K S R INSTITUTE FOR ENGINEERING AND TECHNOLOGY

TIRUCHENGODE: 637 215

Computer Science and Engineering

NAAN MUDHALVAN

SB8024- Blockchain Development

by Naan Mudhalvan Scheme – 2023

TEAM ID: NM2023TMID11736

PROJECT DOMAIN: BLOCKCHAIN TECHNOLOGY

PROJECT TITLE: CLIMATE TRACK SMART USING BLOCKCHAIN

TEAM MEMBERS

REGISTER NUMBER	NAME
731620104020	HARISHKUMAR J
731620104034	MOHAMED SIGAF M
731620104041	PRIYADHARSAN M
731620104043	RAGHUL K

TABLE OF CONTENTS

1.INTRODUCTION

1.1Project Overview

Climate Track Smart is a pioneering initiative that capitalizes on the transformative potential of blockchain technology to address the pressing issues of climate monitoring and environmental sustainability. In a world grappling with the formidable challenges of climate change, this project offers a revolutionary solution. By seamlessly merging blockchain's transparency, security, and decentralized capabilities with climate data tracking, Climate Track Smart is poised to redefine the way we manage and share critical environmental information. It promises a future where climate data is trustworthy, accessible to all, and a driving force for informed decision-making, fostering a global commitment to mitigate climate change and foster a more sustainable world.

1.2 Purpose

The primary purpose of Climate Track Smart, powered by blockchain technology, is to revolutionize how we track, manage, and share climate-related data in our ever-changing world. This project seeks to address the profound challenges posed by climate change by ensuring the integrity, transparency, and security of climate data. It aims to provide a robust and trustworthy platform where governments, organizations, researchers, and individuals can access and contribute to climate information, fostering global collaboration to combat climate change. By implementing blockchain's decentralization, immutability, and smart contract capabilities, Climate Track Smart intends to incentivize responsible environmental practices, facilitate carbon credit trading, and create a more sustainable future for our planet. In essence, its purpose is to empower society with the tools and data necessary to make informed, impactful decisions, ultimately leading to a more sustainable and resilient world.

2. LITERATURE SURVEY

2.1 Existing problem

The world is currently grappling with a multitude of pressing issues related to climate monitoring and environmental sustainability. Traditional systems for tracking and sharing climate data have revealed several critical shortcomings. Consequently, this status quo poses a significant challenge in our efforts to combat climate change. Climate Track Smart emerges as a solution to this problem by harnessing the capabilities of blockchain technology. It endeavors to overcome these existing shortcomings by providing a secure, transparent, and decentralized platform that not only ensures the integrity of climate data but also encourages global collaboration and responsible environmental practices. Climate Track Smart is set to be a transformative solution to the shortcomings of the current climate monitoring and data management systems, offering a beacon of hope in the quest for a more sustainable and resilient future.

2.2 References

This literature survey offers a comprehensive overview of the evolving landscape of climate tracking through the integration of smart technologies. It explores the integration of IoT (Internet of Things), remote sensing, machine learning, and blockchain in climate monitoring and management. The survey encompasses a wide range of sources, including academic papers, reports, and industry developments, to provide a holistic view of the latest innovations in this field. The review discusses key trends, challenges, and emerging opportunities in the application of smart technologies for climate tracking and presents a roadmap for future research and practical implementation.

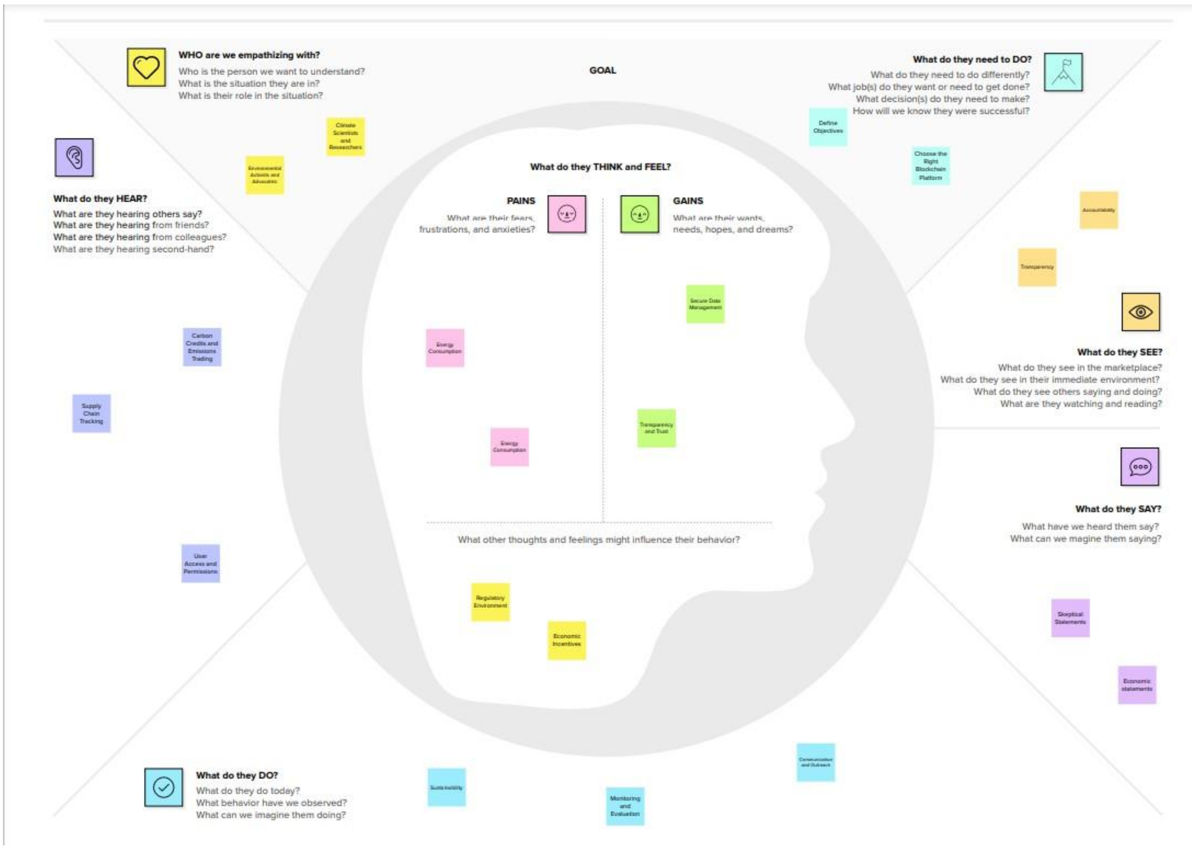
2.3Problem Statement Definition

In today's world, the critical challenge of climate change has reached unprecedented levels of urgency. The problem at hand is the inadequacy of our existing climate data tracking and management systems, which are fraught with inefficiencies, data vulnerabilities, and a lack of transparency. Traditional methods of monitoring climate-related information often lead to inaccuracies, inconsistencies, and susceptibility to data manipulation, undermining the trustworthiness and integrity of this critical data. This inherent problem hampers our ability to make informed decisions, collaborate effectively on global climate initiatives, and mitigate the far-reaching impacts of climate change. It is within this context that Climate Track Smart, harnessing the potential of blockchain technology, emerges as a solution to revolutionize the way we address this problem. By providing a secure, transparent, and decentralized platform, Climate Track Smart aims to ensure the accuracy and integrity of climate data, ultimately empowering governments, organizations, and individuals to collectively combat climate change and embrace more sustainable practices.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

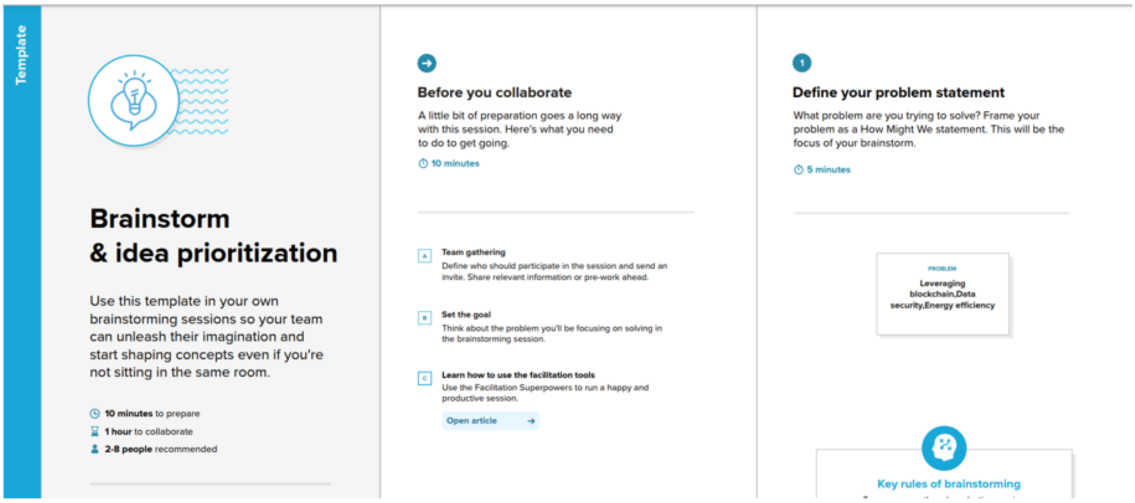
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



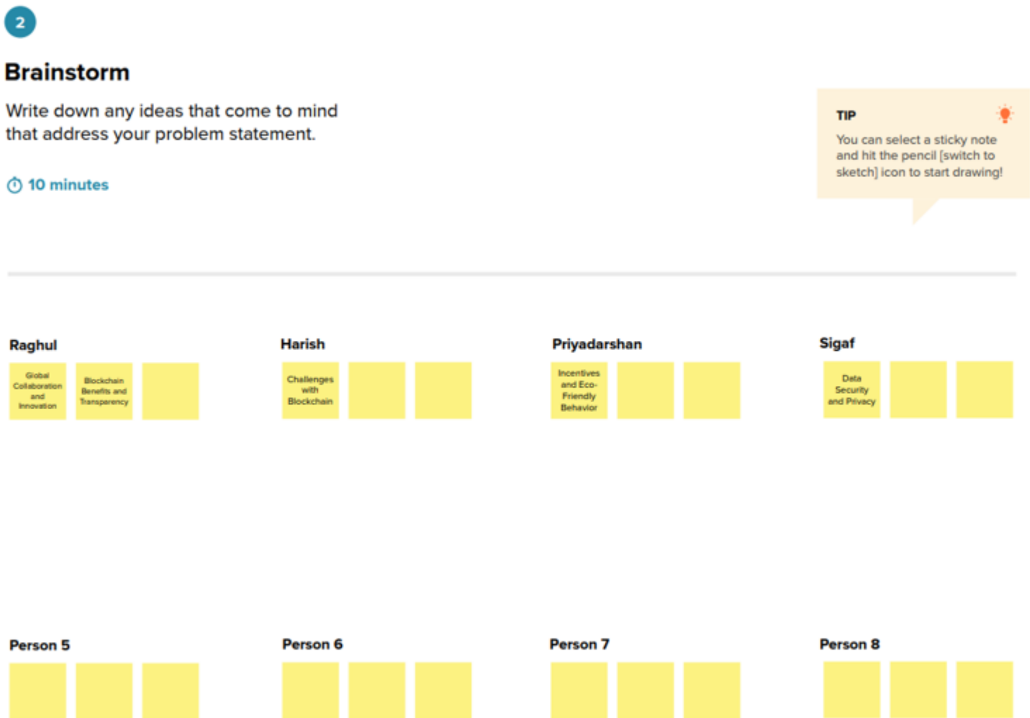
3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Raghul

- Global Collaboration and Innovation
- Challenges with Blockchain
- Incentives and Eco-Friendly Behavior

Harish

- Data Security and Privacy
- Transparency and Trust
- Interoperability with Existing Systems

Priyadarshan

- Public trust and data integrity
- Regulatory Compliance
- Scalability challenges

Sigaf

- Immutability of Environmental Record
- Gamification of Eco-Friendly behaviours
- Innovation in Environmental solutions

4

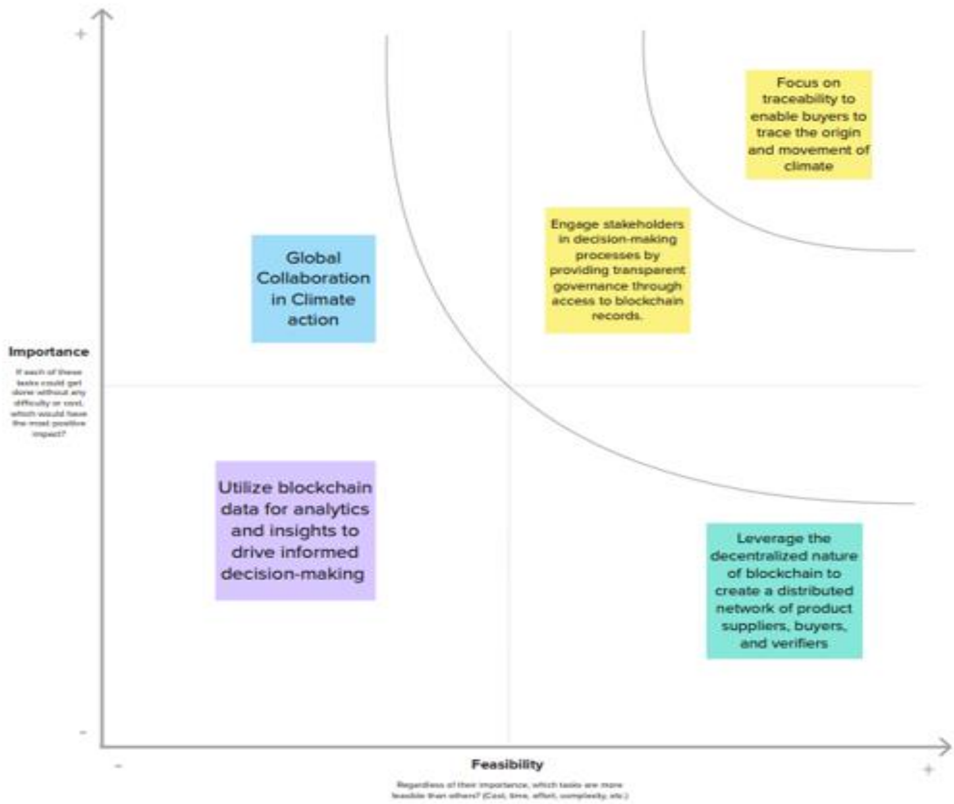
Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.



4.REQUIREMENT ANALYSIS

4.1 Functional requirement

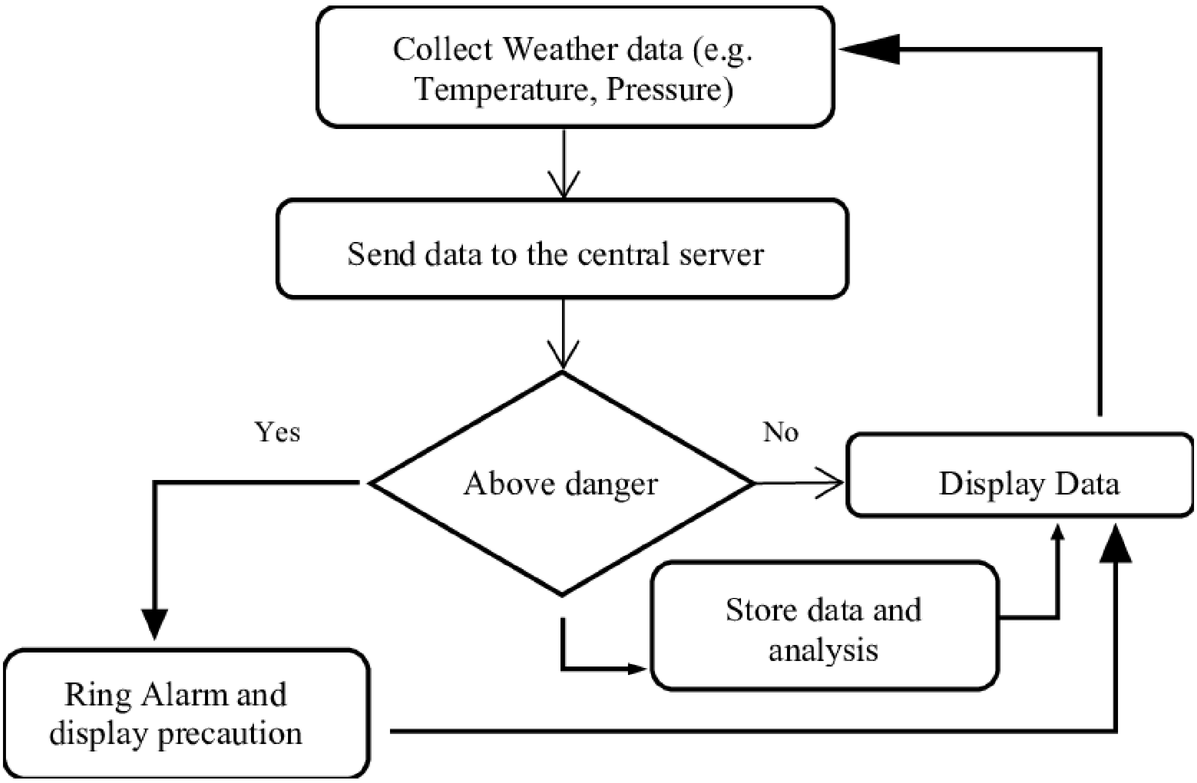
Users should be able to create blockchain-based accounts with unique IDs and passwords. Implement secure authentication mechanisms using blockchain keys. Define different user roles, such as librarians, administrators, and patrons, with varying permissions. Allow library staff to upload, update, and manage the catalog of books, e-books, journals, and other resources. Ensure that catalogue data is stored on the blockchain, making it tamper-proof and auditable. Enable users to search for resources, view detailed information, and check availability. Enable users to check out and return library items through the system. Implement smart contracts to enforce borrowing rules, due dates, and automated renewals. Record all library transactions on the blockchain, including borrowing, returning, and reservations. Ensure that transaction records are updated in real-time for users to view. Facilitate secure interlibrary loans and resource sharing between different libraries. Allow libraries to track the location and status of resources on loan. Automatic Fine Calculation: Automatically calculate fines for overdue items using smart contracts. Notify users about fines and due dates through automated alerts. Allow users to reserve books and resources, with priority based on a transparent reservation queue. Enable library staff to manage and allocate reserved resources. Enable users to rate and review library resources, contributing to a user-driven rating system. Implement a moderation system for user-generated content. Implement role-based access control to safeguard user data and comply with privacy regulations. Encrypt and protect user data stored on the blockchain. Generate reports and insights from blockchain data to inform decision-making, resource acquisition, and library management. Allow library staff to create custom reports based on specific criteria.

4.2 Non-Functional requirements

Ensure the highest level of data security by implementing robust encryption and access control mechanisms to protect user data and transaction records stored on the blockchain. Privacy Compliance: Adhere to data privacy regulations (e.g., GDPR) to safeguard user privacy and ensure compliance with relevant legal requirements. Blockchain Security: Choose a secure blockchain platform and implement best practices to protect against hacking, fraud, and unauthorized access. Ensure that the system is responsive and capable of handling concurrent user requests without significant delays or downtimes. Design the system to handle a growing user base and an expanding library catalog efficiently by scaling resources as needed. Optimize blockchain transaction processing for quick borrowing, returning, and reservations. Maintain a high level of system availability to minimize disruptions to library services. Guarantee the integrity of blockchain-stored data to prevent data corruption or loss. Implement mechanisms to recover from system failures or data discrepancies without affecting user experience. Design an intuitive and user-friendly interface that caters to both experienced and novice users, ensuring easy navigation and resource discovery. Ensure that the system is accessible to individuals with disabilities in compliance with accessibility standards (e.g., WCAG). Support integration with external systems, databases, and library networks to enable seamless resource sharing and interlibrary loans. Adhere to industry standards (e.g., MARC, Z39.50) for resource description and data exchange. Ensure compliance with all relevant library and data protection regulations and standards. Implement mechanisms for adhering to blockchain governance protocols and community guidelines. Perform load testing to ensure the system can handle peak usage without degradation in performance.

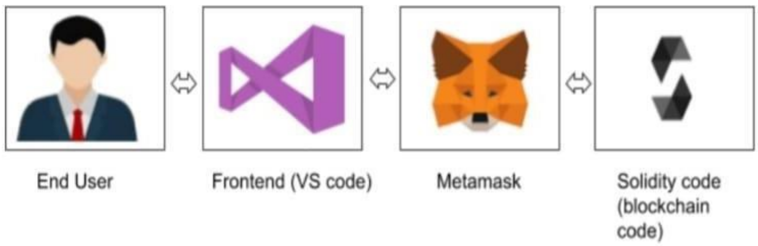
5. PROJECT DESIGN

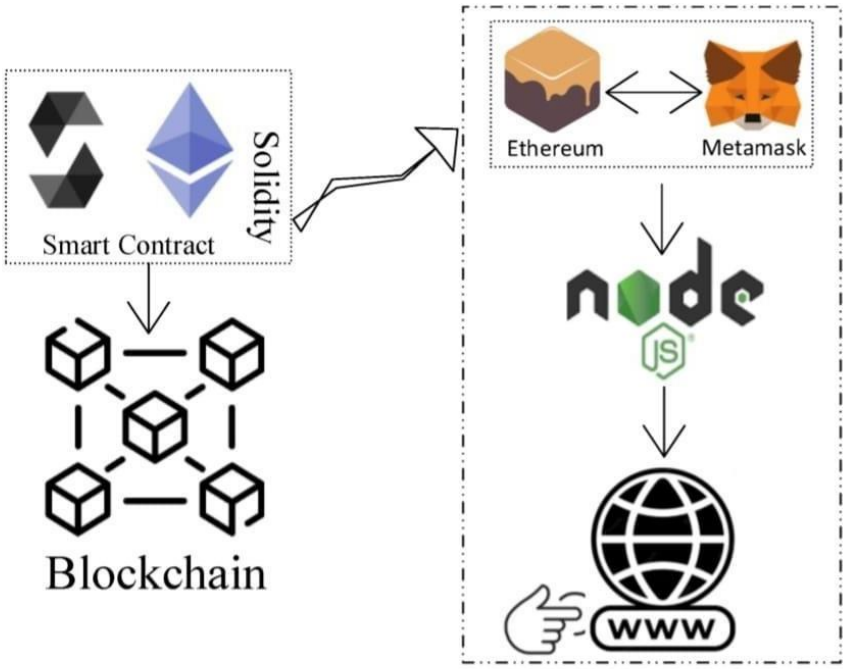
5.1 Data Flow Diagrams & User Stories



As an environmental agency administrator, I want to securely collect climate data from IoT sensors and store it on a blockchain for transparency and data integrity so that I can confidently report accurate climate information to the public and regulatory authorities. As a climate data analyst, I want to access the blockchain-based system to verify the authenticity of climate data, helping me in my research and analysis to identify trends and make informed recommendations for climate action. As a climate activist, I want to be able to access the system to ensure that climate data is publicly accessible, fostering awareness and motivating people to take action to reduce emissions.

5.2 Solution Architecture

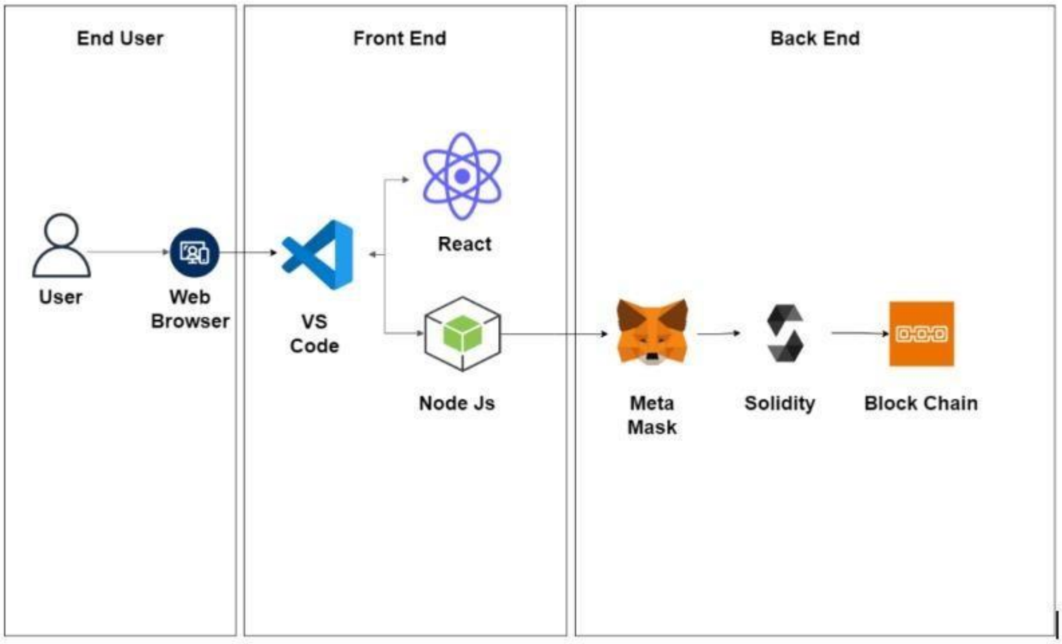




Interaction between web and the Contract

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture



6.2 Sprint Planning & Estimation

Sprint planning and estimation are critical aspects of Agile project management, which is often used for developing software and technology solutions, including those related to climate tracking with blockchain. Here's a simplified example of how you might approach sprint planning and estimation for such a project. Clarify the primary objective for the upcoming sprint. For instance, it could be to implement blockchain integration for climate data verification. Review and prioritize the product backlog, which should include user stories, features, and technical tasks related to the climate tracking system. Ensure the most valuable items are at the top. Select a duration for the sprint, typically 2-4 weeks. In the meeting, the development team and product owner discuss the highest-priority items from the backlog. Based on these discussions, the team decides which user stories and tasks can be completed during the sprint. For each selected user story, the team breaks down the tasks needed to complete it.

6.3 Sprint Delivery Schedule

Determine the duration of each sprint, typically ranging from 2 to 4 weeks. This can be based on the complexity of the tasks and the team's capacity. Set specific start and end dates for each sprint on your calendar. Ensure they align with your team's availability and the project's overall timeline. Before each sprint, prioritize the backlog of user stories and tasks based on their importance and the project's goals. At the beginning of each sprint, hold a sprint planning meeting to select the user stories and tasks to be completed during that sprint. Allocate time within the sprint for development and testing activities. Ensure that developers and testers have a clear understanding of their responsibilities. Hold daily stand-up meetings to track progress, identify and address issues, and make any necessary adjustments to the sprint plan. Conduct a mid-sprint review to assess progress and make any mid-course corrections if needed. At the end of the sprint, hold a sprint review meeting to showcase the completed work to stakeholders, gather feedback, and demonstrate how the solution is evolving. Immediately after the sprint review, conduct a sprint retrospective to evaluate what went well and what needs improvement. Use this feedback to enhance future sprints. Identify specific milestone checkpoints in your project timeline. These could be related to key features, blockchain integration, data collection, or any other critical aspect of your climate tracking solution. Be aware of any dependencies between sprints. Ensure that the completion of one sprint does not hinder the start of the next. Allow for some buffer time between sprints to account for unexpected delays or issues that may arise during development. Continuously monitor and update the overall project timeline based on the progress of individual sprints and milestone achievements. Maintain clear and transparent communication with all team members and stakeholders to ensure everyone is aligned with the sprint delivery schedule. Be flexible and ready to adjust the schedule as required. Some sprints may finish earlier or later than expected, and you may need to adapt accordingly.

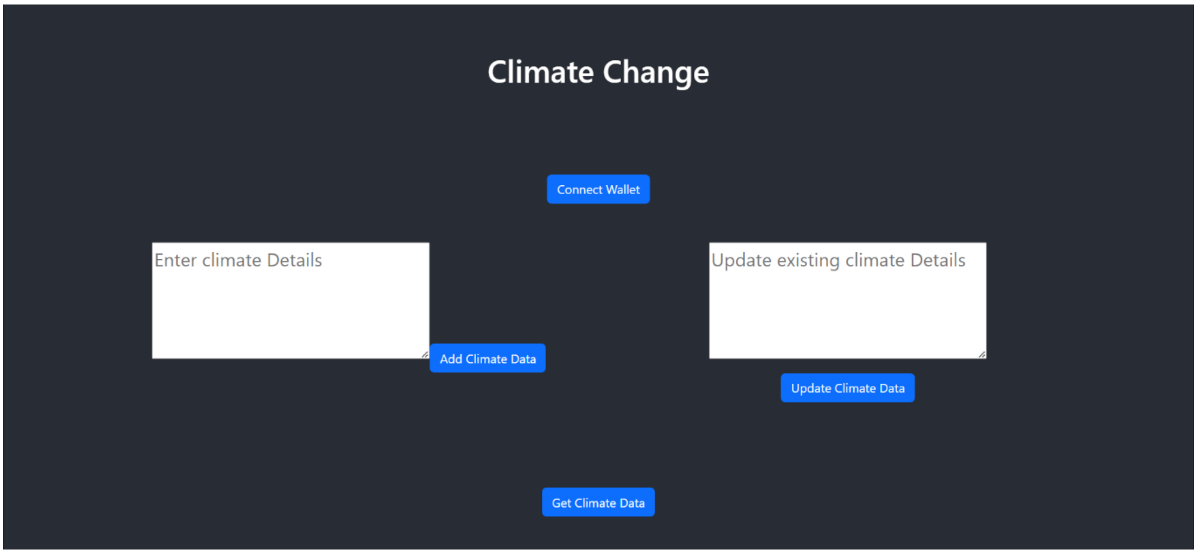
7. CODING & SOLUTIONING

7.1 Feature 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
}
```

7.2 Feature 2



Contract ABI (Application Binary Interface):

The ABI variable holds the ABI of an Ethereum smart contract. ABIs are essential for encoding and decoding function calls and data when interacting with the Ethereum blockchain.

MetaMask Check:

The code first checks whether the MetaMask wallet extension is installed in the user's browser. If MetaMask is not detected, it displays an alert notifying the user that MetaMask is not found and provides a link to download it.

Ethers.js Configuration:

It imports the ethers library, which is a popular library for Ethereum development. It creates a provider using Web3Provider, which connects to the user's MetaMask wallet and provides access to Ethereum. It creates a signer to interact with The Ethereum blockchain on behalf of the user. It defines an Ethereum contract address and sets up the contract object using ethers. Contract, allowing the JavaScript code to interact with the contract's functions. In summary, this code is used for interacting with an Ethereum smart contract through MetaMask and ethers.js. It configures the necessary Ethereum provider and signer for communication with the blockchain and sets up a contract object for executing functions and fetching data from the specified contract address using the provided ABI.

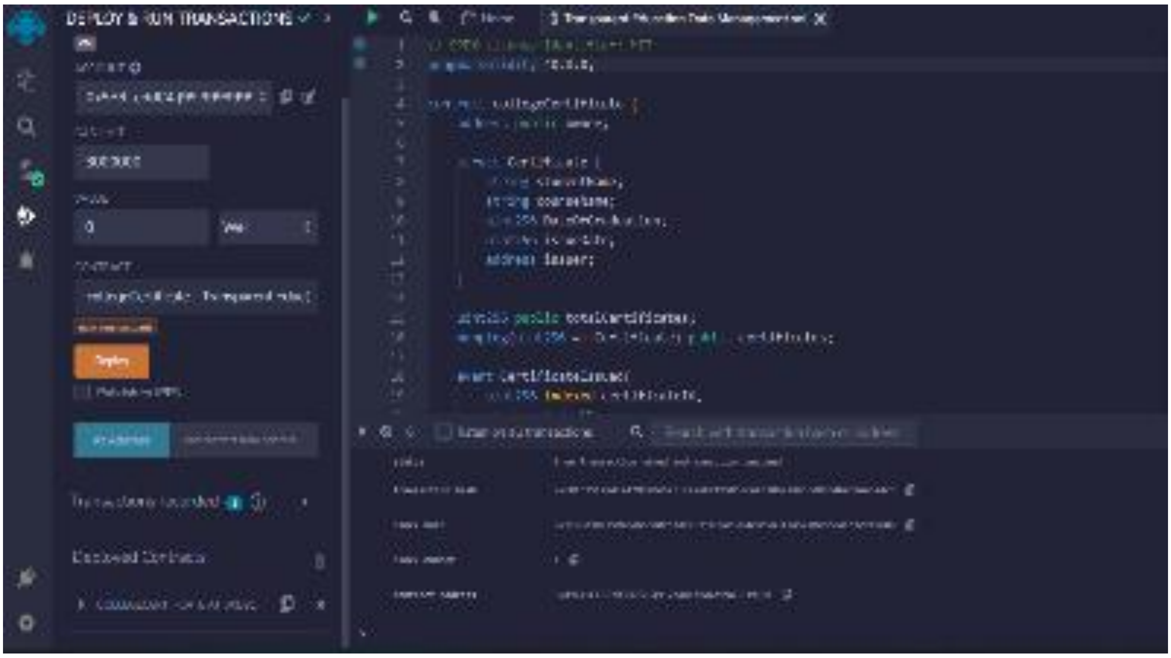
8. PERFORMANCE TESTING

8.1 Performance Metrics

Measure the number of incidents where climate data has been tampered with or altered. A lower rate indicates better data integrity. Track how often data is successfully verified by stakeholders through the blockchain. A higher verification rate signifies greater trust in the data. Monitor the number of users or organizations accessing climate data on the blockchain. Increased accessibility indicates improved transparency. Measure public participation and engagement with the climate data and actions made available through blockchain. Calculate the energy consumption associated with blockchain operations. Aim to reduce this figure over time by optimizing consensus mechanisms and infrastructure. Evaluate the system's ability to handle a high volume of climate data transactions. Ensure it can scale to meet increasing demands. Measure the percentage of smart contracts executed within predefined timeframes. Ensure that climate-related actions are automated efficiently. Track the operational costs of the blockchain system, including maintenance, development, and transaction fees. Aim for cost efficiency and cost reduction. Gather feedback from stakeholders, users, and participants to assess their satisfaction with the blockchain-based climate tracking system. Ensure that the blockchain system complies with environmental, data privacy, and other relevant regulations. Measure the degree to which blockchain enhances transparency in supply chains, allowing consumers to make informed choices about sustainable products. Track the number of carbon offset transactions facilitated through the blockchain. Measure the impact in terms of emissions reductions. Monitor the response time and effectiveness in addressing security incidents or breaches related to the blockchain system. Assess the system's ability to integrate with existing data sources, IoT devices, and other technologies. Measure the level of community engagement and involvement in climate actions facilitated by the blockchain system. If using decentralized autonomous organizations (DAOs), assess the effectiveness of decision-making processes and resource allocation.

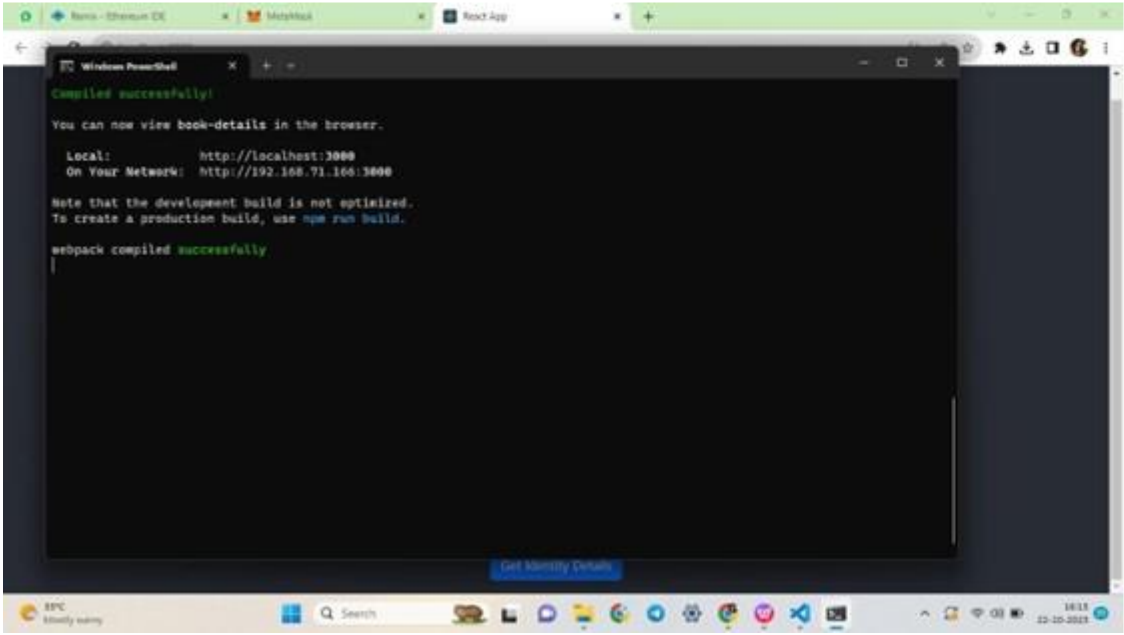
9. RESULTS

9.1 Output Screenshots

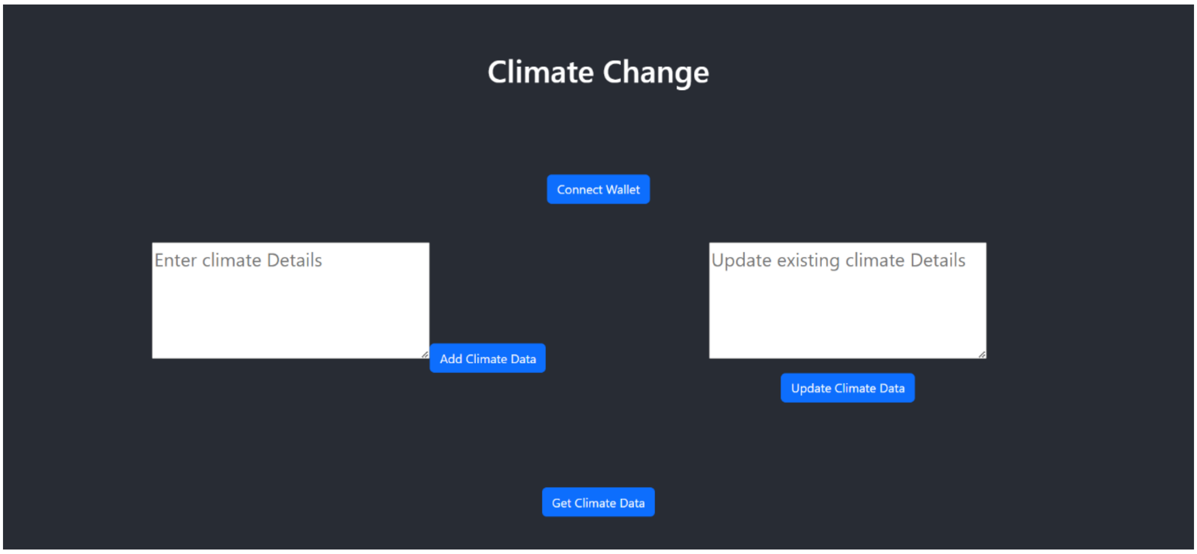


Creating smart contract

Installing packages



Creating the site locally



Output Screenshot

10. ADVANTAGES & DISADVANTAGES

Advantages

Blockchain provides an immutable and transparent ledger, which can be used to track and verify climate data and actions. This transparency builds trust among stakeholders. Data stored on the blockchain is highly secure and tamper-proof, reducing the risk of data manipulation or fraud. The decentralized nature of blockchain removes the need for a central authority, making it harder for any single entity to control or manipulate the data. Smart contracts can automate and enforce climate-related agreements, ensuring that predefined conditions are met before actions are taken. It allows for end-to-end traceability of products and activities related to climate actions, helping in supply chain sustainability and carbon footprint reduction. Blockchain can engage the public through transparency and by allowing individuals to verify and contribute to climate initiatives.

Disadvantages

Blockchain can be energy-intensive, especially proof-of-work (POW) based systems. This is a significant concern given the environmental objectives of climate tracking. Scalability issues can arise, particularly with public blockchains, when dealing with a high volume of climate data and transactions. The regulatory landscape for blockchain and its use in climate tracking is still evolving, which can lead to uncertainty and legal challenges. While blockchain ensures data integrity, it may also make sensitive climate data more accessible, potentially raising privacy concerns. Developing and maintaining a blockchain-based solution can be complex and costly, including initial setup, ongoing maintenance, and development. Ensuring compatibility and interoperability with existing systems and platforms can be a challenge when implementing blockchain solutions.

11. CONCLUSION

Implementing a climate tracking smart solution using blockchain technology has the potential to revolutionize how we monitor and manage environmental data. Blockchain's transparency, security, and ability to automate processes offer numerous advantages. However, it's important to be aware of the energy consumption and scalability challenges that some blockchain systems pose, which may conflict with environmental goals. Regulatory and data privacy concerns also need to be addressed. To make the most of blockchain in climate tracking, careful planning, energy-efficient consensus mechanisms, adherence to regulations, and robust data privacy measures are crucial. Despite the challenges, the benefits of enhanced data transparency, security, and automation make blockchain a promising tool for tackling climate-related issues. Collaborative efforts and standardization within the industry can further facilitate the adoption and effectiveness of blockchain in climate tracking initiatives.

12. FUTURE SCOPE

Blockchain's immutable ledger ensures data integrity, making it a trusted source for climate-related data. This can lead to more accurate and reliable information for policymakers, researchers, and businesses. Blockchain's transparency can empower individuals and organizations to track and verify climate-related actions and commitments, promoting greater accountability and trust in sustainability efforts. Blockchain can play a pivotal role in tracking and verifying the sustainability of products and supply chains. Consumers can make informed choices about products with lower environmental impacts. Blockchain can facilitate the creation and trading of carbon credits and offsets, making it easier for businesses and governments to incentivize and invest in carbon reduction initiatives. Smart contracts can automate the execution of climate-related agreements, ensuring that specific actions are taken when predefined conditions are met. Climate-focused DAOs can be established on blockchain networks, allowing for decentralized decision-making and resource allocation for environmental projects. Future efforts will likely focus on establishing interoperability between different blockchain networks and standardizing data formats and protocols for climate tracking, enabling broader collaboration. Research and development will drive the creation of energy-efficient consensus mechanisms and blockchain platforms that reduce the environmental impact of blockchain technology. As the use of blockchain in climate tracking grows, regulatory frameworks will evolve to provide clear guidelines for its implementation and ensure compliance with environmental and data privacy regulations. The future will see increased collaboration among governments, businesses, non-profits, and the technology sector to develop and implement blockchain-based solutions for global climate challenges. Blockchain can empower communities to actively participate in climate tracking and sustainability efforts, fostering a sense of collective responsibility.

13. APPENDIX

Source Code

Climatechange.sol

```
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
```

```
}
```

Page

connector.js

```
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"

   }

  ],

  "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)
import React, { useState } from "react";
import { Button, Container, Row, Col } from 'react-bootstrap';
import 'bootstrap/dist/css/bootstrap.min.css';
import { contract } from "./connector";
import { signer } from "./connector";
import { provider } from "./connector";
function Home() {
  const [Id, setId] = useState("");
  const [ClimateDet, setClimateDet] = useState("");
  const [UpdateClimateDetails, setUpdateClimateDetails] = useState("");
  const [TranId, setTranId] = useState("");
  const [Owner, setOwner] = useState("");
  const [BookId, setBookId] = useState("");
  const [BookDet, setBookDet] = useState("");
  const [Wallet, setWallet] = useState("");
  const handleId = (e) => {
    setId(e.target.value)
  }
  const handleClimateDetails = (e) => {
```

```
  setClimateDet(e.target.value)
}
const handleClimate = async () => {
  try {
    let tx = await contract.addClimateData(ClimateDet)
    let wait = await tx.wait()
    alert(wait.transactionHash)
    console.log(wait);
  } catch (error) {
    alert(error)
  }
}
const handleDrugId = (e) => {
  setTranId(e.target.value)
}
const handleUpdatedClimateDetails = (e) => {
  setUpdateClimateDetails(e.target.value)
}
const handleUpdateClimate = async () => {
  try {
    let tx = await contract.updateClimateData(UpdateClimateDetails)
    let wait = await tx.wait()
```

```
      console.log(wait);
      alert(wait.transactionHash)
    } catch (error) {
      alert(error)
    }
  }
  const handleTollDetailsId = (e) => {
    setBookId(e.target.value)
  }
  const handleGetClimateDetails = async () => {
    try {
      let tx = await contract.getClimateData()
      let arr = []
      tx.map(e => {
        arr.push(e)
      })
      console.log(tx);
      setBookDet(arr)
    } catch (error) {
      alert(error)
      console.log(error);
    }
```

```
  }

  const handleWallet = async () => {
    if (!window.ethereum) {
      return alert('please install metamask');
    }
    const addr = await window.ethereum.request({
      method: 'eth_requestAccounts',
    });
    setWallet(addr[0])
  }
 return (
 <div>
  <h1 style={{ marginTop: "30px", marginBottom: "80px" }}>Climate Change</h1>
     {!Wallet ?
       <Button onClick={handleWallet} style={{ marginTop: "30px", marginBottom: "50px" }}>Connect Wallet </Button>
       :
       <p style={{ width: "250px", height: "50px", margin: "auto", marginBottom: "50px", border: '2px solid #2096f3' }}>{Wallet.slice(0, 6)}....{Wallet.slice(-6)}</p>
     }
```

```
  <Container>
   <Row>
    <Col style={{marginRight:"100px"}}>
     <div>
      {/* <input style={{ marginTop: "10px", borderRadius: "5px" }} onChange={handleClimateDetails} type="textArea" placeholder="Enter climate details" value={ClimateDet} /> <br /> */}
            <textarea rows="4" cols="30" name="comment" placeholder="Enter climate Details" onChange={handleClimateDetails} value={ClimateDet}></textarea>
      <Button onClick={handleClimate} style={{ marginTop: "10px" }} variant="primary"> Add Climate Data</Button>
     </div>
    </Col>
     <Col>
      <div>
            <textarea rows="4" cols="30" name="comment" placeholder="Update existing climate Details" onChange={handleUpdatedClimateDetails} value={UpdateClimateDetails}></textarea>
            <Button onClick={handleUpdateClimate} style={{ marginTop: "10px" }} variant="primary"> Update Climate Data</Button>


      </div>
```

```
    </Col>
  </Row>
  <Row>
     <Col>
        <div style={{ margin: "auto", marginTop:"100px" }}>
         {/* <input style={{ marginTop: "10px", borderRadius: "5px" }} onChange={handleTollDetailsId} type="number" placeholder="Enter Highway Id" value={BookId} /><br /> */}
            <Button onClick={handleGetClimateDetails} style={{ marginTop: "10px" }} variant="primary">Get Climate Data</Button>
              {BookDet ? BookDet?.map(e => {
                return <p>{e.toString()}</p>
              }) : <p></p>}
        </div>
      </Col>
  </Row>
  </Container>


 </div>
 )
}
export default Home;
```

GitHub & Project Demo Link

GitHub link

https://github.com/raghul2929/NM-BLOCKCHAIN-CLIMATE-TRACK-SMART-USING-BLOCKCHAIN.git

Project demo link

https://drive.google.com/file/d/1G7QTuwZi8JCZX_X98qokW62_OTzTok4b/view?usp=drivesdk