

## project Development Phase

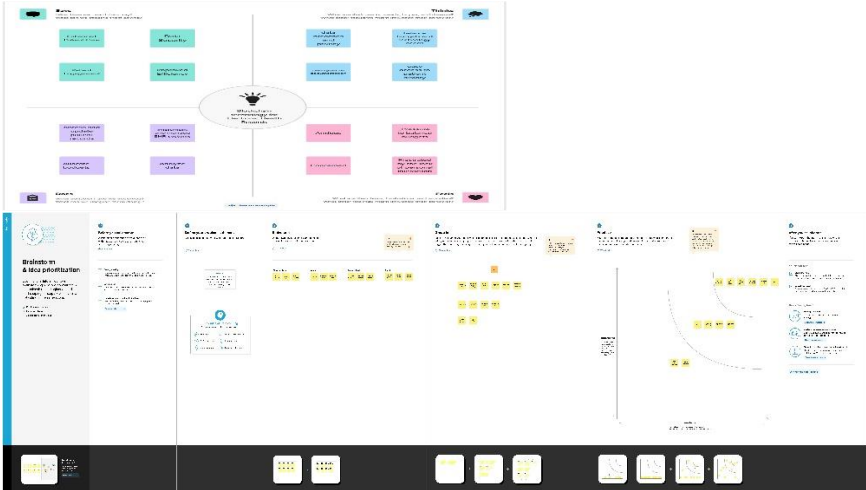
### Model Performance Test

Date	25-10-2023
Team ID	NM2023TMID00147
Project Name	Blockchain Technology for Electronic Health Records
Maximum Marks	10 Marks

### Model Performance Testing:

Project team shall fill the following information when working for blockchain.

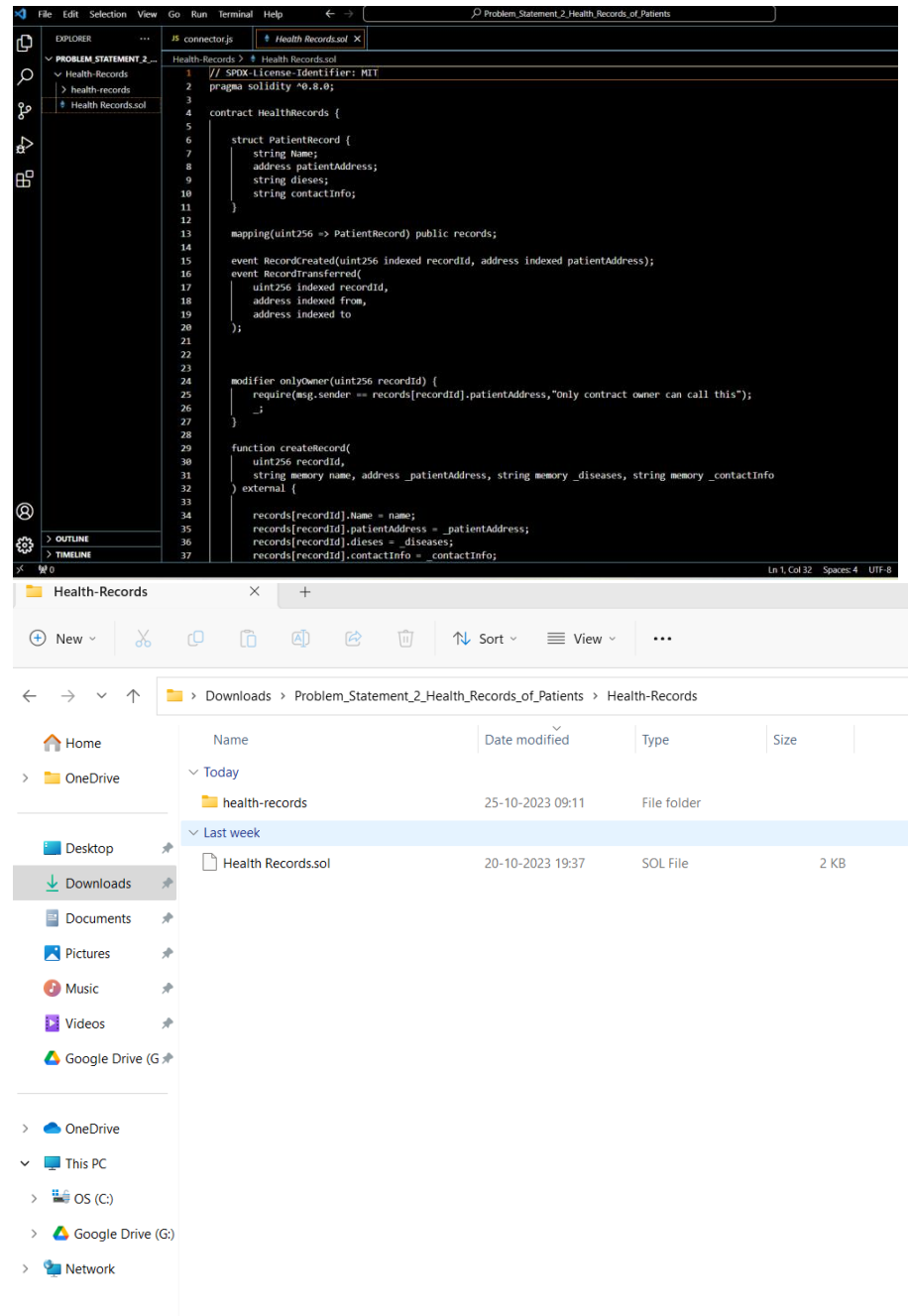
S.No.	Parameter	Values	Screenshot
-------	-----------	--------	------------

1.	Information gathering	Setup all the Prerequisite:	 <p>The screenshot displays a software interface with three main sections. The top section features a hierarchical diagram with a central node labeled 'Information Systems' and several sub-nodes in colored boxes (green, blue, purple, pink). The bottom-left section contains a table with columns for 'System', 'Version', 'Status', and 'Date'. The bottom-right section shows a flowchart with nodes and connecting lines. The interface has a dark blue header and a sidebar on the left.</p>
----	-----------------------	-----------------------------	---

2.

Extract the  
zip files

Open to vs  
code



3.

Remix Ide  
platform  
  
exploring

Deploy the  
smart contract  
code

Deploy and  
run the  
transaction.  
By selecting  
the  
environment -  
inject the  
MetaMask.

The image shows two windows side-by-side. The top window is the Remix IDE, displaying a Solidity smart contract named 'HealthRecords.sol'. The code defines a contract 'HealthRecords' with a 'PatientRecord' struct, a 'records' array, and a 'createRecord' function. The bottom window is a Windows File Explorer showing the 'Downloads' folder. It lists the 'HealthRecords.sol' file, which is a 2 KB SOL File, created on 20-10-2023 at 19:37.

**Remix IDE - HealthRecords.sol**

```

1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract HealthRecords {
5
6     struct PatientRecord {
7         string Name;
8         address patientAddress;
9         string diseases;
10        string contactInfo;
11    }
12
13    mapping(uint256 => PatientRecord) public records;
14
15    event RecordCreated(uint256 indexed recordId, address indexed patientAddress);
16    event RecordTransferred(
17        uint256 indexed recordId,
18        address indexed from,
19        address indexed to
20    );
21
22
23    modifier onlyOwner(uint256 recordId) {
24        require(msg.sender == records[recordId].patientAddress, "only contract owner can call this");
25        _;
26    }
27
28    function createRecord(
29        uint256 recordId,
30        string memory name, address _patientAddress, string memory _diseases, string memory _contactInfo
31    ) external {
32
33        records[recordId].Name = name;
34        records[recordId].patientAddress = _patientAddress;
35        records[recordId].diseases = _diseases;
36        records[recordId].contactInfo = _contactInfo;
37    }
38

```

**Windows File Explorer - Downloads**

Name	Date modified	Type	Size
Today			
health-records	25-10-2023 09:11	File folder	
Last week			
Health Records.sol	20-10-2023 19:37	SOL File	2 KB

SOLIDITY COMPILER

COMPILER +

0.8.18+commit.87161d96

Include nightly builds

Auto compile

Hide warnings

Advanced Configurations

Compile electronichealthreco...

Compile and Run script

CONTRACT

HealthRecords (electronichealthreco:

Publish on ipfs

Publish on Swarm

Compilation Details

ABI

Bytecode

1

// SPDX-License-Identifier: MIT

2

pragma solidity ^0.8.0;

3

contract HealthRecords {

4

5

6

struct PatientRecord {

7

string Name;

8

address patientAddress;

9

string diagnoses;

10

string contactInfo;

11

}

12

mapping(uint256 => PatientRecord) public records;

13

14

15

event RecordCreated(uint256 indexed recordId, address indexed patientAddress);

16

event RecordTransferred(

17

uint256 indexed recordId,

18

address indexed from,

19

address indexed to

20

);

21

22

23

24

25

26

27

28

modifier onlyOwner(uint256 recordId) {

require(msg.sender == records[recordId].patientAddress, "only contract owner can call this");

};

0

listen on all transactions

Search with transaction hash or address

Welcome to Remix 0.36.3

Your files are stored in IndexedDB, 1.5 MB / 275.67 GB used

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

Injected Provider - MetaMask

Custom (Rinkeby) network

ACCOUNT

0x09D...c141B (0.49258411)

GAS LIMIT

3000000

VALUE

0

Wei

CONTRACT

HealthRecords - electronichealthreco:

even renders paths

Deploy

Publish to IPFS

As Address

Load contract from Address

Transactions recorded

Deployed Contracts

Currently you have no contract instances to interact with.

1

// SPDX-License-Identifier: MIT

2

pragma solidity ^0.8.0;

3

contract HealthRecords {

4

5

6

struct PatientRecord {

7

string Name;

8

address patientAddress;

9

string diagnoses;

10

string contactInfo;

11

}

12

mapping(uint256 => PatientRecord) public records;

13

14

15

event RecordCreated(uint256 indexed recordId, address indexed patientAddress);

16

event RecordTransferred(

17

uint256 indexed recordId,

18

address indexed from,

19

address indexed to

20

);

21

22

23

24

25

26

27

28

modifier onlyOwner(uint256 recordId) {

require(msg.sender == records[recordId].patientAddress, "only contract owner can call this");

};

0

listen on all transactions

Search with transaction hash or address

rinkeby-12

remix

Type the library name to see available commands.

remix.ethereum.org/#lang=ens&optimizer=false&runs=200&evmVersion=null&version=solcom-v0.8.18+commit.87161d96.js

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

Injected Provider - MetaMask

Custom (Rinkeby) network

ACCOUNT

0x09D...c141B (0.49258411)

GAS LIMIT

3000000

VALUE

0

Wei

CONTRACT

HealthRecords - electronichealthreco:

even renders paths

Deploy

Publish to IPFS

As Address

Load contract from Address

Transactions recorded

Deployed Contracts

Currently you have no contract instances to interact with.

1

// SPDX-License-Identifier: MIT

2

pragma solidity ^0.8.0;

3

contract HealthRecords {

4

5

6

struct PatientRecord {

7

string Name;

8

address patientAddress;

9

string diagnoses;

10

string contactInfo;

11

}

12

mapping(uint256 => PatientRecord) public records;

13

14

15

event RecordCreated(uint256 indexed recordId, address indexed patientAddress);

16

event RecordTransferred(

17

uint256 indexed recordId,

18

address indexed from,

19

address indexed to

20

);

21

22

23

24

25

26

27

28

modifier onlyOwner(uint256 recordId) {

require(msg.sender == records[recordId].patientAddress, "only contract owner can call this");

};

0

listen on all transactions

Search with transaction hash or address

rinkeby-12

remix

Type the library name to see available commands.

creation of HealthRecords pending...

Account 1

New contract

https://remix.ethereum.org

CONTRACT DEPLOYMENT

\$0.00

DETAILS

DATA

Site suggested >

Gas (estimated)

\$0.00

Very likely in < 15 seconds

Max fee: 0.0024796 MATIC

Total

\$0.00

Amount + gas fee

Max amount: 0.0024796 MATIC

Reject

Confirm

HealthRecords - electronichealthrecon<

view version path

Deploy

Publish to IPFS

All Address

Load verified from Address

Transactions recorded

Deployed Contracts

HEALTHRECORDS AT 0x7E7...4E7C

Welcome to Remix 0.36.3

Your files are stored in indexedDB, 1.35 MB / 275.57 GB used

You can use this terminal to:

• Check transactions details and start debugging.

• Execute JavaScript scripts

• Launch a script directly in the command line interface

• Select a JavaScript file in the file explorer and then run 'remix.execute()' or 'remix.execute()' in the command line interface

• Right click on a JavaScript file in the file explorer and then click 'Run'

The following libraries are accessible:

• web3 version 1.2.3

• ethers.js

• remix

Type the library name to see available commands.

creation of HealthRecords pending...

[black:41617226 txIndex:1] #from: 0x090...c1418 to: HealthRecords.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0x608...a5daf

Debug

EXPLORER

▼ PROBLEM\_STATEMENT\_2\_...

▼ Health-Records

▼ health-records

▼ node\_modules

▼ public

▼ src

▼ Page

JS connector.js

JS Home.js

# App.css

JS App.js

JS App.test.js

# index.css

JS index.js

logo.svg

JS reportWebVitals.js

JS setupTests.js

.gitignore

package-lock.json

package.json

README.md

JS connector.js X

Health Records.sol

Health-Records > health-records > src > Page > JS connector.js > ...

180 ],

181 "name": "transferRecord",

182 "outputs": [],

183 "stateMutability": "nonpayable",

184 "type": "function"

185 }

186 }

187

188 if (!window.ethereum) {

189 alert('Meta Mask Not Found')

190 window.open("https://metamask.io/download/")

191 }

192

193 export const provider = new ethers.providers.Web3Provider(window.ethereum);

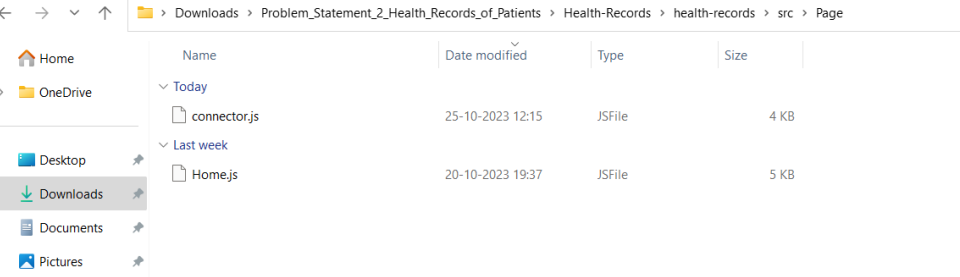
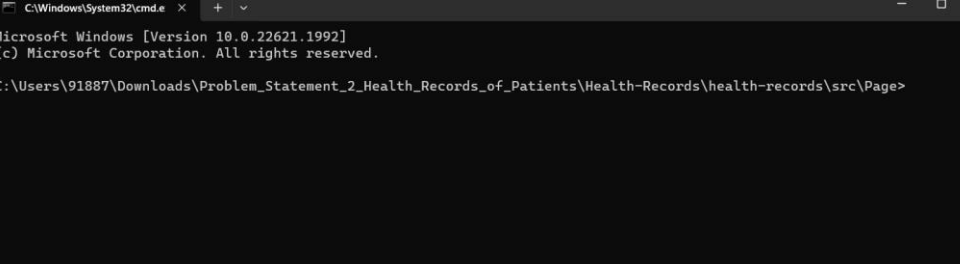
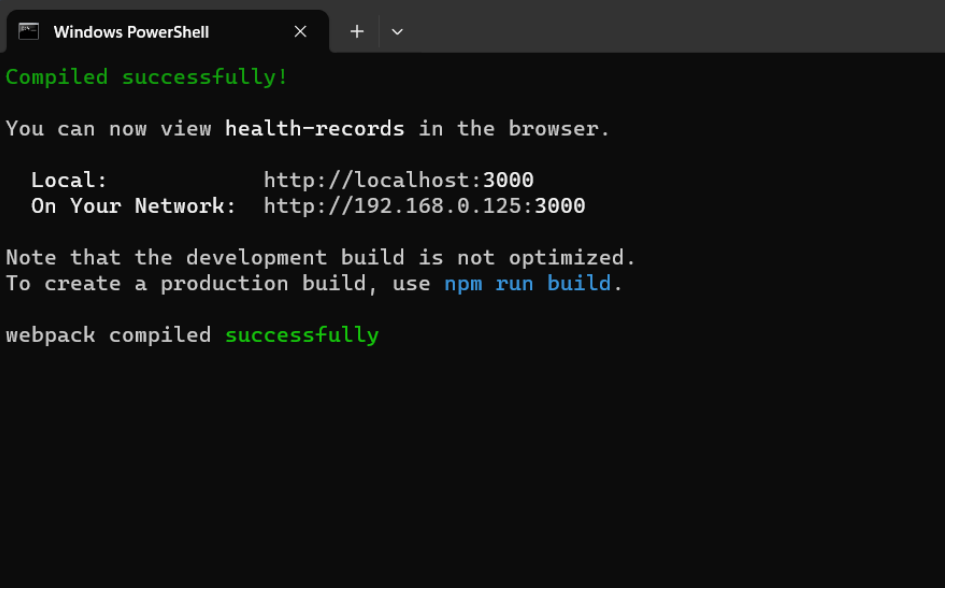
194 export const signer = provider.getSigner();

195 export const address = "0x7E7194411c2E82892C620496f57BC2F04d94e7cd"

196

197 export const contract = new ethers.Contract(address, abi, signer)

198

4	Open file explorer	<p>Open the extracted file and click on the folder.</p> <p>Open src, and search for utiles.</p> <p>Open cmd enter commands</p> <p>1.npm install</p> <p>2.npm bootstrap</p> <p>3. npm start</p>	  
---	--------------------	--	---

5			<div><div><div>Health Records Using Blockchain</div><div>0x09d7....4c1418</div><div><div><div>Enter Record Id</div><div>Enter name</div><div>Enter patient Address</div><div>Enter disease</div><div>Enter contact Info</div><div>Create Record</div></div><div><div>Enter new record Id</div><div>Enter new owner metama</div><div>Transfer Record</div></div></div></div></div> <div><div>Health Records Using Blockchain</div><div><div>Connect Wallet</div><div><div><div>Enter Record Id</div><div>Enter name</div><div>Enter patient Address</div><div>Enter disease</div><div>Enter contact Info</div><div>Create Record</div></div><div><div>Enter new record Id</div><div>Enter new owner metama</div><div>Transfer Record</div></div></div></div><div><div>Enter Id</div><div>Get Record Data</div></div></div>
---	--	--	---

{LOCALHOSTIP  
ADDRESS

copy the  
address and  
open it to  
chrome so you  
can see the  
front end of  
your project.