Source Code for IPR:

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract IPRegistry {

//enum ipType{copy\_rights, patent, trade\_mark}

struct IP {

uint id;

string ipType;

string description;

address currentOwner;

}

uint IPcounter = 0;

mapping (uint => IP) IPrecords;

mapping (uint => address[]) OwnerHistory;

event IPRegistered (uint IP\_ID, address currentOwner);

event ownerChanged (uint IP\_ID, address currentOwner);

modifier onlyOwner (uint \_IPcounter) {

require(msg.sender == IPrecords[\_IPcounter].currentOwner,"You are not the owner of this IP");

\_;

}

function registerIP(string memory \_IPType, string memory \_description) public

{

IPcounter++;

IPrecords[IPcounter] = IP(IPcounter, \_IPType,\_description, msg.sender);

OwnerHistory[IPcounter].push(msg.sender);

emit IPRegistered(IPcounter, msg.sender);

}

function changeOwner( uint IP\_ID, address \_newOwner) public onlyOwner(IP\_ID)

{

IPrecords[IP\_ID].currentOwner = \_newOwner;

OwnerHistory[IP\_ID].push(\_newOwner);

emit ownerChanged(IP\_ID, \_newOwner);

}

function getHistory(uint IP\_ID) public view returns (address[] memory){

return OwnerHistory[IP\_ID];

}

function getIPDetails(uint IP\_ID) public view returns (uint IP\_id, string memory IP\_Type, string memory IP\_desc, address IP\_currentOwner){

IP memory ip = IPrecords[IP\_ID];

return(ip.id, ip.ipType, ip.description, ip.currentOwner);

}

}