



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

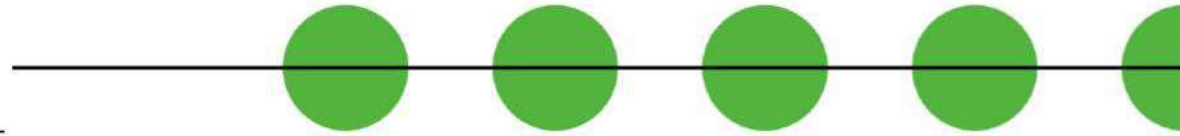
SEMESTER:04

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

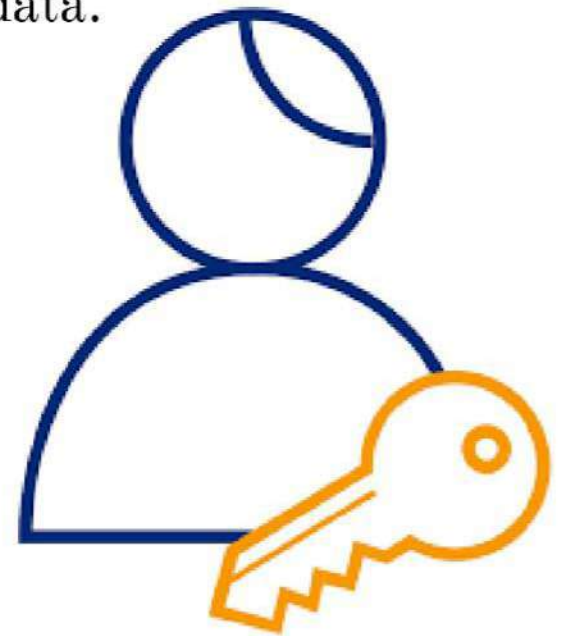
GUIDED BY:Ms. S.H. ANNIE SILVIYA

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

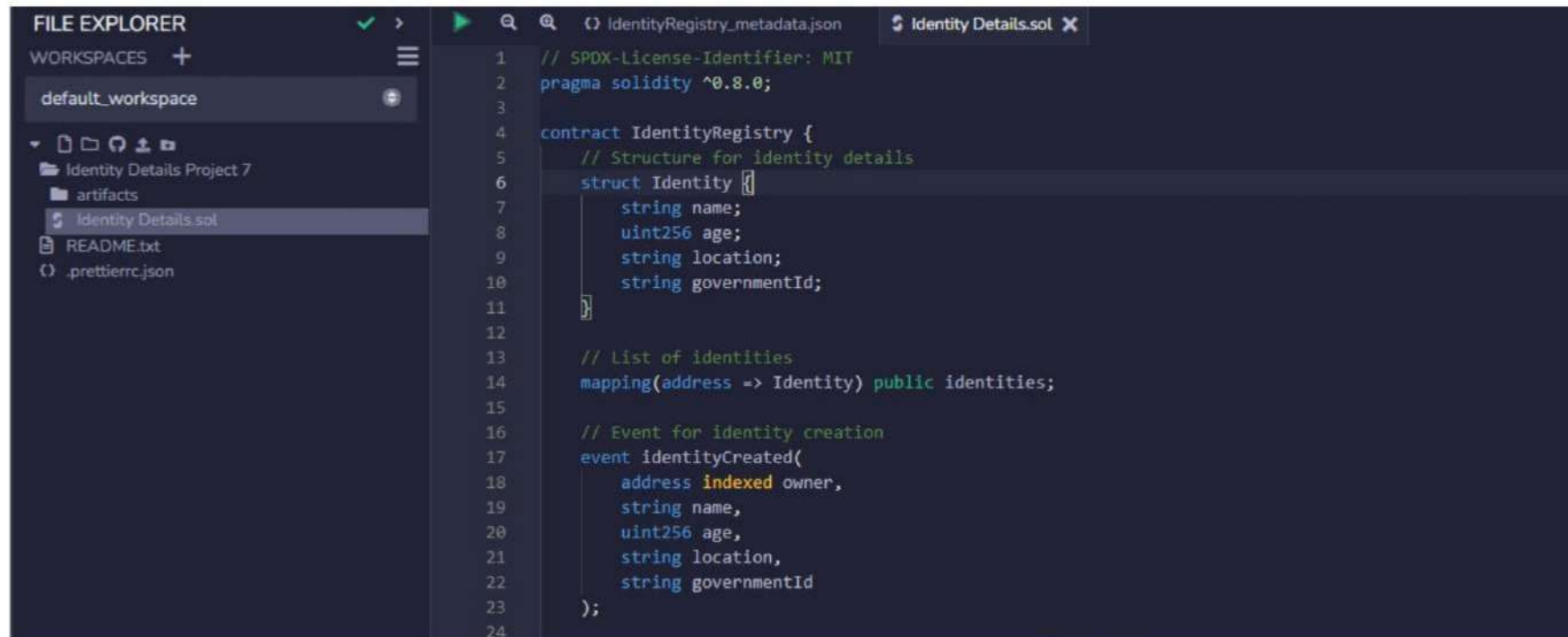


With a blockchain-based identity system, individuals can create and manage their own digital identities without the need for intermediaries. This can give users greater control over their personal data and how it is shared.



code work on solidity

Elaborate on what you want to discuss.



The screenshot shows a code editor with a dark theme. On the left is the 'FILE EXPLORER' panel showing a workspace named 'default_workspace'. Inside, there is a folder 'Identity Details Project 7' containing 'artifacts', 'Identity Details.sol' (selected), 'README.txt', and '.prettierrc.json'. The main editor area displays the content of 'IdentityDetails.sol'. The code is Solidity and defines a contract 'IdentityRegistry'. It includes a license comment, a pragma statement for Solidity version ^0.8.0, a struct 'Identity' with fields 'name', 'age', 'location', and 'governmentId', a public mapping 'identities' indexed by address, and an event 'identityCreated' with parameters 'indexed owner', 'name', 'age', 'location', and 'governmentId'.

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract IdentityRegistry {
5     // Structure for identity details
6     struct Identity {
7         string name;
8         uint256 age;
9         string location;
10        string governmentId;
11    }
12
13    // List of identities
14    mapping(address => Identity) public identities;
15
16    // Event for identity creation
17    event identityCreated(
18        address indexed owner,
19        string name,
20        uint256 age,
21        string location,
22        string governmentId
23    );
24 }
```

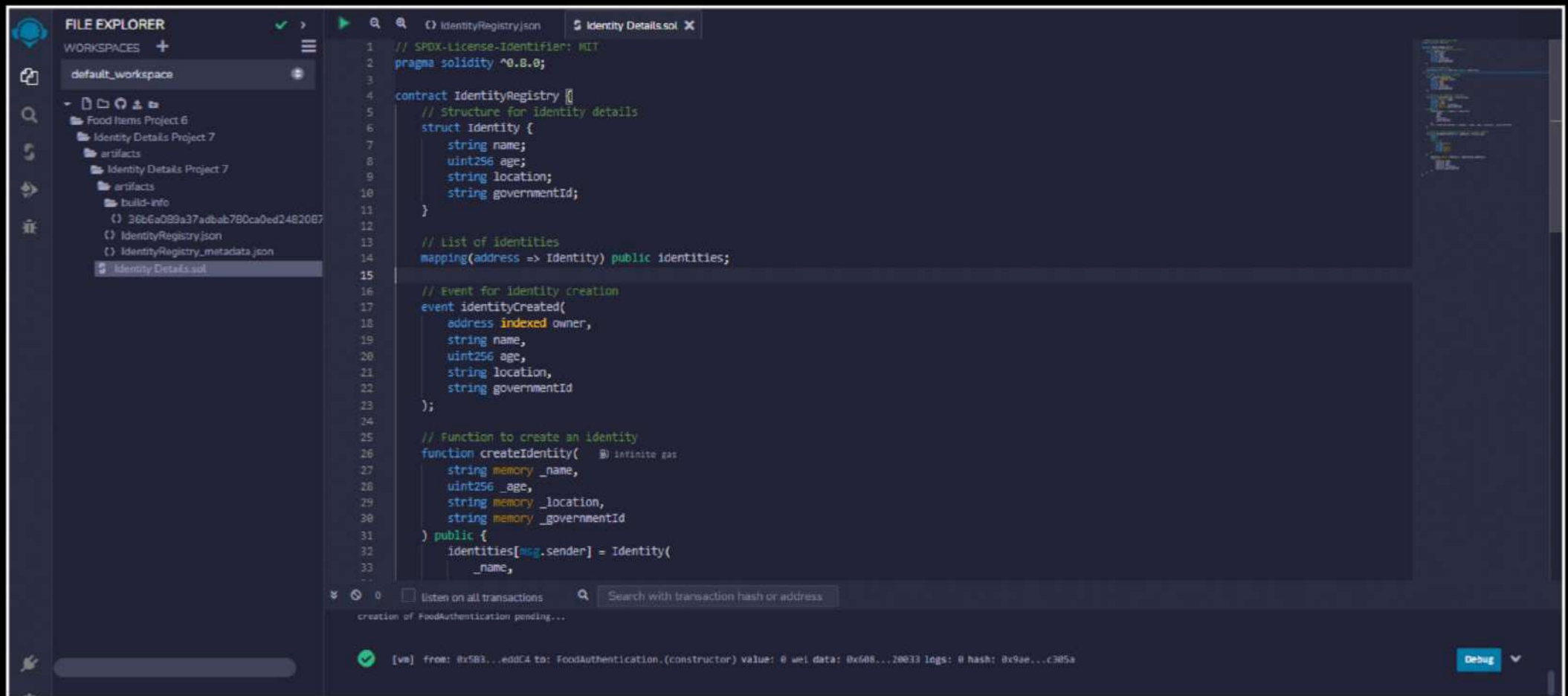
My Work

The screenshot displays the Remix IDE interface. On the left, the 'Deployed Contracts' panel shows a contract named 'IDENTITYREGISTRY-AT 0XD91...3'. The contract's balance is 0 ETH. It features two functions: 'createIdentity' and 'getIdentity'. The 'getIdentity' function is currently selected, with its parameters set to: `_address: 0x5B38Da6a701c568545dCfC8C...`. Below the function parameters, there is a 'call' button and a 'Transact' button. The 'Low level interactions' section shows the 'CALLDATA' field.

The main panel on the right displays the terminal output. It starts with a message: 'Your files are stored in indexedDB, 978.48 KB / 516.63 MB used'. Below this, it lists the libraries accessible: 'web3 version 1.5.2', 'ethers.js', and 'remix'. The terminal shows the following execution steps:

- A VM execution from `0x5B3...eddC4` to `IdentityRegistry.(constructor)` with a value of 0 wei and data `0x608...a0033`. The logs show `0` and the hash `0xf92...2727a`. The transaction is pending.
- A VM execution from `0x5B3...eddC4` to `IdentityRegistry.createIdentity(string,uint256,string,string)` with a value of 0 wei and data `0x37c...00000`. The logs show `1` and the hash `0xd9a...c6f61`. The transaction is pending.
- A call from `0x5B38Da6a701c568545dCfC803Fc8875f56beddC4` to `IdentityRegistry.getIdentity(address)` with data `0x2fe...eddc4`.

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



CODE:

FILE EXPLORER

WORKSPACES +

default_workspace

Food Items Project 6

Identity Details Project 7

Identity Details Project 7

artifacts

build-info

36b6a089a37adbab780ca0ed2482087

IdentityRegistry.json

IdentityRegistry_metadata.json

IdentityDetails.sol

IdentityRegistry.json

IdentityDetails.sol

```
30 string memory _governmentId
31 } public {
32     identities[msg.sender] = Identity(
33         _name,
34         _age,
35         _location,
36         _governmentId
37     );
38     emit identityCreated(msg.sender, _name, _age, _location, _governmentId);
39 }
40
41 // Function to get identity details for a given address
42 function getIdentity(address _address) infinite gas
43 public
44 view
45 returns (
46     string memory,
47     uint256,
48     string memory,
49     string memory
50 )
51 {
52     Identity memory identity = identities[_address];
53     return (
54         identity.name,
55         identity.age,
56         identity.location,
57         identity.governmentId
58     );
59 }
60
61
```

0

☐ Listen on all transactions

creation of FoodAuthentication pending...

✓ [vm] from: 0x5B3...eddC4 to: FoodAuthentication.(constructor) value: 0 wei data: 0x608...20E33 logs: 0 hash: 0x9ae...c305a

Debug

finally our

work &

output

Deployed Contracts

IDENTITYREGISTRY AT 0X7EF...8

Balance: 0 ETH

createIdentity

_name:

arun

_age:

19

_location:

Chennai

_governmentId:

2117

Calldata

Parameters

transact

getIdentity

_address:

0x5B38Da6a701c568545dCfc8C

Calldata

Parameters

call

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

Identities

0x5B38Da6a701c568545dCfc8C

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

Thank You

