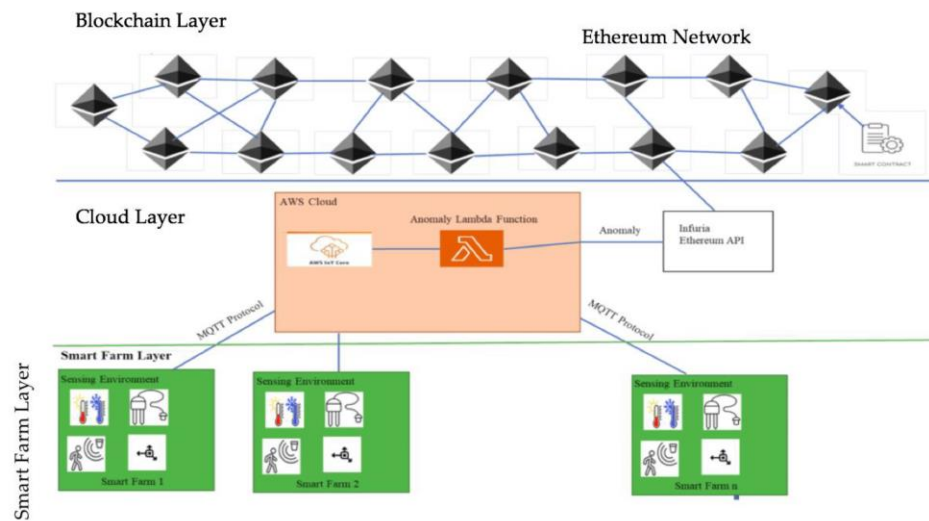


## Solution Architecture

Date	4 Nov 2023
Team ID	NM2023TMID04532
Project Name	Climate tracksmart using blockchain
Maximum Marks	4 Marks

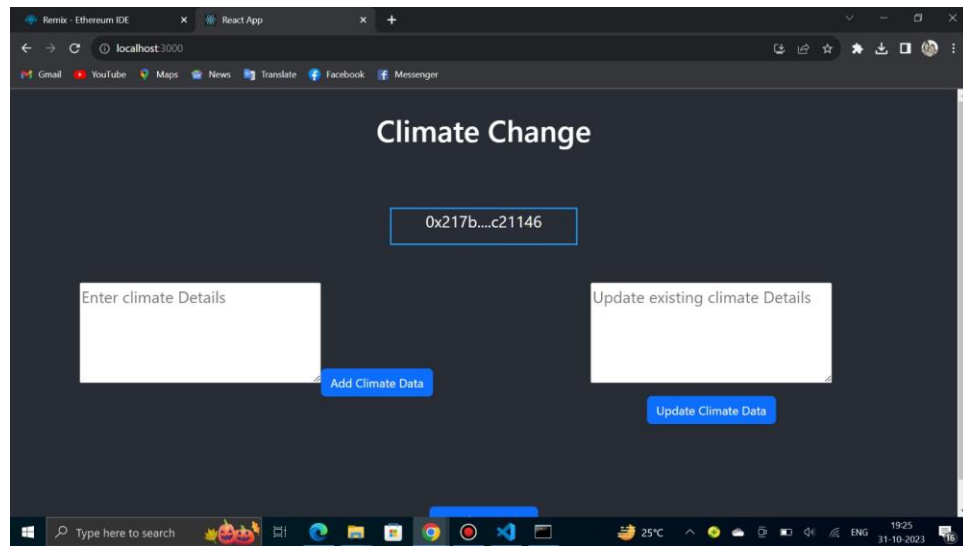
Solution Architecture:



Output Screenshot:

Local Host IP Address: <http://localhost:3000/>

Screenshot of Output:



Video Demo Link: <https://youtu.be/lwk3jLIREnw?si=xOYdPgSHKVZbwdob>



**Source Code 1:**

```

pragma solidity ^0.8.0;

contract climateChange{

    struct ClimateData {
        uint timestamp;
        string details;
    }

    mapping(address => ClimateData) public climateRecords;

    function addClimateData(string memory details) public {
        ClimateData memory newData = ClimateData(block.timestamp, details);
        climateRecords[msg.sender] = newData;
    }

    function getClimateData() public view returns (ClimateData memory) {
        return climateRecords[msg.sender];
    }

    function updateClimateData(string memory details) public {
        climateRecords[msg.sender].details = details;
    }

    // These are very basic functions written to carry out the operation

}

```

### Source Code 2:

```

const { ethers } = require("ethers");

```

```
const abi = [
{
  "inputs": [
    {
      "internalType": "string",
      "name": "details",
      "type": "string"
    }
  ],
  "name": "addClimateData",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
},
{
  "inputs": [
    {
      "internalType": "address",
      "name": "",
      "type": "address"
    }
  ],
  "name": "climateRecords",
  "outputs": [
    {
      "internalType": "uint256",
      "name": "timestamp",
      "type": "uint256"
    }
  ]
}
```

```
    },  
    {  
      "internalType": "string",  
      "name": "details",  
      "type": "string"  
    }  
  ],  
  "stateMutability": "view",  
  "type": "function"  
},  
{  
  "inputs": [],  
  "name": "getClimateData",  
  "outputs": [  
    {  
      "components": [  
        {  
          "internalType": "uint256",  
          "name": "timestamp",  
          "type": "uint256"  
        },  
        {  
          "internalType": "string",  
          "name": "details",  
          "type": "string"  
        }  
      ],  
      "type": "tuple"  
    }  
  ],  
  "internalType": "struct climateChange.ClimateData",  
  "name": "",
```

```

    "type": "tuple"
  }
],
"stateMutability": "view",
"type": "function"
},
{
  "inputs": [
    {
      "internalType": "string",
      "name": "details",
      "type": "string"
    }
  ],
  "name": "updateClimateData",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
}
]

```

```

if (!window.ethereum) {
  alert('Meta Mask Not Found')
  window.open("https://metamask.io/download/")
}

```

```

export const provider = new ethers.providers.Web3Provider(window.ethereum);
export const signer = provider.getSigner();
export const address = "0x1F57236Ac53e8960eaeFe82d1E2ccaed0833cf09"

```

```
export const contract = new ethers.Contract(address, abi, signer)
```