FARMER INSURANCE CHAIN

A project report

Submitted by,

M.guna selvi

B.selvagokila

B.vithya

A.vincila jothi

Team id : NM2023TMID03185

Oct-23

Jayaraj annapackiam CSI college of Engineering

Nazareth-628617

Farmer Insurance Chain

1.Introduction

1.1project overview

- Provide a brief introduction to the project, explaining its purpose and goals.
- Clearly outline the specific objectives of the project, such as providing insurance services to farmers, improving their financial security, or mitigating agricultural risks.
- Define the target audience, which in this case would be farmers or agricultural stakeholders.
- Detail the geographical scope and coverage of the insurance chain, specifying the regions or areas where it will operate.
- Describe the types of insurance products and services offered, such as crop insurance, livestock insurance, or weather-related risk coverage.
- Identify any key partnerships with agricultural organizations, government agencies, or financial institutions that will support the project.
- Outline the risks associated with the project, including market volatility, climate-related risks, and regulatory challenges.
- Provide a project timeline with key milestones and deadlines.

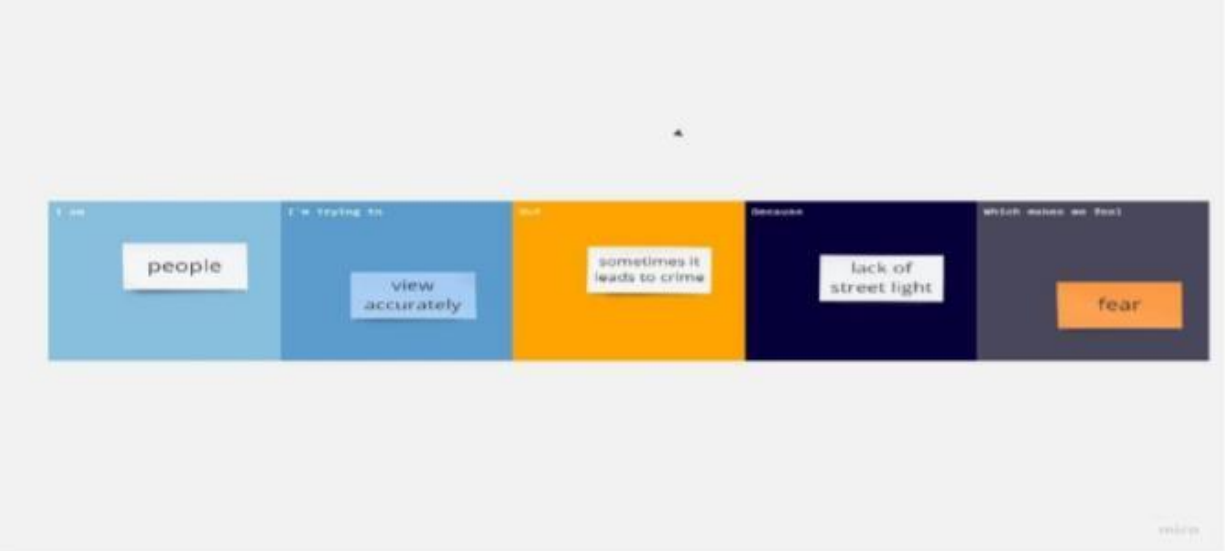
1.2purpose

- Farmers face various risks, such as crop failures due to weather, pest infestations, or market fluctuations. Farmer insurance helps mitigate these risks by providing compensation in case of adverse events.
- It helps farmers maintain financial stability by offering a safety net when their agricultural income is impacted by unexpected losses.
- .Farmer insurance encourages farmers to invest in their operations, knowing that they have a safety net in place. This can lead to increased agricultural productivity and innovation.
- By reducing the financial vulnerability of farmers, insurance contributes to overall food security by helping maintain a consistent and sufficient food supply.
- Farmer insurance supports the long-term sustainability of agricultural practices by helping farmers recover from setbacks and continue their operations.
- Having insurance can make it easier for farmers to access credit and investment, as it provides a level of security for lenders and investors.
- A stable agricultural sector contributes to broader economic stability, as it is a significant part of many economies, especially in developing countries.
- : Farmer insurance can improve the welfare of farming communities, reducing poverty and vulnerability.
- Insurance can play a role in disaster preparedness and response, ensuring that farmers have the means to recover and rebuild after natural disasters.

2.LITERATURE SURVEY:

S. no	Title of the paper	Authors	Algorithm	Advantage	Disadvantage
1.	AGRICULTURE INSURANCE	ArupChatterjee Arman Oza	Extreme weather events can lead to significant adversely, agriculture	Agriculture insurance it reduce the problem of the farmer.	The disadvantage of the system where increase the policy
2.	Insurance Value Chain Chat bot for Farmer	1.M.L. Mishawaka Institute of Technology, Pune,	The agricultural method to reduce the importants and to produced the method of insurance	It to reduced the insurance	The former want to increase the agricultural insurance
3.	Block chain based crop insurance	Dr. Deepika swranra faculity in engineering USWA	A unique identified to more then reduced the quality	They were to increase the farmer member	They also given to reduced the methods of agricultural maintains
4.	Apply block chain technology in the farmers insurance	S. Kayalvizhi D. Amirtha sughi	The basic methods are included	It to improve the methods of farmer insurance	The importance of the techniques were improved.
5.	Farmer insurance tracking using block chain	Huanhuan Fenga.	A radio frequency Identification (RFID)-based sensor,MQTT,IOT. Aiding in weight management, and helping meet	The advanced to reduced the problem of insurance	They were to improve the problems of insurance

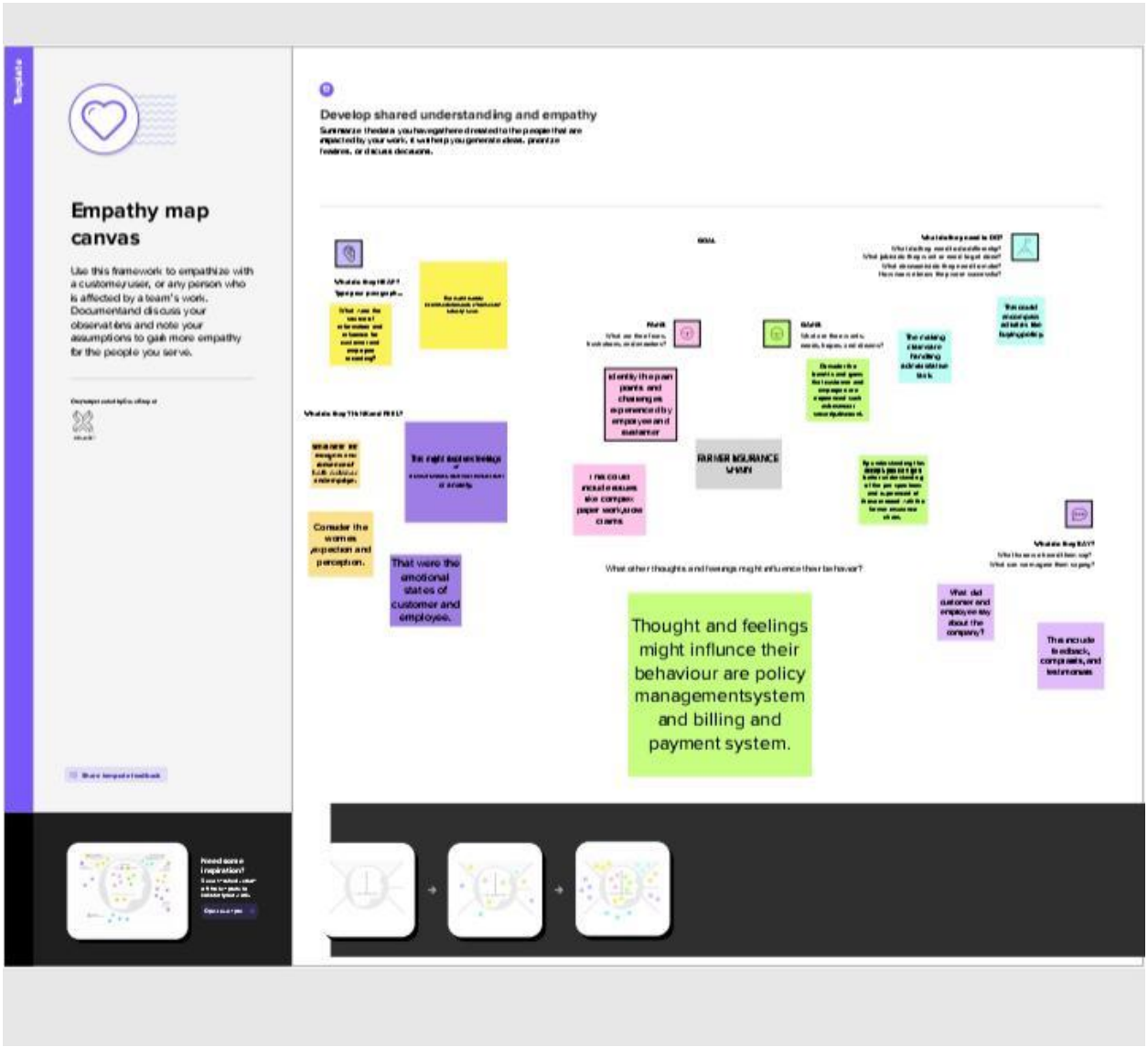
3.**PROBLEM STATEMENT:**



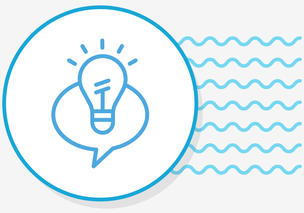
4. IDEATION AND PROPOSED SOLUTION:

4.1 EMPATHY MAP CANVAS:

4.2IDEATION AND BRAINSTORMING:



Template

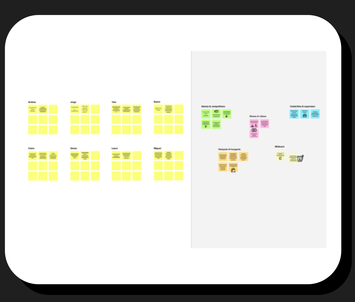


Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Share template feedback



Need some inspiration?

See a finished version of this template to kickstart your work.

Open example →

Before you collaborate

A little bit of preparation goes a long way with this session. Here’s what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

Eco-Friendly Insurance**: Develop insurance products that promote eco-friendly practices and sustainability

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Defer judgment.
- Go for volume.
- Encourage wild ideas.
- Listen to others.
- If possible, be visual.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Person 1

Diversify Services**: Explore opportunities to diversify beyond traditional insurance, perhaps into financial planning, healthcare services, or technology solutions.

*Digital Transformation**: Consider how the chain could have embraced digital technologies to enhance customer experiences, streamline operations, and improve data security.

*Specialized Niche Markets**: Identify underserved niche markets and create tailored insurance products for them, such as insurance for specific professions or unique assets

Person 2

Eco-Friendly Insurance**: Develop insurance products that promote eco-friendly practices and sustainability

*Collaborations**: Explore partnerships with other companies or platforms to extend the reach of insurance services

Customer-Centric Approach**: Brainstorm ways to make the customer experience more seamless and transparent, from quotes to claims

Person 3

AI and Automation**: Investigate how AI and automation could have improved operational efficiency and customer service.

Data Analytics**: Utilize data analytics to better understand customer behaviors and needs, enabling more accurate pricing and risk assessment.

Insurance for Emerging Technologies**: Develop insurance products for emerging tech fields like autonomous veh

Person 4

In 1928, two men shared a dream of providing a quality insurance product at a reasonable price. In the decades that followed we've grown and adapted to meet the changing needs of Americans. But one constant has remained:

We have an unwavering commitment to uphold our founding ideals to provide industry-leading products and first-rate services to the customers we're privileged to serve.

Farmers not only prides itself on helping you plan wisely for the unexpected, but also on helping restore order when it occurs, so you can keep moving along the road of your life's plans.

Person 5

Farmers sent 300 agents to assess damages while policy holders were evacuated. Farmers also donated $100,000 for the city's emergency operations center and two megawatt generators that restored power to Beaumont. Whatever year or decade a disaster occurs, you can count on us to be there for you.

Farmers is proud to serve more than 10 million households with more than 19 million individual policies across all 50 states through the efforts of over 48,000 exclusive and independent agents and nearly 21,000 employees.

The Farmers Exchanges are three reciprocal insurers (Farmers Insurance Exchange, Fire Insurance Exchange and Truck Insurance Exchange) owned by their policyholders

Person 6

Person 7

Person 8

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.

In 1928, two men shared a dream of providing a quality insurance product at a reasonable price. In the decades that followed we've grown and adapted to meet the changing needs of Americans. But one constant has remained:

*Collaborations**: Explore partnerships with other companies or platforms to extend the reach of insurance services

*Digital Transformation**: Consider how the chain could have embraced digital technologies to enhance customer experiences, streamline operations, and improve data security.

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

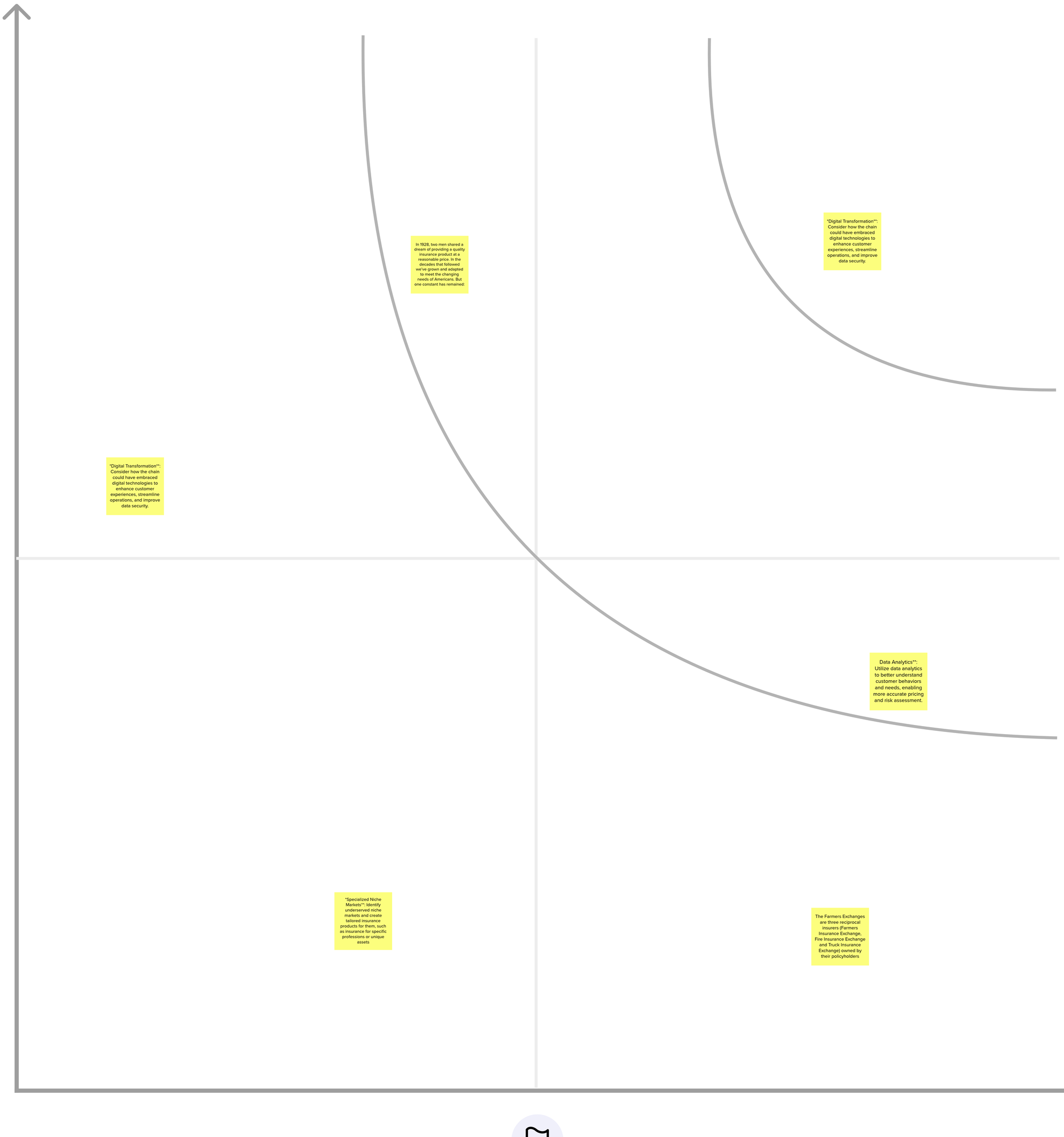
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

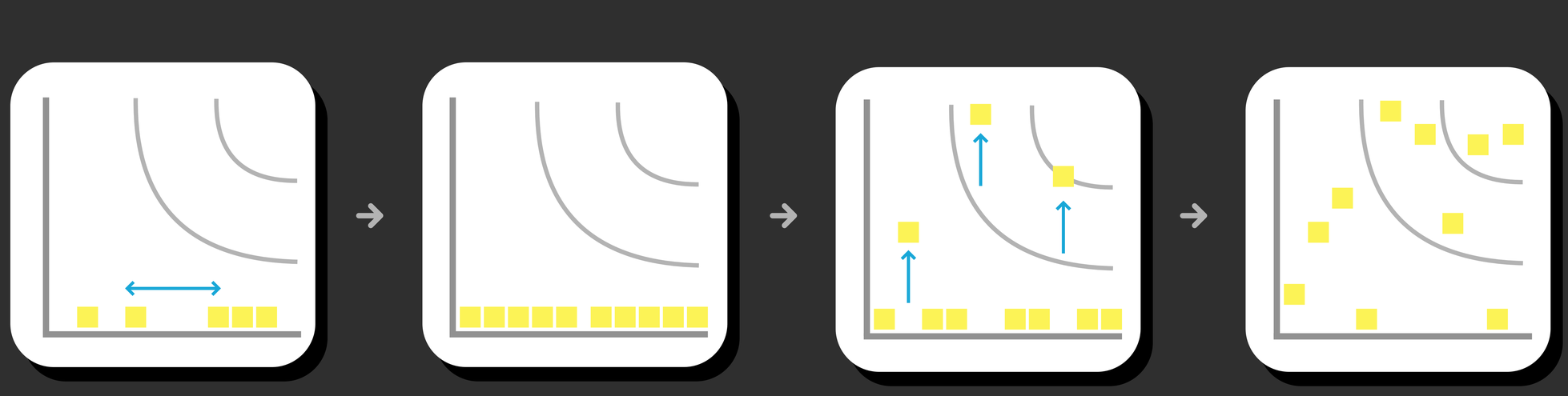
Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)





After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.

Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.

Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.

Open the template →

Share template feedback

5.PROJECT PHASE
Solution(requirement and non-requirment)

Date	20 oct 2023
Team ID	NM2023TMID03182
Project Name	Project –Farmer Insurance chain

Functional Requirements:

To develop a former insurance chain solution, you would need to consider several key requirements

1.**Data Management**

Effective data storage and retrieval systems for policyholder information

Integration with legacy insurance systems for data migration.

2.**Claims processing**

A claims processing module for handling claims efficiently and accurately.

Automated claims assessment tools to streamline the process.

3.**Policy Management:**

Policy creation and management capabilities Premium calculation and billing functionalities.

4.**Compliance and Regulation:**

Compliance with industry regulations and legal requirements. Builtin features for reporting and auditing.

5.**Customer Service:**

Customer relationship management (CRM) tools for communication and support.
Self-service portals for policyholders and agents.

6.**Workflow Automation**

Workflow automation for underwriting, policy issuance, and renewals.
Task management and notification systems.

7.**Reporting and Analytics:**

Comprehensive reporting and analytics to assess performance. Data. visualization tools for insights.

8.**Integration:**

Integration with third-party data sources and service Compatibility. with industry standards and APIs

9. Security:

Robust security measures to protect sensitive customer data
Data encryption and access control

10.**scalability**

The ability to scale the solution as the business grows Support for multi-location and multi-brand operations.

11. **Cloud Compatibility:**

Consideration of cloud-based solutions for flexibility and scalability
Data backup and disaster recover

These requirements will depend on the specific needs of the former insurance chain and the type of insurance products the offer. Customization and careful planning are essential to meet the unique demands of the business.

Non-functional requirements

a farmer insurance chain typically focus on the system's performance, reliability, security, and scalability. Here are some non-functional requirements that might apply:

1. Performance:

Response Time: The system should respond to user requests within a specified time frame to ensure efficient service.

Throughput: It should be capable of handling a certain number of transactions or claims per second to meet peak demands.

2. Reliability:

Availability: The system should be available 24/7 with minimal downtime.

Fault Tolerance: It should continue to function in the presence of hardware or software failures.

3. Security:

Data Encryption: Sensitive data, such as customer information and financial records, should be encrypted to protect against unauthorized access.

Access Control: Only authorized personnel should be able to access and modify data and system settings.

Compliance: The system should adhere to industry and legal standards for data protection and privacy.

4. Scalability:

Elasticity: The system should be able to scale up or down to accommodate changes in demand, such as during planting or harvesting seasons.
Load Balancing: Distribute incoming requests evenly across servers to prevent overloading.

5. Usability:

User Interface Design: The user interface should be intuitive and user-friendly for both farmers and insurance agents.

Accessibility: Ensure that the system is accessible to users with disabilities.

6. Data Backup and Recovery:

Regular data backups should be performed to prevent data loss in case of system failures.

Effective data recovery mechanisms should be in place to restore data in case of failures.

7.Compliance:

Ensure the system complies with relevant regulatory and legal requirements related to insurance services and data handling.

8.Audit Trail:

Maintain an audit trail of all system activities to track and review actions performed within the system for accountability and compliance purposes.

9.Monitoring and Reporting:

Implement monitoring tools to track system performance and generate reports on system health and usage.

10.Integration:

The system should be capable of integrating with external systems and data sources, such as weather forecasts, government databases, and financial institutions.

These non-functional requirements are critical for ensuring that a farmer insurance chain operates efficiently, securely, and reliably while meeting the needs of both farmers and the insurance providers.

6. PROJECT PLANNING &SCHEDULING

6.1 TECHNICAL ARCHITECTURE:

• Block chain Network: Climate Track utilizes a distributed length technology (DLT) based on block chain to ensure the integrity immutability of climate-related data. This network consists of multiple nodes that participate in the validation and consensus process.

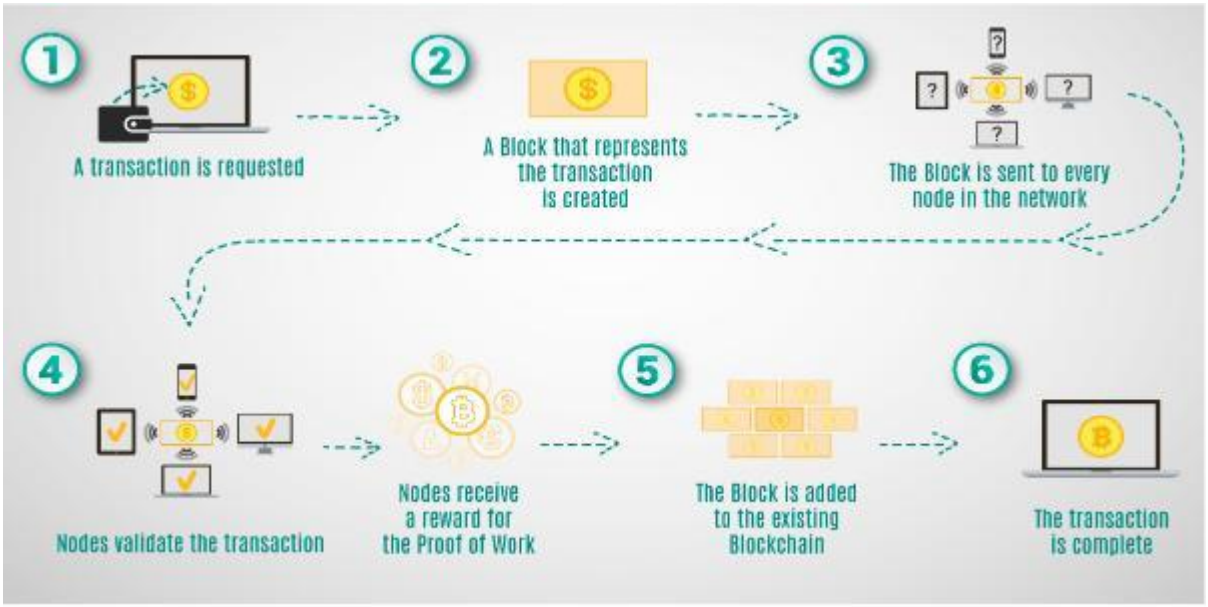
• Smart Contracts: Climate Track employs smart contracts, which are self-executing agreements with predefined rules and conditions. These contracts automate the execution of climate-related transactions, such as carbon credit trading or renewable energy certificate issuance.

• Data Storage: Climate-related data, including emissions data, sustainability initiatives, and renewable energy generation, is securely stored on the block chain network. This decentralized storage ensures data integrity and prevents unauthorized modifications.

• APIs and Integration: Climate Track provides APIs (Application Programming Interfaces) to enable seamless integration with external systems and data sources. This allows for the collection and verification of climate data from various stakeholders, such as IOT devices, energy providers, and government databases.

• User Interface: Climate Track offers a user-friendly interface that allows stakeholders to interact with the platform. This interface enables data submission, access to reports and analytics, and the ability to track progress towards climate goal.

6.2 Sprint Planning and Estimation:



Sprint planning and estimation for Climate Track would involve breaking down the Development tasks and features into manageable units of work that can be completed within a sprint, Typically a time-boxed period of 1-4 weeks.

Here is a general outline of the process:

Product Backlog: Start by maintaining a prioritized list of features, enhancements, and bug fixe In the product backlog. This backlog should be regularly reviewed and updated based on user Feedback, market demands, and business priorities.

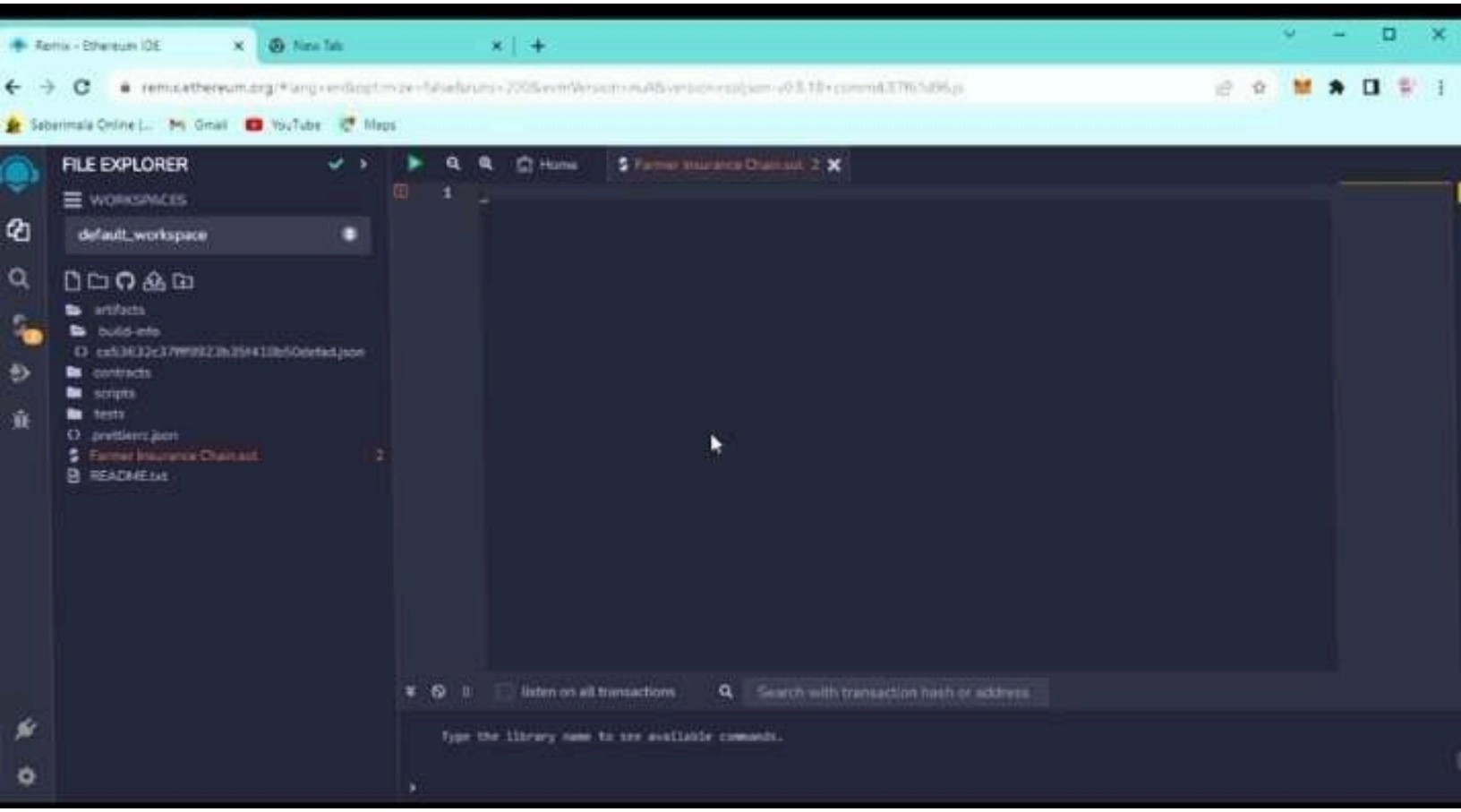
- Sprint Goal: Define a clear goal for each sprint that aligns with the overall objectives of Climate Track. This goal should be specific, measurable, achievable, relevant, and time-bound (SMART).

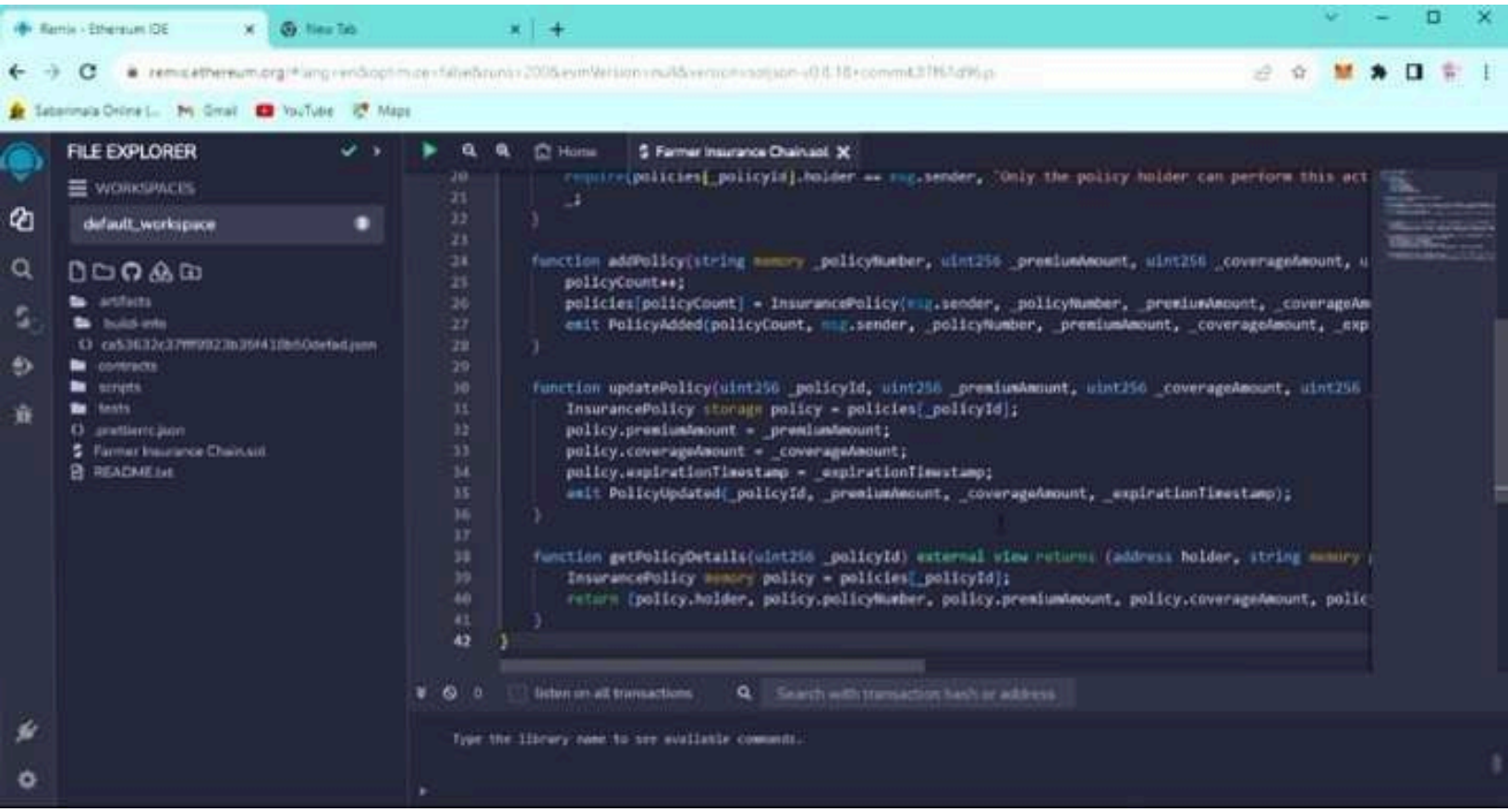
- Sprint Planning Meeting: Conduct a sprint planning meeting where the development team, Product owner, and other stakeholders collaborate to select the top-priority items from the Product backlog for the upcoming sprint. The team should discuss and understand the Requirements, dependencies, and acceptance criteria for each selected item.

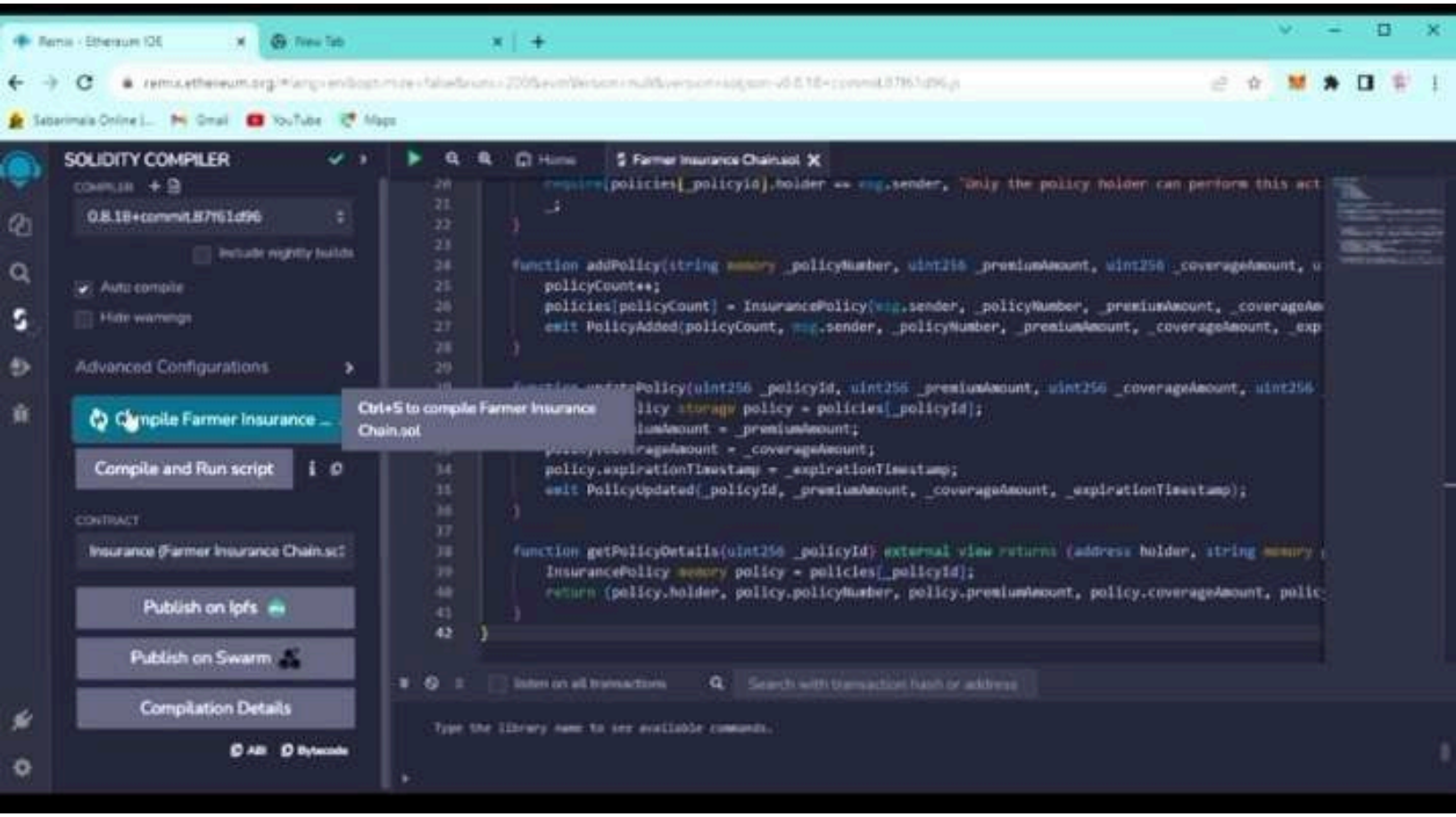
7.CODING &SOLUTIONING:

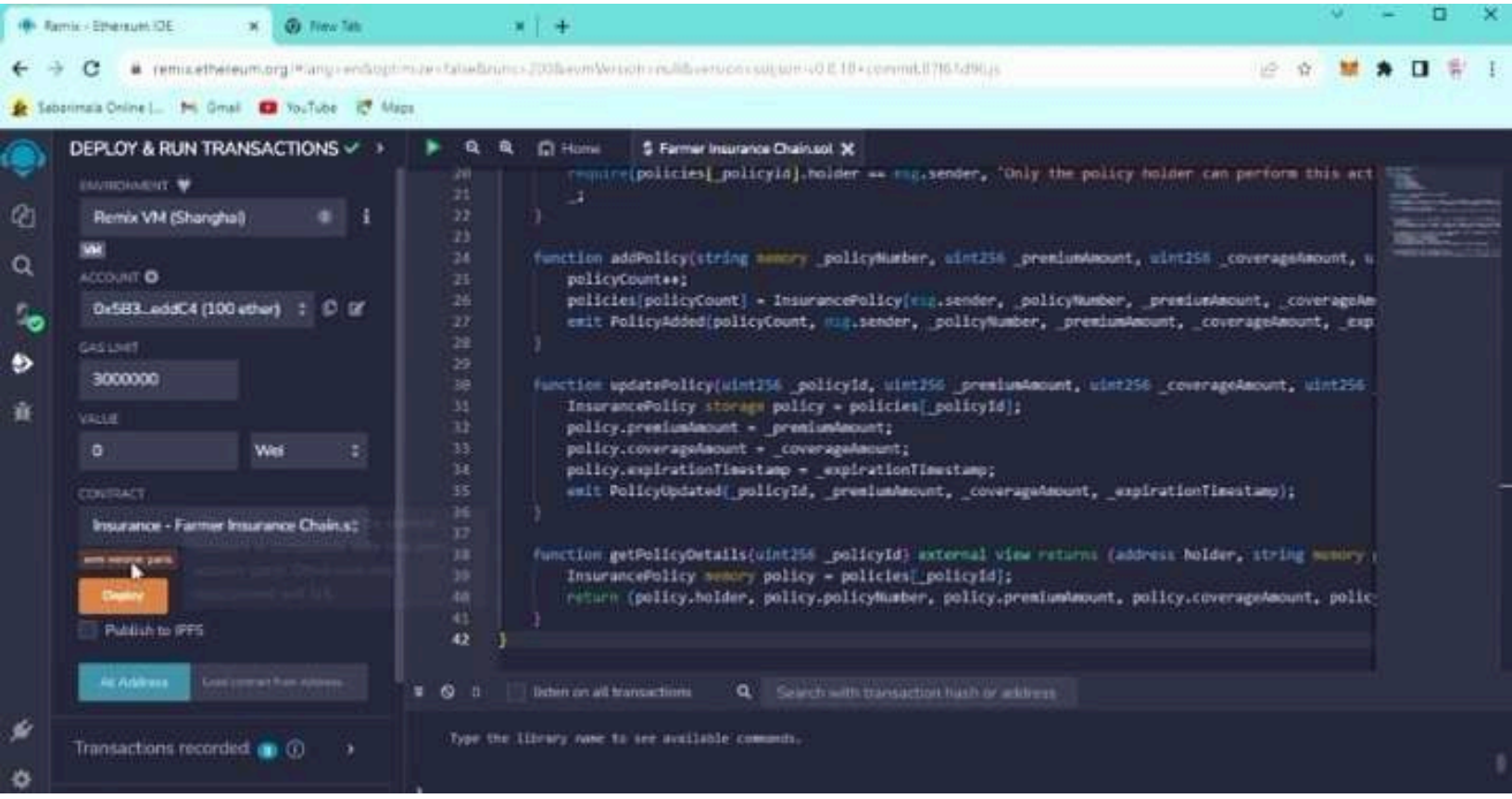
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Insurance {
    struct InsurancePolicy {
        address holder;
        string policyNumber;
        uint256 premiumAmount;
        uint256 coverageAmount;
        uint256 expirationTimestamp;
    }

    mapping(uint256 => InsurancePolicy) public policies;
    uint256 public policyCount;

    event PolicyAdded(uint256 policyId, address holder, string policyNumber, uint256 premiumAmount,
uint256 coverageAmount, uint256 expirationTimestamp);
    event PolicyUpdated(uint256 policyId, uint256 premiumAmount, uint256 coverageAmount, uint256
expirationTimestamp);

    modifier onlyHolder(uint256 _policyId) {
        require(policies[_policyId].holder == msg.sender, "Only the policy holder can perform this action");
        _;
    }
```

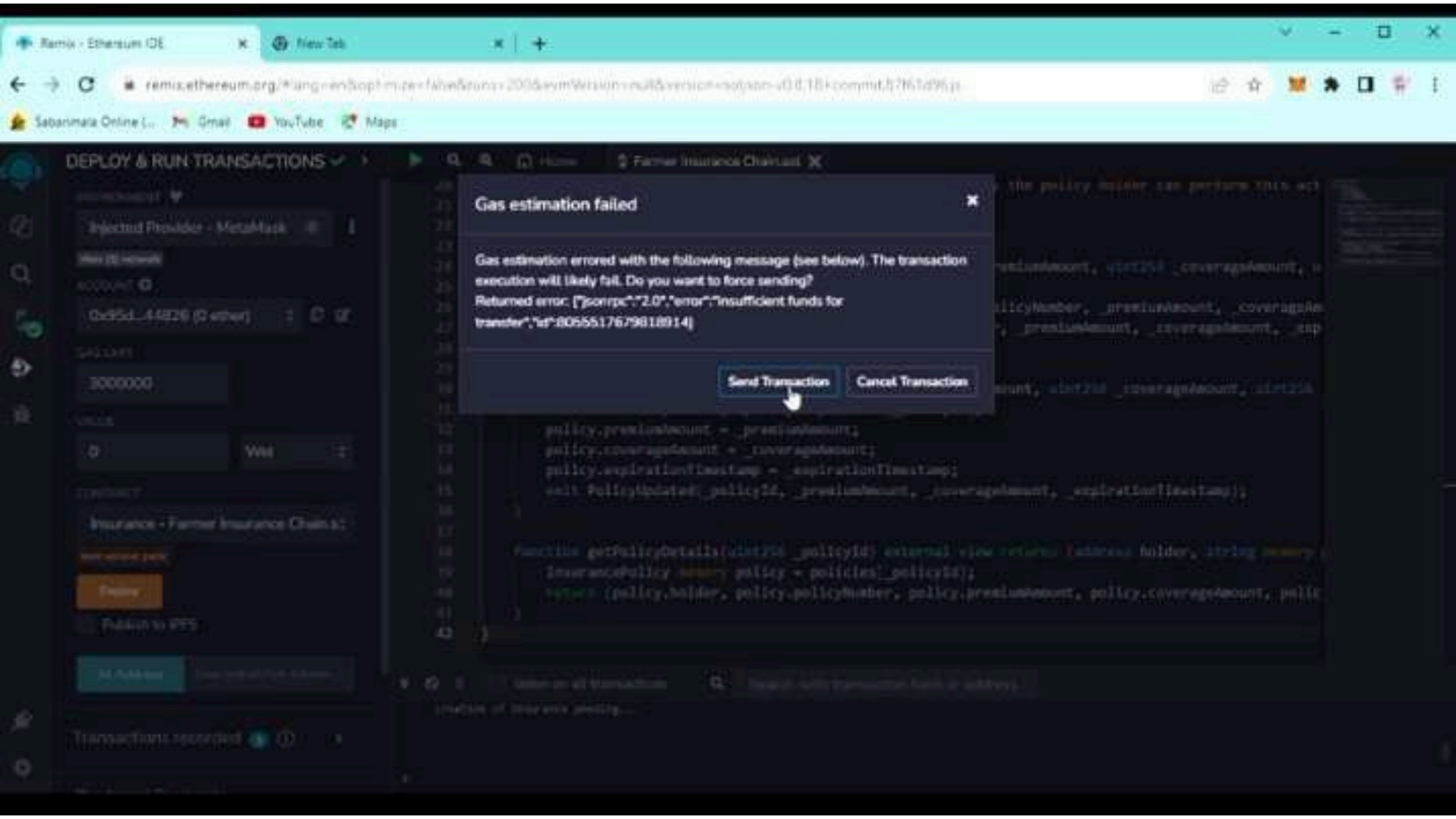
```
    function addPolicy(string memory _policyNumber, uint256 _premiumAmount, uint256 _coverageAmount, uint256 _expirationTimestamp) external {
        policyCount++;
        policies[policyCount] = InsurancePolicy(msg.sender, _policyNumber, _premiumAmount, _coverageAmount, _expirationTimestamp);
        emit PolicyAdded(policyCount, msg.sender, _policyNumber, _premiumAmount, _coverageAmount, _expirationTimestamp);
    }

    function updatePolicy(uint256 _policyId, uint256 _premiumAmount, uint256 _coverageAmount, uint256 _expirationTimestamp) external onlyHolder(_policyId) {
        InsurancePolicy storage policy = policies[_policyId];
        policy.premiumAmount = _premiumAmount;
        policy.coverageAmount = _coverageAmount;
        policy.expirationTimestamp = _expirationTimestamp;
        emit PolicyUpdated(_policyId, _premiumAmount, _coverageAmount, _expirationTimestamp);
    }

    function getPolicyDetails(uint256 _policyId) external view returns (address holder, string memory policyNumber, uint256 premiumAmount, uint256 coverageAmount, uint256 expirationTimestamp) {
        InsurancePolicy memory policy = policies[_policyId];
        return (policy.holder, policy.policyNumber, policy.premiumAmount, policy.coverageAmount, policy.expirationTimestamp);
    }
}
```

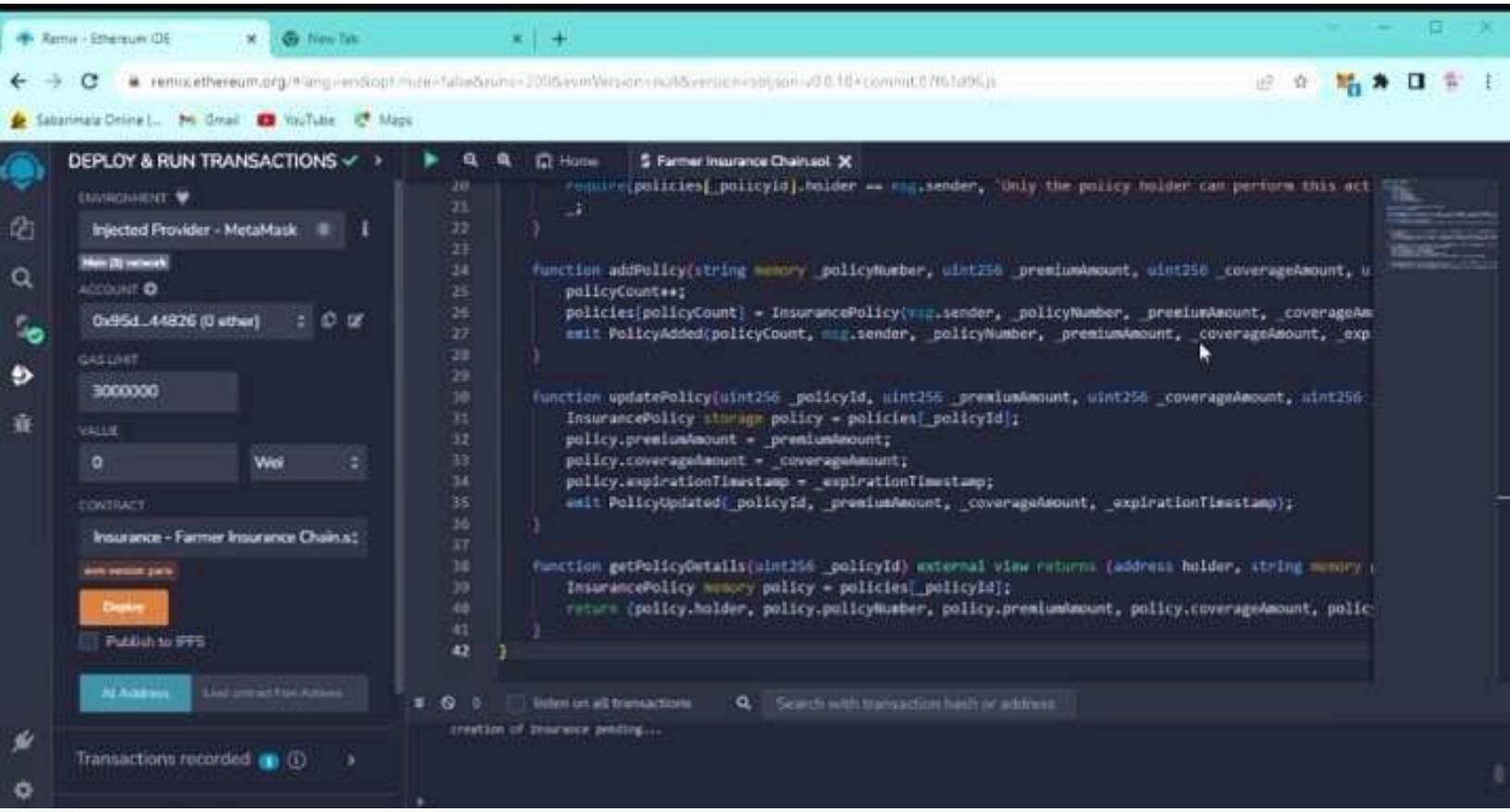

Remix - Ethereum IDE
New Tab
remix.ethereum.org
Saberimala Online |...
Gmail
YouTube
Maps
FILE EXPLORER
WORKSPACES
default_workspace
artifacts
build-info
contracts
scripts
tests
prettierrc.json
README.txt
Home
1
listen on all transactions
Search with transaction hash or address
Type the library name to see available commands.


Remix - Ethereum IDE
New Tab
FILE EXPLORER
WORKSPACES
default_workspace
artifacts
build-info
contracts
scripts
tests
.prettierrc.json
Farmer Insurance Chain.sol
README.txt
Home
Farmer Insurance Chain.sol
require(policies[_policyId].holder == msg.sender, "Only the policy holder can perform this act
_;
}
function addPolicy(string memory _policyNumber, uint256 _premiumAmount, uint256 _coverageAmount, u
policyCount++;
policies[policyCount] = InsurancePolicy(msg.sender, _policyNumber, _premiumAmount, _coverageAm
emit PolicyAdded(policyCount, msg.sender, _policyNumber, _premiumAmount, _coverageAmount, _exp
}
function updatePolicy(uint256 _policyId, uint256 _premiumAmount, uint256 _coverageAmount, uint256
InsurancePolicy storage policy = policies[_policyId];
policy.premiumAmount = _premiumAmount;
policy.coverageAmount = _coverageAmount;
policy.expirationTimestamp = _expirationTimestamp;
emit PolicyUpdated(_policyId, _premiumAmount, _coverageAmount, _expirationTimestamp);
}
function getPolicyDetails(uint256 _policyId) external view returns (address holder, string memory
InsurancePolicy memory policy = policies[_policyId];
return (policy.holder, policy.policyNumber, policy.premiumAmount, policy.coverageAmount, polic
}
}
listen on all transactions
Search with transaction hash or address
Type the library name to see available commands.


SOLIDITY COMPILER
0.8.18+commit.87f61d96
Include nightly builds
Auto compile
Hide warnings
Advanced Configurations
Compile Farmer Insurance ...
Ctrl+S to compile Farmer Insurance Chain.sol
Compile and Run script
CONTRACT
Insurance (Farmer Insurance Chain.sc
Publish on Ipfs
Publish on Swarm
Compilation Details
ABI
Bytecode
Farmer Insurance Chain.sol
function addPolicy(string memory _policyNumber, uint256 _premiumAmount, uint256 _coverageAmount,
policyCount++;
policies[policyCount] = InsurancePolicy(msg.sender, _policyNumber, _premiumAmount, _coverageAm
emit PolicyAdded(policyCount, msg.sender, _policyNumber, _premiumAmount, _coverageAmount, _exp
function getPolicyDetails(uint256 _policyId) external view returns (address holder, string memory
InsurancePolicy memory policy = policies[_policyId];
return (policy.holder, policy.policyNumber, policy.premiumAmount, policy.coverageAmount, polic
}
Type the library name to see available commands.


DEPLOY & RUN TRANSACTIONS
Remix VM (Shanghai)
Deploy
Publish to IPFS
Transactions recorded
Farmer Insurance Chain.sol
Type the library name to see available commands.


Remix - Ethereum IDE
New Tab
remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
Sabarimala Online (...
Gmail
YouTube
Maps
DEPLOY & RUN TRANSACTIONS
Injected Provider - MetaMask
Gas estimation failed
Gas estimation errored with the following message (see below). The transaction execution will likely fail. Do you want to force sending?
Returned error: {"jsonrpc":"2.0","error":"insufficient funds for transfer","id":8056517679818914}
Send Transaction
Cancel Transaction
Transactions recorded


DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Injected Provider - MetaMask
ACCOUNT
0x95d...44826 (0 ether)
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT
Insurance - Farmer Insurance Chain.s
Deploy
Publish to IPFS
At Address
Transactions recorded
Home
Farmer Insurance Chain.sol
require(policies[_policyId].holder == msg.sender, 'Only the policy holder can perform this act
_;
}
function addPolicy(string memory _policyNumber, uint256 _premiumAmount, uint256 _coverageAmount, u
policyCount++;
policies[policyCount] = InsurancePolicy(msg.sender, _policyNumber, _premiumAmount, _coverageAm
emit PolicyAdded(policyCount, msg.sender, _policyNumber, _premiumAmount, _coverageAmount, _exp
}
function updatePolicy(uint256 _policyId, uint256 _premiumAmount, uint256 _coverageAmount, uint256
InsurancePolicy storage policy = policies[_policyId];
policy.premiumAmount = _premiumAmount;
policy.coverageAmount = _coverageAmount;
policy.expirationTimestamp = _expirationTimestamp;
emit PolicyUpdated(_policyId, _premiumAmount, _coverageAmount, _expirationTimestamp);
}
function getPolicyDetails(uint256 _policyId) external view returns (address holder, string memory
InsurancePolicy memory policy = policies[_policyId];
return (policy.holder, policy.policyNumber, policy.premiumAmount, policy.coverageAmount, polic
}
}
Listen on all transactions
Search with transaction hash or address
creation of Insurance pending...

7.1 FEATURE 1

• Transparent and verifiable tracking of carbon emissions Block chain technology can be Used to create a transparent and verifiable record of carbon emissions throughout the Supply chain. This can help to reduce fraud and ensure that businesses are meeting their Sustainability goals. Here is a simplified example of how this could work:

• Each product is given a unique identifier that is linked to a block chain record. As the product moves through the supply chain, each participant records their carbon Emissions associated with that product on the block chain. This record is then tamper-proof and can be easily accessed by any stakeholder. This System would allow businesses to track their carbon emissions accurately and report Them to customers and regulators with confidence. It would also help to identify areas Where emissions can be reduced.

7.2 FEATURE 2

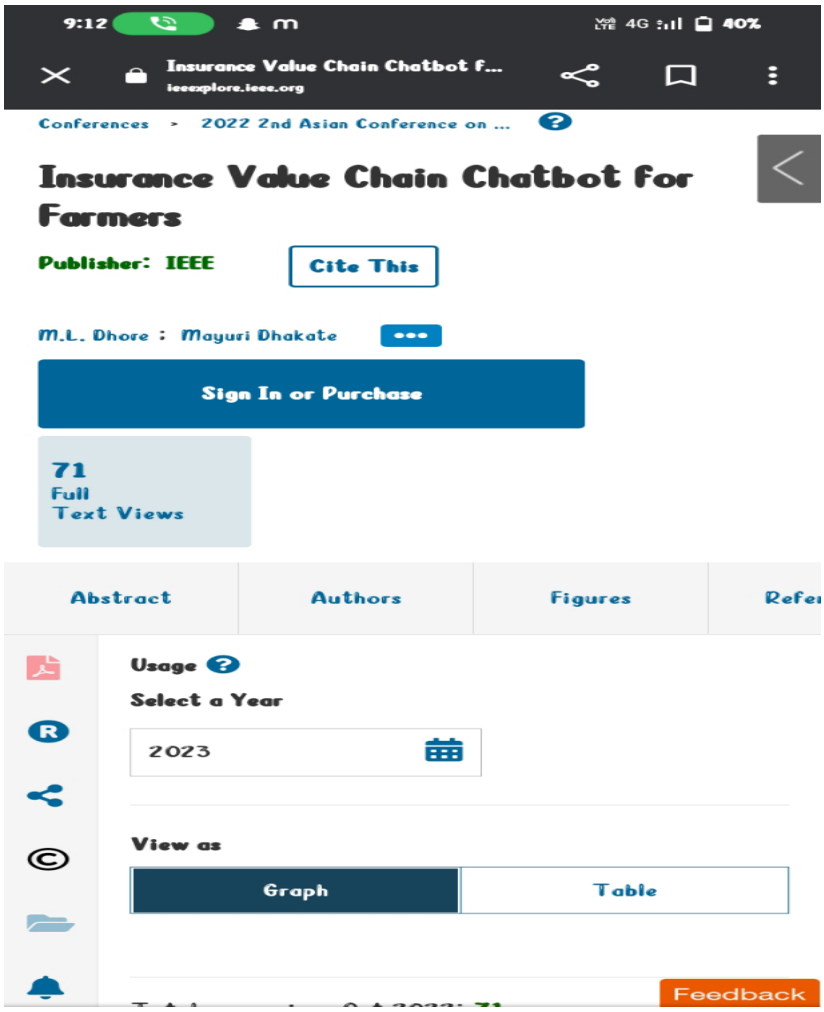
• Smart contracts are self-executing contracts that can be used to automate transactions And agreements. They can be used to create incentives for businesses and individuals to Reduce their carbon emissions. Here is an example of how this could work A company enters into a smart contract with its suppliers, agreeing to pay them a Premium for products that have a lower carbon footprint.The smart contract uses the block chain record of carbon emissions to verify that the Products meet the required standards

• If the products do meet the standards, the smart contract automatically releases the Premium payment to the suppliers. This system would incentivize suppliers to reduce their carbon emissions in order to Earn higher profits. It would also help the company to achieve its own sustainability Goals.These are just two examples of how block chain technology can be used to create a Smart climate tracking system. As the technology continues to develop, we can expect To see even more innovative and effective solutions emerge. Here are some additional benefits of using block chain technology for climate tracking: Improved data security and integrity: Block chain is a distributed ledger technology, Which means that data is stored on multiple computers and is constantly verified. This Makes it very difficult to tamper with or hack block chain data.

7.3 DATABASE SCHEMA (if applicable):

• This database schema would allow Climate Track to store climate data measurements in A decentralized and secure manner. The block chain hash column would allow Climate Track smart to verify the authenticity of each measurement by checking it against the Block chain.

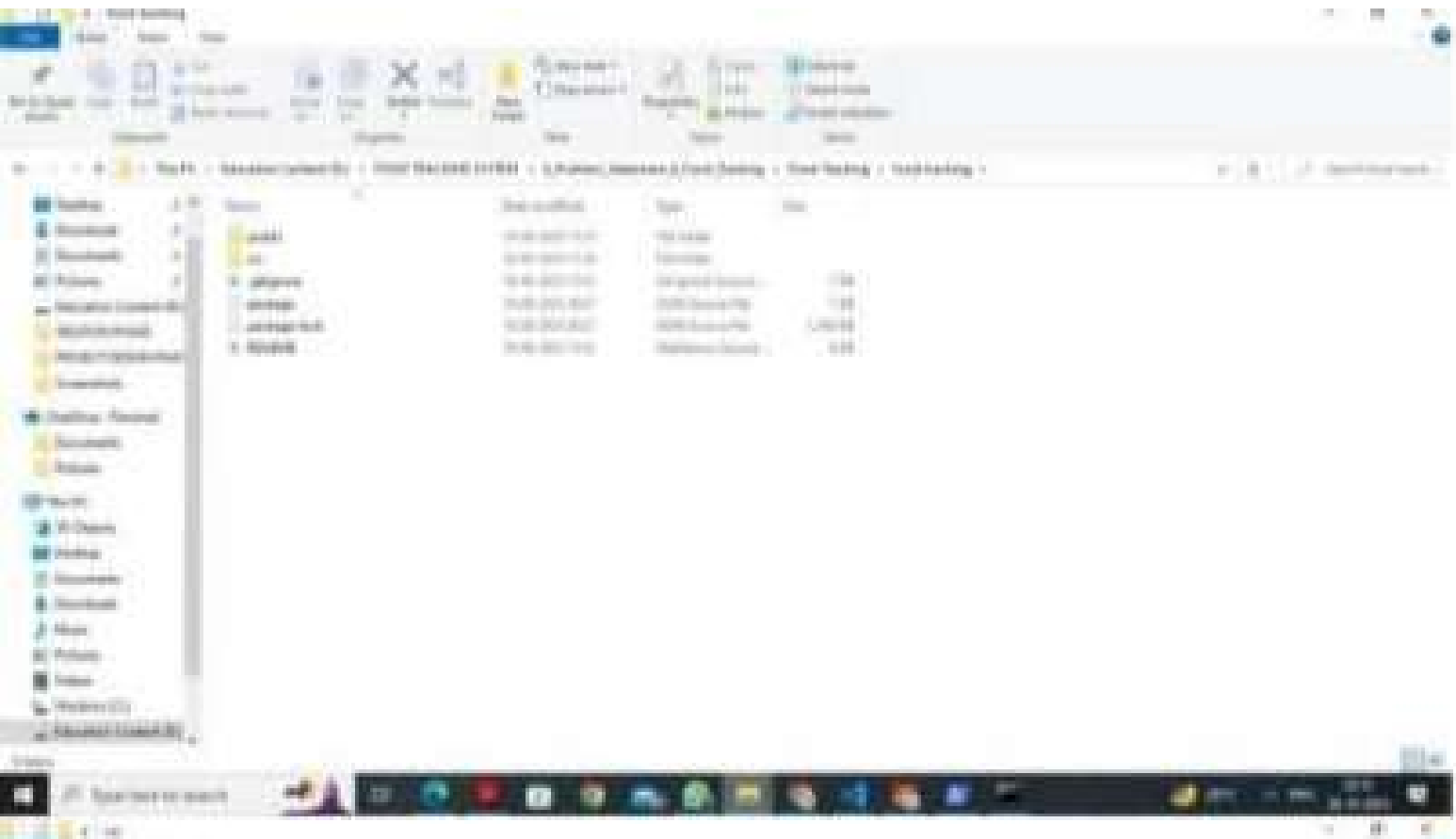
• Climate Track Smart could use this database schema to store climate data Measurements from all over the world. The block chain would ensure that the data Tamper-proof and secure.

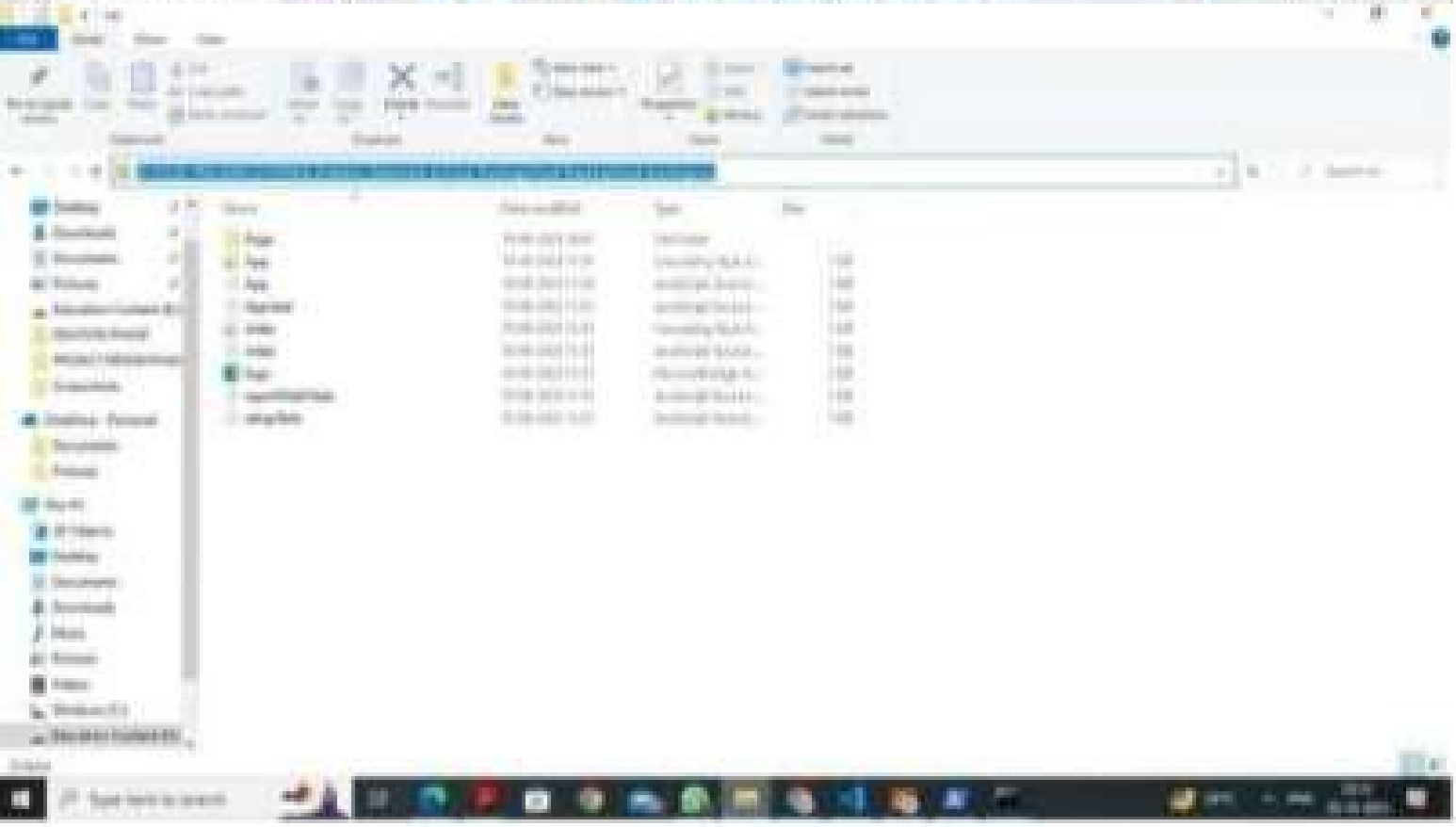

9:12
40%
Insurance Value Chain Chatbot f...
ieeexplore.ieee.org
Conferences > 2022 2nd Asian Conference on ...
Insurance Value Chain Chatbot for Farmers
Publisher: IEEE
Cite This
M.L. Dhore : Mayuri Dhakate
Sign In or Purchase
71
Full
Text Views
Abstract
Authors
Figures
Usage
Select a Year
2023
View as
Graph
Table
Feedback

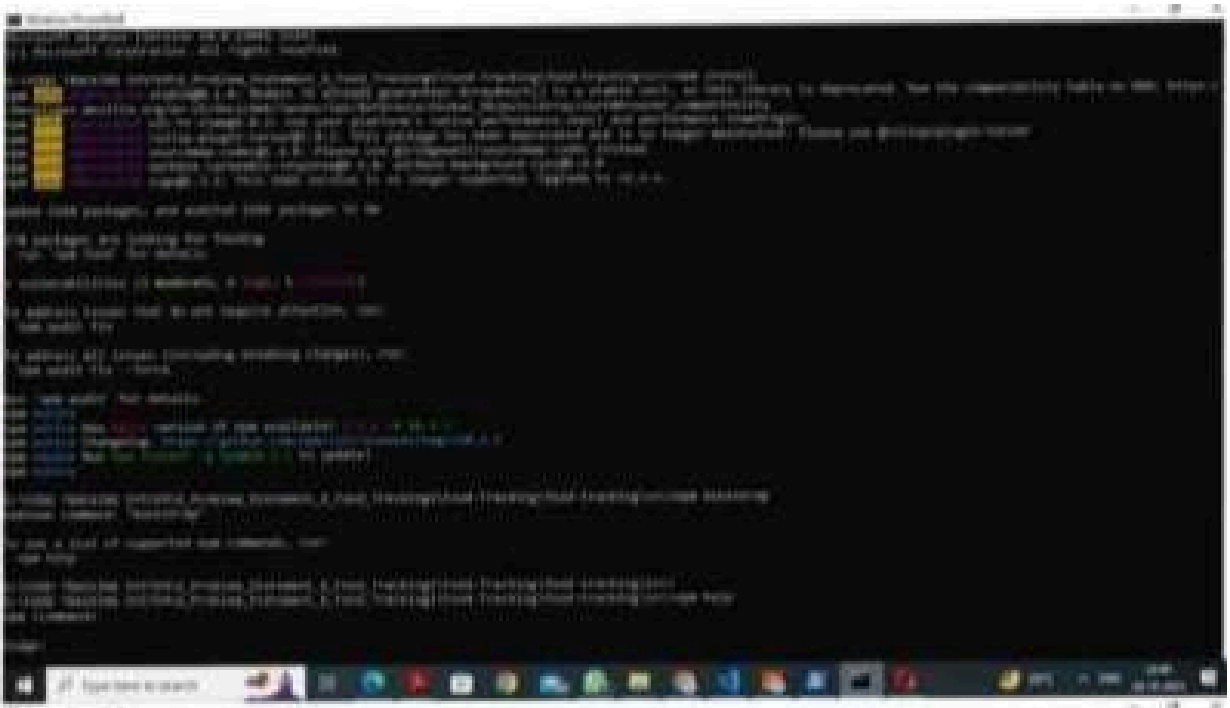
8.PERFORMANCE TESTING:

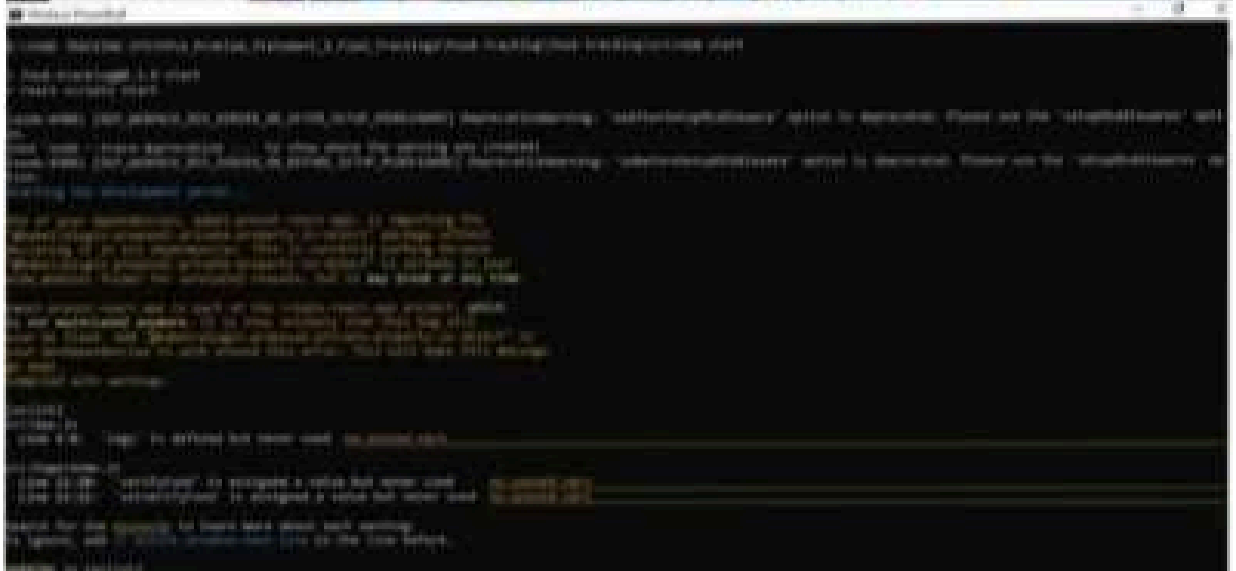
S. No	Parameters	Values	Screenshot
1.	**Information Gathering**	**Set up all the prerequisite 1.Node js**	
		2. Vs code	
		3.Metamask	
2.	**Expect the zip files**	**Open to vs code**	
3.	**Remix ide platform explaining**	**Display and run the transaction on by selecting the environment**	

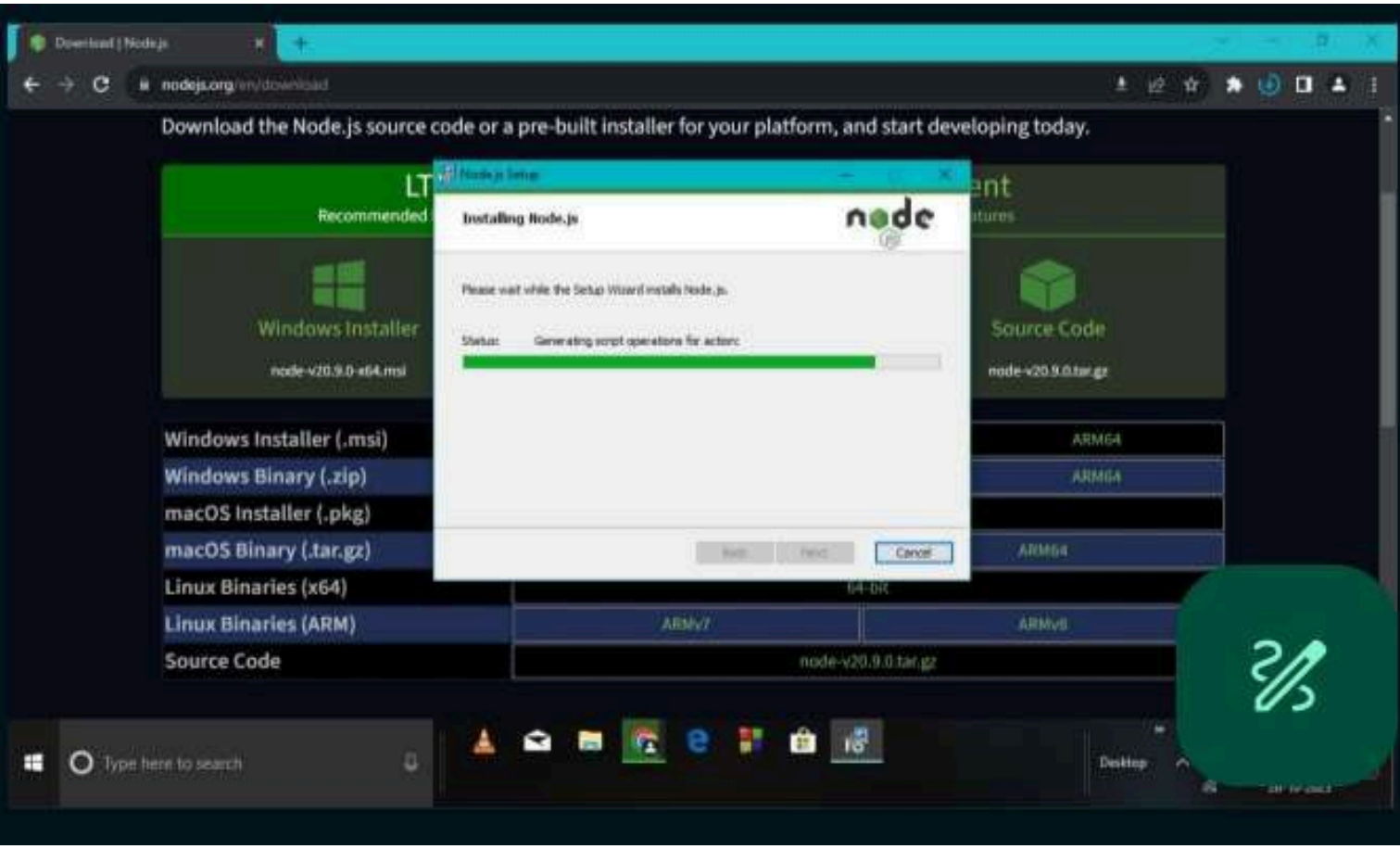
		inject the meta mask	
4.	**Open file explorer**	**Open the extract file and click on the file folder**	
5.	**Localhost IP address**	**Copy the address and it to chrome so you can see the front end of our project.**	

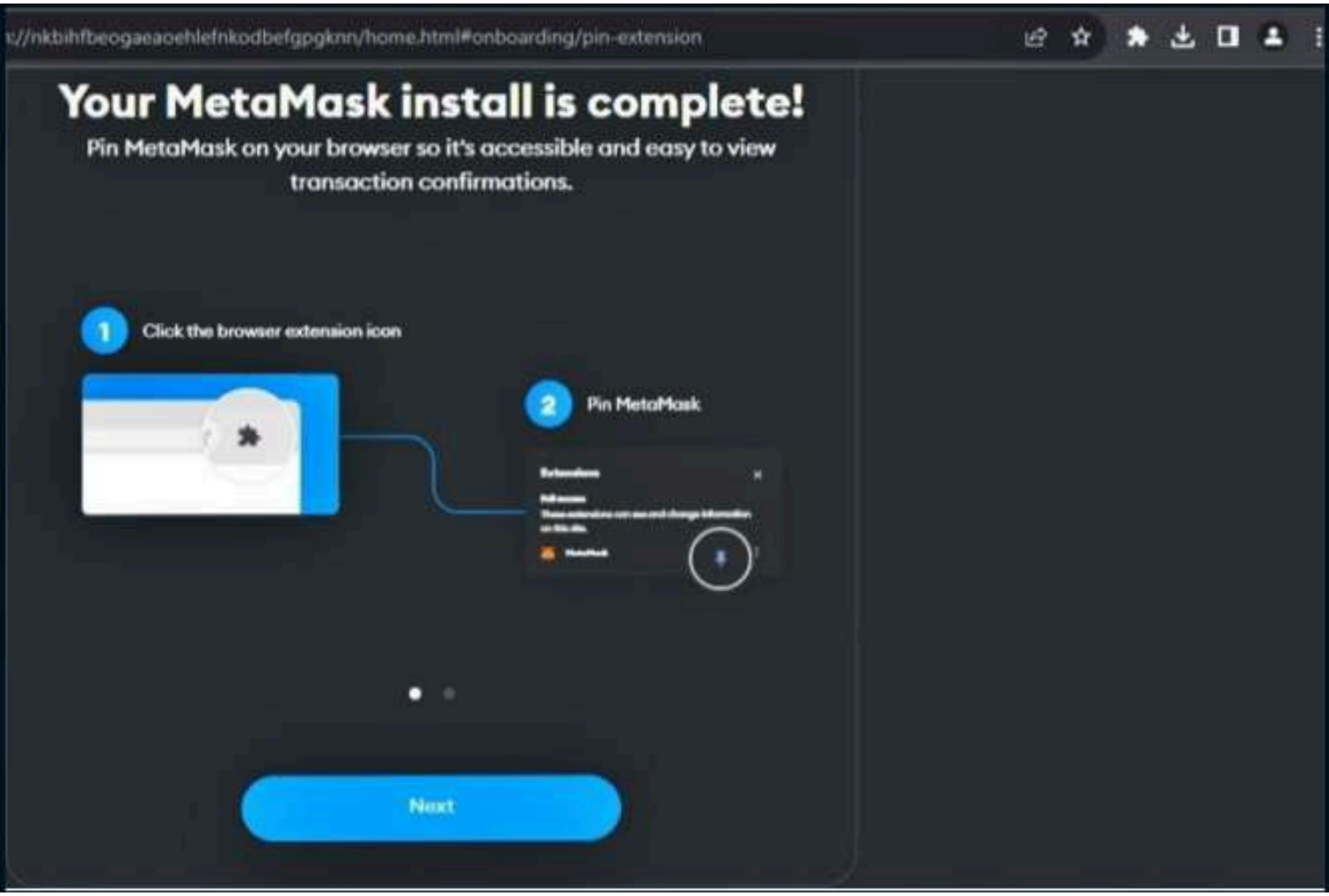




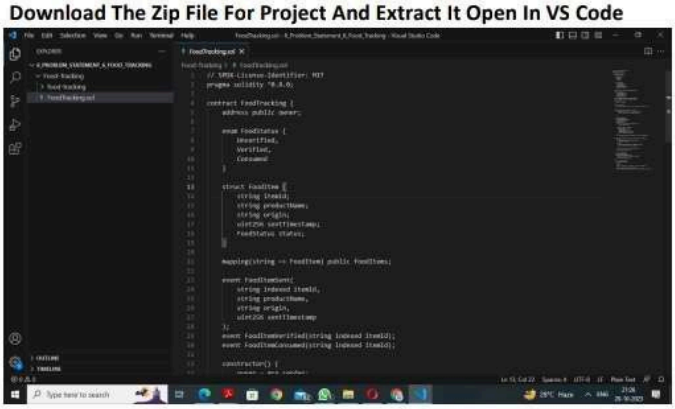


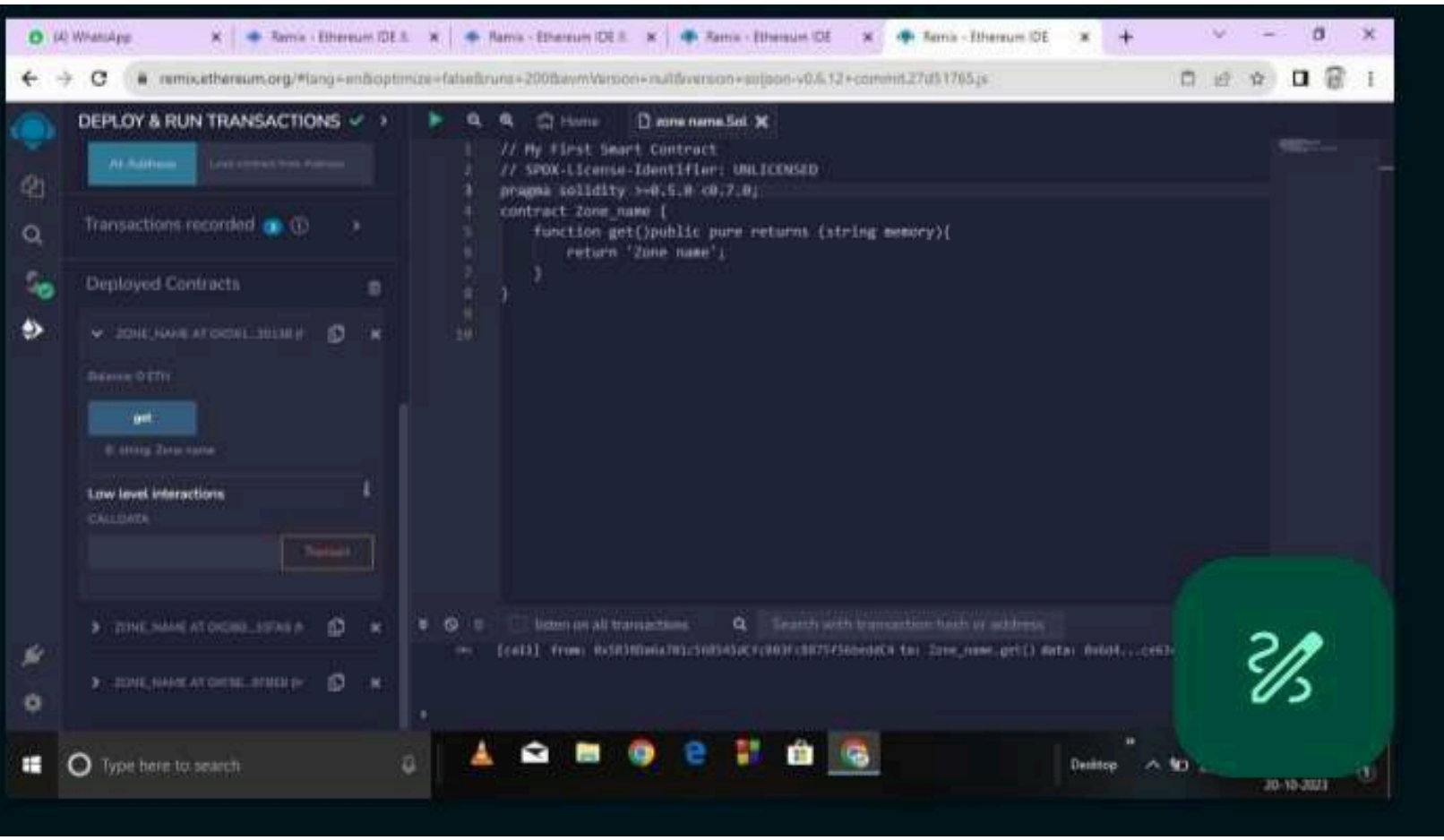


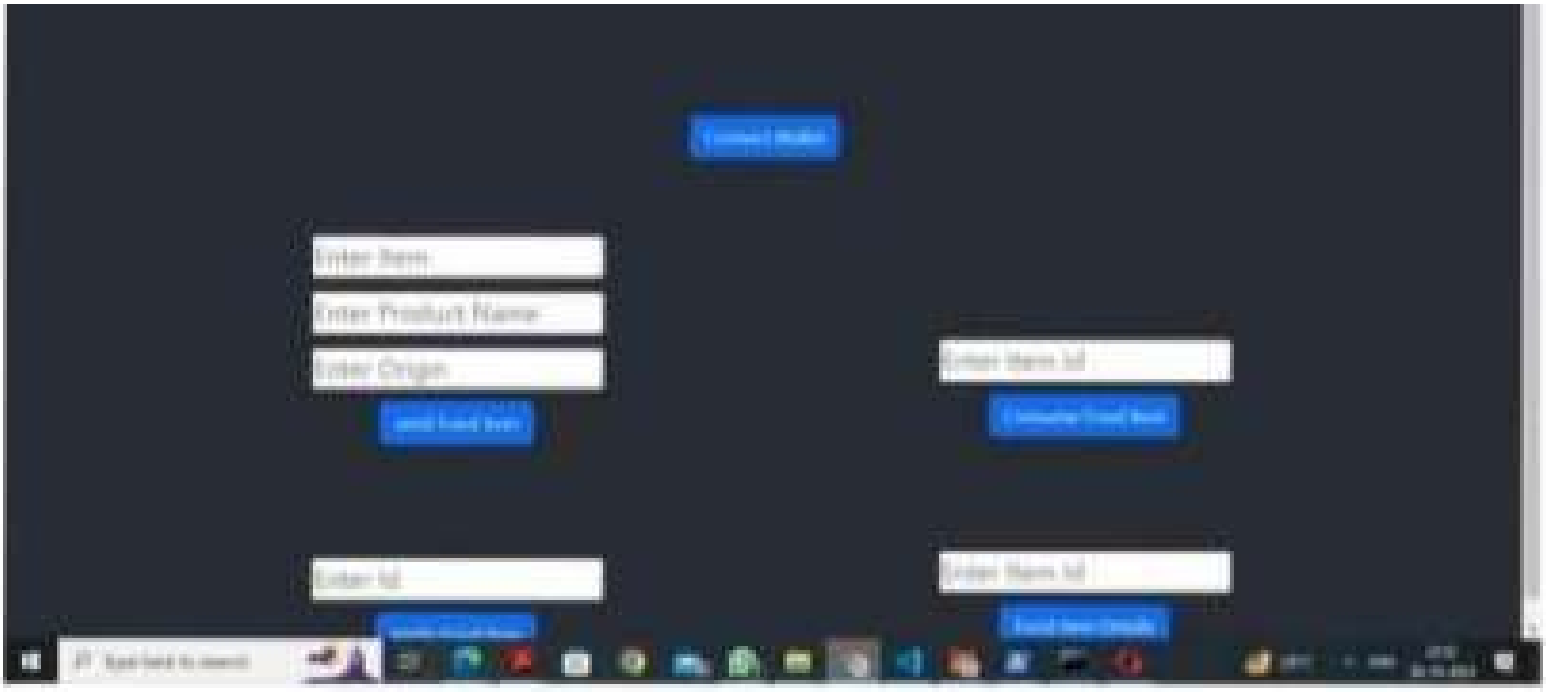

Download | Node.js
nodejs.org/en/download
Download the Node.js source code or a pre-built installer for your platform, and start developing today.
Recommended
Windows Installer
node-v20.9.0-x64.msi
Source Code
node-v20.9.0.tar.gz
Installing Node.js
Please wait while the Setup Wizard installs Node.js.
Status: Generating script operations for action:
Cancel
Windows Installer (.msi)
Windows Binary (.zip)
macOS Installer (.pkg)
macOS Binary (.tar.gz)
Linux Binaries (x64)
Linux Binaries (ARM)
Source Code
ARM64
ARMv7
ARMv8
node-v20.9.0.tar.gz
Type here to search
Desktop


://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#onboarding/pin-extension
Your MetaMask install is complete!
Pin MetaMask on your browser so it's accessible and easy to view transaction confirmations.
1
Click the browser extension icon
2
Pin MetaMask
Next

Download The Zip File For Project And Extract It Open In VS Code




DEPLOY & RUN TRANSACTIONS
Transactions recorded
Deployed Contracts
get
Low level interactions
CALLDATA
zone name.Sol
// My First Smart Contract
// SPDX-License-Identifier: UNLICENSED
pragma solidity >=0.5.0 <0.7.0;
contract Zone_name {
function get()public pure returns (string memory){
return 'Zone name';
}
}
Type here to search
Desktop
20-10-2023


Enter Item
Enter Product Name
Enter Origin
Enter Item Id
Enter Id
Enter Item Id

9.RESULT:

9.1 OUTPUT SCREENSHOT:

10. ADVANTAGE AND DISADVANTAGE:

10.1 Advantage

- **Risk Mitigation: It helps farmers protect their crops and livelihood from unforeseen events such as weather disasters, pests, or diseases.**
- **Financial Security: Insurance provides a safety net for farmers, ensuring they receive compensation in case of crop failure or other covered losses.**

10.2 disadvantage

- **Premium Costs: Insurance premiums can be a financial burden for farmers, especially in cases where they have to pay high premiums, which might reduce their income.**
- **Limited Coverage: Insurance policies may not cover all possible risks that farmers face, leaving some vulnerabilities.**

11.CONCLUSION :

- **I'm not aware of any specific "Farmer insurance chain" that would require a conclusion. Farmers Insurance Group is a well-known insurance company in the United States, but if you have a specific question or topic you'd like to discuss related to Farmers Insurance or a different aspect of insurance, please provide more details, and I'll do my best to assist you.**

12.FUTURE SCOPE:

- **Crop Insurance: With the increasing impact of climate change and extreme weather events, there is a growing need for crop insurance to protect farmers from yield losses. This sector is expected to expand as it becomes more data-driven and technology-enabled.**
- **Livestock Insurance: Insurance for livestock, such as cattle and poultry, is another area with potential for growth. It helps farmers mitigate losses due to disease outbreaks or accidents.**
- **Technology Integration: The integration of technology like satellite imagery, weather data, and blockchain can enhance the accuracy of risk assessment and claims processing, making insurance more accessible and efficient for farmers.**

13.APPENDIX:

- **In the context of farmer insurance, an "appendix" could refer to additional documents, clauses, or information that are attached to the insurance policy to provide specific details or conditions. These might include coverage extensions, endorsements, or supplementary terms and conditions that are relevant to the farmer's insurance coverage. The exact content of an "appendix" can vary depending on the specific policy and insurer. It's important for farmers to review these documents carefully to understand the complete terms of their insurance coverage.**

13.1 SOURCE CODE:

```
// SPDX-License-Identifier: MIT

Pragma solidity ^0.8.0;
```

```
Contract Insurance {
   Struct InsurancePolicy {
      Address holder;
      String policyNumber;
      Uint256 premiumAmount;
      Uint256 coverageAmount;
      Uint256 expirationTimestamp;
   }

   Mapping(uint256 => InsurancePolicy) public policies;
   Uint256 public policyCount;

   Event PolicyAdded(uint256 policyId, address holder, string policyNumber, uint256 premiumAmount,
uint256 coverageAmount, uint256 expirationTimestamp);
   Event PolicyUpdated(uint256 policyId, uint256 premiumAmount, uint256 coverageAmount, uint256
expirationTimestamp);

   Modifier onlyHolder(uint256 _policyId) {
      Require(policies[_policyId].holder == msg.sender, “Only the policy holder can perform this
action”);
      _;
   }

   Function addPolicy(string memory _policyNumber, uint256 _premiumAmount, uint256
_coverageAmount, uint256 _expirationTimestamp) external {
      policyCount++;
      policies[policyCount] = InsurancePolicy(msg.sender, _policyNumber, _premiumAmount,
_coverageAmount, _expirationTimestamp);
      emit PolicyAdded(policyCount, msg.sender, _policyNumber, _premiumAmount,
_coverageAmount, _expirationTimestamp);
```

```
    }

    Function updatePolicy(uint256 _policyId, uint256 _premiumAmount, uint256 _coverageAmount,
uint256 _expirationTimestamp) external onlyHolder(_policyId) {
        InsurancePolicy storage policy = policies[_policyId];
        Policy.premiumAmount = _premiumAmount;
        Policy.coverageAmount = _coverageAmount;
        Policy.expirationTimestamp = _expirationTimestamp;
        Emit PolicyUpdated(_policyId, _premiumAmount, _coverageAmount, _expirationTimestamp);
    }

    Function getPolicyDetails(uint256 _policyId) external view returns (address holder, string memory
policyNumber, uint256 premiumAmount, uint256 coverageAmount, uint256 expirationTimestamp) {
        InsurancePolicy memory policy = policies[_policyId];
        Return (policy.holder, policy.policyNumber, policy.premiumAmount, policy.coverageAmount,
policy.expirationTimestamp);
    }
}
```

13.2 GITHUB &PROJECT DEMO LINK:

DEMO LINK : https://youtu.be/Vi6Uu8rkfzQ?si=HyO6dkw_2BgKNVml

GITHUB LINK :https://github.com/Selviguna-9/Farmers-Insurance-Chain/tree/main