows the users having defined roles to have access to the system and its functions. So, our system would not only ensure security of patient records but would also make sure the access control of entities associated with it. This parameter also ensures that the security of the patient’s personal medical data is not compromised and the access is provided to only the authorized users of the system.
4. INFORMATION CONFIDENTIALITY The patient medical records stored on the blockchain should be secured from any third party access to ensure the confidentiality of the patients‟ record. The patient’s data include the important information of patient such as the patient medical history, blood group, records, lab results, X-rays reports, MRI results and many other related results and reports. All of this information is critical not only to patients but also to the hospital. Smart contracts are a really helpful element in this system as they ensure transparency, precision and trust on the transactions being performed. The record being stored and accessed in the system are only accessible by the trusted parties. Any untrusted third party trying to access the system is denied access by the system. With the information being kept as confidential from third party access the framework would ensure that it would the aspect of privacy as well

**CONCLUSION**

Nowadays, with the increasing number of medical services providers and consequently the number of appointments, with the lack of interoperability and the lack of patient centricity that is adjacent to it, comes the need to find a solution with a secure and viable way to store healthcare data and process transactions in this sector. This solution must be capable of mitigating these difficulties and improve usability for the patients as well for the professionals. This is where blockchain comes in. Being a technology that ensures immutable and secure transactions, it becomes a possible resolution for the health sectors. Considering these two major factors, after analysing the state of the art of blockchain technology and of smart contracts, this work presents a healthcare application blockchain-based – MedBlock – as a solution to mitigate such problems. To summarize, this research combines the blockchain features that are beneficial for healthcare with the smart contracts strengths, within the context of the MedClick. This result in a healthcare application based on blockchain and smart contracts that gives patients the possibility to save their health data plus interact with all the health providers and professionals they choose to, in one single platform. Besides, being implemented in Hyperledger Fabric, a blockchain platform that has a modular architecture, allows this work to take advantage of the blockchain strengths in a business development context. Currently the authors are finishing the implementation of MedBlock and the evaluation of results is planned.

In the EHR system the patient can access their report and can use the report for their lifetime with security. The private key is used for the patient which can be used for the further use of the reports. The one who don’t have the private key cannot involve in the process of retrieving data. Hence the Health Records of the patients are more secured with the BlockChain and can used with their own private key as well as they can make use of that for further reference. Currently, there is a huge obstacle for using ethereum platform, which lays in difficulty of obtaining ETH units. The units can be obtained either by mining or purchased for fiat or cryptocurrency like Bitcoin. Once ETH units are available, the publishing and execution of smart contract is really smooth. At this stage, the solidity code has been deployed and verified. The deployed contract is accessed with the