

Project Design Phase-I

Solution Architecture

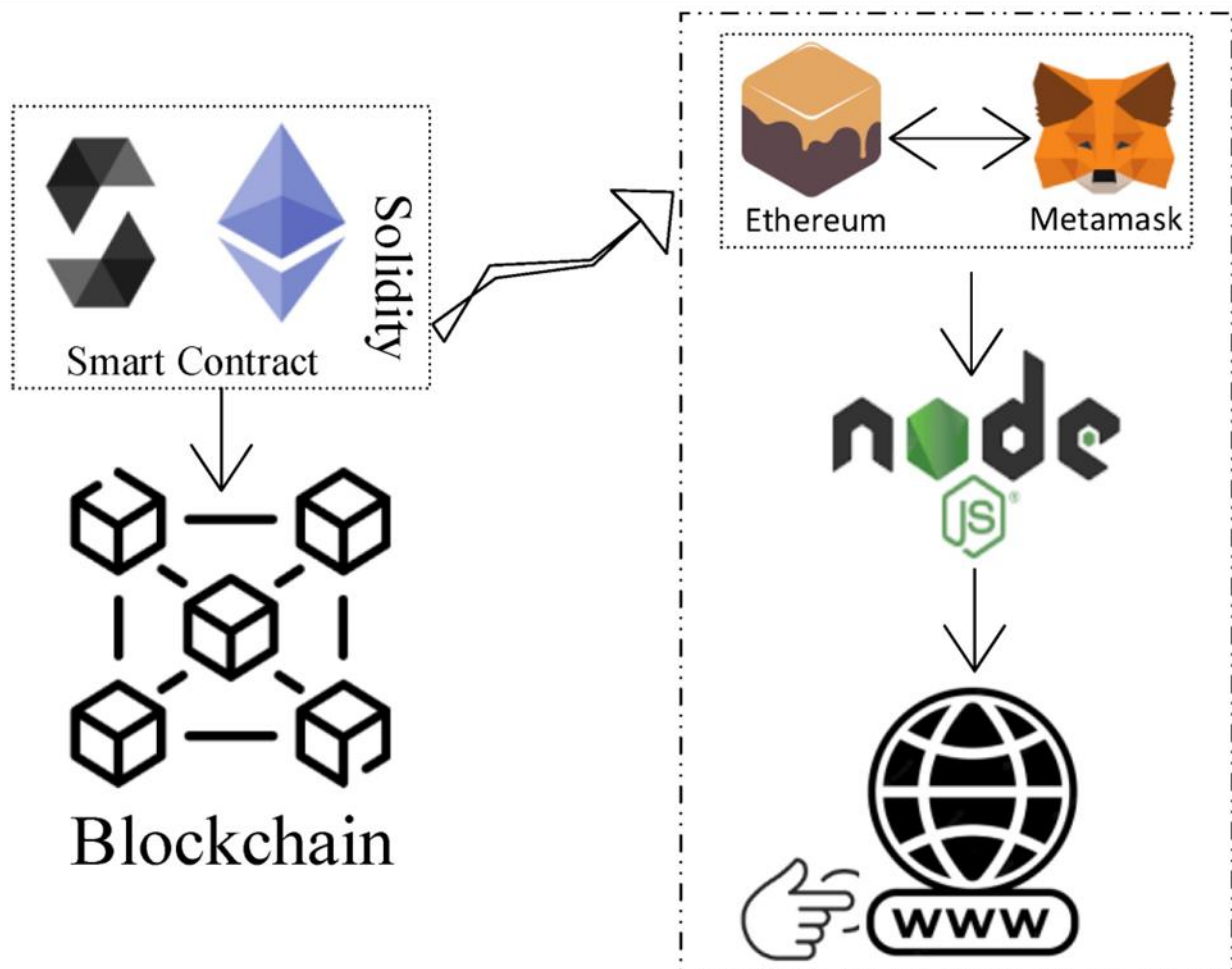
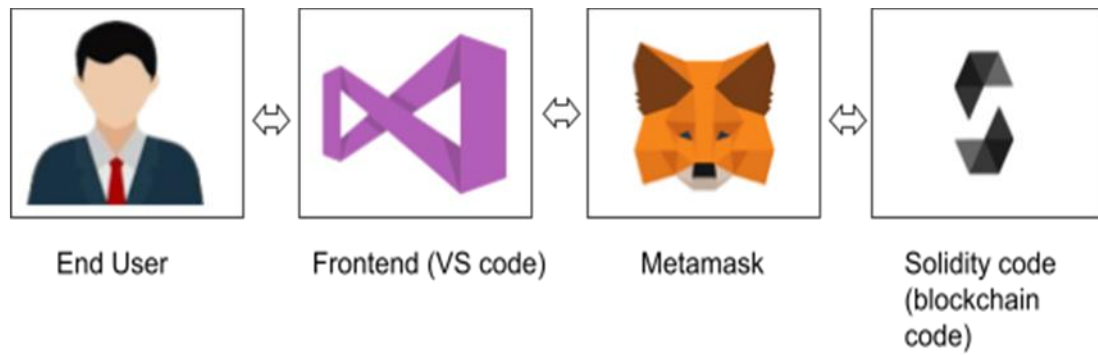
Date	19 September 2022
TEAM ID	75AAE42A61B3A6C9B9D4C495DC4F2542
Project Name	Transparent Education Data Management
Maximum Marks	4 Marks

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

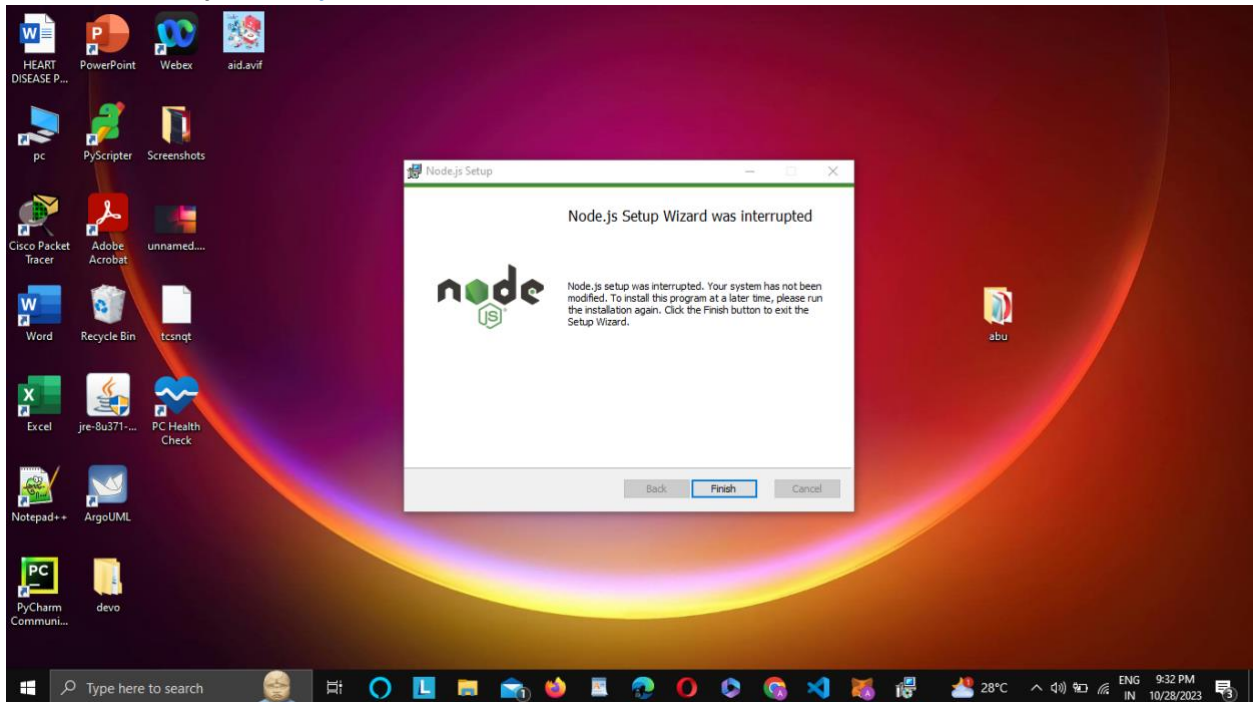
- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram:

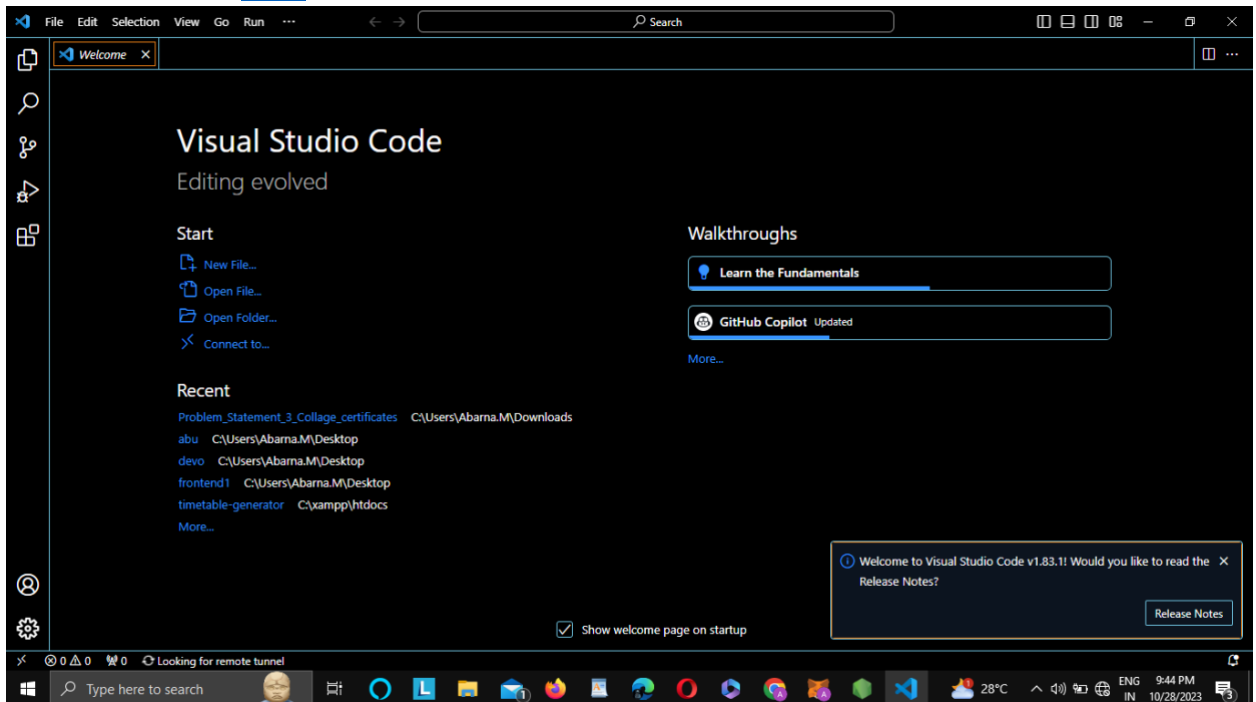


Prerequisite

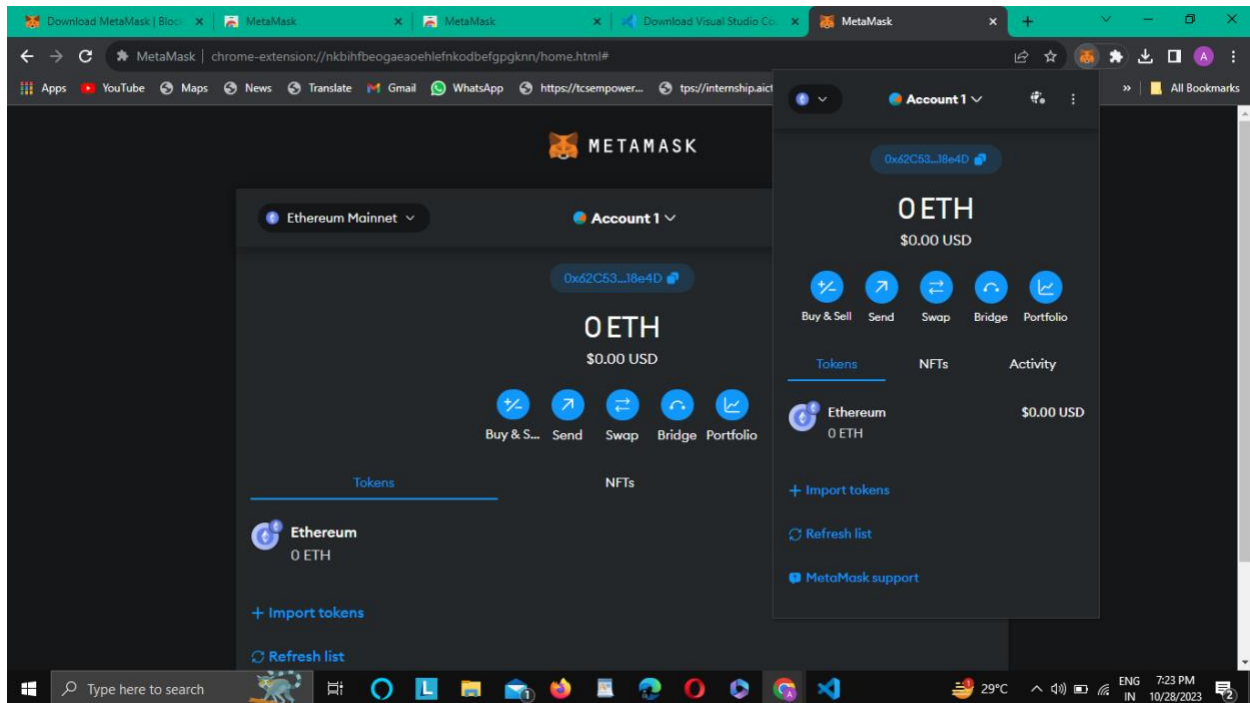
1 download node.js : [Node.js](#)



2 download vs code: [Li4nk](#)



3 download metamask : <https://metamask.io/>

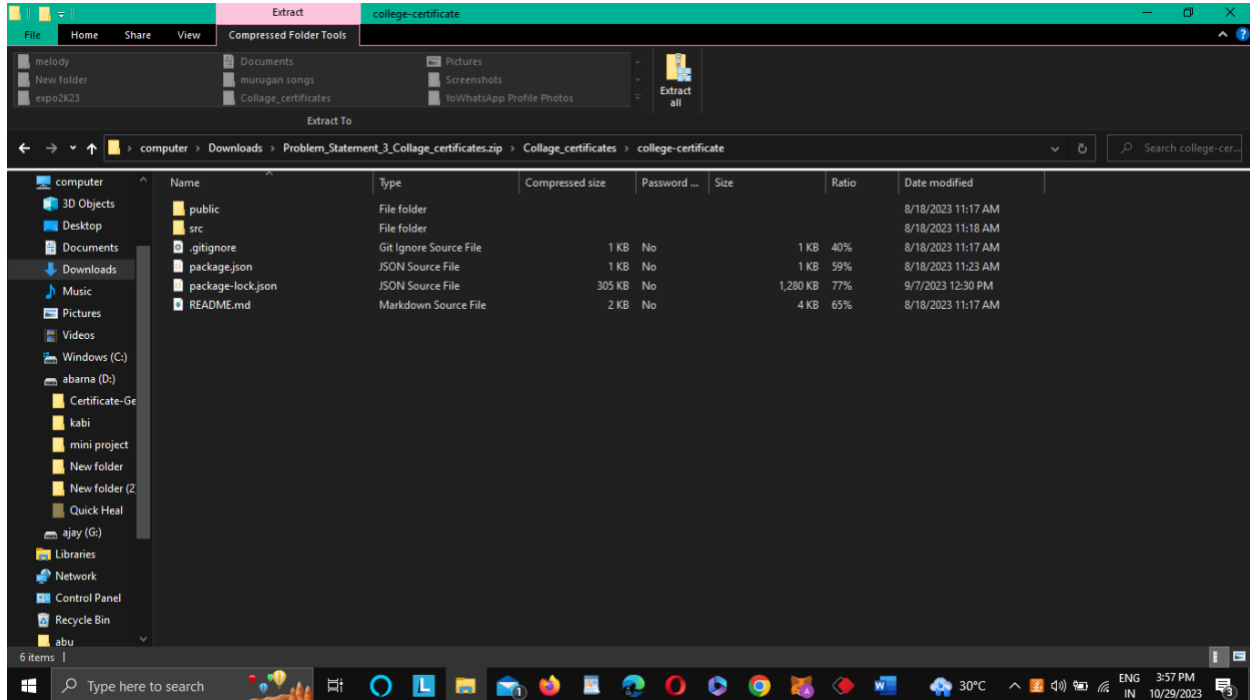


Steps to complete the project

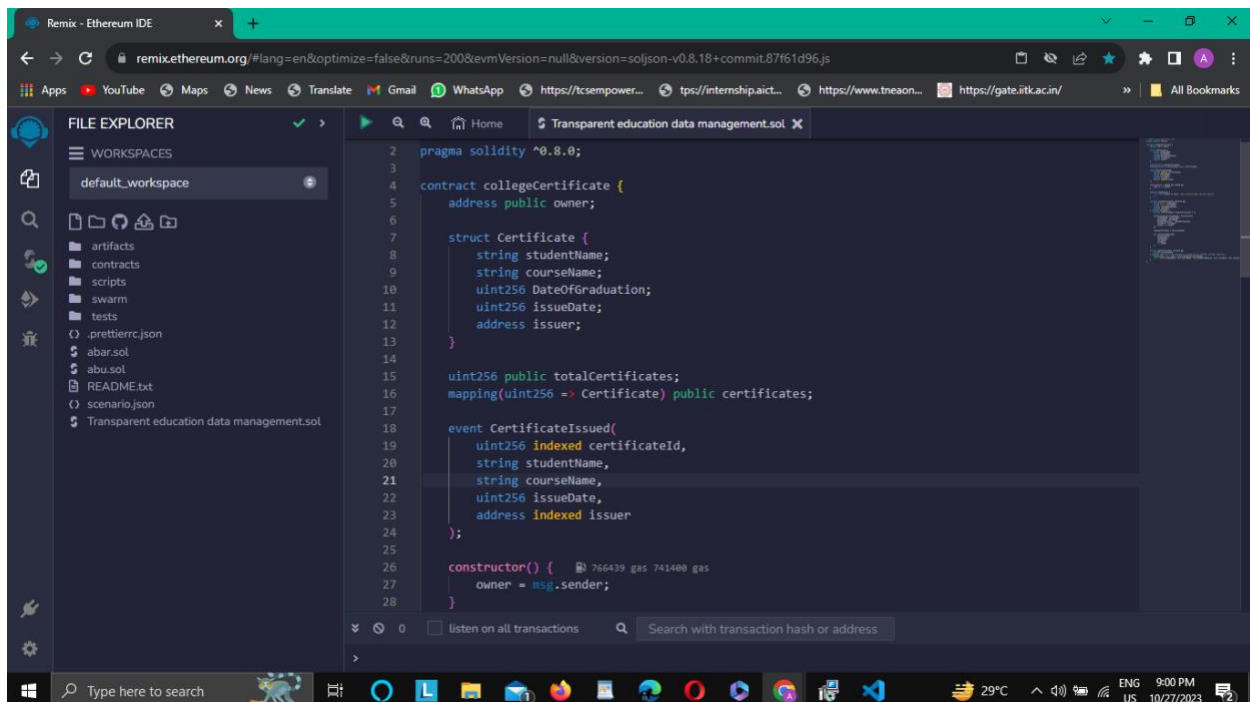
Step 1:-

1. Open the Zip file and download the zip file.

Extract all zip files

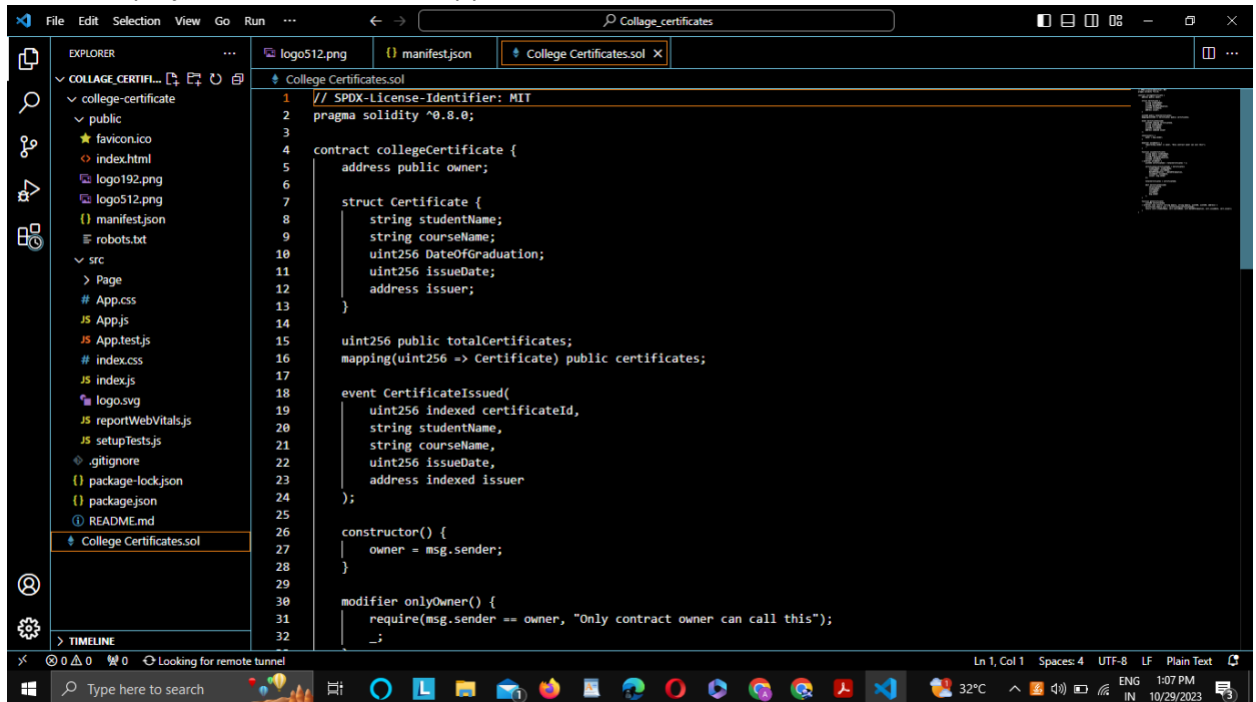


Step 2 : Select the projectname.sol file and copy the code.



Step 2 :

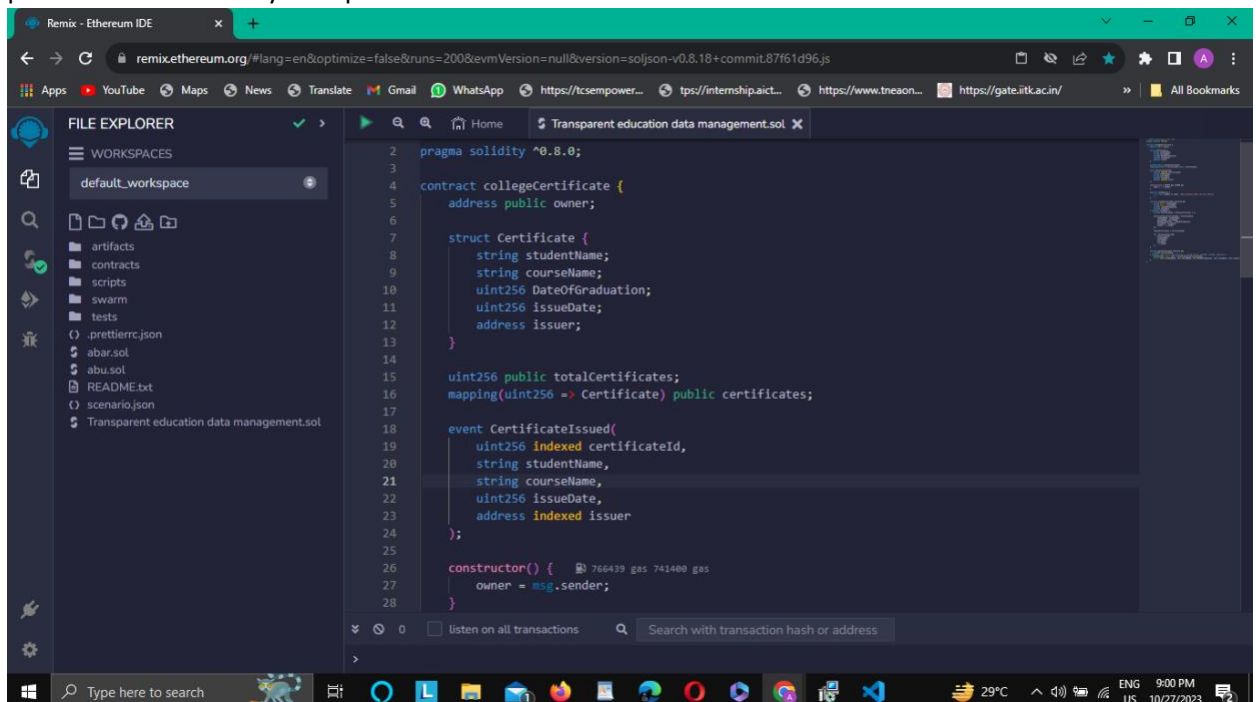
1. Open vs code in the left top select open folder. Select extracted file and open .
- 2 Select the projectname.sol file and copy the code.



The screenshot shows the Visual Studio Code interface with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a folder named 'COLLAGE_CERTIFI...' containing files like 'favicon.ico', 'index.html', 'logo192.png', 'logo512.png', 'manifest.json', 'robots.txt', and a 'src' folder. The code editor displays the content of 'College Certificates.sol'.

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract collegeCertificate {
5     address public owner;
6
7     struct Certificate {
8         string studentName;
9         string courseName;
10        uint256 DateOfGraduation;
11        uint256 issueDate;
12        address issuer;
13    }
14
15    uint256 public totalCertificates;
16    mapping(uint256 => Certificate) public certificates;
17
18    event CertificateIssued(
19        uint256 indexed certificateId,
20        string studentName,
21        string courseName,
22        uint256 issueDate,
23        address indexed issuer
24    );
25
26    constructor() {
27        owner = msg.sender;
28    }
29
30    modifier onlyOwner() {
31        require(msg.sender == owner, "Only contract owner can call this");
32    }
33 }
```

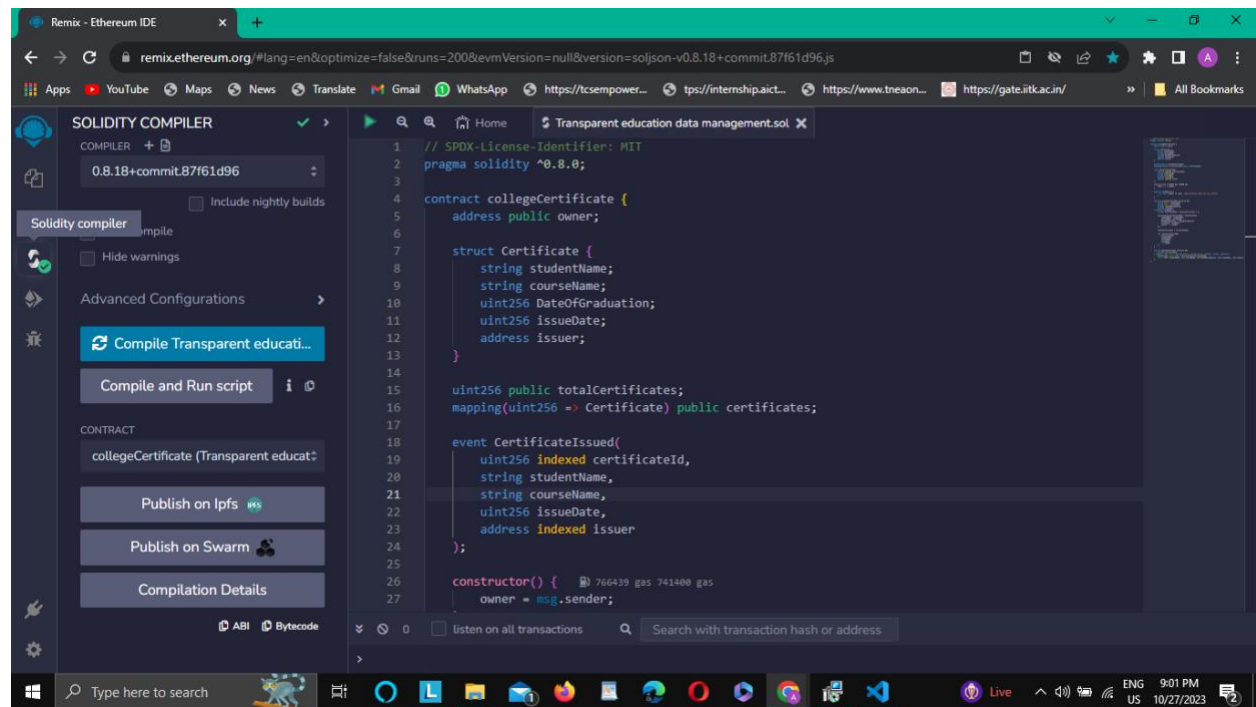
- 3 Open the remix ide platform and create a new file by giving the name of projectname.sol and paste the code which you copied from vs code.



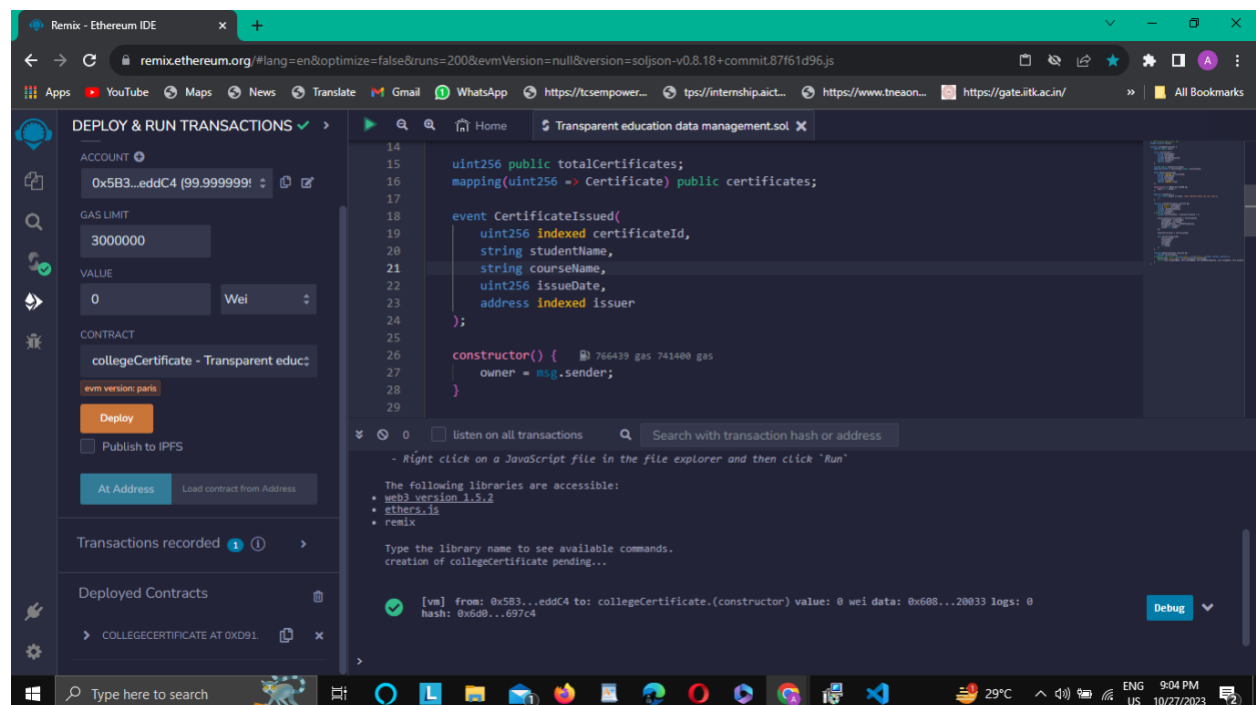
The screenshot shows the Remix IDE interface with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a folder named 'default_workspace' containing files like 'artifacts', 'contracts', 'scripts', 'swarm', 'tests', '.prettierrc.json', 'abar.sol', 'abu.sol', 'README.txt', 'scenario.json', and 'Transparent education data management.sol'. The code editor displays the content of 'Transparent education data management.sol'.

```
2 pragma solidity ^0.8.0;
3
4 contract collegeCertificate {
5     address public owner;
6
7     struct Certificate {
8         string studentName;
9         string courseName;
10        uint256 DateOfGraduation;
11        uint256 issueDate;
12        address issuer;
13    }
14
15    uint256 public totalCertificates;
16    mapping(uint256 => Certificate) public certificates;
17
18    event CertificateIssued(
19        uint256 indexed certificateId,
20        string studentName,
21        string courseName,
22        uint256 issueDate,
23        address indexed issuer
24    );
25
26    constructor() {
27        owner = msg.sender;
28    }
29 }
```

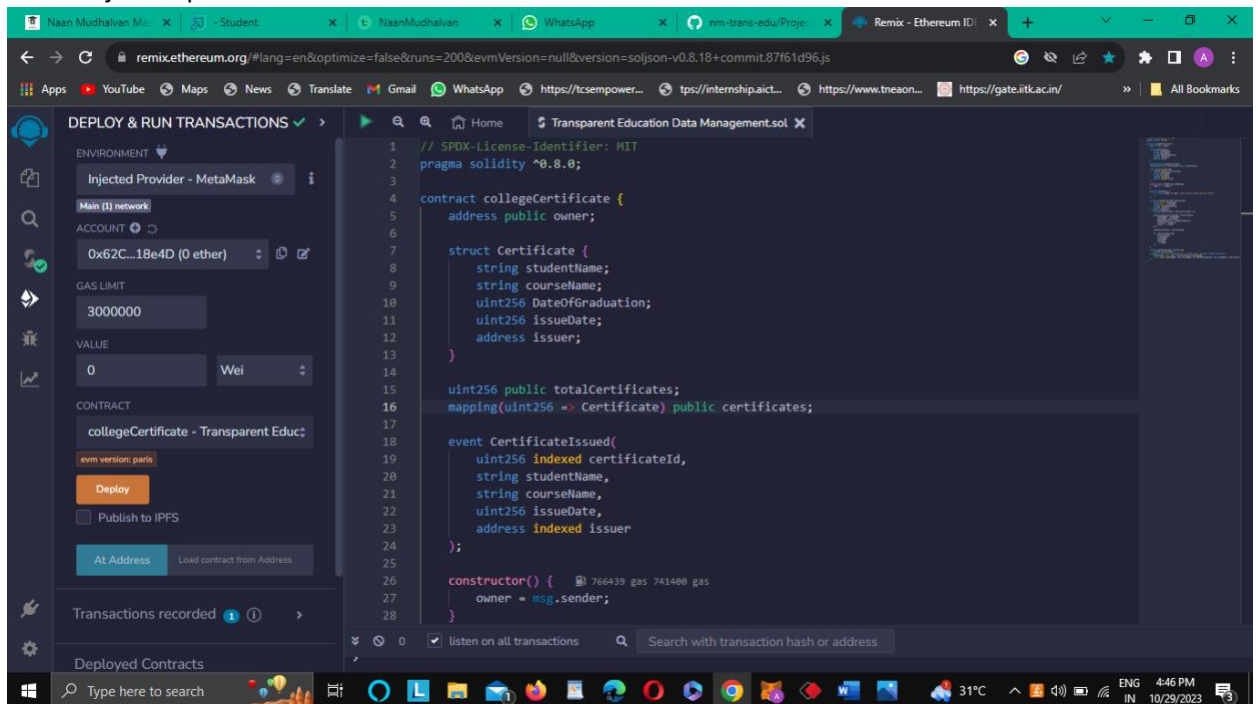
4 Click on solidity compiler and click compile the projectname.sol



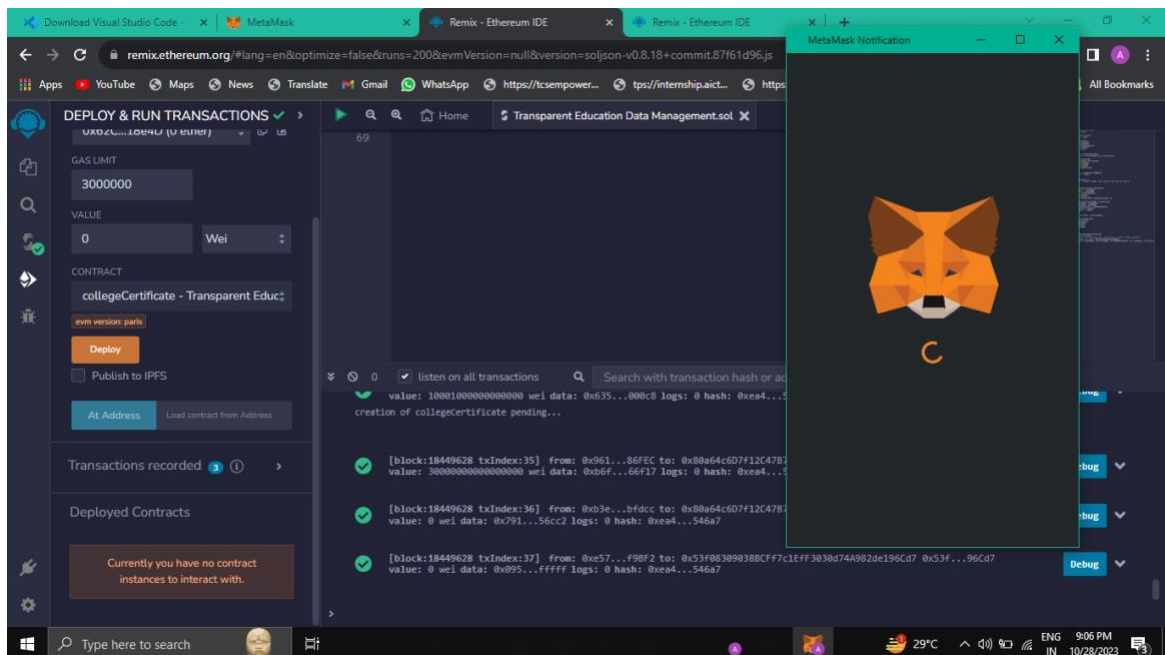
5 Deploy the smart contract by clicking on the deploy and run transaction.



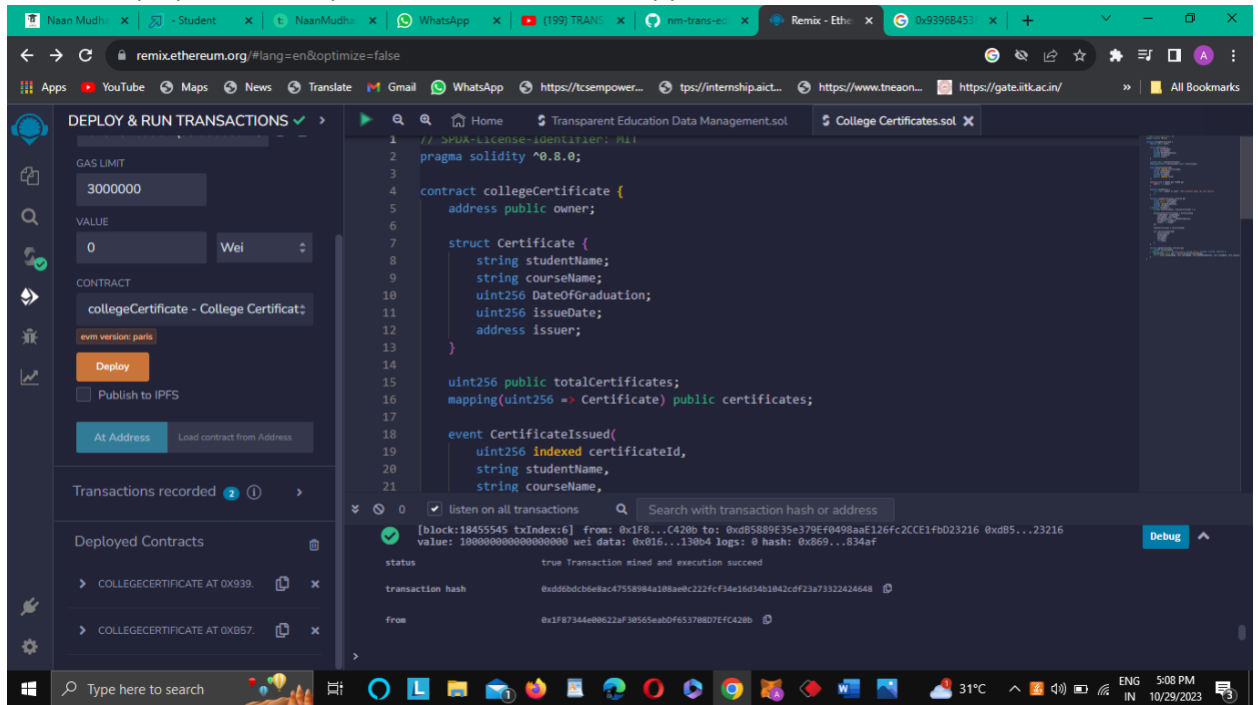
6 select injected provider - MetaMask. In environment



7 Click on deploy. Automatically MetaMask will open and give confirmation. You will get a pop up click on ok



- 8 In the Deployed contract you can see one address copy the address.

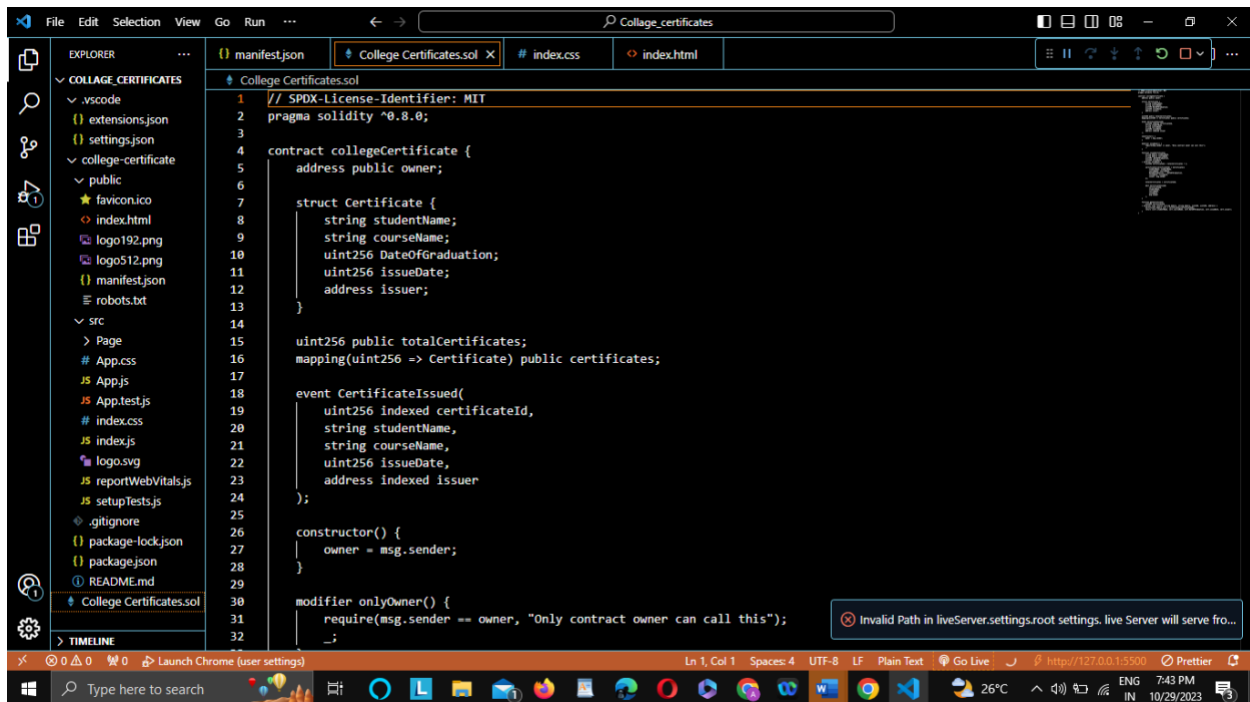


- 9 Open vs code and search for the connector.js. In contract.js you can paste the address at the bottom of the code. In export const address.

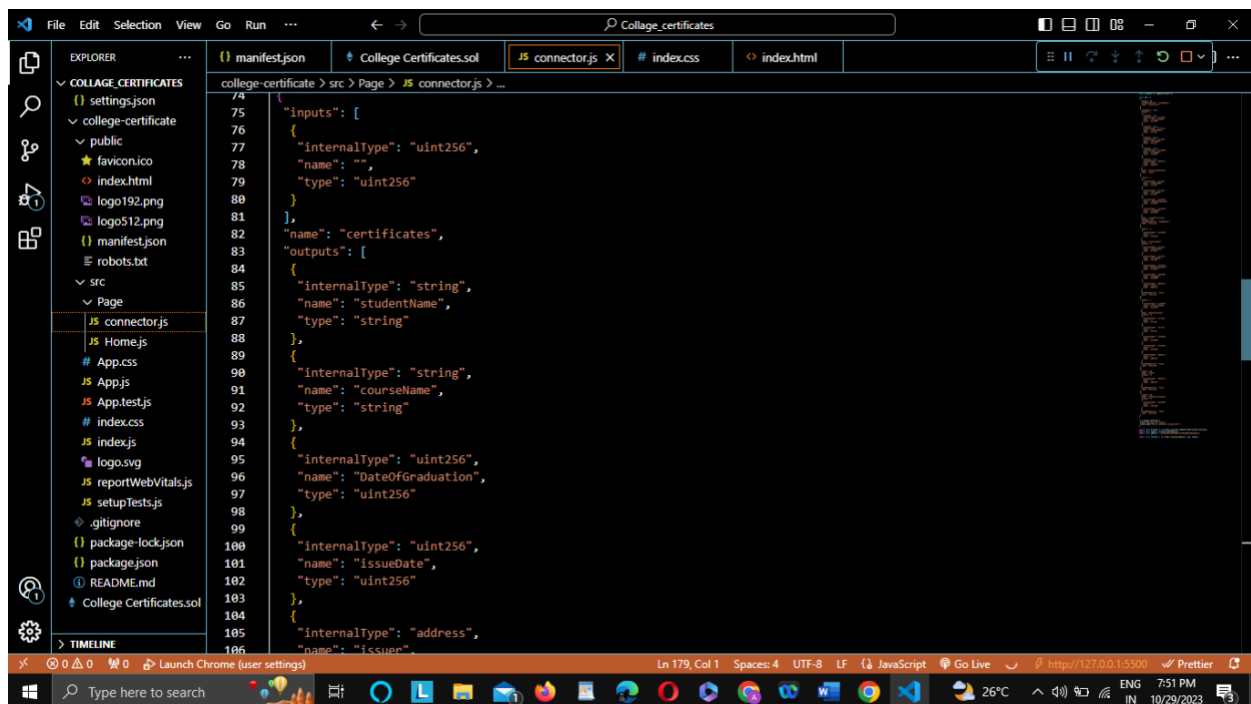
- 10 Save the code

Step 3: open file explorer

1. Open the extracted file and click on the folder.



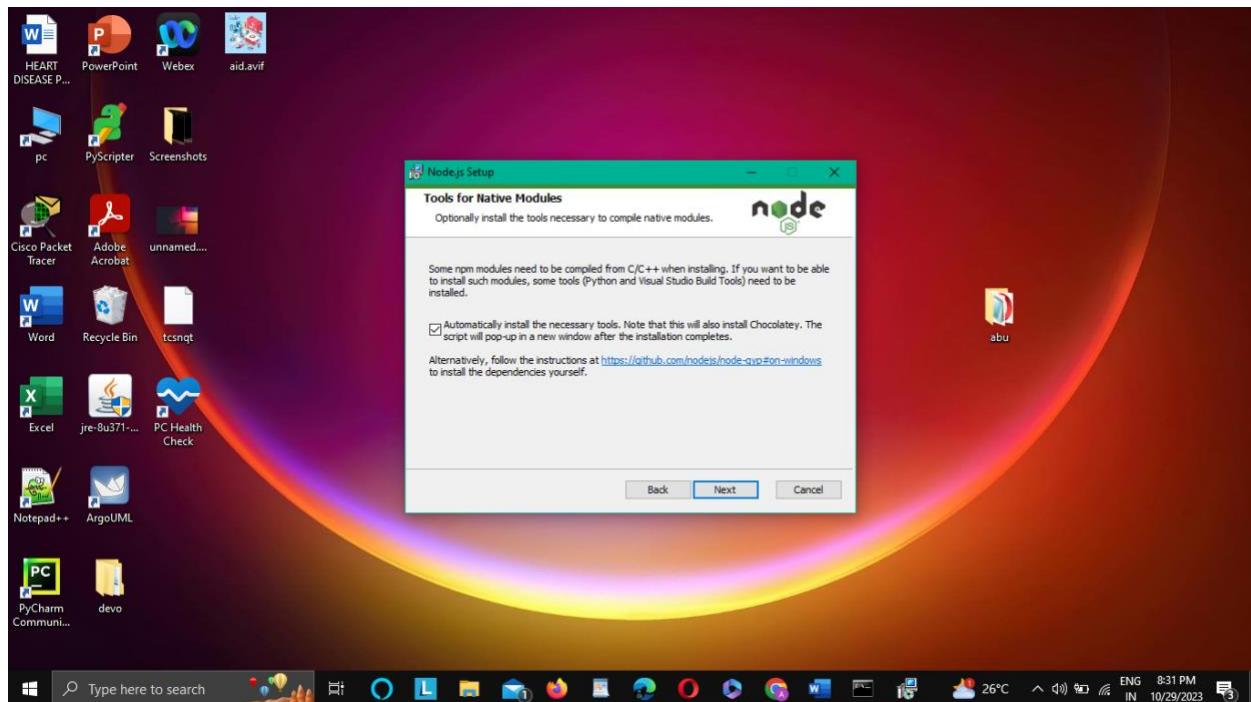
2. Open src, and search for utiles.



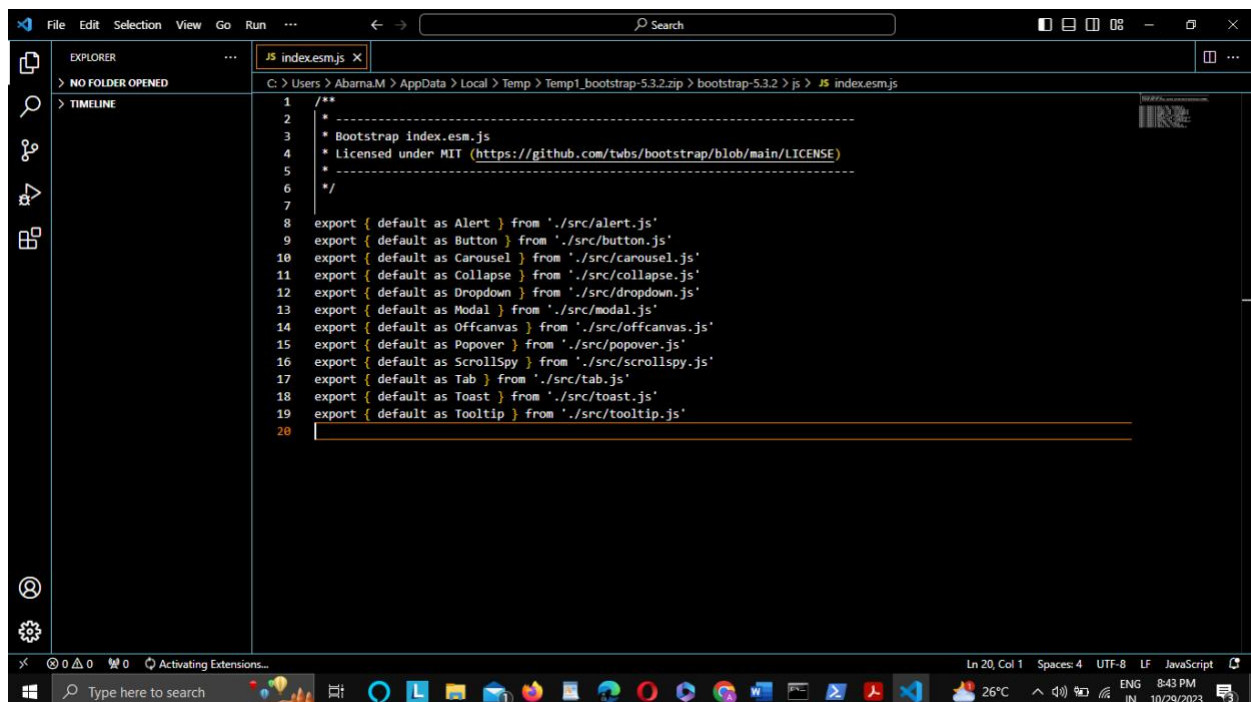
3. You can see the frontend files. Select all the things at the top in the search bar by clicking alt+ A. Search for cm

3. Open cmd enter commands

npm install

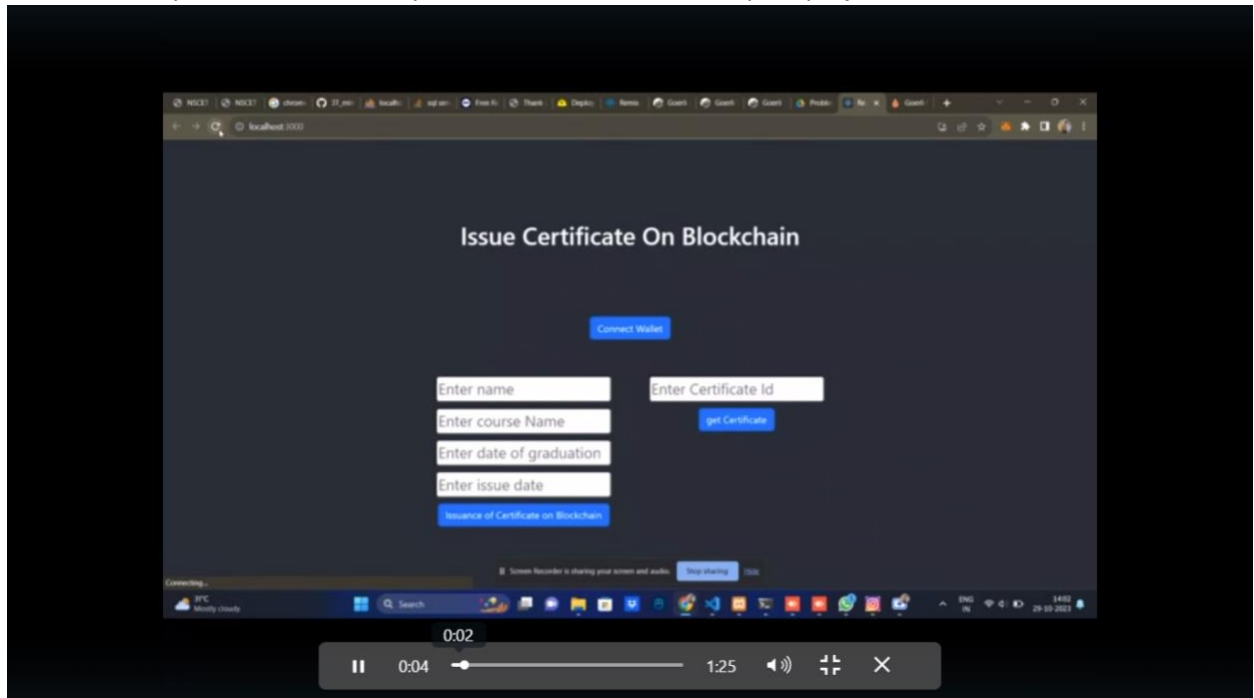


npm bootstrap



npm start

4. It will install all the packages and after completing it will open {LOCALHOST IP ADDRESS} copy the address and open it to chrome so you can see the frontend of your project



5.

