COURSE NAME: **BLOCKCHAIN**

GROUP NUMBER:**07**

PROJECT TITLE:**BLOCKCHAIN IN IDENTITY IN MANAGEMENT**

PROJECT SUBMITTED TO: **ANNA UNIVERSITY/NAAN MUDHALVAN**

YEAR: **II YEAR (2022-2023)**

DEPARTMENT: **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

SEMESTER:**04**

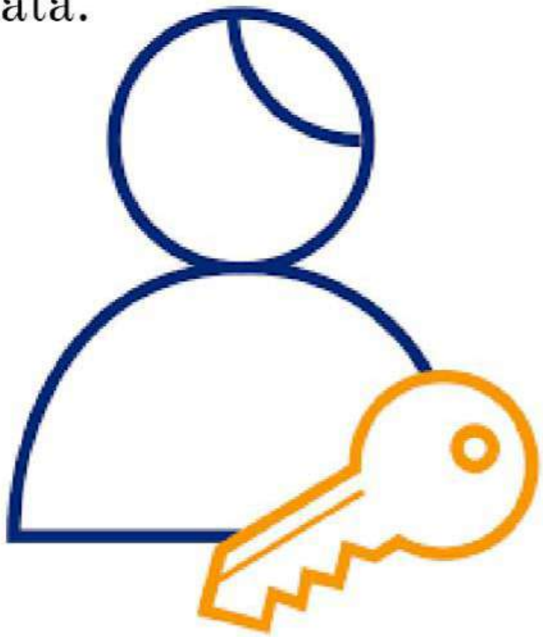
GROUP MEMBERS: **AMIRTHAP P, GOWTHAM BABU V , KRISHNA KANTH,V , MOHAMMED NAVEED SHARIFF R, NAVEEN BALA M**

GUIDED BY:**Ms. S.H. ANNIE SILVIYA**

SPOC NAME: **Dr. SRIDHAR S**

Identity Management

Identity management refers to the processes and technologies used to manage and protect digital identities. This includes the management of user authentication, authorization, and access control for computer systems, networks, applications, and data.



Personal Data Theft

Proving your identity

Produce and provide identity documents

Prone to hacking

Lack of transparency & control

- Lack of transparency
- No clarity on the usage and access rights of the data of the consumer
- Users' data sold to third parties without their consent and knowledge
- Data, then, used for marketing and selling by these parties

Decentralized identity

With a blockchain-based identity system, users can have complete control over their own identity data, rather than relying on centralized authorities such as government agencies or corporations. This can provide greater privacy and security for users.

Verification of identity

Blockchain technology can be used to verify the authenticity of identity documents such as passports or driver's licenses. This can help prevent fraud and identity theft.

Immutable records

Once data is recorded on a blockchain, it cannot be altered or deleted. This can ensure that identity data remains accurate and trustworthy.

Interoperability

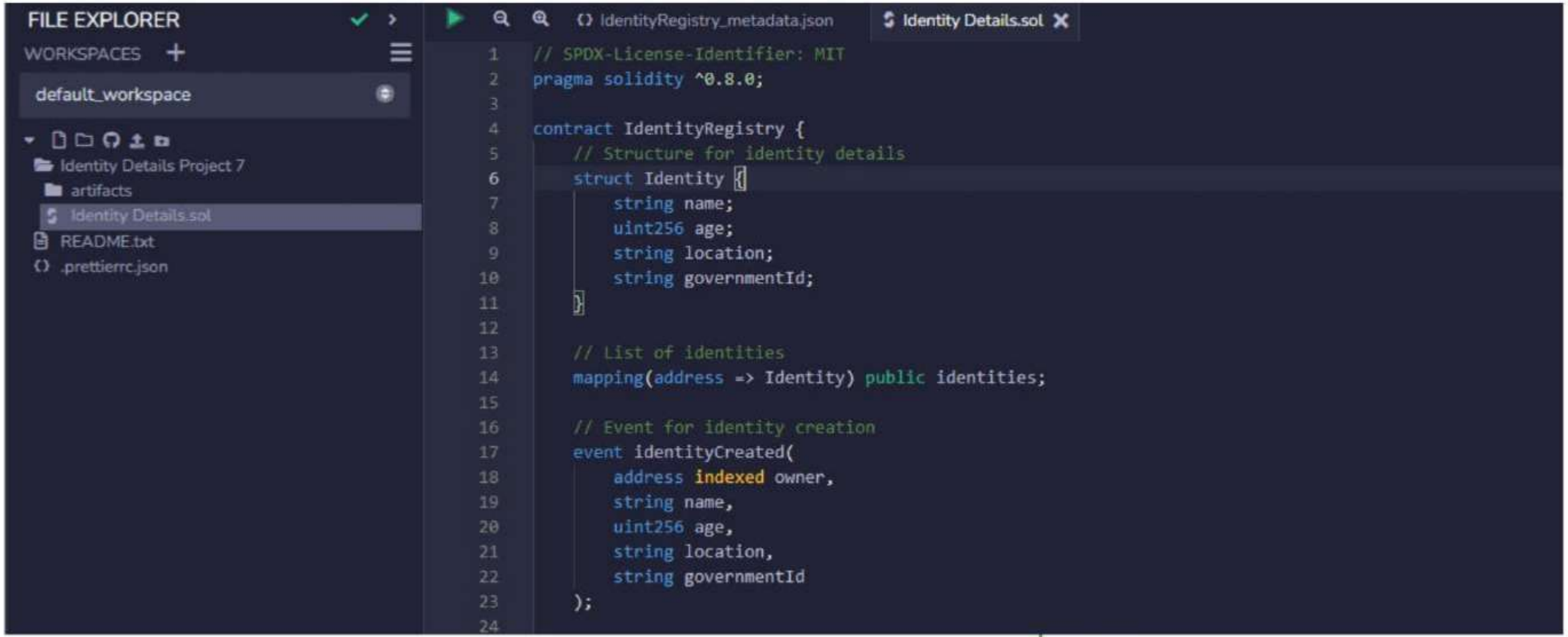
Blockchain-based identity systems can enable interoperability between different identity systems, allowing users to easily access and use their identity data across different platforms.

Self-sovereign identity

With a blockchain-based identity system, individuals can create and manage their own digital identities without the need for intermediaries. This can give users greater control over their personal data and how it is shared.

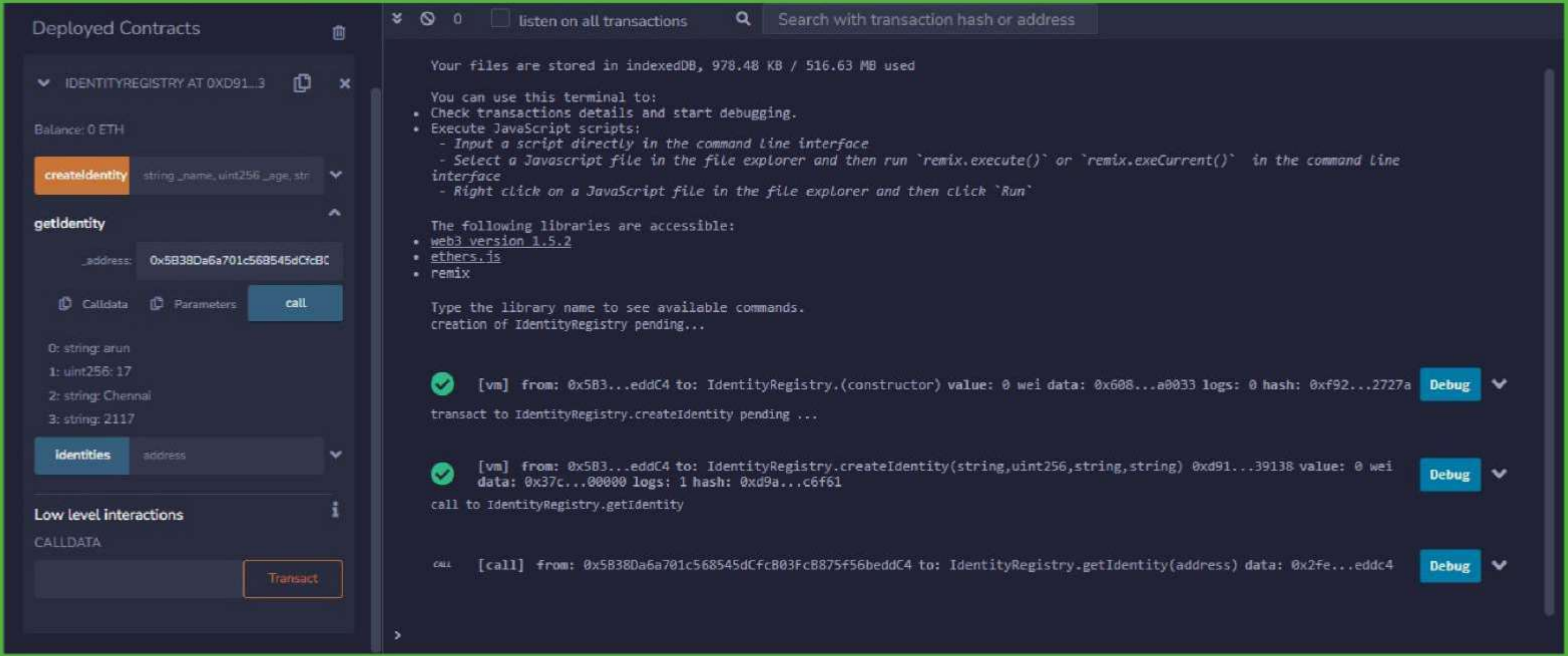
code work on solidity

Elaborate on what you want to discuss.



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract IdentityRegistry {
    // Structure for identity details
    struct Identity {
        string name;
        uint256 age;
        string location;
        string governmentId;
    }

    // List of identities
    mapping(address => Identity) public identities;

    // Event for identity creation
    event identityCreated(
        address indexed owner,
        string name,
        uint256 age,
        string location,
        string governmentId
    );

```

My Work



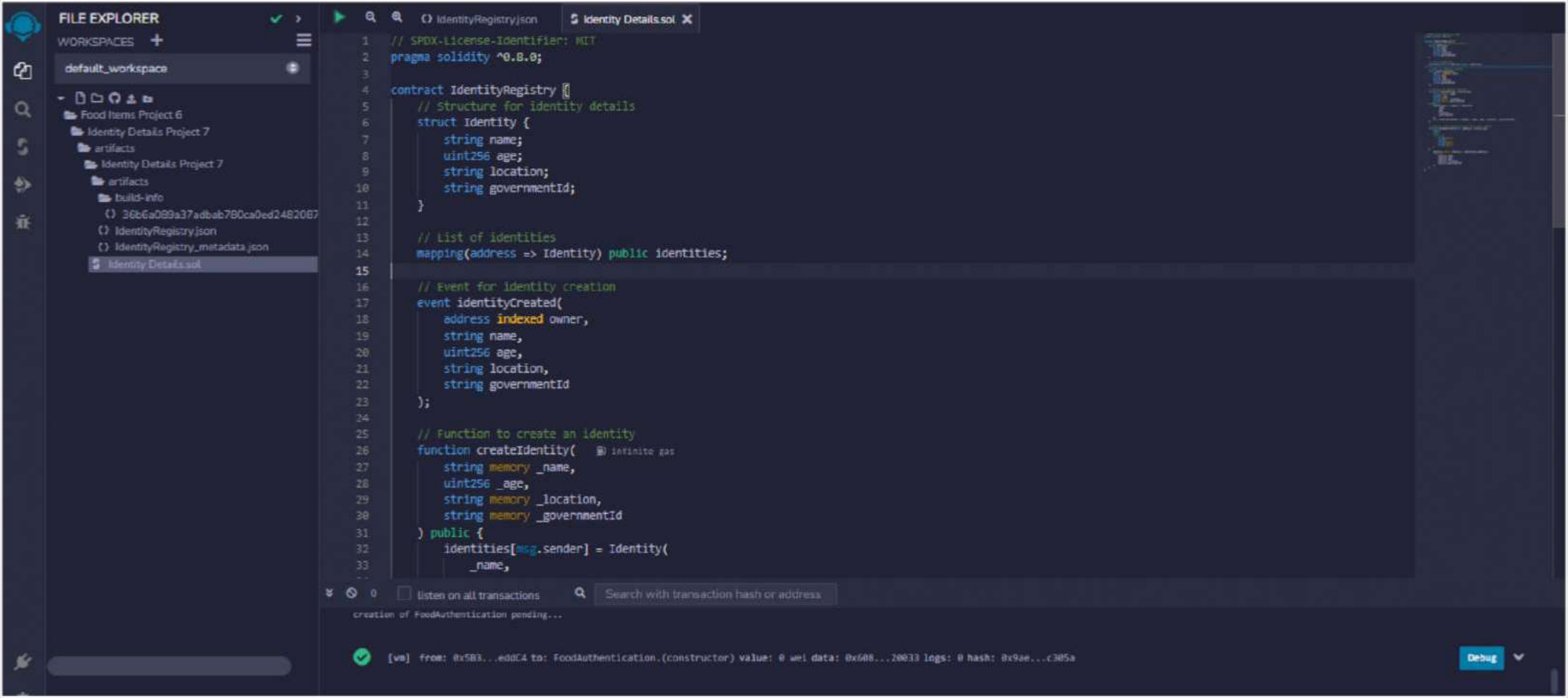
Balance: 0 ETH

_address: 0x5B38Da6a701c568545dCfcB0

```
0: string: arun
1: uint256: 17
2: string: Chennai
3: string: 2117
```

```
Your files are stored in indexedDB, 978.48 KB / 516.63 MB used

You can use this terminal to:
• Check transactions details and start debugging.
• Execute JavaScript scripts:
   - Input a script directly in the command line interface
   - Select a Javascript file in the file explorer and then run `remix.execute()` or `remix.exeCurrent()`  in the command line
interface
   - Right click on a JavaScript file in the file explorer and then click `Run`

The following libraries are accessible:
• web3 version 1.5.2
• ethers.js
• remix

Type the library name to see available commands.
creation of IdentityRegistry pending...

   [vm]  from: 0x5B3...eddC4 to: IdentityRegistry.(constructor) value: 0 wei data: 0x608...a0033 logs: 0 hash: 0xf92...2727a
transact to IdentityRegistry.createIdentity pending ...

   [vm]  from: 0x5B3...eddC4 to: IdentityRegistry.createIdentity(string,uint256,string,string) 0xd91...39138 value: 0 wei
   data: 0x37c...00000 logs: 1 hash: 0xd9a...c6f61
call to IdentityRegistry.getIdentity

CALL [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: IdentityRegistry.getIdentity(address) data: 0x2fe...eddc4

>
```

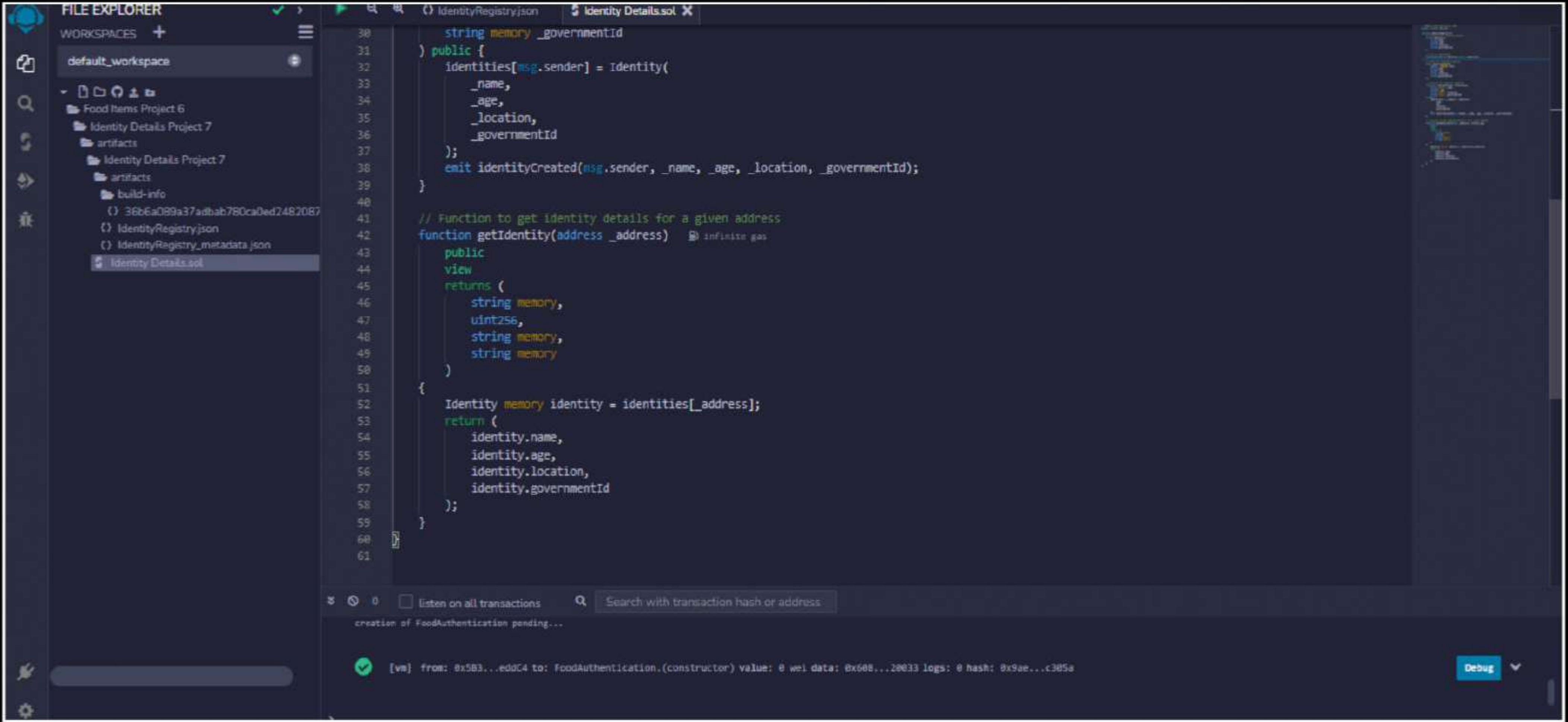
What can you say about
your projects? Share it here!



```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract IdentityRegistry {
    // Structure for identity details
    struct Identity {
        string name;
        uint256 age;
        string location;
        string governmentId;
    }

    // List of identities
    mapping(address => Identity) public identities;

    // Event for identity creation
    event identityCreated(
        address indexed owner,
        string name,
        uint256 age,
        string location,
        string governmentId
    );

    // Function to create an identity
    function createIdentity(
        string memory _name,
        uint256 _age,
        string memory _location,
        string memory _governmentId
    ) public {
        identities[msg.sender] = Identity(
            _name,
```

```
creation of FoodAuthentication pending...

[vm] from: 0x5B3...eddC4 to: FoodAuthentication.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0x9ae...c305a
```

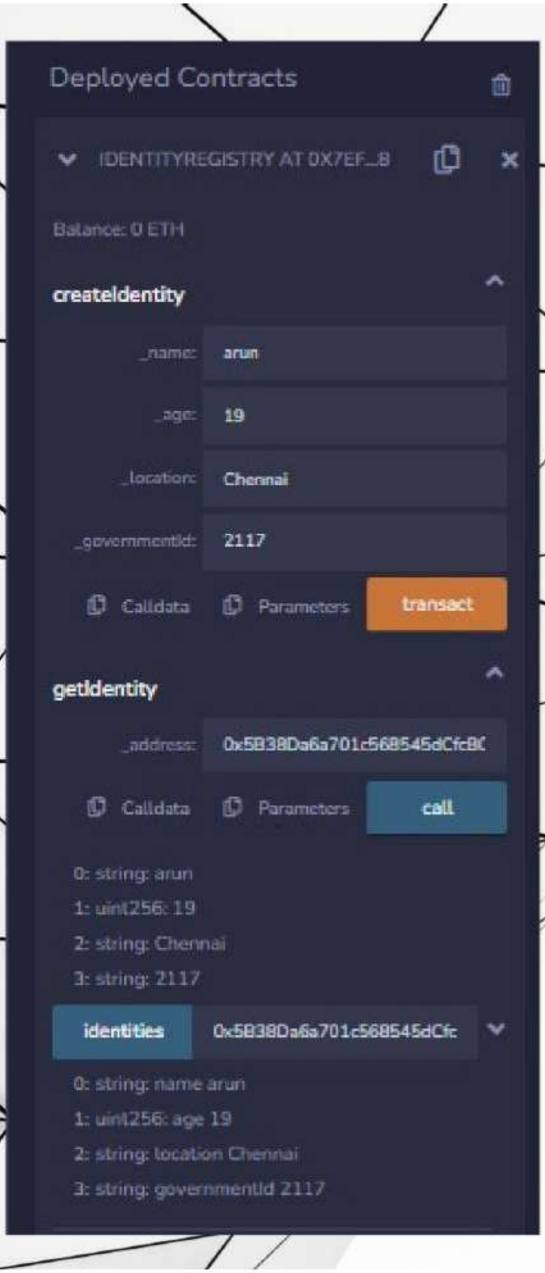
CODE :



```
        string memory _governmentId
    ) public {
        identities[msg.sender] = Identity(
            _name,
            _age,
            _location,
            _governmentId
        );
        emit identityCreated(msg.sender, _name, _age, _location, _governmentId);
    }

    // Function to get identity details for a given address
    function getIdentity(address _address)
        public
        view
        returns (
            string memory,
            uint256,
            string memory,
            string memory
        )
    {
        Identity memory identity = identities[_address];
        return (
            identity.name,
            identity.age,
            identity.location,
            identity.governmentId
        );
    }
}
```

```
creation of FoodAuthentication pending...

[vm] from: 0x5B3...eddC4 to: FoodAuthentication.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0x9ae...c305a
```

finally our work & output



Balance: 0 ETH

createIdentity

_name: arun

_age: 19

_location: Chennai

_governmentId: 2117

getIdentity

_address: 0x5B38Da6a701c568545dCfcB0

```
0: string: arun
1: uint256: 19
2: string: Chennai
3: string: 2117
```

identities 0x5B38Da6a701c568545dCfc

```
0: string: name arun
1: uint256: age 19
2: string: location Chennai
3: string: governmentId 2117
```

Thank You