Smart contract :

// SPDX-License-Identifier: MIT

// OpenZeppelin Contracts v4.4.1 (utils/Context.sol)

pragma solidity >=0.7.0 <0.9.0;

contract FinTastic{

    uint bal;

    address owner;

    uint number\_of\_accounts=0;

    struct KYC{

            uint customer\_ID;

            string full\_name;

            string profession;

            string Date\_of\_Birth;

        // 2 mapping

        }

    mapping(address => uint256) public account\_balances;

    mapping(address => KYC) public account\_info;

    modifier onlyRegistered(address tempAddress){

        require(account\_info[tempAddress].customer\_ID > 0, "This function is restricted to the contract's owner");

        \_;

       }

    function registerAccount(address  \_addr, string memory \_full\_name,string memory \_profession,string memory \_Date\_of\_Birth) public

     {

        number\_of\_accounts+=1 ;

        account\_info[\_addr].customer\_ID = number\_of\_accounts;

        account\_info[\_addr].full\_name = \_full\_name;

        account\_info[\_addr].profession = \_profession;

        account\_info[\_addr].Date\_of\_Birth = \_Date\_of\_Birth;

    }

    function getAccountInfo(address \_addr) public view onlyRegistered( \_addr) onlyRegistered(msg.sender) returns (KYC memory) {

        return account\_info[\_addr];

    }

    function getBalance(address \_addr) public view returns (uint) {

        // Mapping always returns a value.

        // If the value was never set, it will return the default value.

        return account\_balances[\_addr];

    }

 /\*

    function deposit(address \_addr) public {

        // Update the value at this address

        account\_balances[\_addr] +=  msg.value;

    }

     receive()\*/

     receive() external payable onlyRegistered(msg.sender) {

       account\_balances[msg.sender] +=  msg.value;

       }

    function withdraw(uint \_amt) public payable onlyRegistered(msg.sender) {

        // Reset the value to the default value.

        //require

         require(account\_balances[msg.sender] > \_amt, "No Suffecient Fund !");

        account\_balances[msg.sender]= account\_balances[msg.sender] - \_amt;

        payable(msg.sender).transfer(\_amt);

    }

        //msg.sender, token\_count \* (10\*\*18)

    function transfer(address \_recipient, uint \_amt) public  onlyRegistered( \_recipient) onlyRegistered( msg.sender) {

        require(account\_balances[msg.sender] > 0); // You must have some tokens.

        account\_balances[msg.sender] -= \_amt;

        account\_balances[\_recipient] += \_amt;

    }

}

ABI:

[

{

"inputs": [

{

"internalType": "address",

"name": "",

"type": "address"

}

],

"name": "account\_balances",

"outputs": [

{

"internalType": "uint256",

"name": "",

"type": "uint256"

}

],

"stateMutability": "view",

"type": "function"

},

{

"inputs": [

{

"internalType": "address",

"name": "",

"type": "address"

}

],

"name": "account\_info",

"outputs": [

{

"internalType": "uint256",

"name": "customer\_ID",

"type": "uint256"

},

{

"internalType": "string",

"name": "full\_name",

"type": "string"

},

{

"internalType": "string",

"name": "profession",

"type": "string"

},

{

"internalType": "string",

"name": "Date\_of\_Birth",

"type": "string"

}

],

"stateMutability": "view",

"type": "function"

},

{

"inputs": [

{

"internalType": "address",

"name": "\_addr",

"type": "address"

}

],

"name": "getAccountInfo",

"outputs": [

{

"components": [

{

"internalType": "uint256",

"name": "customer\_ID",

"type": "uint256"

},

{

"internalType": "string",

"name": "full\_name",

"type": "string"

},

{

"internalType": "string",

"name": "profession",

"type": "string"

},

{

"internalType": "string",

"name": "Date\_of\_Birth",

"type": "string"

}

],

"internalType": "struct FinTastic.KYC",

"name": "",

"type": "tuple"

}

],

"stateMutability": "view",

"type": "function"

},

{

"inputs": [

{

"internalType": "address",

"name": "\_addr",

"type": "address"

}

],

"name": "getBalance",

"outputs": [

{

"internalType": "uint256",

"name": "",

"type": "uint256"

}

],

"stateMutability": "view",

"type": "function"

},

{

"inputs": [

{

"internalType": "address",

"name": "\_addr",

"type": "address"

},

{

"internalType": "string",

"name": "\_full\_name",

"type": "string"

},

{

"internalType": "string",

"name": "\_profession",

"type": "string"

},

{

"internalType": "string",

"name": "\_Date\_of\_Birth",

"type": "string"

}

],

"name": "registerAccount",

"outputs": [],

"stateMutability": "nonpayable",

"type": "function"

},

{

"inputs": [

{

"internalType": "address",

"name": "\_recipient",

"type": "address"

},

{

"internalType": "uint256",

"name": "\_amt",

"type": "uint256"

}

],

"name": "transfer",

"outputs": [],

"stateMutability": "nonpayable",

"type": "function"

},

{

"inputs": [

{

"internalType": "uint256",

"name": "\_amt",

"type": "uint256"

}

],

"name": "withdraw",

"outputs": [],

"stateMutability": "payable",

"type": "function"

},

{

"stateMutability": "payable",

"type": "receive"

}

]