External Practical Exam

Aim:

Develop a secure and decentralized land registration smart contract that allows:

- The government authority (land registrar) to register land properties.
- Owners to transfer land ownership to other individuals securely.
- LandRegistrar: The government authority that registers land and approves transfers.
- Owner (User): Person who owns a piece of land and can transfer it.
- Buyer/New Owner: A valid Ethereum address to whom land is transferred.

Functional requirements:

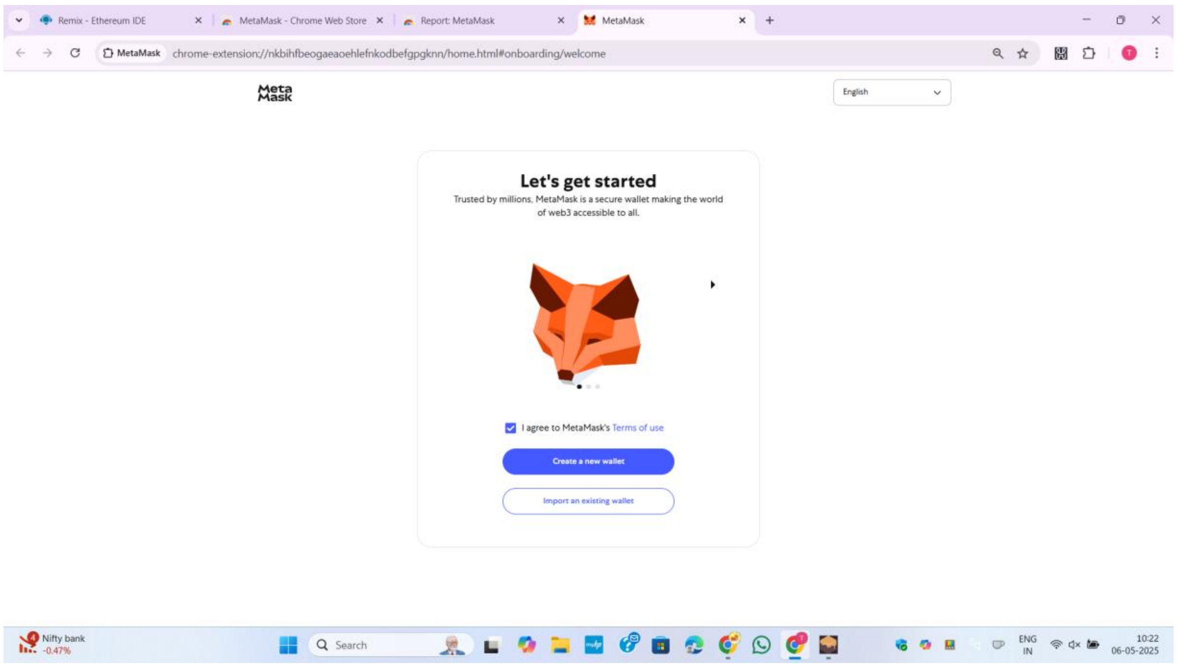
- Only the LandRegistrar (contract deployer) can register a new land property.
- Land properties must include: Land ID, Location (string), Current Owner, Area (in square meters),
- Owner of a land can initiate a transfer to another address.
- LandRegistrar must approve the transfer to finalize ownership change.
- System must prevent unauthorized land registration or transfer.

Code : (LandRegistry.sol)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract LandRegistry {
    address public landRegistrar;

    event LandRegistered(uint256 landId, string location, address indexed
owner, uint256 area);
    event TransferInitiated(uint256 landId, address indexed newOwner);
    event TransferApproved(uint256 landId, address indexed newOwner);

    constructor() {
        landRegistrar = msg.sender;
    }

    struct Land {
        uint256 landId;
        string location;
        address currentOwner;
        uint256 area;
        bool exists;
    }
    struct PendingTransfer {
        address newOwner;
        bool exists;
    }

    mapping(uint256 => Land) public lands;
    mapping(uint256 => PendingTransfer) public pendingTransfers;

    modifier onlyRegistrar() {
        require(msg.sender == landRegistrar, "Only registrar can perform
this action");
        _;
    }

    modifier onlyOwner(uint256 landId) {
        require(lands[landId].exists, "Land not found");
        require(msg.sender == lands[landId].currentOwner, "Only the current
owner can perform this action");
        _;
    }

    function registerLand(
        uint256 landId,
        string memory location,
        address owner,
        uint256 area
    ) public onlyRegistrar {
        require(!lands[landId].exists, "Land already registered");

        lands[landId] = Land({
            landId: landId,
            location: location,
            currentOwner: owner,
            area: area,
            exists: true
        });

        emit LandRegistered(landId, location, owner, area);
    }

    function initiateTransfer(uint256 landId, address newOwner) public
onlyOwner(landId) {
        require(newOwner != address(0), "Invalid address");
        pendingTransfers[landId] = PendingTransfer({newOwner: newOwner,
exists: true});
```

```
        emit TransferInitiated(landId, newOwner);
    }

    function approveTransfer(uint256 landId) public onlyRegistrar {
        require(pendingTransfers[landId].exists, "No transfer initiated");
        lands[landId].currentOwner = pendingTransfers[landId].newOwner;
        delete pendingTransfers[landId];
        emit TransferApproved(landId, lands[landId].currentOwner);
    }

    function getLandDetails(uint256 landId) public view returns (string
memory location, address currentOwner, uint256 area) {
        require(lands[landId].exists, "Land not found");
        Land memory land = lands[landId];
        return (land.location, land.currentOwner, land.area);
    }
}
```

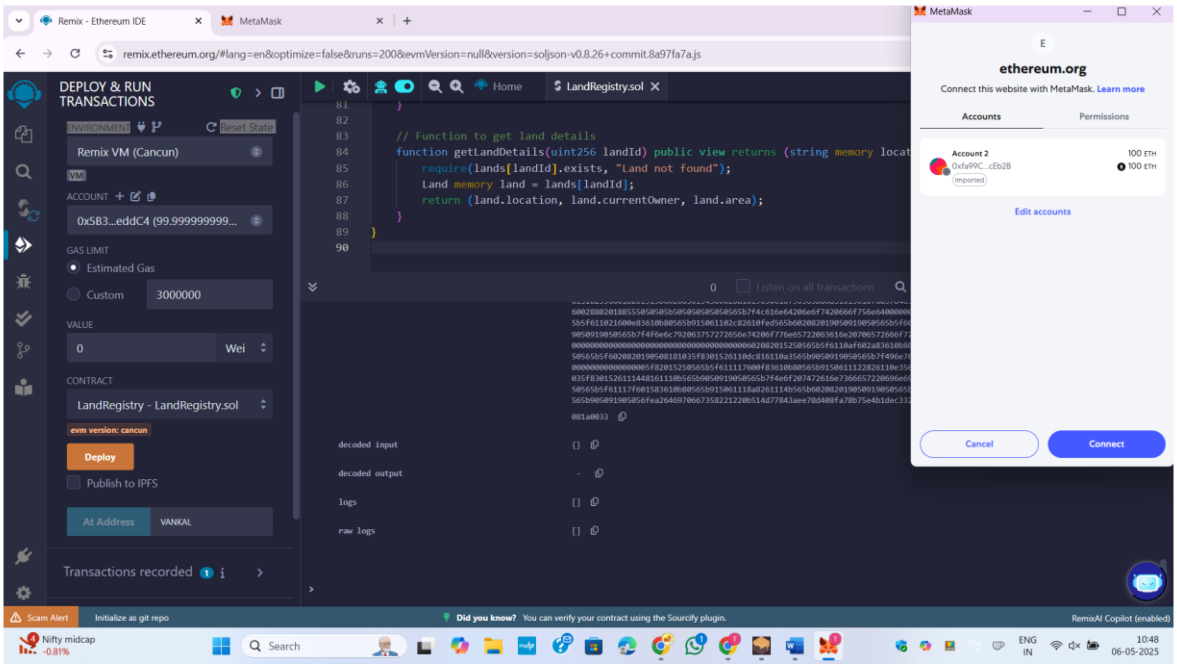
Output:


Let's get started

Trusted by millions, MetaMask is a secure wallet making the world of web3 accessible to all.

I agree to MetaMask's Terms of use
Remix - Ethereum IDE
MetaMask - Chrome Web Store
Report: MetaMask
MetaMask
chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#onboarding/welcome
English
Create a new wallet
Import an existing wallet
Nifty bank -0.47%
Search
ENG IN
10:22 06-05-2025

Figure 1:Install MetaMask



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.26+commit.8a97fa7a.js

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Remix VM (Cancun)

ACCOUNT: 0x5B3...eddC4 (99.999999999...

GAS LIMIT: Estimated Gas; Custom 3000000

VALUE: 0 Wei

CONTRACT: LandRegistry - LandRegistry.sol

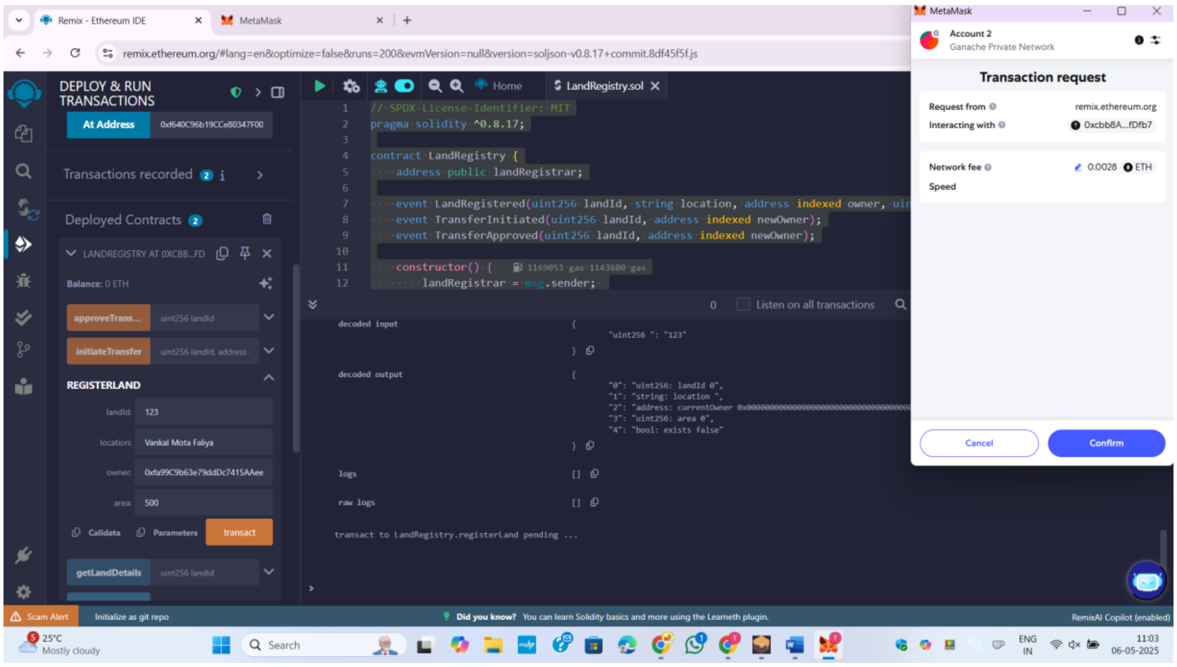
```
    }

    // Function to get land details
    function getLandDetails(uint256 landId) public view returns (string memory locat
        require(lands[landId].exists, "Land not found");
        Land memory land = lands[landId];
        return (land.location, land.currentOwner, land.area);
    }
}
```

MetaMask — ethereum.org

Connect this website with MetaMask.

Account 2 0xfa99C...cEb2B 100 ETH

Figure 2:Connect REMIX IDE with MetaMask



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.17+commit.8df45f5f.js

REGISTERLAND

landId: 123

location: Vankal Mota Faliya

owner: 0xfa99C9b63e79ddDc7415AAee

area: 500

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract LandRegistry {
    address public landRegistrar;

    event LandRegistered(uint256 landId, string location, address indexed owner, uin
    event TransferInitiated(uint256 landId, address indexed newOwner);
    event TransferApproved(uint256 landId, address indexed newOwner);

    constructor() {    1169053 gas 1143600 gas
        landRegistrar = msg.sender;
```

```
decoded input         {
                              "uint256 ": "123"
                      }
decoded output        {
                              "0": "uint256: landId 0",
                              "1": "string: location ",
                              "2": "address: currentOwner 0x000000000000000000000000000000000
                              "3": "uint256: area 0",
                              "4": "bool: exists false"
                      }
logs                  []
raw logs              []

transact to LandRegistry.registerLand pending ...
```

MetaMask — Account 2, Ganache Private Network

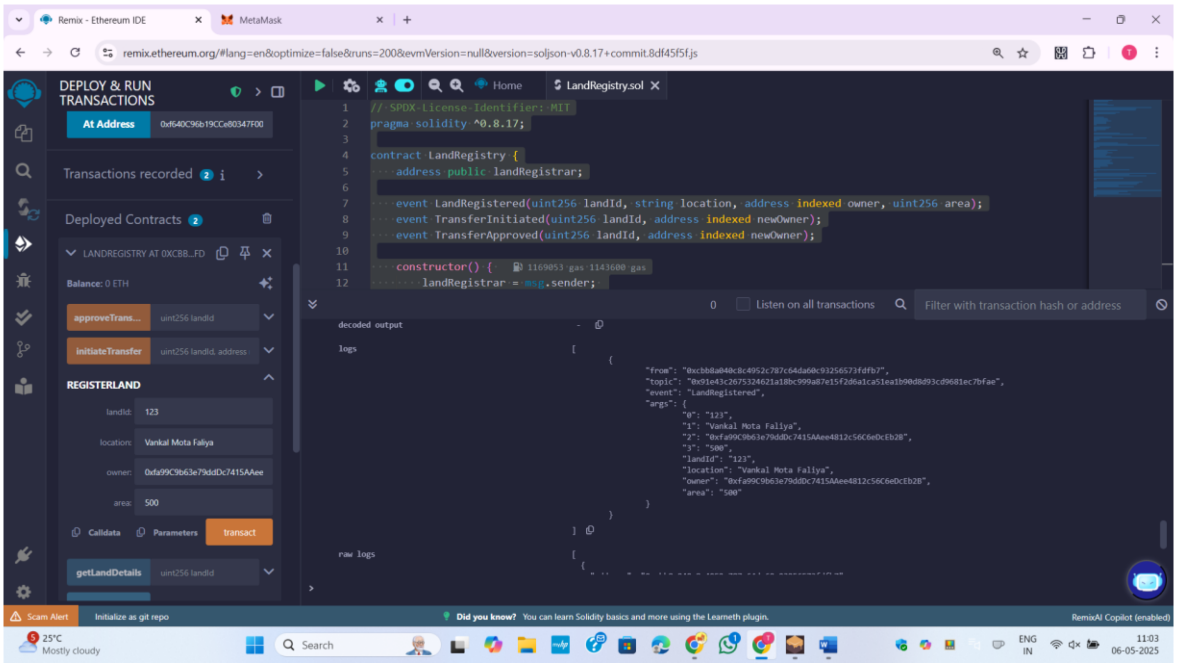
Transaction request

Request from: remix.ethereum.org

Interacting with: 0xcbb8A...fDfb7

Network fee: 0.0028 ETH

Figure 3:Register land with landid = 123



REGISTERLAND

landId: 123

location: Vankal Mota Faliya

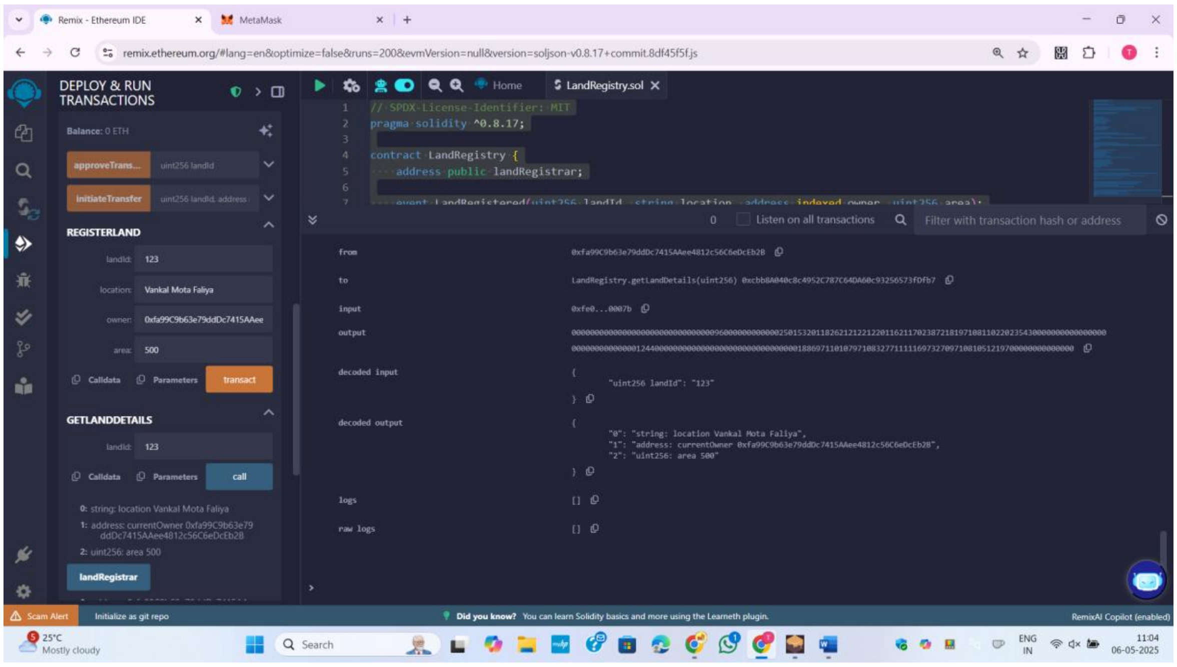
owner: 0xfa99C9b63e79ddDc7415AAee

area: 500

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract LandRegistry {
    address public landRegistrar;

    event LandRegistered(uint256 landId, string location, address indexed owner, uint256 area);
    event TransferInitiated(uint256 landId, address indexed newOwner);
    event TransferApproved(uint256 landId, address indexed newOwner);

    constructor() {    1169053 gas 1143600 gas
        landRegistrar = msg.sender;
```

```
decoded output          -
logs                    [
                            {
                                "from": "0xcbb8a040c8c4952c787c64da60c93256573fdfb7",
                                "topic": "0x91e43c2675324621a18bc999a87e15f2d6a1ca51ea1b90d8d93cd9681ec7bfae",
                                "event": "LandRegistered",
                                "args": {
                                        "0": "123",
                                        "1": "Vankal Mota Faliya",
                                        "2": "0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B",
                                        "3": "500",
                                        "landId": "123",
                                        "location": "Vankal Mota Faliya",
                                        "owner": "0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B",
                                        "area": "500"
                                }
                            }
                        ]
raw logs                [
```

Figure 4:Log of Land Registration



REGISTERLAND

landId: 123

location: Vankal Mota Faliya

owner: 0xfa99C9b63e79ddDc7415AAee

area: 500

GETLANDDETAILS

landId: 123

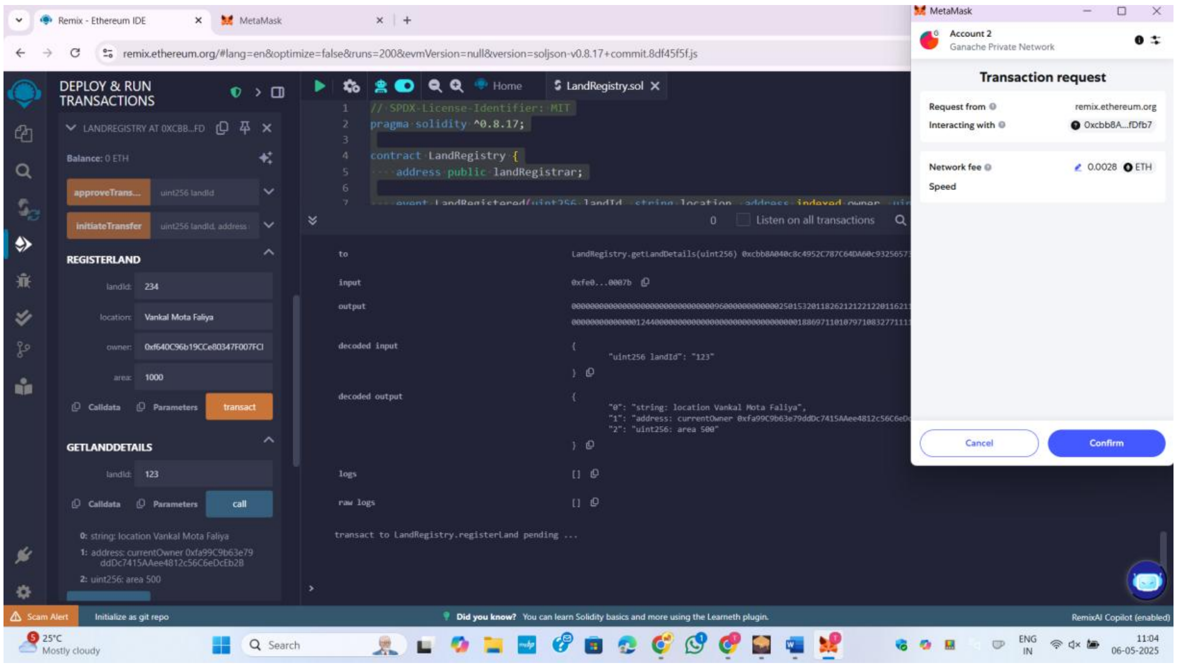
0: string: location Vankal Mota Faliya

1: address: currentOwner 0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B

2: uint256: area 500

```
from            0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B
to              LandRegistry.getLandDetails(uint256) 0xcbb8A040c8c4952C787C64DA60c93256573fDfb7
input           0xfe0...0007b
output          0000000000000000000000000000000960000000000000250153201182621212212201162117023872181971081102202354300000000000000000
                000000000000012440000000000000000000000000000000001886971101079710832771111169732709710810512197000000000000000
decoded input   {
                        "uint256 landId": "123"
                }
decoded output  {
                        "0": "string: location Vankal Mota Faliya",
                        "1": "address: currentOwner 0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B",
                        "2": "uint256: area 500"
                }
logs            []
raw logs        []
```

Figure 5:Fetch Land Details

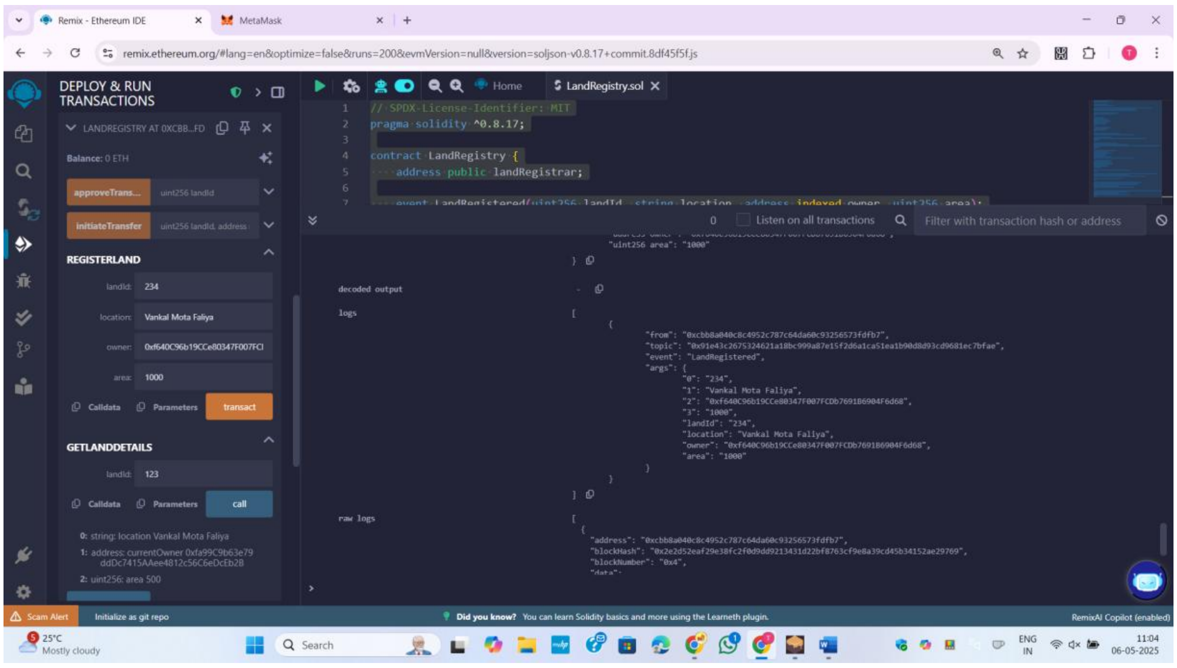


```
REGISTERLAND
landId: 234
location: Vankal Mota Faliya
owner: 0xf640C96b19CCe80347F007FCl
area: 1000

GETLANDDETAILS
landId: 123
0: string: location Vankal Mota Faliya
1: address: currentOwner 0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb28
2: uint256: area 500
```

```
to              LandRegistry.getLandDetails(uint256) 0xcbb8A040c8c4952C787C64DA60c9325657
input           0xfe0...0007b
output          000000000000000000000000000000096000000000000250153201182621212212201162
                000000000000001244000000000000000000000000000000001886971101079710832771111
decoded input   {
                        "uint256 landId": "123"
                }
decoded output  {
                        "0": "string: location Vankal Mota Faliya",
                        "1": "address: currentOwner 0xfa99C9b63e79ddDc7415AAee4812c56C6eD
                        "2": "uint256: area 500"
                }
logs            []
raw logs        []

transact to LandRegistry.registerLand pending ...
```

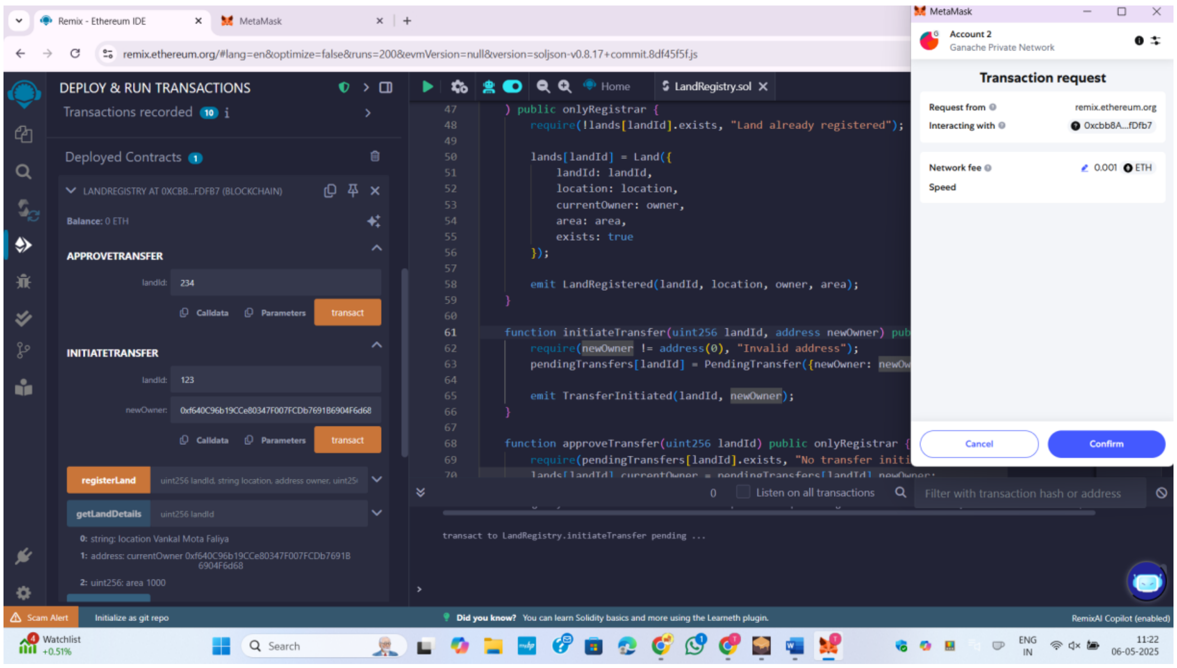
MetaMask — Account 2, Ganache Private Network
Transaction request
Request from: remix.ethereum.org
Interacting with: 0xcbb8A...fDfb7
Network fee: 0.0028 ETH
Speed
Cancel | Confirm

Figure 6:Add new Land Details



```
                        "uint256 area": "1000"
                }
decoded output  -
logs            [
                        {
                                "from": "0xcbb8a040c8c4952c787c64da60c93256573fdfb7",
                                "topic": "0x91e43c2675324621a18bc999a87e15f2d6a1ca51ea1b90d8d93cd9681ec7bfae",
                                "event": "LandRegistered",
                                "args": {
                                        "0": "234",
                                        "1": "Vankal Mota Faliya",
                                        "2": "0xf640C96b19CCe80347F007FCDb769186904F6d68",
                                        "3": "1000",
                                        "landId": "234",
                                        "location": "Vankal Mota Faliya",
                                        "owner": "0xf640C96b19CCe80347F007FCDb769186904F6d68",
                                        "area": "1000"
                                }
                        }
                ]
raw logs        [
                  {
                    "address": "0xcbb8a040c8c4952c787c64da60c93256573fdfb7",
                    "blockHash": "0x2e2d52eaf29e38fc2f0d9dd9213431d22bf8763cf9e8a39cd45b34152ae29769",
                    "blockNumber": "0x4",
```

Figure 7:Land Details where landid=234



Transaction request

Request from remix.ethereum.org
Interacting with 0xcbb8A...fDfb7
Network fee 0.001 ETH

APPROVETRANSFER landId: 234

INITIATETRANSFER landId: 123 newOwner: 0xf640C96b19CCe80347F007FCDb7691B6904F6d68

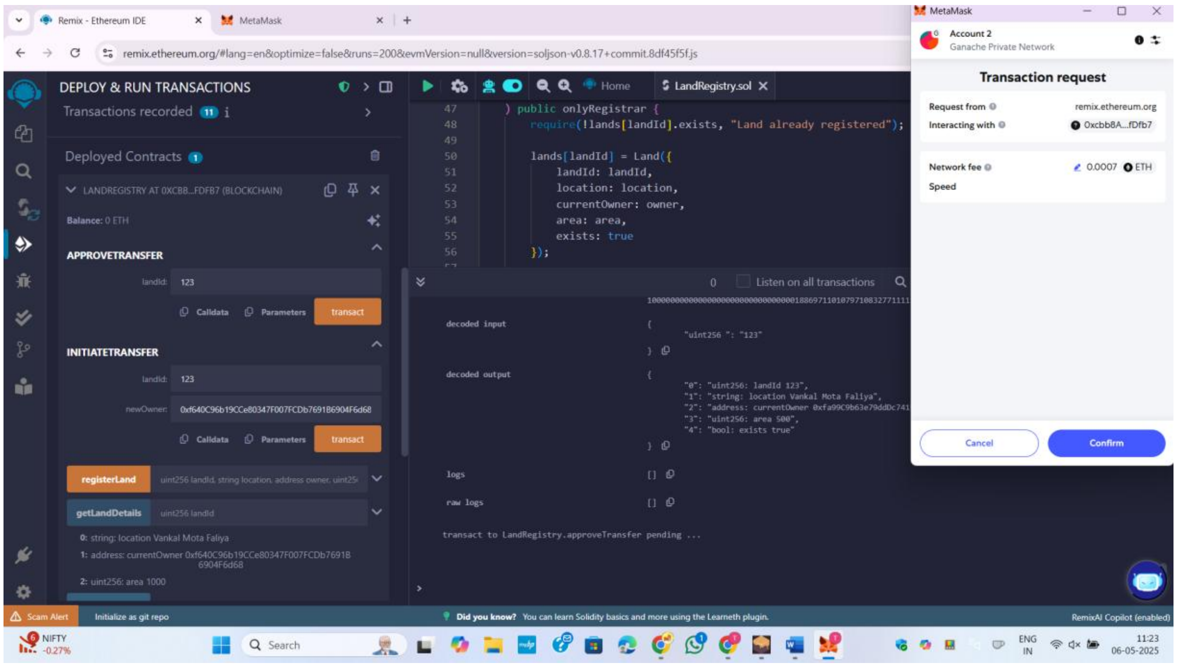
getLandDetails

0: string: location Vankal Mota Faliya
1: address: currentOwner 0xf640C96b19CCe80347F007FCDb7691B6904F6d68
2: uint256: area 1000

```
    ) public onlyRegistrar {
        require(!lands[landId].exists, "Land already registered");

        lands[landId] = Land({
            landId: landId,
            location: location,
            currentOwner: owner,
            area: area,
            exists: true
        });

        emit LandRegistered(landId, location, owner, area);
    }

    function initiateTransfer(uint256 landId, address newOwner) pub
        require(newOwner != address(0), "Invalid address");
        pendingTransfers[landId] = PendingTransfer({newOwner: newOw

        emit TransferInitiated(landId, newOwner);
    }

    function approveTransfer(uint256 landId) public onlyRegistrar {
        require(pendingTransfers[landId].exists, "No transfer initi
```

```
transact to LandRegistry.initiateTransfer pending ...
```

Figure 8:Ask metamak for transaction



Transaction request

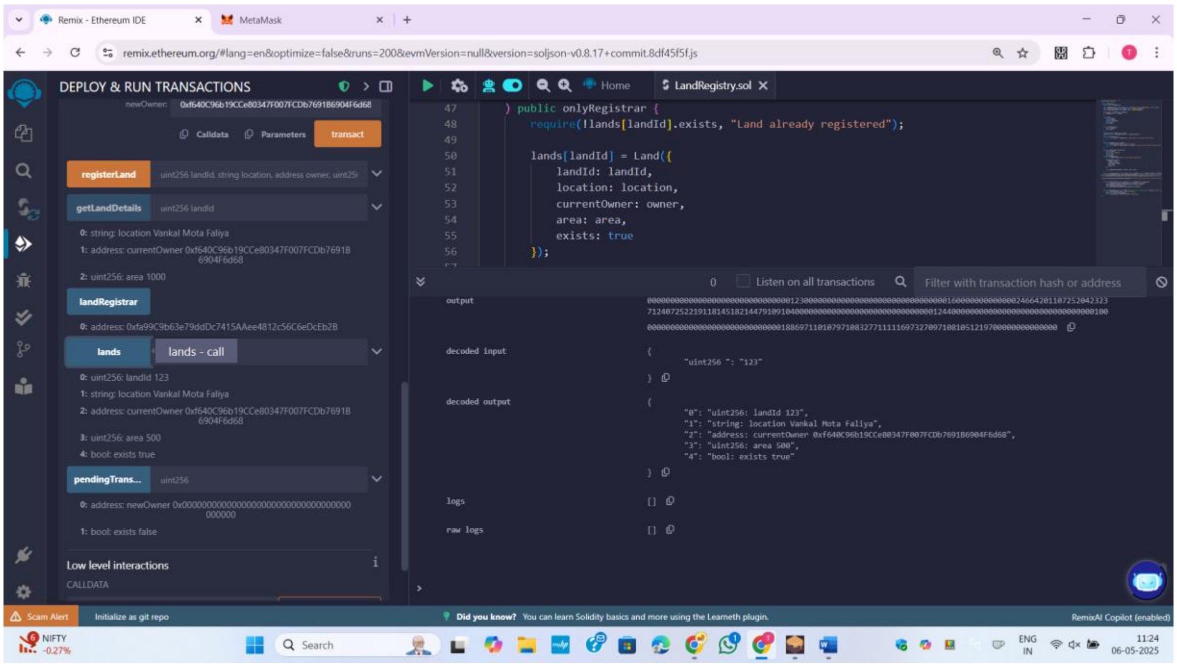
Request from remix.ethereum.org
Interacting with 0xcbb8A...fDfb7
Network fee 0.0007 ETH

APPROVETRANSFER landId: 123

INITIATETRANSFER landId: 123 newOwner: 0xf640C96b19CCe80347F007FCDb7691B6904F6d68

```
decoded input                  {
                                       "uint256 ": "123"
                               }
decoded output                 {
                                       "0": "uint256: landId 123",
                                       "1": "string: location Vankal Mota Faliya",
                                       "2": "address: currentOwner 0xfa99C9b63e79ddDc741
                                       "3": "uint256: area 500",
                                       "4": "bool: exists true"
                               }
logs                           []
raw logs                       []

transact to LandRegistry.approveTransfer pending ...
```

Figure 9:Approve the transaction for Land 123



DEPLOY & RUN TRANSACTIONS

newOwner: 0xf640C96b19CCe80347F007FCDb7691B6904F6d68

registerLand: uint256 landId, string location, address owner, uint25

getLandDetails: uint256 landId

```
0: string: location Vankal Mota Faliya
1: address: currentOwner 0xf640C96b19CCe80347F007FCDb7691B
6904F6d68
2: uint256: area 1000
```

landRegistrar

```
0: address: 0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B
```

lands - call

```
0: uint256: landId 123
1: string: location Vankal Mota Faliya
2: address: currentOwner 0xf640C96b19CCe80347F007FCDb7691B
6904F6d68
3: uint256: area 500
4: bool: exists true
```

pendingTrans...: uint256

```
0: address: newOwner 0x0000000000000000000000000000000000
000000
1: bool: exists false
```

Low level interactions

CALLDATA

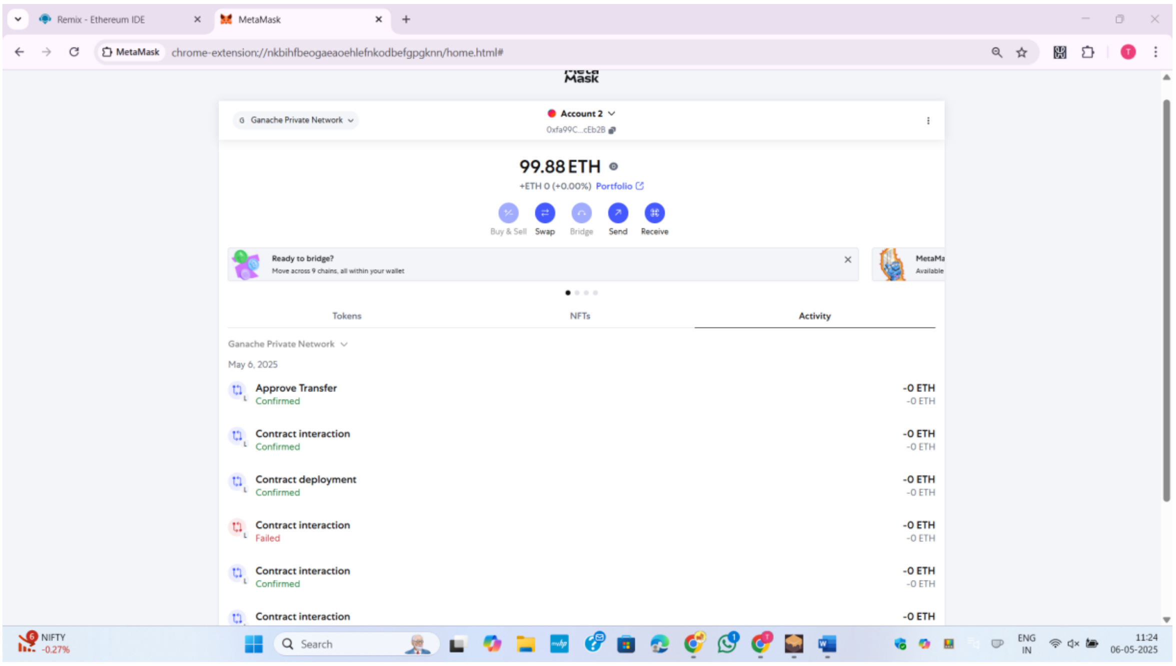
LandRegistry.sol

```
    ) public onlyRegistrar {
        require(!lands[landId].exists, "Land already registered");

        lands[landId] = Land({
            landId: landId,
            location: location,
            currentOwner: owner,
            area: area,
            exists: true
        });
```

```
output          00000000000000000000000000000001230000000000000000000000000000001600000000000002466420110725204232371240725221911814518214479109104000000000000000000000000000124400000000000000000000000000000010000000000000000000000000000018869711010797108327711111697327097108105121970000000000000000

decoded input   {
                        "uint256 ": "123"
                }

decoded output  {
                        "0": "uint256: landId 123",
                        "1": "string: location Vankal Mota Faliya",
                        "2": "address: currentOwner 0xf640C96b19CCe80347F007FCDb7691B6904F6d68",
                        "3": "uint256: area 500",
                        "4": "bool: exists true"
                }

logs            []

raw logs        []
```

Did you know? You can learn Solidity basics and more using the Learneth plugin.

Figure 10:New Owner Change for landid 123



Ganache Private Network

Account 2

0xfa99C...cEb2B

99.88 ETH

+ETH 0 (+0.00%) Portfolio

Ready to bridge?

Move across 9 chains, all within your wallet

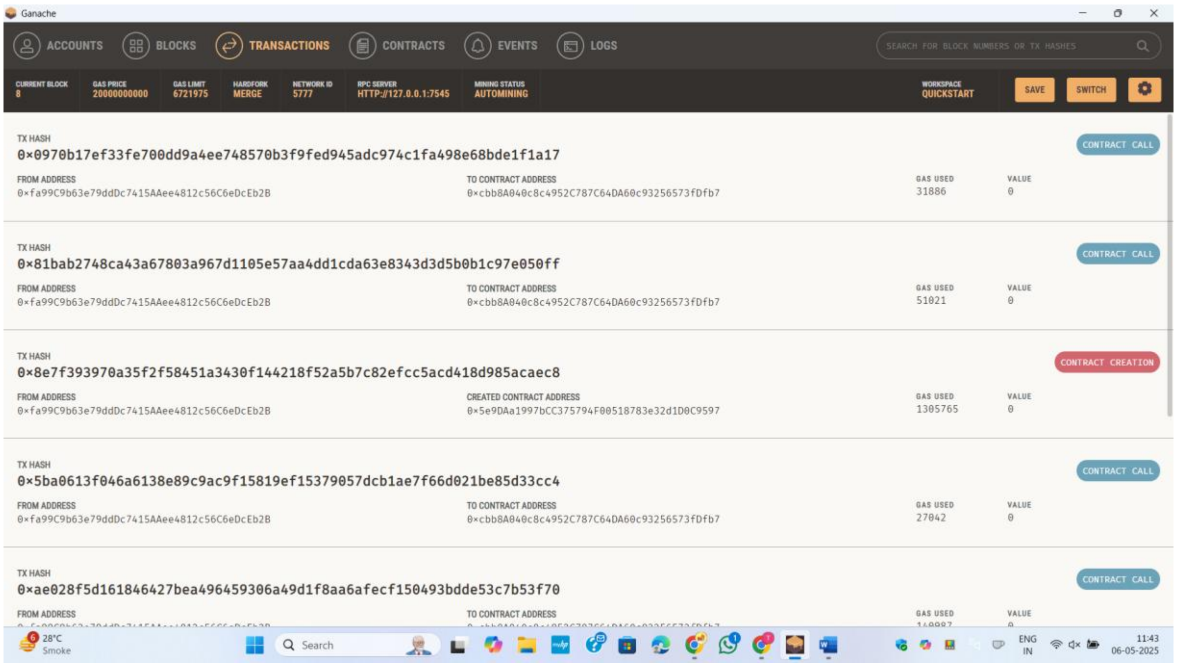
Tokens | NFTs | Activity

Ganache Private Network

May 6, 2025

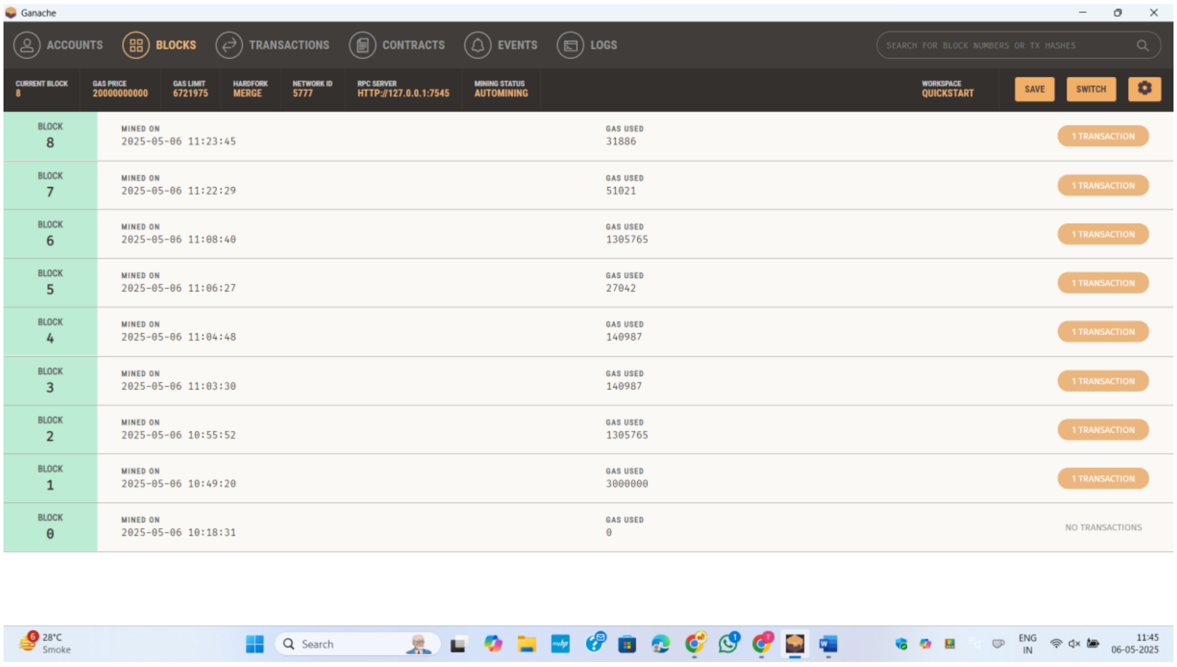
Approve Transfer Confirmed	-0 ETH -0 ETH
Contract interaction Confirmed	-0 ETH -0 ETH
Contract deployment Confirmed	-0 ETH -0 ETH
Contract interaction Failed	-0 ETH -0 ETH
Contract interaction Confirmed	-0 ETH -0 ETH
Contract interaction	-0 ETH

Figure 11:Transaction in MetaMask



```
CURRENT BLOCK  GAS PRICE    GAS LIMIT  HARDFORK  NETWORK ID  RPC SERVER             MINING STATUS  WORKSPACE
8              20000000000  6721975    MERGE     5777        HTTP://127.0.0.1:7545  AUTOMINING     QUICKSTART

TX HASH                                                                                          CONTRACT CALL
0x0970b17ef33fe700dd9a4ee748570b3f9fed945adc974c1fa498e68bde1f1a17
FROM ADDRESS                                  TO CONTRACT ADDRESS                          GAS USED  VALUE
0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B    0xcbb8A040c8c4952C787C64DA60c93256573fDfb7   31886     0

TX HASH                                                                                          CONTRACT CALL
0x81bab2748ca43a67803a967d1105e57aa4dd1cda63e8343d3d5b0b1c97e050ff
FROM ADDRESS                                  TO CONTRACT ADDRESS                          GAS USED  VALUE
0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B    0xcbb8A040c8c4952C787C64DA60c93256573fDfb7   51021     0

TX HASH                                                                                          CONTRACT CREATION
0x8e7f393970a35f2f58451a3430f144218f52a5b7c82efcc5acd418d985acaec8
FROM ADDRESS                                  CREATED CONTRACT ADDRESS                     GAS USED  VALUE
0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B    0x5e9DAa1997bCC375794F00518783e32d1D0C9597   1305765   0

TX HASH                                                                                          CONTRACT CALL
0x5ba0613f046a6138e89c9ac9f15819ef15379057dcb1ae7f66d021be85d33cc4
FROM ADDRESS                                  TO CONTRACT ADDRESS                          GAS USED  VALUE
0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B    0xcbb8A040c8c4952C787C64DA60c93256573fDfb7   27042     0

TX HASH                                                                                          CONTRACT CALL
0xae028f5d161846427bea496459306a49d1f8aa6afecf150493bdde53c7b53f70
FROM ADDRESS                                  TO CONTRACT ADDRESS                          GAS USED  VALUE
```

Figure 12:Transaction Ganache



```
CURRENT BLOCK  GAS PRICE    GAS LIMIT  HARDFORK  NETWORK ID  RPC SERVER             MINING STATUS  WORKSPACE
8              20000000000  6721975    MERGE     5777        HTTP://127.0.0.1:7545  AUTOMINING     QUICKSTART

BLOCK  MINED ON             GAS USED
8      2025-05-06 11:23:45  31886     1 TRANSACTION
7      2025-05-06 11:22:29  51021     1 TRANSACTION
6      2025-05-06 11:08:40  1305765   1 TRANSACTION
5      2025-05-06 11:06:27  27042     1 TRANSACTION
4      2025-05-06 11:04:48  140987    1 TRANSACTION
3      2025-05-06 11:03:30  140987    1 TRANSACTION
2      2025-05-06 10:55:52  1305765   1 TRANSACTION
1      2025-05-06 10:49:20  3000000   1 TRANSACTION
0      2025-05-06 10:18:31  0         NO TRANSACTIONS
```

Figure 13:Trasaction block in Ganache

AIM:

Develop a decentralized smart contract system to manage patient medical records on the Ethereum blockchain. The contract should allow doctors to create records, patients to access them, and administrators to manage permissions.

- Admin (Hospital or Health Authority): Deployer of the contract, manages doctor and patient
- registration.
- Doctor: Authorized to create and update medical records.
- Patients: Can view their own medical records only.

Functional requirements:

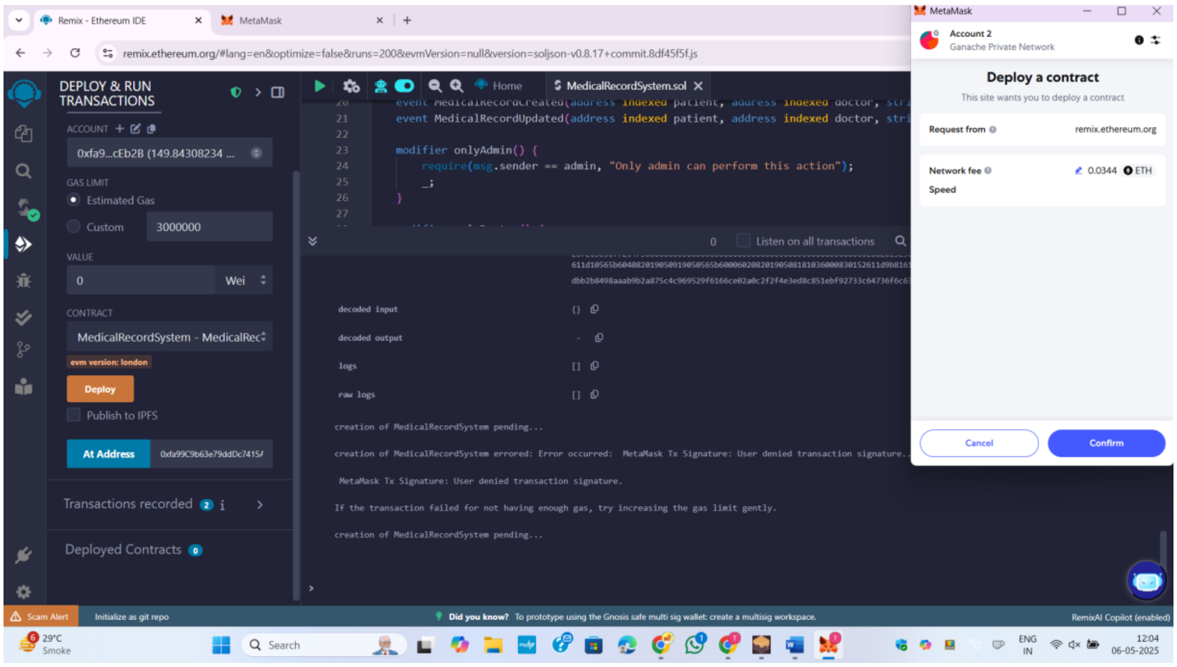
- The admin can register new doctors and patients.
- Only authorized doctors can add or update a patient's medical record.
- Only registered patients can view their own records.
- A medical record should contain: Patient ID (address), Doctor ID (address), Diagnosis (string),
- Prescription (string), Timestamp
- Ensure access control so that no unauthorized access or modification occurs.

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract MedicalRecordSystem {
    address public admin;

    struct MedicalRecord {
        address doctorId;
        string diagnosis;
        string prescription;
        uint256 timestamp;
    }

    mapping(address => bool) public doctors;
    mapping(address => bool) public patients;
    mapping(address => MedicalRecord[]) public records;

    event DoctorRegistered(address indexed doctor);
    event PatientRegistered(address indexed patient);
    event MedicalRecordCreated(address indexed patient, address indexed
doctor, string diagnosis, string prescription, uint256 timestamp);
```

```
    event MedicalRecordUpdated(address indexed patient, address indexed
doctor, string diagnosis, string prescription, uint256 timestamp);

    modifier onlyAdmin() {
        require(msg.sender == admin, "Only admin can perform this action");
        _;
    }

    modifier onlyDoctor() {
        require(doctors[msg.sender] == true, "Only authorized doctors can
perform this action");
        _;
    }

    modifier onlyPatient(address patient) {
        require(patients[patient] == true, "Only registered patients can
view records");
        _;
    }


    constructor() {
        admin = msg.sender;
    }

    function registerDoctor(address doctor) public onlyAdmin {
        require(doctors[doctor] == false, "Doctor already registered");
        doctors[doctor] = true;
        emit DoctorRegistered(doctor);
    }

    function registerPatient(address patient) public onlyAdmin {
        require(patients[patient] == false, "Patient already registered");
        patients[patient] = true;
        emit PatientRegistered(patient);
    }

    function createMedicalRecord(address patient, string memory diagnosis,
string memory prescription) public onlyDoctor {
        require(patients[patient] == true, "Patient not registered");

        MedicalRecord memory newRecord = MedicalRecord({
            doctorId: msg.sender,
            diagnosis: diagnosis,
            prescription: prescription,
            timestamp: block.timestamp
        });

        records[patient].push(newRecord);

        emit MedicalRecordCreated(patient, msg.sender, diagnosis,
prescription, block.timestamp);
    }

    function updateMedicalRecord(address patient, uint256 recordIndex,
string memory diagnosis, string memory prescription) public onlyDoctor {
        require(patients[patient] == true, "Patient not registered");
        require(records[patient].length > recordIndex, "Invalid record
index");

        MedicalRecord storage record = records[patient][recordIndex];
        record.diagnosis = diagnosis;
        record.prescription = prescription;
        record.timestamp = block.timestamp;

        emit MedicalRecordUpdated(patient, msg.sender, diagnosis,
prescription, block.timestamp);
    }

    function viewMedicalRecords() public view onlyPatient(msg.sender)
returns (MedicalRecord[] memory) {
        return records[msg.sender];
    }

    function isDoctor(address doctor) public view returns (bool) {
        return doctors[doctor];
    }

    function isPatient(address patient) public view returns (bool) {
        return patients[patient];
    }
}
```

Output :

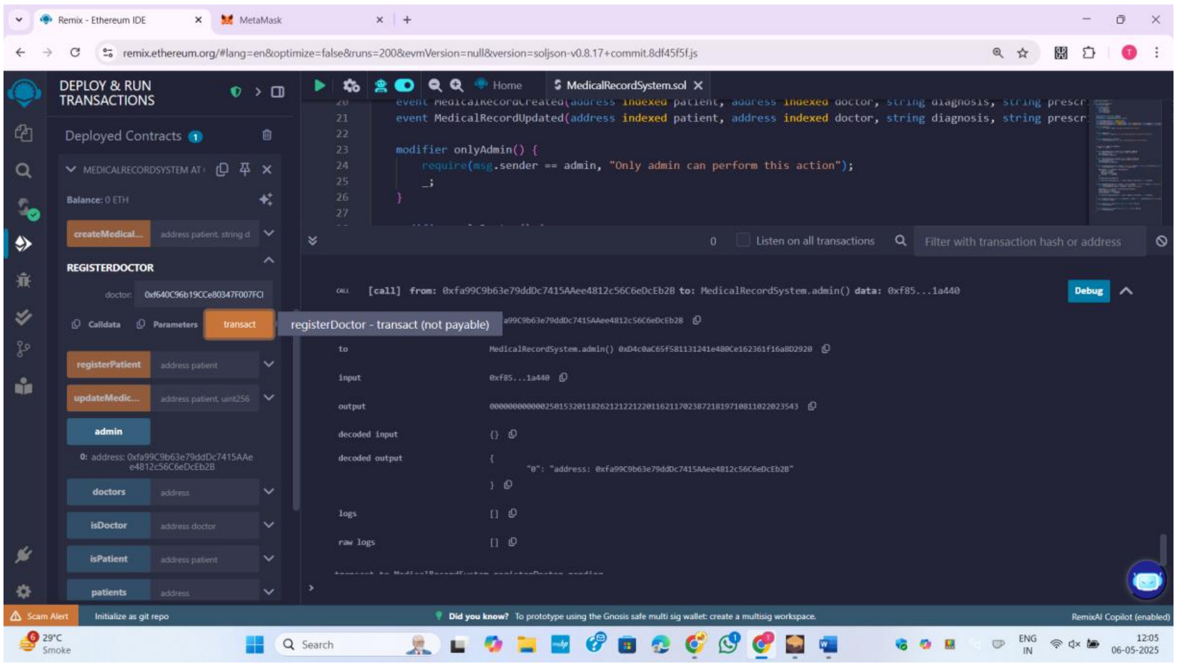


```
event MedicalRecordUpdated(address indexed patient, address indexed doctor, stri

modifier onlyAdmin() {
    require(msg.sender == admin, "Only admin can perform this action");
    _;
}
```

```
611d10565b604082019050919050565b6000602082019050818103600083015261 1d9b816
dbb2b8498aaab9b2a875c4c969529f6166ce02a0c2f2f4e3ed8c851ebf92733c64736f6c6

decoded input      {}
decoded output     -
logs               []
raw logs           []

creation of MedicalRecordSystem pending...

creation of MedicalRecordSystem errored: Error occurred:  MetaMask Tx Signature: User denied transaction signature..

 MetaMask Tx Signature: User denied transaction signature.

If the transaction failed for not having enough gas, try increasing the gas limit gently.

creation of MedicalRecordSystem pending...
```

DEPLOY & RUN TRANSACTIONS
ACCOUNT
0xfa9...cEb2B (149.84308234 ...
GAS LIMIT
Estimated Gas
Custom 3000000
VALUE
0 Wei
CONTRACT
MedicalRecordSystem - MedicalRec
evm version: london
Deploy
Publish to IPFS
At Address
Transactions recorded 2
Deployed Contracts 0
MetaMask
Account 2
Ganache Private Network
Deploy a contract
This site wants you to deploy a contract
Request from remix.ethereum.org
Network fee 0.0344 ETH
Speed
Cancel
Confirm
Did you know? To prototype using the Gnosis safe multi sig wallet: create a multisig workspace.

Figure 14:Deploy the contract

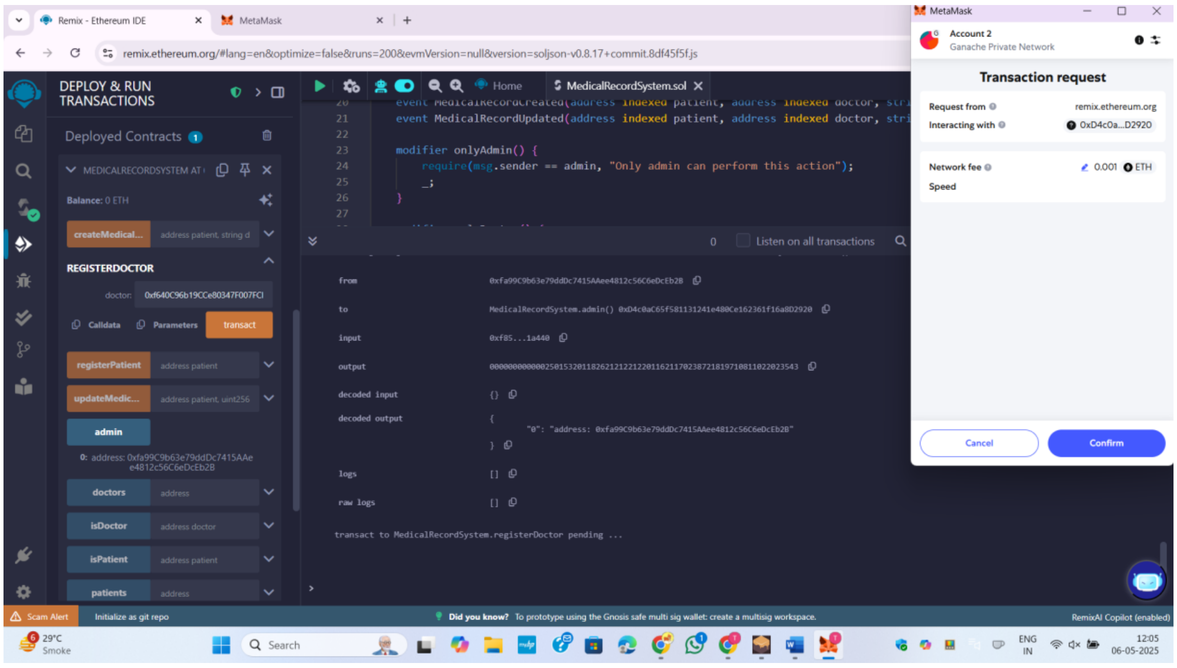


```
event MedicalRecordUpdated(address indexed patient, address indexed doctor, string diagnosis, string prescr

modifier onlyAdmin() {
    require(msg.sender == admin, "Only admin can perform this action");
    _;
}
```

```
[call] from: 0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B to: MedicalRecordSystem.admin() data: 0xf85...1a440

from            0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B
to              MedicalRecordSystem.admin() 0xD4c0aC65f581131241e480Ce162361f16a8D2920
input           0xf85...1a440
output          0000000000002501532011826212122122011621170238721819710811022023543
decoded input   {}
decoded output  {
                     "0": "address: 0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B"
                }
logs            []
raw logs        []
```

DEPLOY & RUN TRANSACTIONS
Deployed Contracts 1
MEDICALRECORDSYSTEM AT
Balance: 0 ETH
createMedical... address patient, string d
REGISTERDOCTOR
doctor: 0xf640C96b19CCe80347F007FCl
Calldata
Parameters
transact
registerDoctor - transact (not payable)
registerPatient address patient
updateMedic... address patient, uint256
admin
0: address: 0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B
doctors address
isDoctor address doctor
isPatient address patient
patients address
Debug
Did you know? To prototype using the Gnosis safe multi sig wallet: create a multisig workspace.

Figure 15:Get Admin details



```
event MedicalRecordUpdated(address indexed patient, address indexed doctor, stri

modifier onlyAdmin() {
    require(msg.sender == admin, "Only admin can perform this action");
    _;
}
```

```
from            0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B
to              MedicalRecordSystem.admin() 0xD4c0aC65f581131241e480Ce162361f16a8D2920
input           0xf85...1a440
output          0000000000002501532011826212122122011621170238721819710811022023543
decoded input   {}
decoded output  {
                        "0": "address: 0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B"
                }
logs            []
raw logs        []

transact to MedicalRecordSystem.registerDoctor pending ...
```

REGISTERDOCTOR — doctor: 0xf640C96b19CCe80347F007FCl

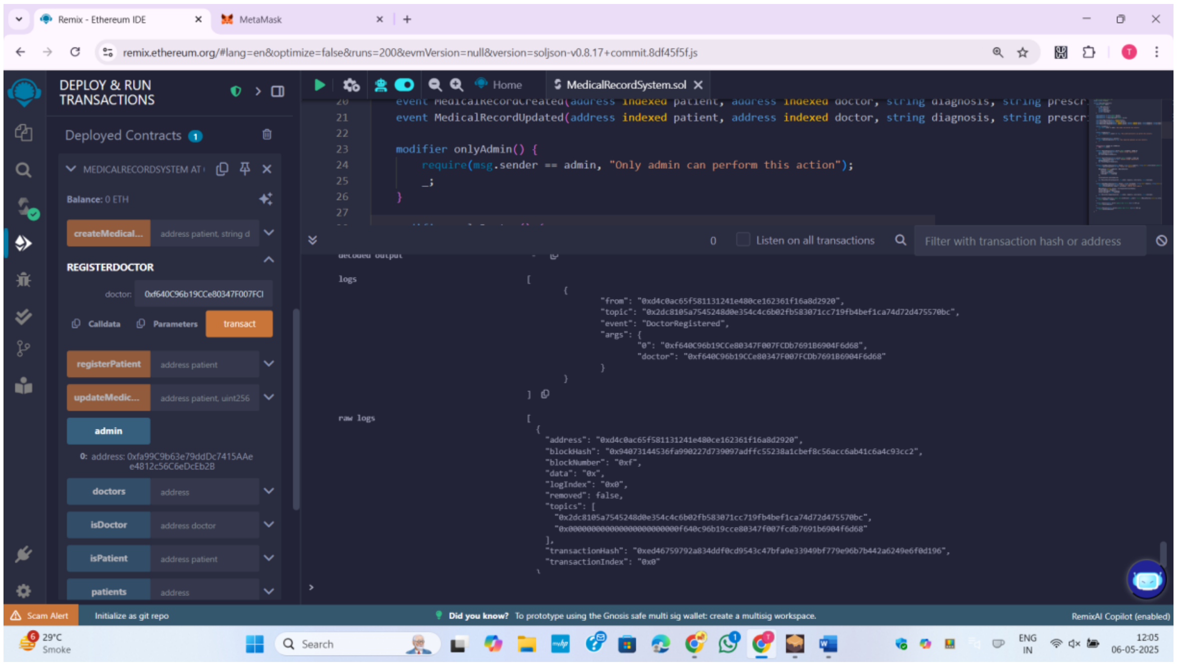
admin — 0: address: 0xfa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B

MetaMask — Account 2, Ganache Private Network

Transaction request

Request from	remix.ethereum.org
Interacting with	0xD4c0a...D2920
Network fee	0.001 ETH
Speed	

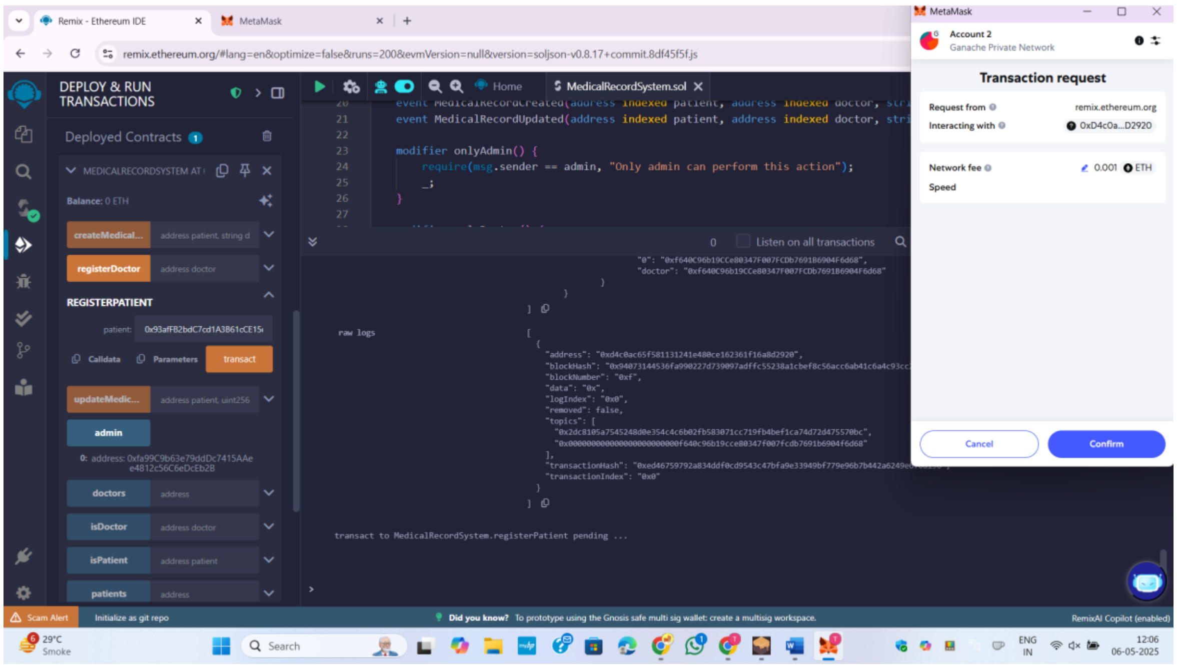
Figure 16:Admin register the Docter



```
event MedicalRecordUpdated(address indexed patient, address indexed doctor, string diagnosis, string prescr

modifier onlyAdmin() {
    require(msg.sender == admin, "Only admin can perform this action");
    _;
}
```

```
logs            [
                        {
                                "from": "0xd4c0ac65f581131241e480ce162361f16a8d2920",
                                "topic": "0x2dc8105a7545248d0e354c4c6b02fb583071cc719fb4bef1ca74d72d475570bc",
                                "event": "DoctorRegistered",
                                "args": {
                                        "0": "0xf640C96b19CCe80347F007FCDb7691B6904F6d68",
                                        "doctor": "0xf640C96b19CCe80347F007FCDb7691B6904F6d68"
                                }
                        }
                ]

raw logs        [
                  {
                    "address": "0xd4c0ac65f581131241e480ce162361f16a8d2920",
                    "blockHash": "0x94073144536fa990227d739097adffc55238a1cbef8c56acc6ab41c6a4c93cc2",
                    "blockNumber": "0xf",
                    "data": "0x",
                    "logIndex": "0x0",
                    "removed": false,
                    "topics": [
                      "0x2dc8105a7545248d0e354c4c6b02fb583071cc719fb4bef1ca74d72d475570bc",
                      "0x000000000000000000000000f640c96b19cce80347f007fcdb7691b6904f6d68"
                    ],
                    "transactionHash": "0xed46759792a834ddf0cd9543c47bfa9e33949bf779e96b7b442a6249e6f0d196",
                    "transactionIndex": "0x0"
```

Figure 17:Registration of Docter successfully completed



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.17+commit.8df45f5f.js

```
event MedicalRecordCreated(address indexed patient, address indexed doctor, stri
event MedicalRecordUpdated(address indexed patient, address indexed doctor, stri

modifier onlyAdmin() {
    require(msg.sender == admin, "Only admin can perform this action");
    _;
}
```

```
                "0": "0xf640C96b19CCe80347F007FCDb7691B6904F6d68",
                "doctor": "0xf640C96b19CCe80347F007FCDb7691B6904F6d68"
            }
        }
    ]
raw logs    [
      {
        "address": "0xd4c0ac65f581131241e480ce162361f16a8d2920",
        "blockHash": "0x94073144536fa990227d739097adffc55238a1cbef8c56acc6ab41c6a4c93cc
        "blockNumber": "0xf",
        "data": "0x",
        "logIndex": "0x0",
        "removed": false,
        "topics": [
          "0x2dc8105a7545248d0e354c4c6b02fb583071cc719fb4bef1ca74d72d475570bc",
          "0x000000000000000000000000f640c96b19cce80347f007fcdb7691b6904f6d68"
        ],
        "transactionHash": "0xed46759792a834ddf0cd9543c47bfa9e33949bf779e96b7b442a6249e
        "transactionIndex": "0x0"
      }
    ]

transact to MedicalRecordSystem.registerPatient pending ...
```

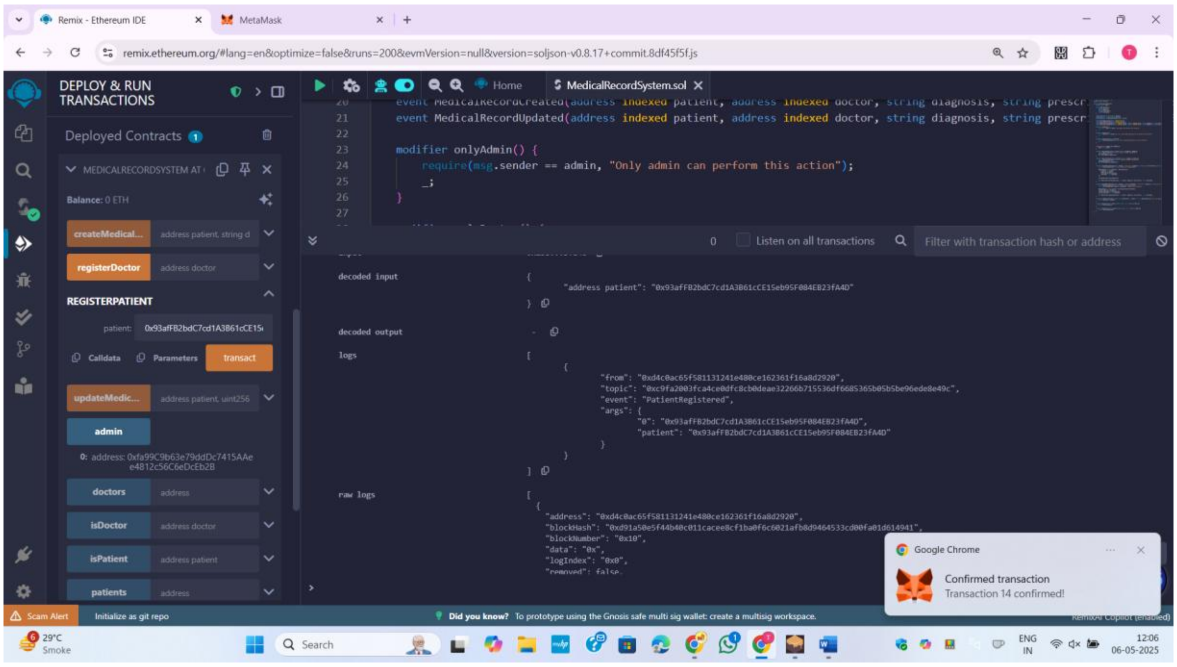
MetaMask — Account 2, Ganache Private Network

Transaction request

Request from: remix.ethereum.org
Interacting with: 0xD4c0a...D2920
Network fee: 0.001 ETH
Speed

Cancel | Confirm

Figure 18:Registration of patient

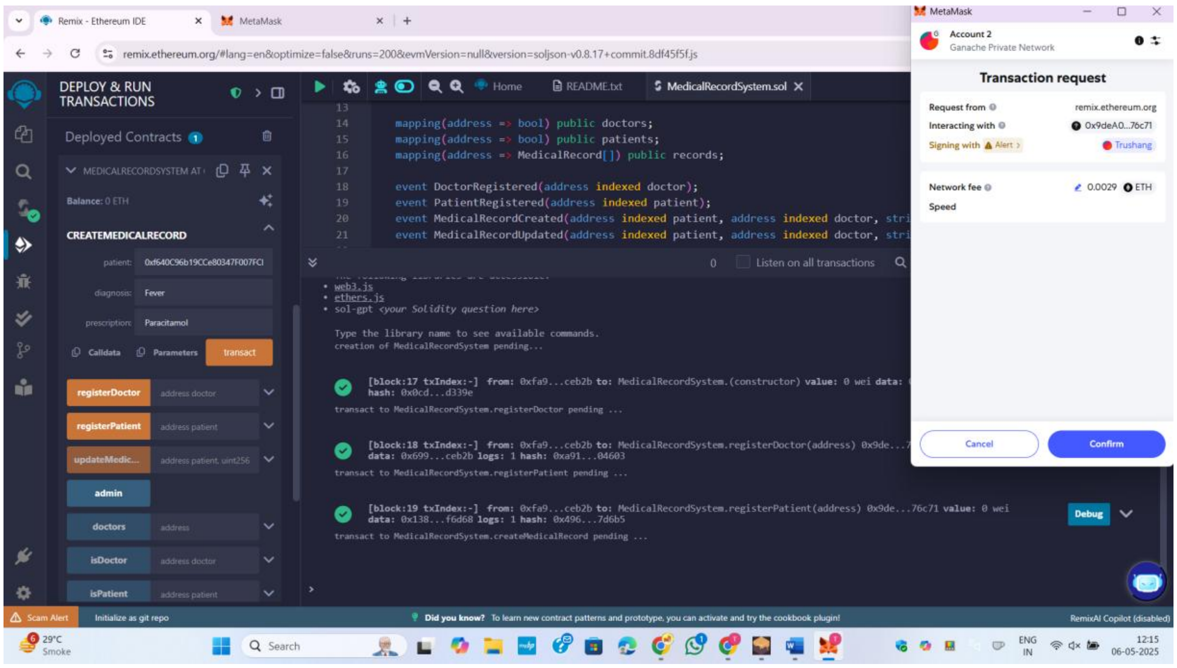


```
event MedicalRecordCreated(address indexed patient, address indexed doctor, string diagnosis, string prescr
event MedicalRecordUpdated(address indexed patient, address indexed doctor, string diagnosis, string prescr

modifier onlyAdmin() {
    require(msg.sender == admin, "Only admin can perform this action");
    _;
}
```

```
decoded input    {
                     "address patient": "0x93afFB2bdC7cd1A3B61cCE15eb95F084EB23fA4D"
                 }
decoded output   -
logs             [
                     {
                         "from": "0xd4c0ac65f581131241e480ce162361f16a8d2920",
                         "topic": "0xc9fa2003fca4ce0dfc8cb0deae32266b715536df6685365b05b5be96ede8e49c",
                         "event": "PatientRegistered",
                         "args": {
                             "0": "0x93afFB2bdC7cd1A3B61cCE15eb95F084EB23fA4D",
                             "patient": "0x93afFB2bdC7cd1A3B61cCE15eb95F084EB23fA4D"
                         }
                     }
                 ]
raw logs         [
                   {
                     "address": "0xd4c0ac65f581131241e480ce162361f16a8d2920",
                     "blockHash": "0xd91a50e5f44b40c011cacee8cf1ba0f6c6021afb8d9464533cd00fa01d614941",
                     "blockNumber": "0x10",
                     "data": "0x",
                     "logIndex": "0x0",
                     "removed": false,
```

Google Chrome — Confirmed transaction. Transaction 14 confirmed!

Figure 19:Transaction completed for Registration of patient



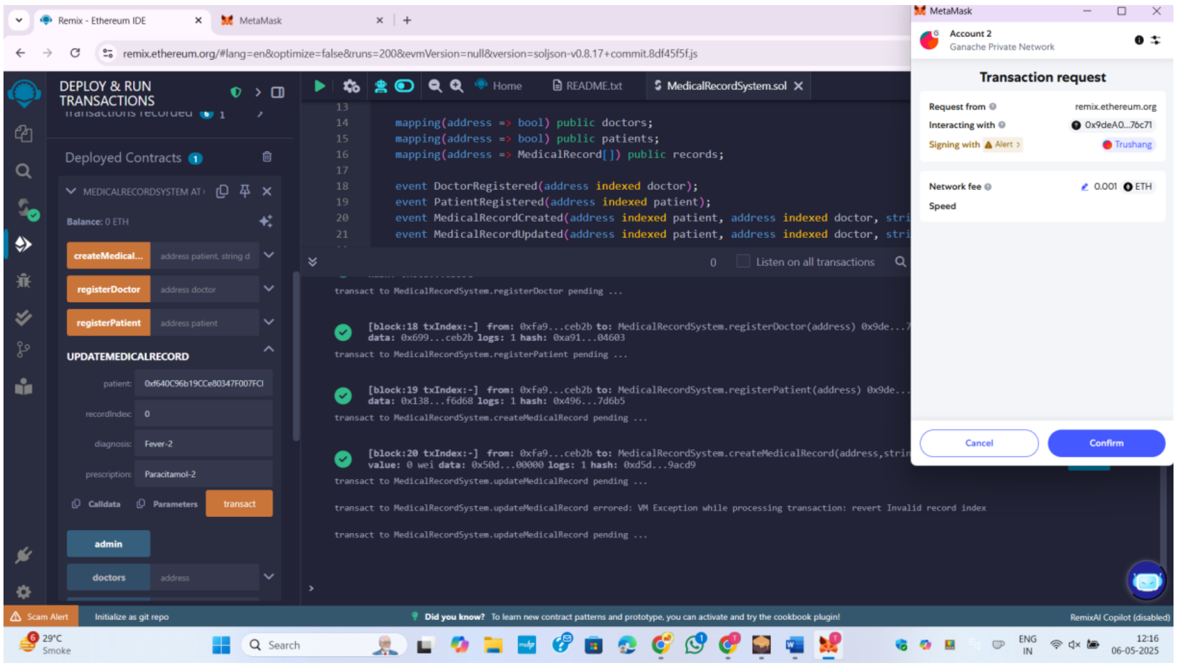
CREATEMEDICALRECORD
patient: 0xf640C96b19CCe80347F007FCl
diagnosis: Fever
prescription: Paracitamol

```
    mapping(address => bool) public doctors;
    mapping(address => bool) public patients;
    mapping(address => MedicalRecord[]) public records;

    event DoctorRegistered(address indexed doctor);
    event PatientRegistered(address indexed patient);
    event MedicalRecordCreated(address indexed patient, address indexed doctor, stri
    event MedicalRecordUpdated(address indexed patient, address indexed doctor, stri
```

```
 • web3.js
 • ethers.js
 • sol-gpt <your Solidity question here>

 Type the library name to see available commands.
 creation of MedicalRecordSystem pending...

 [block:17 txIndex:-] from: 0xfa9...ceb2b to: MedicalRecordSystem.(constructor) value: 0 wei data:
 hash: 0x0cd...d339e
 transact to MedicalRecordSystem.registerDoctor pending ...

 [block:18 txIndex:-] from: 0xfa9...ceb2b to: MedicalRecordSystem.registerDoctor(address) 0x9de...7
 data: 0x699...ceb2b logs: 1 hash: 0xa91...04603
 transact to MedicalRecordSystem.registerPatient pending ...

 [block:19 txIndex:-] from: 0xfa9...ceb2b to: MedicalRecordSystem.registerPatient(address) 0x9de...76c71 value: 0 wei
 data: 0x138...f6d68 logs: 1 hash: 0x496...7d6b5
 transact to MedicalRecordSystem.createMedicalRecord pending ...
```

MetaMask — Account 2, Ganache Private Network
Transaction request
Request from: remix.ethereum.org
Interacting with: 0x9deA0...76c71
Signing with: Trushang
Network fee: 0.0029 ETH

Figure 20:Docter Adding patient details

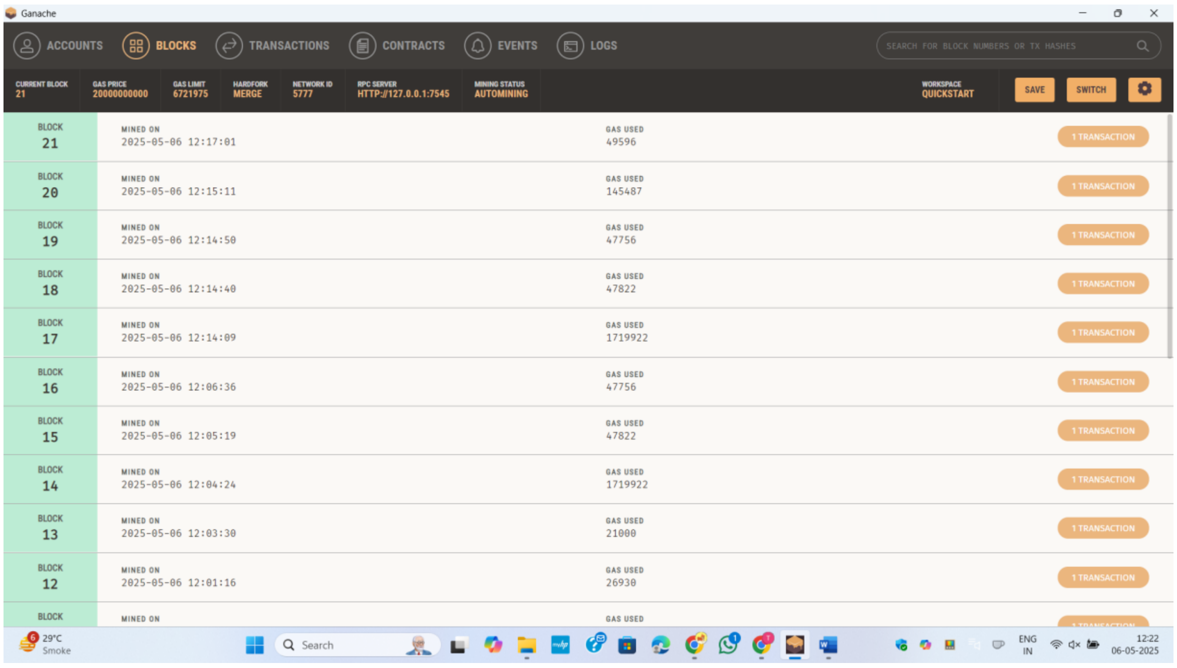


UPDATEMEDICALRECORD
patient: 0xf640C96b19CCe80347F007FCl
recordIndex: 0
diagnosis: Fever-2
prescription: Paracitamol-2

```
 transact to MedicalRecordSystem.registerDoctor pending ...

 [block:18 txIndex:-] from: 0xfa9...ceb2b to: MedicalRecordSystem.registerDoctor(address) 0x9de...7
 data: 0x699...ceb2b logs: 1 hash: 0xa91...04603
 transact to MedicalRecordSystem.registerPatient pending ...

 [block:19 txIndex:-] from: 0xfa9...ceb2b to: MedicalRecordSystem.registerPatient(address) 0x9de...
 data: 0x138...f6d68 logs: 1 hash: 0x496...7d6b5
 transact to MedicalRecordSystem.createMedicalRecord pending ...

 [block:20 txIndex:-] from: 0xfa9...ceb2b to: MedicalRecordSystem.createMedicalRecord(address,strin
 value: 0 wei data: 0x50d...00000 logs: 1 hash: 0xd5d...9acd9
 transact to MedicalRecordSystem.updateMedicalRecord pending ...

 transact to MedicalRecordSystem.updateMedicalRecord errored: VM Exception while processing transaction: revert Invalid record index

 transact to MedicalRecordSystem.updateMedicalRecord pending ...
```

MetaMask — Account 2, Ganache Private Network
Transaction request
Request from: remix.ethereum.org
Interacting with: 0x9deA0...76c71
Signing with: Trushang
Network fee: 0.001 ETH

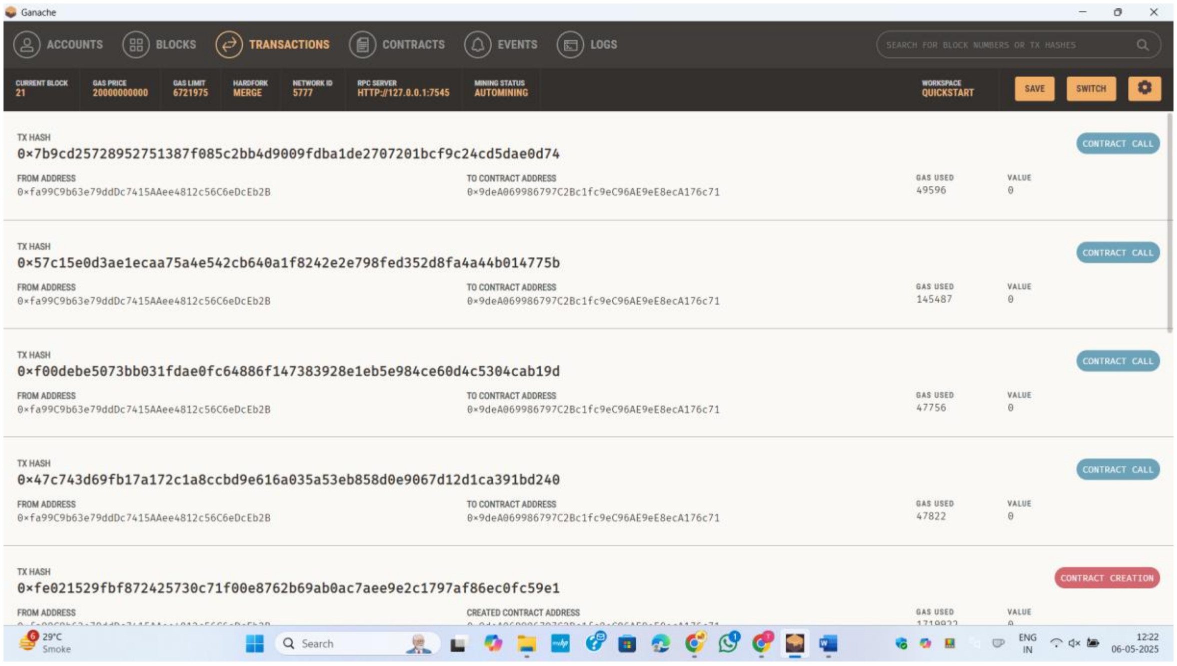
Figure 21:Update patient details



CURRENT BLOCK 21 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MERGE | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE QUICKSTART

BLOCK	MINED ON	GAS USED	
21	2025-05-06 12:17:01	49596	1 TRANSACTION
20	2025-05-06 12:15:11	145487	1 TRANSACTION
19	2025-05-06 12:14:50	47756	1 TRANSACTION
18	2025-05-06 12:14:40	47822	1 TRANSACTION
17	2025-05-06 12:14:09	1719922	1 TRANSACTION
16	2025-05-06 12:06:36	47756	1 TRANSACTION
15	2025-05-06 12:05:19	47822	1 TRANSACTION
14	2025-05-06 12:04:24	1719922	1 TRANSACTION
13	2025-05-06 12:03:30	21000	1 TRANSACTION
12	2025-05-06 12:01:16	26930	1 TRANSACTION

Figure 22:Ganache blocks



CURRENT BLOCK 21 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MERGE | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE QUICKSTART

TX HASH
0×7b9cd25728952751387f085c2bb4d9009fdba1de2707201bcf9c24cd5dae0d74 — CONTRACT CALL
FROM ADDRESS 0×fa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B
TO CONTRACT ADDRESS 0×9deA069986797C2Bc1fc9eC96AE9eE8ecA176c71
GAS USED 49596
VALUE 0

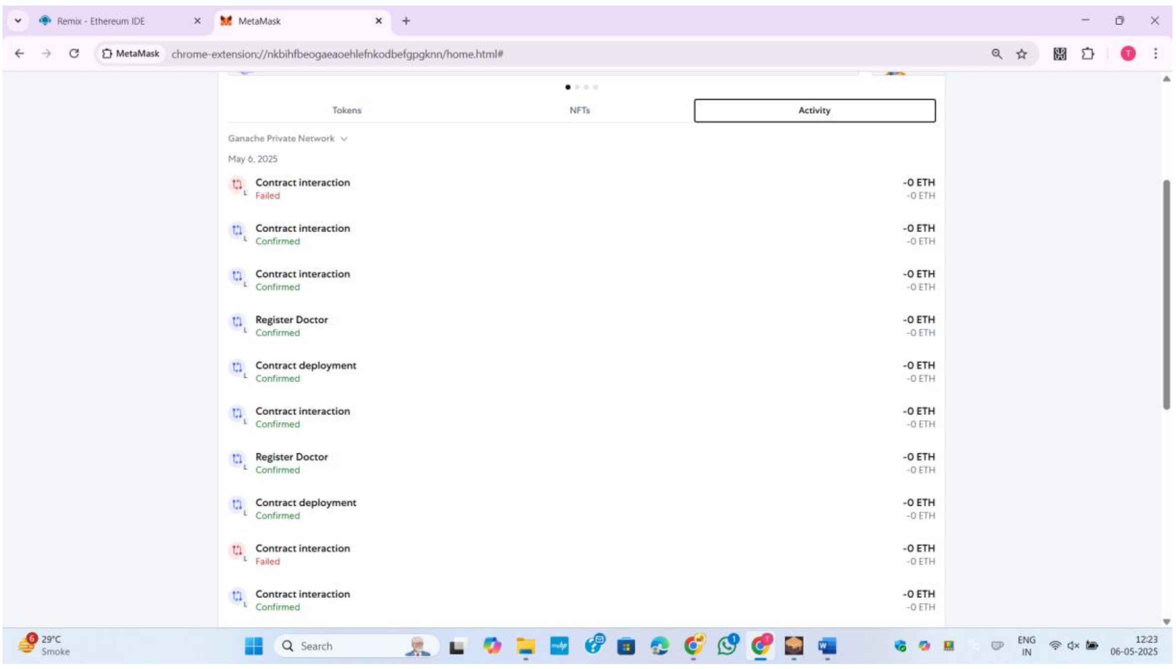
TX HASH
0×57c15e0d3ae1ecaa75a4e542cb640a1f8242e2e798fed352d8fa4a44b014775b — CONTRACT CALL
FROM ADDRESS 0×fa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B
TO CONTRACT ADDRESS 0×9deA069986797C2Bc1fc9eC96AE9eE8ecA176c71
GAS USED 145487
VALUE 0

TX HASH
0×f00debe5073bb031fdae0fc64886f147383928e1eb5e984ce60d4c5304cab19d — CONTRACT CALL
FROM ADDRESS 0×fa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B
TO CONTRACT ADDRESS 0×9deA069986797C2Bc1fc9eC96AE9eE8ecA176c71
GAS USED 47756
VALUE 0

TX HASH
0×47c743d69fb17a172c1a8ccbd9e616a035a53eb858d0e9067d12d1ca391bd240 — CONTRACT CALL
FROM ADDRESS 0×fa99C9b63e79ddDc7415AAee4812c56C6eDcEb2B
TO CONTRACT ADDRESS 0×9deA069986797C2Bc1fc9eC96AE9eE8ecA176c71
GAS USED 47822
VALUE 0

TX HASH
0×fe021529fbf872425730c71f00e8762b69ab0ac7aee9e2c1797af86ec0fc59e1 — CONTRACT CREATION
FROM ADDRESS
CREATED CONTRACT ADDRESS
GAS USED
VALUE

Figure 23:Ganache transaction



May 6, 2025

Contract interaction Failed -0 ETH -0 ETH
Contract interaction Confirmed -0 ETH -0 ETH
Contract interaction Confirmed -0 ETH -0 ETH
Register Doctor Confirmed -0 ETH -0 ETH
Contract deployment Confirmed -0 ETH -0 ETH
Contract interaction Confirmed -0 ETH -0 ETH
Register Doctor Confirmed -0 ETH -0 ETH
Contract deployment Confirmed -0 ETH -0 ETH
Contract interaction Failed -0 ETH -0 ETH
Contract interaction Confirmed -0 ETH -0 ETH

Figure 24:MetaMask Transaction