pragma solidity ^0.7.4;

pragma experimental ABIEncoderV2;

import "./dCheckToken.sol";

contract dcheck{

dCheckToken instance;

address owner;

constructor ( address \_instance ){

instance = dCheckToken(\_instance);

owner = msg.sender;}

// \*\*\*\*\*\*\*\*Mappings \*\*\*\*\*\*\*\*\*

// Employee mapping

mapping (uint => employeeinfo) employee;

uint employeeKey= 0;

// Student Mappings

mapping (uint => studentinfo) student;

uint studentkey = 0;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Structs \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//employee structs

struct employeeinfo{

string employeename;

string employername;

string employeedesignation;

string employeetenure; }

struct studentinfo{

string studentName;

string program;

string duration;

string graduationtime; }

///\*\*\*\*\*\*\*\*\*\*\*\*\*Events \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

event employeeadded(address ,employeeinfo);

event studentadded(address ,studentinfo);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Functions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Employee functions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Adding Employees \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function addemployeeinfo( string memory \_employeename, string memory \_employername, string memory \_employeedesignation, string memory \_employeetenure) public {

employee[employeeKey] = employeeinfo(\_employeename, \_employername, \_employeedesignation, \_employeetenure);

instance.transferFrom(owner,msg.sender, 1);

emit employeeadded(msg.sender, employee[employeeKey]);

employeeKey++; }

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Finding Employee\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function findemployee(uint employeeKey) public returns(employeeinfo memory){

require (instance.balanceOf(msg.sender) >=5 , "insufficinet balance");

instance.transferFrom(msg.sender,owner,5);

return employee[employeeKey]; }

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Studnet function\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*adding student \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function addstudent(string memory \_studentName, string memory \_program, string memory \_duration, string memory \_graduationtime) public {

student[studentkey] = studentinfo(\_studentName ,\_program, \_duration, \_graduationtime);

instance.transferFrom(owner,msg.sender, 1);

emit studentadded(msg.sender, student[studentkey]);

studentkey++; }

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*finding students\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function findstudent(uint studentId) public returns(studentinfo memory){

require (instance.balanceOf(msg.sender) >=5 , "insufficinet balance");

instance.transferFrom(msg.sender,owner,5);

return student[studentId]; }

}