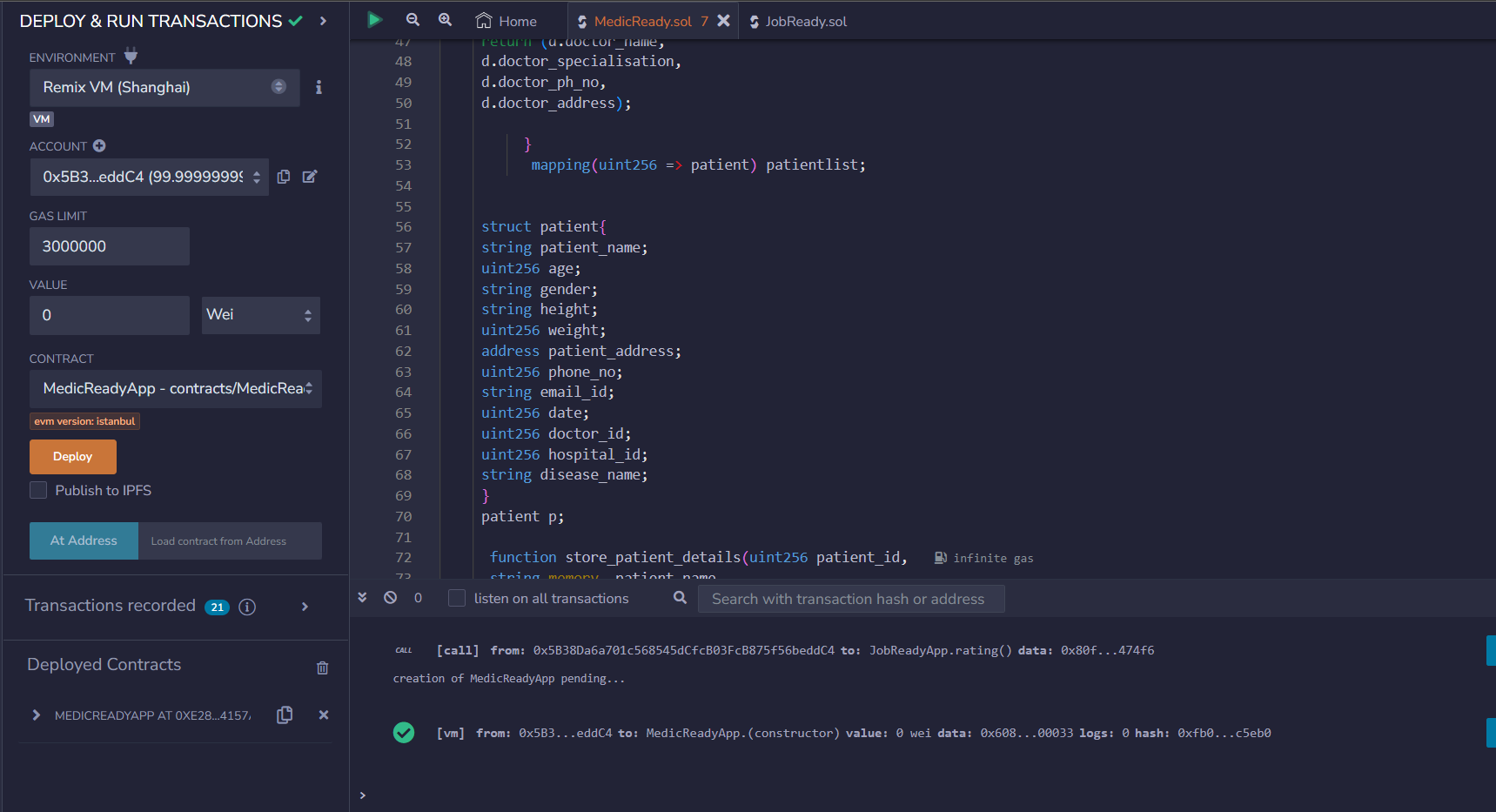
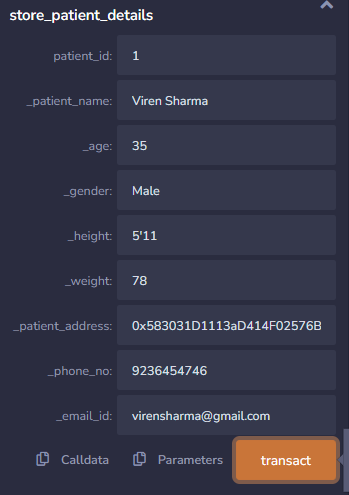
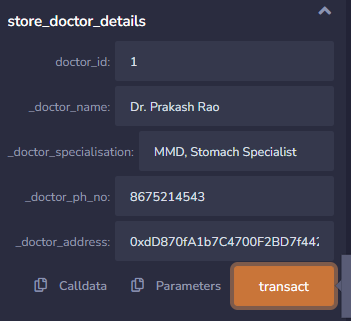
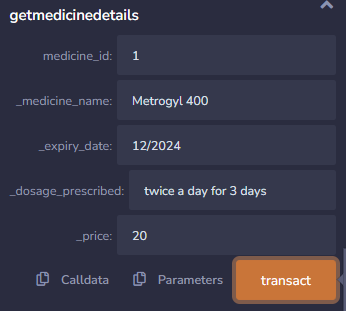
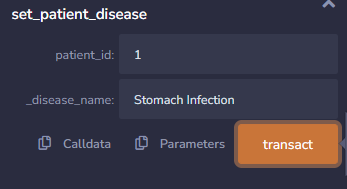
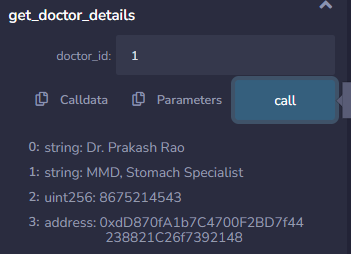
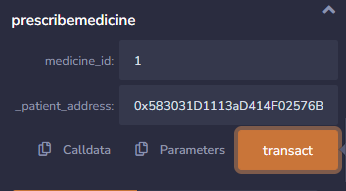
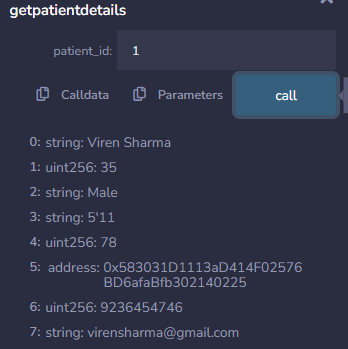
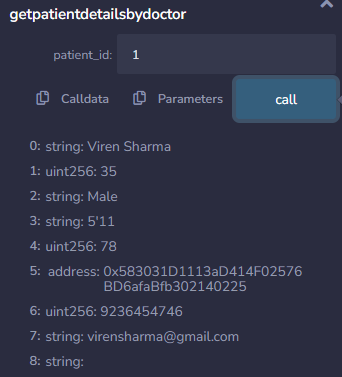
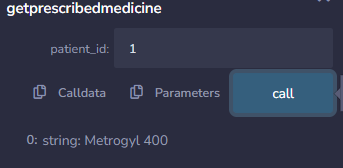
Writeup

I have designed and analysed the Medical Record Smart Contract named as ‘MedicReady’. As part of my development, I created three structures- doctor, patient and medicine. I also applied mapping, modifiers wherever required and I am able to implement the functions as per the need for the contract. Correcting errors was one of the most difficult parts but I did it well and thoroughly.

Screenshots



Source Code

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract MedicReadyApp {

    mapping(uint256 => doctor) doctorlist;

     struct doctor{

         string doctor\_name;

         string doctor\_specialisation;

         uint256 doctor\_ph\_no;

         address doctor\_address;

     }

     doctor d;

     address owner;

      constructor()  public {

          owner = 0xdD870fA1b7C4700F2BD7f44238821C26f7392148;

      }

       modifier isOwner() {

         \_;

     }

   function store\_doctor\_details(

    uint16 doctor\_id,

   string memory \_doctor\_name,

   string memory \_doctor\_specialisation,

   uint256 \_doctor\_ph\_no,

   address \_doctor\_address)public isOwner {

         d.doctor\_name = \_doctor\_name;

         d.doctor\_specialisation = \_doctor\_specialisation;

         d.doctor\_ph\_no = \_doctor\_ph\_no;

         d.doctor\_address = \_doctor\_address;

        doctorlist[doctor\_id] = d;

   }

         function get\_doctor\_details(uint16 doctor\_id) public view returns (

            string memory,

            string memory,

            uint256,

            address) {

     doctor memory d = doctorlist[doctor\_id];

     return (d.doctor\_name,

     d.doctor\_specialisation,

     d.doctor\_ph\_no,

     d.doctor\_address);

          }

           mapping(uint256 => patient) patientlist;

     struct patient{

     string patient\_name;

     uint256 age;

     string gender;

     string height;

     uint256 weight;

     address patient\_address;

     uint256 phone\_no;

     string email\_id;

     uint256 date;

     uint256 doctor\_id;

     uint256 hospital\_id;

     string disease\_name;

     }

     patient p;

      function store\_patient\_details(uint256 patient\_id,

      string memory \_patient\_name,

      uint256 \_age,

      string memory \_gender,

      string memory \_height,

      uint256 \_weight,

      address \_patient\_address,

      uint256 \_phone\_no,

      string memory \_email\_id)public isOwner {

         p.patient\_name=\_patient\_name;

         p.age=\_age;

         p.gender=\_gender;

         p.height=\_height;

         p.weight=\_weight;

         p.patient\_address=\_patient\_address;

         p.phone\_no=\_phone\_no;

         p.email\_id=\_email\_id;

        patientlist[patient\_id] = p;}

        function getpatientdetails(uint256 patient\_id) public view returns (

        string memory,

        uint256,

        string memory,

        string memory,

        uint256,

        address,

        uint256,

        string memory)

       { patient memory p = patientlist[patient\_id];

     return (p.patient\_name,

     p.age,

     p.gender,

     p.height,

     p.weight,

     p.patient\_address,

     p.phone\_no,

     p.email\_id);

    }

    function set\_patient\_disease(uint256 patient\_id, string memory \_disease\_name) public isOwner{

        p.disease\_name;

        patientlist[patient\_id] = p;

    }

               mapping(uint256 => medicine) medicinelist;

    struct medicine{

        string medicine\_name;

        string expiry\_date;

        string dosage\_prescribed;

        uint256 price;

    }

    medicine m;

    function getmedicinedetails(

        uint medicine\_id,

        string memory \_medicine\_name,

        string memory \_expiry\_date,

        string memory \_dosage\_prescribed,

        uint256 \_price) public isOwner{

            m.medicine\_name= \_medicine\_name;

            m.expiry\_date= \_expiry\_date;

            m.dosage\_prescribed= \_dosage\_prescribed;

            m.price= \_price;

        medicinelist [medicine\_id]=m;

        }

    function getmedicinedeatails(uint medicine\_id) public view returns(

        string memory,

        string memory,

        string memory,

        uint

    ) { medicine memory m= medicinelist[medicine\_id];

    return

        (m.medicine\_name,

            m.expiry\_date,

            m.dosage\_prescribed,

            m.price);

    }

        function prescribemedicine(uint256 medicine\_id, address \_patient\_address) public isOwner{

            p.patient\_address=\_patient\_address;

            medicinelist [medicine\_id]=m;

        }

        function getprescribedmedicine(uint patient\_id) public view returns(

            string memory

        ){medicine memory m = medicinelist[patient\_id];

        return(m.medicine\_name);

        }

       modifier onlydoctor(){

        \_;

       }

       function getpatientdetailsbydoctor(uint256 patient\_id) public view onlydoctor returns (

        string memory,

        uint256,

        string memory,

        string memory,

        uint256,

        address,

        uint256,

        string memory,

        string memory)

       { patient memory p = patientlist[patient\_id];

     return (p.patient\_name,

     p.age,

     p.gender,

     p.height,

     p.weight,

     p.patient\_address,

     p.phone\_no,

     p.email\_id,

     p.disease\_name);

    }

}

ThankYou.