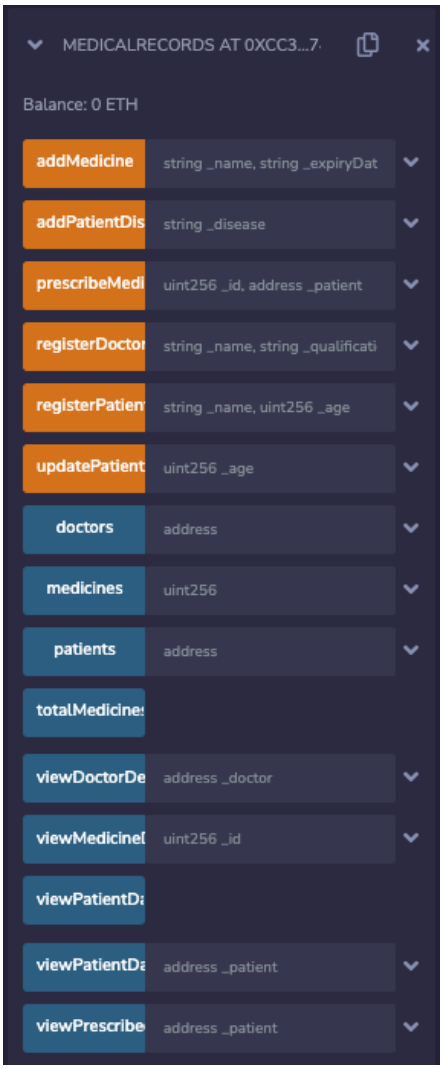
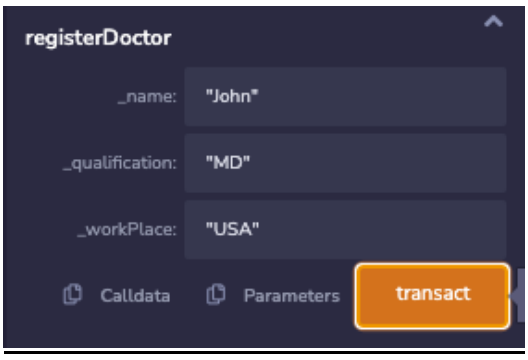
Screenshot

Functions Involved:

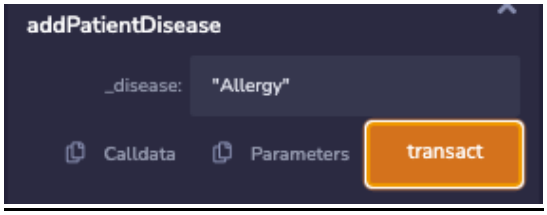


Register Doctor:

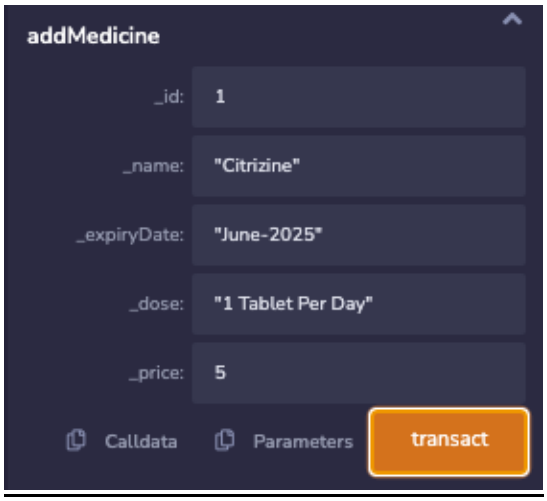


Register Patient:

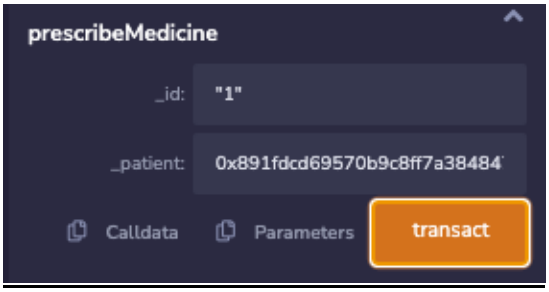
Patient Disease:



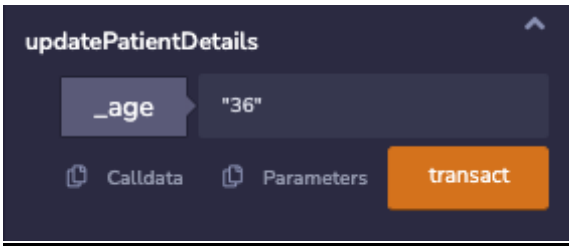
Medicine:



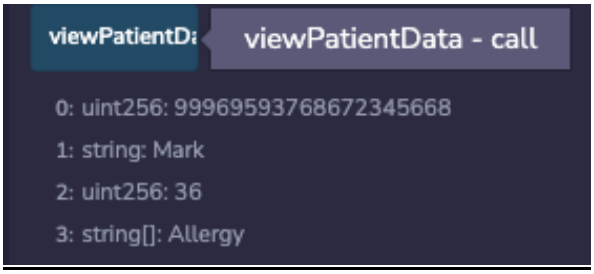
Prescribe Medicine:



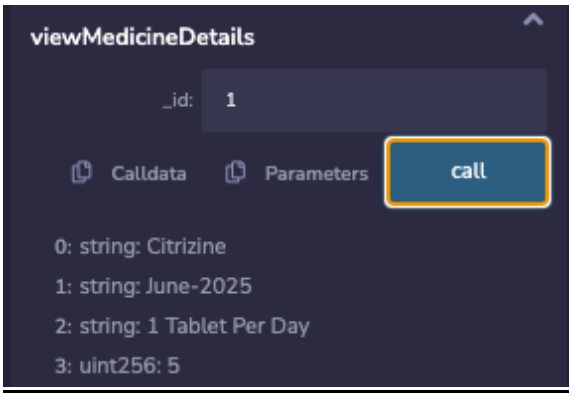
Update Patient Details:



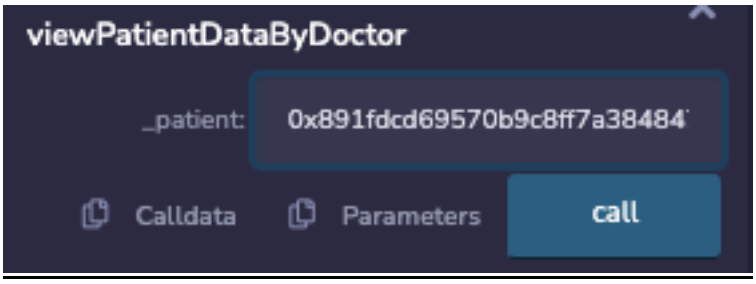
Patient Details:



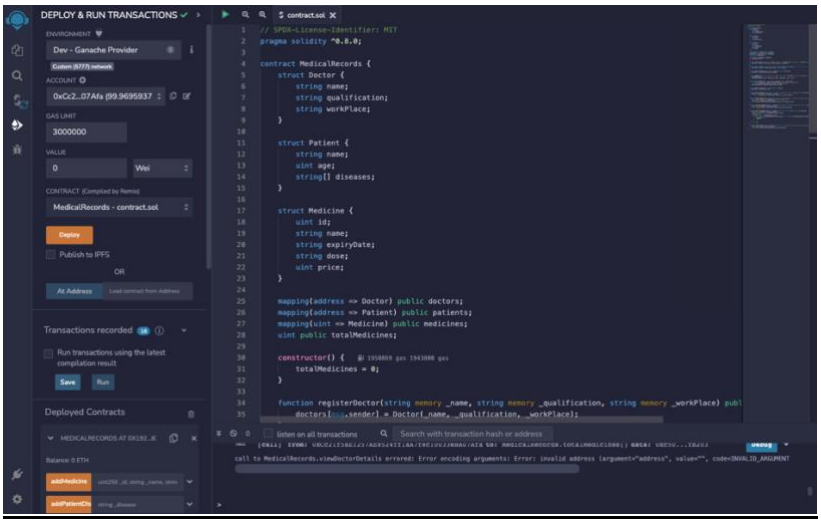
Medicine Details:

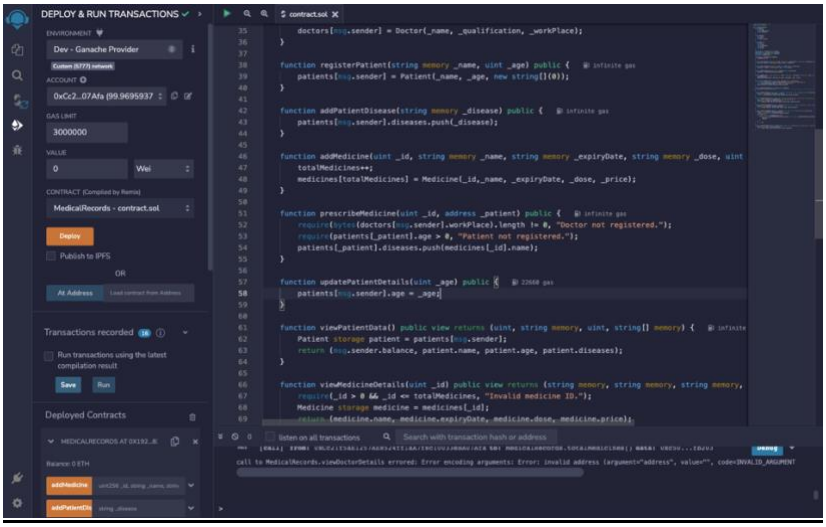


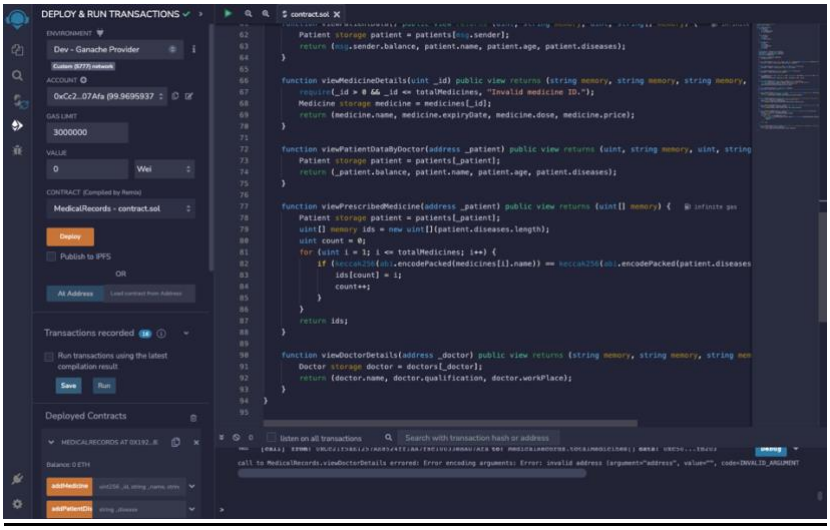
Patient Data By Doctor:



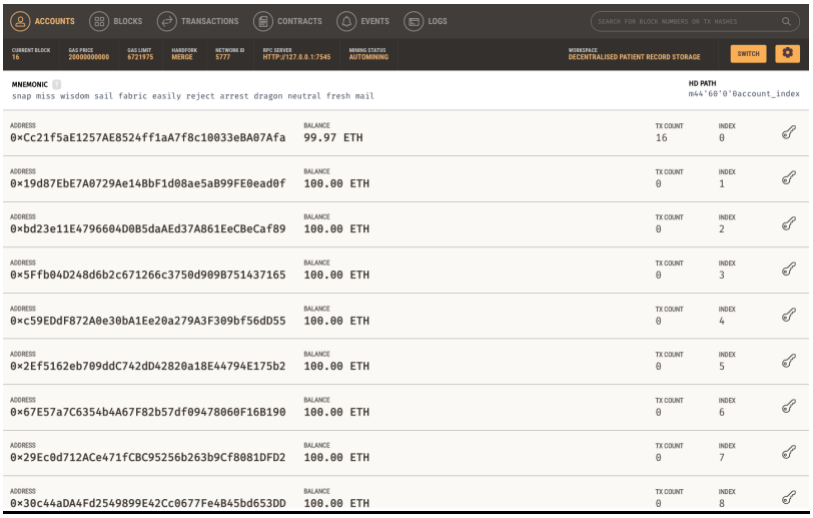
CODE:


DEPLOY & RUN TRANSACTIONS
contract.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract MedicalRecords {
struct Doctor {
string name;
string qualification;
string workPlace;
}
struct Patient {
string name;
uint age;
string[] diseases;
}
struct Medicine {
uint id;
string name;
string expiryDate;
string dose;
uint price;
}
mapping(address => Doctor) public doctors;
mapping(address => Patient) public patients;
mapping(uint => Medicine) public medicines;
uint public totalMedicines;
constructor() {
totalMedicines = 0;
}
function registerDoctor(string memory _name, string memory _qualification, string memory _workPlace) publ
doctors[msg.sender] = Doctor(_name, _qualification, _workPlace);
Dev - Ganache Provider
0xCc2...07Afa (99.9695937
3000000
0
Wei
MedicalRecords - contract.sol
Deploy
Publish to IPFS
OR
At Address
Transactions recorded
Run transactions using the latest compilation result
Save
Run
Deployed Contracts
Balance: 0 ETH
addMedicine
addPatientDis
Search with transaction hash or address
call to MedicalRecords.viewDoctorDetails errored: Error encoding arguments: Error: invalid address (argument="address", value="", code=INVALID_ARGUMENT

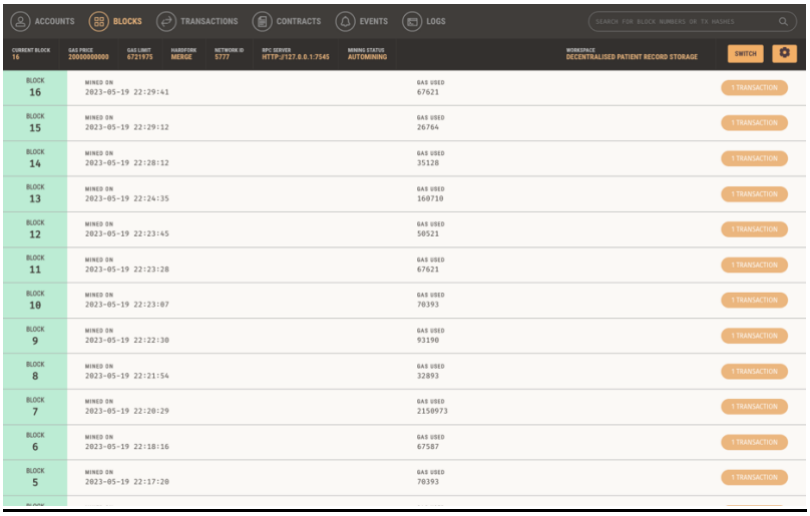

DEPLOY & RUN TRANSACTIONS
contract.sol
doctors[msg.sender] = Doctor(_name, _qualification, _workPlace);
}
function registerPatient(string memory _name, uint _age) public {
patients[msg.sender] = Patient(_name, _age, new string[](0));
}
function addPatientDisease(string memory _disease) public {
patients[msg.sender].diseases.push(_disease);
}
function addMedicine(uint _id, string memory _name, string memory _expiryDate, string memory _dose, uint
totalMedicines++;
medicines[totalMedicines] = Medicine(_id,_name, _expiryDate, _dose, _price);
}
function prescribeMedicine(uint _id, address _patient) public {
require(bytes(doctors[msg.sender].workPlace).length != 0, "Doctor not registered.");
require(patients[_patient].age > 0, "Patient not registered.");
patients[_patient].diseases.push(medicines[_id].name);
}
function updatePatientDetails(uint _age) public {
patients[msg.sender].age = _age;
}
function viewPatientData() public view returns (uint, string memory, uint, string[] memory) {
Patient storage patient = patients[msg.sender];
return (msg.sender.balance, patient.name, patient.age, patient.diseases);
}
function viewMedicineDetails(uint _id) public view returns (string memory, string memory, string memory,
require(_id > 0 && _id <= totalMedicines, "Invalid medicine ID.");
Medicine storage medicine = medicines[_id];
return (medicine.name, medicine.expiryDate, medicine.dose, medicine.price);
Dev - Ganache Provider
0xCc2...07Afa (99.9695937
3000000
0
Wei
MedicalRecords - contract.sol
Deploy
Publish to IPFS
OR
At Address
Transactions recorded
Run transactions using the latest compilation result
Save
Run
Deployed Contracts
Balance: 0 ETH
addMedicine
addPatientDis
Search with transaction hash or address
call to MedicalRecords.viewDoctorDetails errored: Error encoding arguments: Error: invalid address (argument="address", value="", code=INVALID_ARGUMENT


DEPLOY & RUN TRANSACTIONS
contract.sol
Patient storage patient = patients[msg.sender];
return (msg.sender.balance, patient.name, patient.age, patient.diseases);
}
function viewMedicineDetails(uint _id) public view returns (string memory, string memory, string memory,
require(_id > 0 && _id <= totalMedicines, "Invalid medicine ID.");
Medicine storage medicine = medicines[_id];
return (medicine.name, medicine.expiryDate, medicine.dose, medicine.price);
}
function viewPatientDataByDoctor(address _patient) public view returns (uint, string memory, uint, string
Patient storage patient = patients[_patient];
return (_patient.balance, patient.name, patient.age, patient.diseases);
}
function viewPrescribedMedicine(address _patient) public view returns (uint[] memory) {
Patient storage patient = patients[_patient];
uint[] memory ids = new uint[](patient.diseases.length);
uint count = 0;
for (uint i = 1; i <= totalMedicines; i++) {
if (keccak256(abi.encodePacked(medicines[i].name)) == keccak256(abi.encodePacked(patient.diseases
ids[count] = i;
count++;
}
}
return ids;
}
function viewDoctorDetails(address _doctor) public view returns (string memory, string memory, string mem
Doctor storage doctor = doctors[_doctor];
return (doctor.name, doctor.qualification, doctor.workPlace);
}
}
Dev - Ganache Provider
0xCc2...07Afa (99.9695937
3000000
0
Wei
MedicalRecords - contract.sol
Deploy
Publish to IPFS
OR
At Address
Transactions recorded
Run transactions using the latest compilation result
Save
Run
Deployed Contracts
Balance: 0 ETH
addMedicine
addPatientDis
Search with transaction hash or address
call to MedicalRecords.viewDoctorDetails errored: Error encoding arguments: Error: invalid address (argument="address", value="", code=INVALID_ARGUMENT

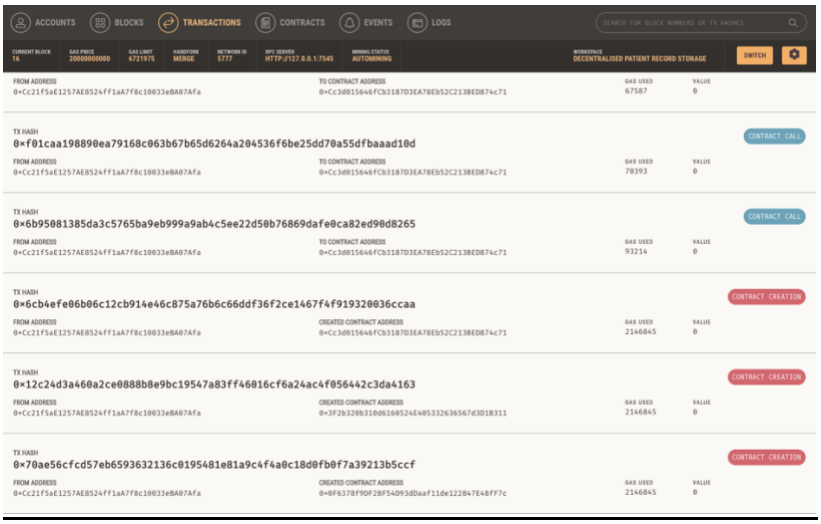
Ganache Accounts



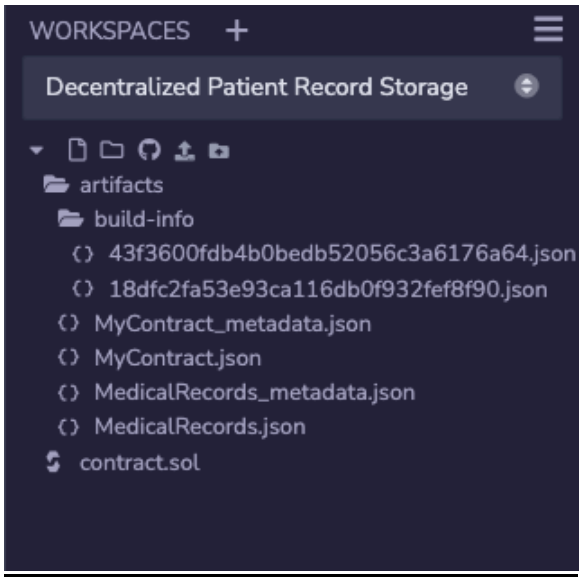
Blocks:



Transactions:



Files Involved:



RPC Server: