BLOCKCHAIN BASED SOLUTION TO FRAUD-PROOF GOVERNMENT AIDED SCHEMES

Sai Nithish Mucherla
17BCE0449
8870665582
mucherlasai.nithish2017@vitstudent.ac.in

Dheeraj Gadwala
17BCE0620
8019334644
sai.dheerajreddy2017@vitstudent.ac.in

Teja Potnuru
17BCE2020
9550225822
potnuru.teja2017@vitstudent.ac.in

Balakrishnan P
Associate Professor Grade 1
7550357648
Balakrishnan.p@vit.ac.in

B.Tech.

in

Computer Science and Engineering

School of Computer Science & Engineering

®

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

DECLARATION

We hereby declare that the thesis entitled *“Blockchain based solution to fraud-proof government aided schemes”* submitted by us, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering* to VIT is a record of bonafide work carried out by us under the supervision of *Balakrishnan P*.

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date : 09.06.2021

Signature of the Candidate
(17BCE0449)

Saidheeraj

Signature of the Candidate
(17BCE0620)

P. teja

Signature of the Candidate
(17BCE2020)

CERTIFICATE

This is to certify that the thesis entitled *"Blockchain based solution to fraud-proof government aided schemes submitted"* submitted by **Sai Nithish Mucherla (17BCE0449), Dheeraj Gadwala (17BCE0620) and Teja Potnuru (17BCE2020),** School of Computer Science and Engineering [SCOPE], VIT, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering*, is a record of bonafide work carried out by them under my supervision during the period, 01. 12. 2018 to 30.04.2019, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : 09.06.2021

Signature of the Guide

Internal Examiner

External Examiner

HOD: Dr. VAIRAMUTHU S
Program: B. Tech CSE

ACKNOWLEDGEMENTS

We wish to take this opportunity to express our gratitude to all the people who have extended their cooperation in various ways during the project work. It’s our pleasure to acknowledge the help of all those individuals.

We would like to thank our project guide, Prof. Balakrishnan P, for his guidance and help throughout the development of this project work by providing us with the required information.

We would also like to thank our Head of the Department, Dr. Vairamuthu S for his guidance and help throughout the development of this project work by providing us with the required information.

We would also like to thank our Dean, SCOPE who is supportive in all ways and helped us in many ways by providing the required information. Without guidance, cooperation and encouragement, we couldn’t have learned many things without his help throughout our engineering career.

We are exceedingly obligated to VIT University, Vellore for their direction and steady supervision and also to provide fundamental data with respect to the task and likewise for their help in finishing the project.

Sai Nithish Mucherla,
Dheeraj Gadwala,
Teja Potnuru.

Executive Summary

The Indian government takes various initiatives to better our nation, for instance, the Mid-day meal scheme. However, due to the lack of transparency, improper record management, and beneficiary identification, it is not uncommon to hear about misuse and mismanagement of resources in schemes like these. This paper proposes a generic architecture to leverage the possible benefits of blockchain technology and smart contracts to address such inefficiencies in government-aided schemes. With keen observation, one can notice a standard structure in these schemes. In general, a government scheme may have the following steps: scheme initiation, fund-transfer/allocation, and or allocation of raw-materials (E.g., food grains in case of Mid-day meal scheme). At the beneficiary end, a process of authentication occurs, followed by updating of records and collection of subsidies. In the current architecture, funds/raw materials are transferred/allocated to the beneficiaries via a layered architecture. The proposed architecture aims to eliminate human intervention from these processes to the maximum extent possible. The authentication process can be digitalized by using sophisticated technologies like biometrics, face-recognition, among others. This process is followed by securely storing the data and transactions using blockchain technology, which is famously tamper-resistant. Combining these technologies can potentially revolutionize such government schemes, allowing them to do what they were truly meant to do; serve the Indian population.

Keywords: Government aided schemes, Blockchain, Smart Contracts.

CONTENTS

S. NO.	CONTENT	PAGE NO.
	Acknowledgement	**i**
	Executive Summary	**ii**
	Table of Contents	**iii**
	List of Figures	**v**
	List of Tables	**xii**
	Abbreviations	**xiii**
	Symbols and Notations	
1	**INTRODUCTION**	**1**
	1.1 Theoretical Background	**2**
	1.2 Motivation	**3**
	1.3 Aim of the Proposed Work	**4**
	1.4 Objective(s) of the Proposed Work	**4**
2	**LITERATURE SURVEY**	**4**
	2.1 Survey of the Existing Models/Work	**4**
	2.2 Summary/Gaps identified in the Survey	**6**
3	**OVERVIEW OF THE PROPOSED SYSTEM**	**7**
	3.1 Introduction	**7**
	3.2 Framework of proposed model	**9**
	3.3 Proposed System Model	**17**
4	**PROPOSED SYSTEM ANALYSIS AND DESIGN**	**19**
	4.1 Introduction	**19**
	4.2 Requirement Analysis	**19**
	4.2.1 Functional Requirements	**19**
	4.2.1.1 Product Perspective	**19**
	4.2.1.2 Product Features	**19**
	4.2.1.3 User Characteristics	**19**

	4.2.1.4 Assumptions and Dependencies	**20**
	4.2.1.5 Domain Requirements	**20**
	4.2.1.6 User Requirements	**20**
	4.2.2 Non-Functional Requirements	**21**
	4.2.2.1 Product Requirements	**21**
	4.2.2.1.1 Efficiency	**21**
	4.2.2.1.2 Reliability	**21**
	4.2.2.1.3 Usability	**21**
	4.2.2.2 Organizational Requirements	**21**
	4.2.2.2.1 Implementation Requirements	**21**
	4.2.2.2.2 Engineering Standard Requirements	**21**
	4.2.2.3 Operational Requirements	**21**
	4.2.2.3.1 Economic	**21**
	4.2.2.3.2 Social	**21**
	4.2.2.3.3 Inspectability	**22**
	4.2.3 System Requirements	**22**
	4.2.3.1 Software Requirements	**22**
5	**RESULTS AND DISCUSSION**	**22**
6	**REFERENCES**	**60**
	APPENDIX	

LIST OF FIGURES

Figure No.	Title	Page No.
1.1	Current Architecture	**1**
3.1.1	Depiction of similar structure in the transfer of resources from one layer to another	**8**
3.1.2	Depiction of the presence of a similar structure in the transactions between the last two layers	**8**
3.2.1	Initiation of beneficiary registration in the prototype	**12**
3.2.2	Distributed validation process for beneficiary registration process in the prototype.	**13**
3.2.3	Beneficiary authentication for transfer of resources in the prototype.	**16**
3.3.1	Proposed System Model	**17**
3.3.2	Configured blockchain network for development	**18**
5.1	Landing Page of the portal	**23**
5.2	Admin login screen	**23**

5.3	Admin login prompt (Uses metamask along with smart contracts (Credential Manager smart contract) to authorize Authorities & here admin)	**24**
5.4	Result: Unauthorized login	**24**
5.5	Selected account for login that resulted in an unauthorized login:0x173897136202ef44a2fe1FCFabE701BBcf2e63Cb	**25**
5.6	Admin Screen/portal on successful login. Account selected:0xf17f52151EbEF6C7334FAD080c5704D77216b732	**25**
5.7	Assign authority to distribution point 1. Authority id/address: 0xAB6e2c81675bb30411054F915e04393FA9D20441	**26**
5.8	Metamask prompt for signing the (add authority) transaction	**26**
5.9	Result: Successful assignment of authority to distribution point 1	**27**
5.10	Result: Authority verification success	**27**
5.11	Result: Verification for unauthorized authority	**28**
5.12	Login screen for distribution points	**28**
5.13	Unauthorized login attempt for distribution point -1	**29**
5.14	Portal for distribution points after successful login (Currently logged in authority: 3C28961eeB55d82b212B848DD779f7467D77a1A4)	**29**

5.15	Result: Invalid data entry for beneficiary registration	**30**
5.16	Captured beneficiary image for registration	**30**
5.17	Metamask Prompt to sign Beneficiary registration transaction	**31**
5.18	Result: Successful Beneficiary registration initiation	**31**
5.19	Registration requests for distribution point -1 authority. (Currently logged in authority: 3C28961eeB55d82b212B848DD779f7467D77a1A4)	**32**
5.20	Registration requests for distribution point -1 authority. (Currently logged in authority: AB6e2c81675bb30411054F915e04393FA9D20441)	**32**
5.21	Registration requests for distribution point -1 authority. (Currently logged in authority: E8C26fC27209066C4d2F0ca9703B2c4B604904B5)	**33**
5.22	Beneficiary (id:123123123123) rejected by authority-3	**33**
5.23	Registration requests for distribution point -1 authority. (Currently logged in authority: AB6e2c81675bb30411054F915e04393FA9D20441)	**34**
5.24	Result: Registration status of beneficiary 123123123123 after approval by authority-1	**35**
5.25	New Beneficiary registration (id:2123213213213) initiated by authority-1	**36**
5.26	Beneficiary registered (id:2123213213213) to the scheme after approval by initiated by authority-2	**37**

5.27	Registration requests for distribution point -1 authority after the registration of beneficiary (id:213213213213) (Currently logged in authority: E8C26fC27209066C4d2F0ca9703B2c4B604904B5)	**38**
5.28	New Beneficiary registration (id:143144145146) initiated by authority-1	**38**
5.29	Result: Registration status of beneficiary 143144145146 after rejection by authority-1	**39**
5.30	Result: Registration status of beneficiary 143144145146 after rejection by authority-2	**40**
5.31	Resource transfer initiation page with from Entity selected: ‘Distribution point’.	**41**
5.32	Image showing registered beneficiaries under distribution point-1	**42**
5.33	Image showing resource-transfer-initiation from ‘Central’ to ‘Andhra Pradesh’	**42**
5.34	Image showing metamask prompt to sign the resource-transfer-transaction (Currently logged account: f17f52151EbEF6C7334FAD080c5704D77216b732 [Authorized as central authority in the smart contract])	**43**
5.35	Image showing successful resource-transfer-initiation transaction txnId: 0x99794cb0d697ab249b6d0feabff2e200bde97b059f13c77f53a5f28d9a61beed	**43**
5.36	Result: transaction list with the transaction b/n central and Andhra Pradesh	**44**
5.37	Assigning authority (0xA95E385E92896A4153bbE46Cd19BABa023Bd9A99) to Andhra Pradesh	**44**

5.38	Authority assignment successful for Andhra Pradesh (0xA95E385E92896A4153bbE46Cd19BABa023Bd9A99) – Authority id for Andhra Pradesh	**45**
5.39	Interface used to acknowledge the receipt of resources using QR codes	**45**
5.40	Image showing an active QR reader	**46**
5.41	Metamask prompt to sign an acknowledgment txn (Currently selected account: f17f52151EbEF6C7334FAD080c5704D77216b732 {Unauthorized for acknowledging since the receipt is supposed to be Andhra Pradesh})	**46**
5.42	Image showing response of an unauthorized transaction	**47**
5.43	Metamask prompt to sign the receipt acknowledgement (Currently selected account: 0xA95E385E92896A4153bbE46Cd19BABa023Bd9A99 {Authorized})	**47**
5.44	Image showing response of a successful receipt acknowledgment	**48**
5.45	Transaction list with updated receiver details for transaction b/n Central & Andhra Pradesh	**48**
5.46	Image showing resource transfer transaction from Central to Andhra Pradesh with selected account: 0xA95E385E92896A4153bbE46Cd19BABa023Bd9A99 (Unauthorized to perform the transaction as the authorized authority should be of Central entity)	**49**
5.47	Response of an unauthorized resource transaction request	**49**
5.48	Transaction list showing Unauthorized transaction request	**50**

5.49	Result: Response of acknowledging an already acknowledged transaction	**50**
5.50	Resource transfer at distribution point-1 with beneficiary image (id:123123123123) captured	**51**
5.51	Resource transfer at distribution point with beneficiary image captured (Selected account: 0xAB6e2c81675bb30411054F915e04393FA9D20441)	**51**
5.52	Result: Successful transaction at distribution point-1 implying a successful beneficiary authentication (Facial-image-recognition-based authentication). Transaction Id: 0xa5cd336b5b532c68c603b2047c3683a149aedef58b14348b823f31362dea774f	**52**
5.53	Transaction list with transaction b/n distribution point-1 and beneficiary (id:123123123123)	**52**
5.54	Distribution point transaction with invalid beneficiary image	**53**
5.55	Result: Image showing failed beneficiary authentication	**53**
5.56	Transaction result showing unauthenticated beneficiary transaction	**54**
5.57	Transaction filter interface	**54**
5.58	Transaction filter applied: {From Entity: central}	**55**
5.59	Transactions from Central entity	**55**

5.60	Transaction filter applied {From entity: dp-2 (distribution point-2)}	**56**
5.61	Transactions by distribution point-2	**56**
5.62	Transaction filter applied {From entity: dp-1 (distribution point-1)} and level selected as Distribution point level	**57**
5.63	Transactions at distribution level	**57**
5.64	Grafana Dashboard visualizing node metrics polled by data source Prometheus	**58**
5.65	Graphical data showing the block time of each node and the number of blocks each Besu node is behind the main chain	**58**
5.66	Graphical representation of CPU utilization by each Besu node	**59**
5.67	Graphical representation showing RAM utilization by each Besu node	**59**

LIST OF TABLES

Table No.	Title	Page No.
2.1	Overview of literature survey and the gaps identified.	**6**
3.1.1	Attributes required for transaction management.	**9**
3.2.1	Attributes collected for beneficiary registration	**11**
3.2.2	Final Status of beneficiary based on percentage of authorities' decisions	**12**
3.2.3	Sample states during beneficiary validation process.	**13**
3.3.1	Comparison of consensus algorithms	**18**
5.1	Table showing beneficiary (id:123123123123) status after authority-3's rejection	**34**
5.2	Table showing beneficiary (id:123123123123) status after authority-1's approval	**35**
5.3	Table showing beneficiary status (id:213213213213) after authority-1's registration initiation	**36**
5.4	Table showing beneficiary status (id:213213213213) after authority-2's approval	**37**
5.5	Table showing beneficiary status (id:143144145146) after authority-3's registration initiation	**39**
5.6	Table showing beneficiary status (id:143144145146) after authority-1's rejection	**40**

LIST OF ABBREVIATIONS

JVM	Java Virtual Machine
PDS	Public Distribution System
FCI	Food Corporation of India
FSC	Food Supply Chain
IoT	Internet of Things
IPFS	Interplanetary File System
QR	Quick Response
AUA	Authentication User Agency
RAM	Random Access Memory
GB	Gigabyte
REST	Representational State Transfer
HTTP	Hypertext Transfer Protocol
IPC	Inter Process Communication
JS	JavaScript
PoW	Proof of Work
PoA	Proof of Authority
IBFT	Istanbul Byzantine Fault Tolerant

1 INTRODUCTION

Government-aided schemes are initiatives taken by the government for the upliftment of various classes of the society and the nation's well-being. A few of these are the Mid-day meal scheme, the public distribution system, and others. These schemes play a crucial role in the development of our nation. The general structure of any government scheme has the following components: a) Initiation, b) Verification, c) Fund or Value transfer. Usually, such schemes follow a layered architecture with every layer comprising of different authorities and units [1]. For instance, any scheme initiated by the central government will have different layers. The central government is at the topmost layer, followed by the state governments, district authorities and other intermediaries at the middle layer and beneficiaries at the bottom-most layer.

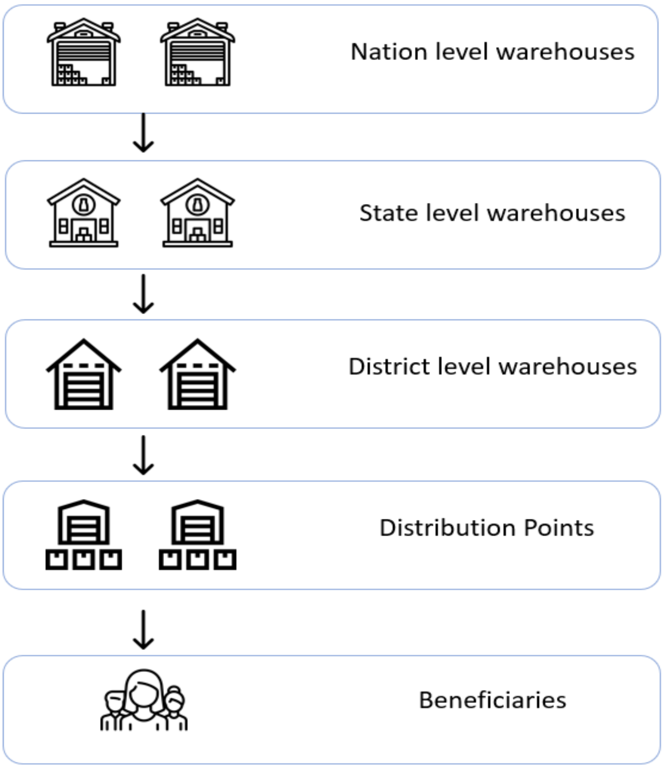


Fig 1.1 Current Architecture

1.1 Theoretical background

a) Current Architecture:

The current architecture is more centralized in terms of the fund/value flow. Transactions at a given level might not be open for other levels. The existing architecture induces quite a few complications, which might include:

- Fund Stagnation: Intentional or Unintentional suppression of funds at the middle levels.
- Obscurity of the process: Due to the central nature of the current architecture, transactions at a certain level may not be available for other levels, creating a lack of trust in the flow.
- Misappropriation of resources:

  Though the current model has monitoring to a certain extent, due to the schemes' manual and centralised nature, there have been instances where the resources were stolen. According to a news article by India Today in 2019, about 155 bags of meals, which were supposed to be a part of Mid-day meal allocation, were found at an animal fodder storage place of Rae Bareli [9]. In another instance, a school's authorities misreported student attendance to purchase more food grains and benefit from it.

b) Introduction to blockchain:

Blockchain is relatively new in the discipline of computer science. It is a decentralized data structure where every node refers to the node before it. Once new blocks are added to a blockchain, it is implausible to tamper with as the transaction is shared with all the network peers. Each block is built using the previous block's nonce and signature and contains transactional data. It is the network's miners' responsibility to add these blocks to the blockchain. Blockchain has many potential use cases because of its immutable and irreversible characteristics, making it exceptionally resilient to tampering. Additionally, this technology can only be used in a decentralized manner, implying that no server/ participant can dominate the network. The authors of [7] present its progression since its arrival and shed light on contemporary research on blockchain.

This paper proposes a model for India's various public distribution systems, utilizing blockchain technology and its assets.

Karan Sachan has presented how using blockchain with aadhar data can be used to make the welfare services provided by the Indian government more effective by eliminating corruptive elements and mismanagement in [5]. The author details the following use cases of blockchain with the Aadhar system that the Indian government can use: identity management, land registration, voting, managing healthcare records, public distributions systems, humanitarian aids and fraud detection while signing insurance premiums, among others. The author has listed various government schemes linked with aadhar that could benefit from blockchain technology.

c) Smart contracts:

Smart contracts are computer protocols that ensure the verification and execution of contracts made on blockchains. Smart contracts are deployed on blockchains and are hence very secure and tamper resistant. These contracts are executed in a decentralized network of anonymous nodes. Smart contracts have a value and a state attribute along with sets of trigger conditions and responses. After deploying a smart contract to a blockchain, the terms of the contract automatically get enforced on adding new blocks, i.e., they can mimic business contracts, consequently reducing administrative overhead. For instance, a smart contract for insurance policies would work as follows, firstly, the terms of the contract are agreed upon by all parties involved. These terms exist in the blockchain and are not alterable. When a specific event triggers the insurance policy, the smart contract policy is executed, and the payment is made instantaneously.

In [8], the authors offer an overview of smart contracts, their operating mechanisms, and their various application scenarios. The model proposed in this paper uses smart contracts in recording transaction of resources and funds provided by the Indian government as they pass through multiple intermediaries to reach the aimed beneficiaries.

1.2 Motivation

Large scams regularly make national news in our nation. As engineering students and individuals with a strong sense of social responsibility, we looked at these issues and started thinking about various possible solutions. These ideas culminated into our prototype application, that uses blockchain technology to maintain records of the transactions that take place in government aided schemes.

1.3 Aim and objectives

Our aim of fraud-proofing government aided schemes and consequently reducing human involvement by leveraging the benefits of blockchain can be accomplished by the following objectives:

- Designing an architecture to eliminate the flaws pointed out in the current architecture.
- Developing a prototype of the proposed architecture:
    1. Implementing necessary authentication modules.
    2. Developing smart contracts for inventory and transaction management.
    3. Creating and hosting the prototype application on a private blockchain network.

2 LITERATURE REVIEW

2.1 Survey of existing work

Saiharsha B. Et. Al. suggest a model for the mid-day meal scheme using fingerprint and blockchain technologies to minimize human intervention with the scheme in [4]. They describe the daily process of the scheme to be happening in three phases by their model. In the first phase, a fingerprint-based attendance system used to capture the attendance data and matched with a secure database which contains the fingerprint data. After verification, this data is uploaded into a cloud database. In the next phase, the backend of the application calculates the resources necessary for the day, from the gathered attendance data and government guidelines. This information is used by the authorities to procure resources from a warehouse of resources. The authors suggest the use of network-based access controls to prevent tampering with the cloud logic that computes the quantity of resources to be released. Finally, when an authority, supposedly a chef/ cook, acquires raw material from the warehouse, he follows the proposed model's protocol of initiating a transaction to a blockchain, which contains data of the quantity of food acquired. These transactions can be reviewed to check for discrepancies and take requisite actions.

In [6], Sandeep Kumar Singh Et. Al. have presented a model for all Indian PDSs using the concept of consortium blockchains. Here, the authors have used consortium blockchain technology in Hyperledger Fabric, which is an open-source permissioned distributed ledger technology platform under Linux Foundation. The write access to this consortium block chain is proposed to be given only to entities like the government of India, the FCI, approved transporters of resources/ food materials, state/ central government warehouses and verified ration shop dealers. These are the peer nodes in the consortium blockchain and have the authority to initiate transactions. There are four processes which require transactional history to be stored, namely: movement, storage, quality control and distribution. These processes can have multiple smart contracts as their components. For instance, the movement chain will have the following smart contracts: transportation contract, loading contract and allocation contract. The accessibility of these processes' chains is given only to the required authorities. In case of the movement process, the FCI headquarters, warehouse managers, transporters, and other such personnel would be allowed access. In layman's terms, each process's data is write-accessible only to specific peers. The authors also propose the use of data analytics for various decision-making processes. The proposed architecture can prevent the use of ghost cards and botched execution of tasks due to human interference.

A blockchain framework has been proposed by Pankaj Joshi Et. Al. in their research work aiming to reduce the possible frauds/embezzlements in fund transfer protocol of different government schemes that are usually launched for the upliftment of society. Any generic government scheme that funds common people will have the cash-flow in three steps: initialization, verification of the scheme, and fund transfer [1]. In the current architecture, funds are transferred to the recipients/beneficiaries via a layered architecture. Each layer being a government organization, the funds are ultimately allotted to the recipients at the last layer. The blockchain framework proposed, claims to solve the issues of fund stagnation (possible at middle layers of the traditional architecture), the obscurity of process, and efficient utilization of funds. The proposed framework has all the governing authorities of the scheme as a part of the blockchain network. The smart contract responsible for processing the scheme is supposed to be deployed by the organization/authority responsible for initiating the scheme. All other parties involved in transferring funds can only update the deployed contract. The proposed framework tries to leverage the potential of blockchain to restraint the frauds in government schemes. Further, the framework provides auditing capabilities to the beneficiaries making the system transparent.

A Smart Contract based blockchain framework has been proposed, by Devi Et. Al., to automate public distribution system. The framework claims to record all the transactions between central government, state government, FCI (Food Corporation of India), ration shops, and the beneficiaries [2]. The proposed system has central & state governments, FCI, fair price shops, and the beneficiaries as participating nodes in the blockchain. All the transactions can be tracked by all the participating nodes at any point in time. The proposed system has been developed on browser based JVM.

Traceability is one of the critical elements of any supply chain management. An architecture based on Blockchain Technology has been modeled by Casino Et. Al. for effective traceability in Food Supply Chain (FSC). [3] The model aims to enable information exchange and transparency among all the members of FSC. The FSC members are classified as Upstream (Food producers, Manufacturers) and Downstream (Wholesalers, Retailers, etc.) members. The model is built to store the upstream data that is generated from IoT and manual entries, in centralized servers and decentralized storage using IPFS. Smart contracts are used to enable permissioned access to the information to all the members of FSC. A link between upstream and downstream data is established with the help of global-product-id and IPFS hash that has product-related data. The model here uses blockchain technology and smart contracts to overcome problems in the traceability of traditional FSC.

Table 2.1 Overview of literature survey and the gaps identified

S. No	Title of the Paper and Year	Aim of the Paper	Algorithms/ techniques Used	Limitations (Gap identified)	Scope for Future Work
1	Cloud Based Data Integrity model for Minimal Human Interv-ention using Smart Contracts [2020]	A model for eliminating human intervention in the mid-day meal scheme.	Smart contracts on Ethereum blockchain, Fingerprint matching	Storage of attendance in database might not be the most secure way.	Integration with a food material dispensing system can eliminate human intervention.
2	A conceptual model for Indian public distribution system using consortium blockchain with on-chain and off-chain trusted data.	Presents a model to eliminate inefficiencies in the Indian PDS using smart contracts and a consortium-based system.	Smart contracts, Consortium Blockchain, Big data analytics, on-chain/ off-chain data.	The framework is highly specific for the ration card scheme.	The framework can be extended to other PDSs or similar schemes like water distribution, direct transfer of benefits to under-privileged people, etc
3	A Blockchain Based Framework for Fraud Detection	Proposed a blockchain framework to reduce the	Smart Contracts	The framework does not include inventory/resource management.	The Framework can be extended to audit and

	(2019)	possible frauds/embezzlements in fund transfer protocol of different government schemes.			manage Inventories.
4	Public Distribution System Based on Blockchain using Solidity. (2021)	Proposed an automated food distribution system that records and logs all the transactions between central & state governments, district level offices, ration shops and beneficiaries.	Smart Contracts (On Ethereum Blockchain)	The proposed system is only deployed on browser based JVM.	An interactive web application can be developed to mimic the functionalities.
5	Modelling food-supply-chain traceability based on blockchain technology	A blockchain model with smart contracts is proposed to tackle the traceability problem in traditional food supply chains.	Smart contracts and IPFS (Inter-Planetary File System)	Only addresses simple FSC network.	Combining the current model with other supply chain optimization approaches and exploring other multi-tier FSC networks.

3 OVERVIEW OF THE PROPOSED SYSTEM

3.1 Introduction

On observation of the various processes in the current architecture, one can identify symmetric processes happening across its various layers. For instance, the authorization by respective authorities on acknowledging the transfer and receipt of resources from one level to another and, the need to store transactions on each level in case they need to be verified in the future.

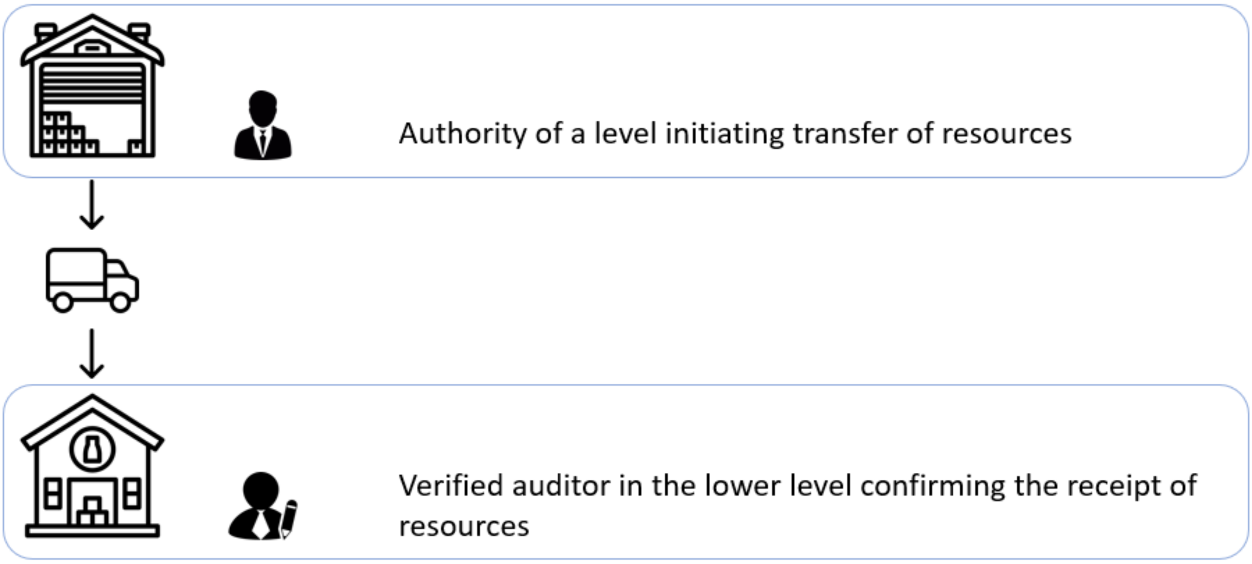

Authority of a level initiating transfer of resources
Verified auditor in the lower level confirming the receipt of resources

Fig 3.1.1 Depiction of similar structure in the transfer of resources from one layer to another

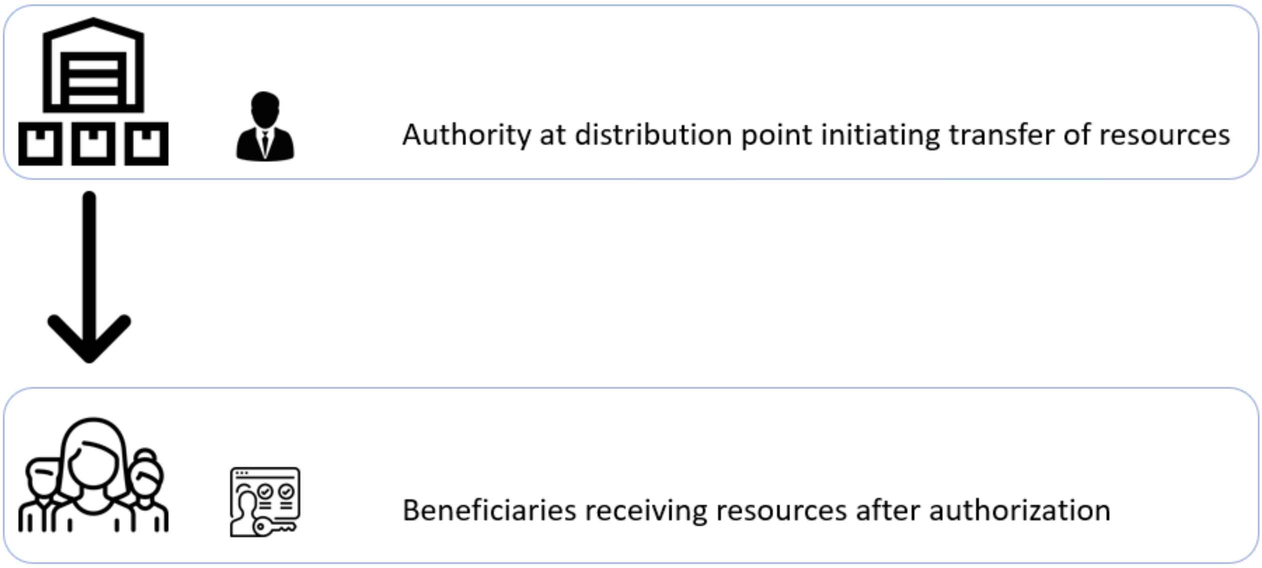

Authority at distribution point initiating transfer of resources
Beneficiaries receiving resources after authorization

Fig 3.1.2 Depiction of the presence of a similar structure in the transactions between the last two layers

Therefore, a common system can be developed for these processes on different levels by collecting transaction data along the following attributes:

Table 3.1.1 Attributes required for transaction management.

Type	Attribute	Sample Data 1	Sample Data 2
ID	Transaction ID	0xa9d8sda8709…. 113a	0xa9d31da8709…. 1d3z
ID	Resource_Transfer_Initiator_ID [Authority who signed off over the transfer of resources]	0y3923aas9a8s…. 231z	0y12iu3hi112b1…. r41e
ID	Resource_Reciever_ID [Authority/ Beneficiary who signed off over the receipt of resources]	0yh1h23i1i23i5…. y8a3	6578927518470934
Entity Name	From_Entity	Hyderabad Warehouse	Judges Colony Dist. Point
Entity Name	To_Entity	Warangal Warehouse	Rohit Sharma
Entity Level	From_Entity_Level	State	Distribution Point
Time Stamp	Sent_Time	24/06/2019	27/02/2020
Time Stamp	Received_Time	27/06/2019	27/02/2020
Map	Resources	{Rice: 560 kg; Flour: 350 Kg}	{Rice: 0.5 Kg; Sugar: 0.4 Kg}

3.2 Framework of the proposed model

These modules are essential for the functioning of a system which minimizes human intervention from the current architecture:

i. Module for authentication:

Includes smart contract for verification of identity, fingerprint, picture, latitude/longitude, based on the scheme.

ii. Module for eligibility verification:

  Includes smart contract for verifying the eligibility conditions, Interfaces for querying the blockchain for reading the records.

iii. Module for managing the resources:

  Includes the smart contract for assessment verification, Includes the smart contract for inventory threshold verification, Interfaces for querying the blockchain for reading the records.

These modules handle various processes and facilitate communication between the client, the server and the blockchain.

a) Assignment of authorities:

  Each level has several authorities, these authorities are assigned to the entities present on a level through a central account where an Ethereum address is allocated to them. Whenever an authority initiates a transaction [examples of transactions: transfer of resources, addition of beneficiaries, resource allocation to beneficiaries, approving or rejecting addition of new beneficiaries], relevant functions are fired in smart contracts and the authority is noted as the signer/ initiator.

b) Distributed Beneficiary Registration:

  Registration of a beneficiary to the scheme essentially has three steps:

  i. Registration initiation

  ii. Beneficiary Validation:

    Approval/Rejection by super majority of the authorities at a distribution point.

  iii. Registering beneficiary to the scheme if approved by authorities (as mentioned in step ii).

  - **Registration Initiation:**

    A beneficiary can be registered at any of the distribution points available. An authority of the beneficiary starts the registration process by verifying the required documents and adding beneficiary data using the portal. Table 3.2.1 shows the essential attributes for smooth and secure functioning of the registration process.

Table 3.2.1 Attributes collected for beneficiary registration.

Attribute	**Type**	**Description**
Id	Uint	Primary key, unique identification for beneficiary
DstnId	Uint	Foreign key, distribution point identifier
Credentials	String	Scheme specific beneficiary authentication data
Approved	Address []	List of authority ids by whom the beneficiary was approved during the registration process
Rejected	Address []	List of authority ids by whom the beneficiary was rejected during the registration process
FinalStatus	Uint	Indicates weather the beneficiary has been registered to the scheme or not.

- **Beneficiary Validation by other authorities:**

  Once a beneficiary registration gets initiated by an authority of a distribution point, all the other authorities will get to either approve or reject the beneficiary using their respective portals. This process is further debriefed in the following sections.

  After the beneficiary data is as a part of registration initiation, it is sent to a server where the credential data (beneficiary image, biometric data etc.) is stored on IPFS. The IPFS returns a hash which allows us to access the stored data. The hash is again encrypted using a secret on the server before sending the response to the client. This encrypted-hash along with other relevant data and the secret key is encrypted and returned to the client as a token for secure and easier management of data. Now, a smart contract function is fired. In the contract, firstly, the data along with the secret key (The same key used during token-generation) is counter checked with the encrypted token and the authority is noted as the initiator then, the person wanting to be registered as a beneficiary is directly approved by the authority. However, this person requires the majority of the distribution point authorities (>50%) to approve him, to be successfully registered. Corresponding events are emitted by the smart contract to facilitate this operation. This process is illustrated in Fig 3.2.1.

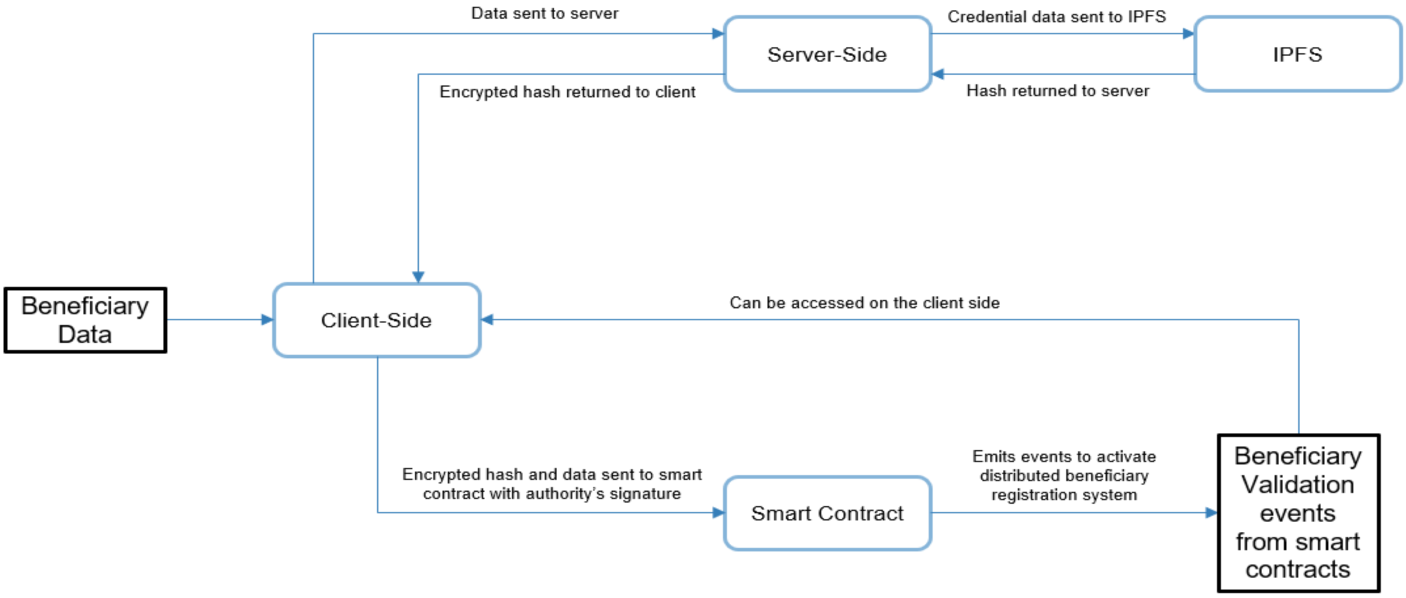

Data sent to server
Server-Side
Credential data sent to IPFS
IPFS
Encrypted hash returned to client
Hash returned to server
Beneficiary Data
Client-Side
Can be accessed on the client side
Encrypted hash and data sent to smart contract with authority's signature
Smart Contract
Emits events to activate distributed beneficiary registration system
Beneficiary Validation events from smart contracts

Fig 3.2.1 Initiation of Beneficiary Registration in the prototype.

These events are detected by the client application. When the authority of a particular distribution point logs into the beneficiary validation portal, they can view all the beneficiaries who have applied to register under their distribution point. If the decision-making process for an upcoming beneficiary is still under progress, the authority can make his decision (accept/ reject) and add comments. This fires a smart contract function where after the authority is verified, his decision is updated.

Table 3.2.2 Final Status of beneficiary based on percentage of authorities' decisions.

Status	**Percentage**
Approved	>50% in approval
Rejected	>=50% in rejection
Pending	otherwise

Then, a query is passed to check if the beneficiary has been approved or rejected by a majority (Table 3.2.2 explains the possible outcomes). If not, then an event is logged about the updated beneficiary status which also indicates the client application to continue with validation process for other authorities of the distribution point. In case a majority has been reached, then the beneficiary is added to the Ethereum storage if the majority had approved the beneficiary and an event is logged indicating the client application to halt the validation process for other authorities. Fig 3.2.2 depicts the flow of the validation process.

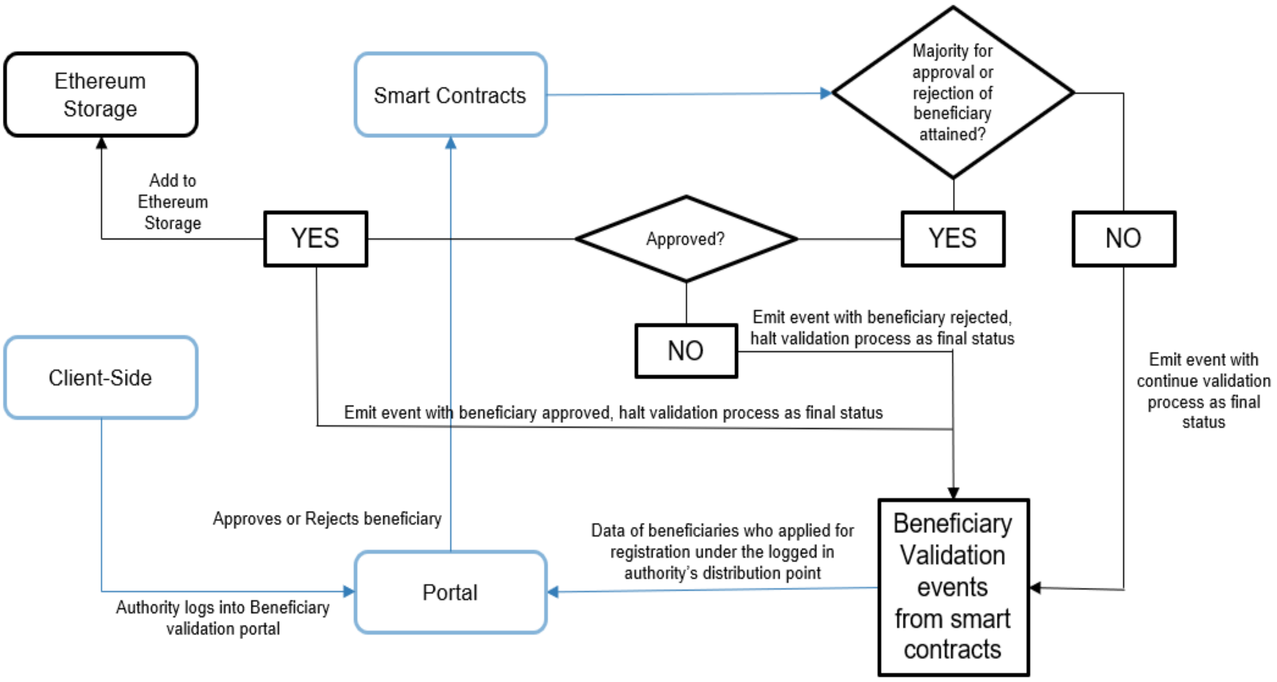

Ethereum Storage
Smart Contracts
Majority for approval or rejection of beneficiary attained?
Add to Ethereum Storage
YES
Approved?
YES
NO
NO
Emit event with beneficiary rejected, halt validation process as final status
Client-Side
Emit event with continue validation process as final status
Emit event with beneficiary approved, halt validation process as final status
Approves or Rejects beneficiary
Data of beneficiaries who applied for registration under the logged in authority's distribution point
Beneficiary Validation events from smart contracts
Portal
Authority logs into Beneficiary validation portal

Fig 3.2.2 Distributed validation process for beneficiary registration process in the prototype.

Table 3.2.3 Sample states during beneficiary validation process.

Authority 1	**Authority 2**	**Authority 3**	**Approval Percentage**	**Rejection Percentage**	**Final Status**
Approved	Pending	Pending	33%	0%	Pending
Approved	Approved	Pending	67%	0%	Approved
Approved	Rejected	Pending	33%	33%	Pending
Approved	Rejected	Approved	67%	33%	Approved
Approved	Rejected	Rejected	33%	67%	Rejected

Table 3.2.3 shows an example of a beneficiary and the possible states one can exist in the distributed registration process at a distribution point with three authorities where

‘Authority 1’ is the initiator. **This data can further be used to assign trust vales to authorities which can increase the system’s reliability.**

c) Transfer of resources to beneficiary and beneficiary authentication:

Multi-factor authentication based on facial data, biometric data and the geo-location can be used to reduce the resource pilferage at distribution point level. In the prototype, facial-image-comparison has been implemented to authenticate beneficiary before resource transfer at distribution level. This can further be extended by integrating with Aadhar API to authenticate beneficiaries with the help of biometric data as well. Moreover, geo-location tagged with beneficiary images captured at distribution points during authentication can be used to validate the source of beneficiary image. Hence, with the help of multi-factor authentication resource pilferage at distribution levels can be reduced.

On receiving beneficiary data and the quantity of resources being allotted to them, the data is sent to the server. The server fetches data from the Ethereum storage, which is in turn used to get previously stored credential data from IPFS. This data along with the input data is compared. In the prototype, an image recognition web API was used to authenticate the beneficiary. After comparison of data, a response is sent to the client which is composed of the data and a token of the data which is encrypted with a secret key. Now, on the client application a smart contract function is called with the authority’s signature. Here, the authority is verified, and the encrypted token is checked against the data with the secret key. If the transaction is valid a record of transaction is created, a response is sent to the client application and an event is logged for future references. Fig 3.2.3 illustrates this process clearly and Table 3.2.4 shows the constituents of the encrypted tokens.

Table 3.2.3 Encrypted token components

Attribute	**Type**	**Description**
fromEntity	String	Name of initiator entity
fromEntityId	Uint	Id of initiator entity
toEntity	String	Receiving entity [Beneficiaries in this level]
toEntityId	Uint	Receiving entity id [aadhar number in the prototype]
entityLevel	String	Always ‘Distribution Point’ for these transactions
timeStamp	Uint	Timestamp when the resources are transferred
resourceQuantities	Uint []	Quantity of various resources being transferred to the beneficiary
secret	String	The secret keyword being used for encryption
similar	Bool	True if the stored data matches with input data else false

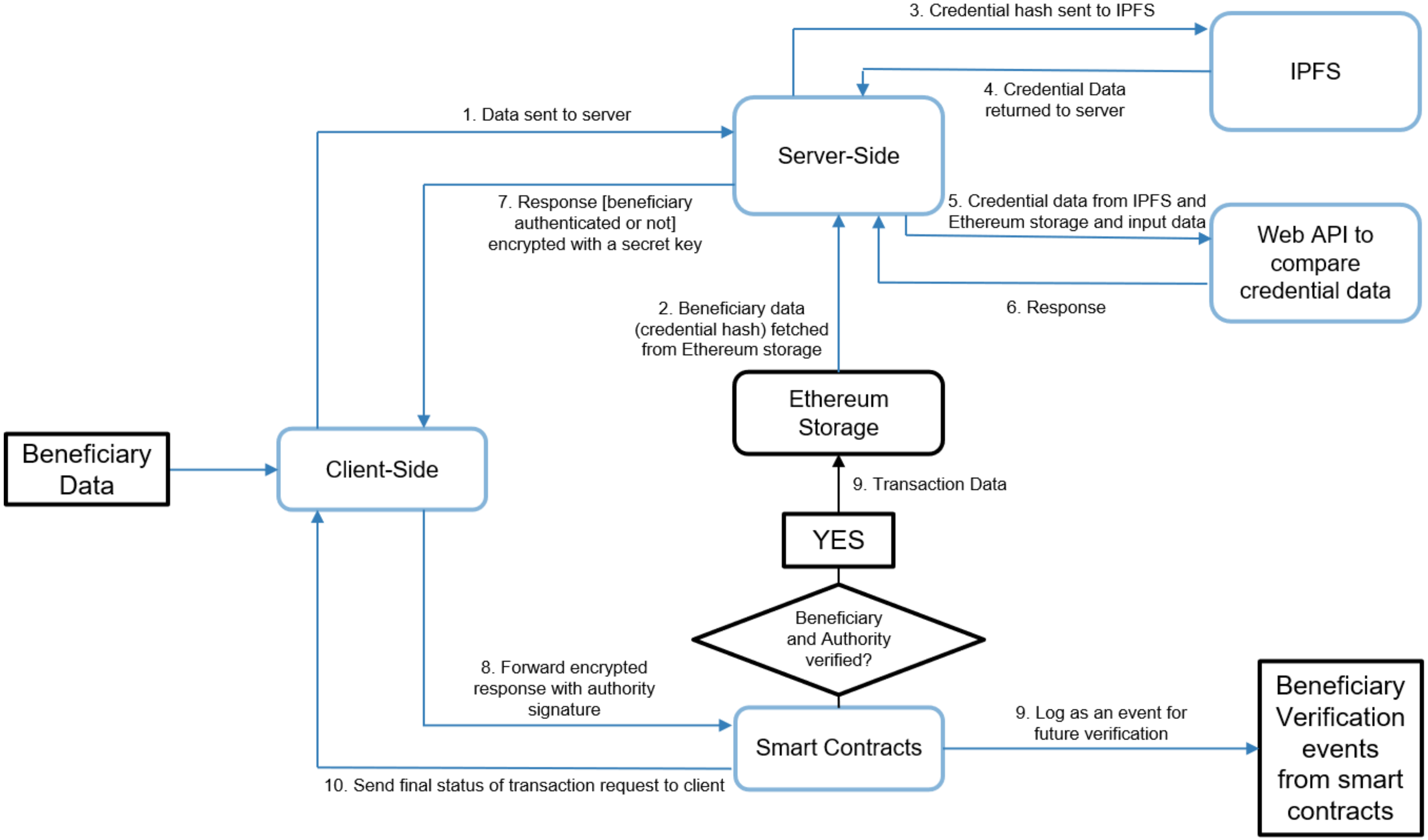

3. Credential hash sent to IPFS
IPFS
4. Credential Data returned to server
1. Data sent to server
Server-Side
7. Response [beneficiary authenticated or not] encrypted with a secret key
5. Credential data from IPFS and Ethereum storage and input data
Web API to compare credential data
6. Response
2. Beneficiary data (credential hash) fetched from Ethereum storage
Ethereum Storage
Beneficiary Data
Client-Side
9. Transaction Data
YES
Beneficiary and Authority verified?
8. Forward encrypted response with authority signature
Smart Contracts
9. Log as an event for future verification
Beneficiary Verification events from smart contracts
10. Send final status of transaction request to client

Fig 3.2.3 Beneficiary authentication for transfer of resources in the prototype.

3.3 *Proposed system model:*

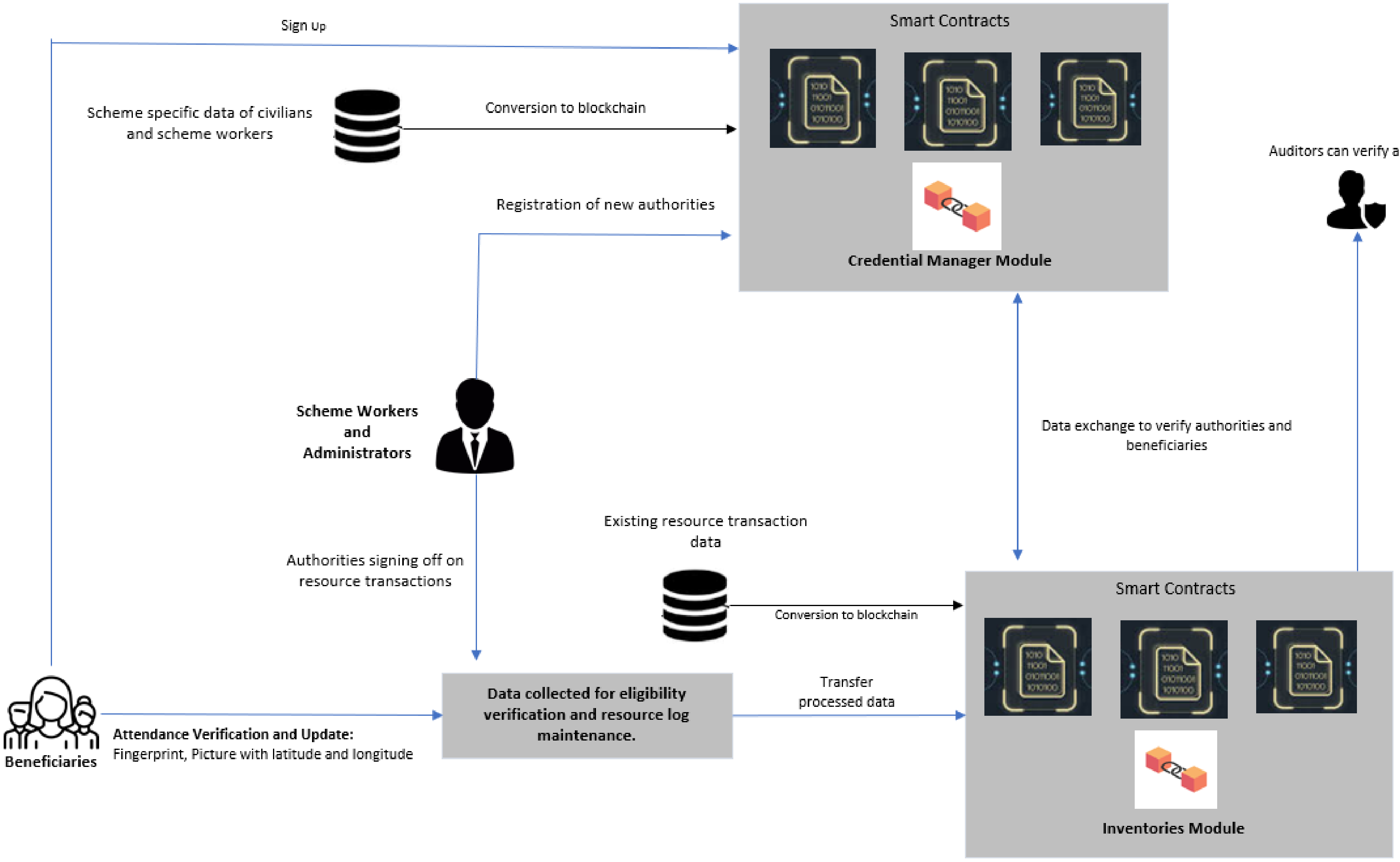

Sign Up
Smart Contracts
Scheme specific data of civilians and scheme workers
Conversion to blockchain
Registration of new authorities
Credential Manager Module
Auditors can verify anytime
Scheme Workers and Administrators
Data exchange to verify authorities and beneficiaries
Existing resource transaction data
Authorities signing off on resource transactions
Conversion to blockchain
Smart Contracts
Transfer processed data
Data collected for eligibility verification and resource log maintenance.
Attendance Verification and Update:
Fingerprint, Picture with latitude and longitude
Beneficiaries
Inventories Module

Fig. 3.3.1 Proposed System Model

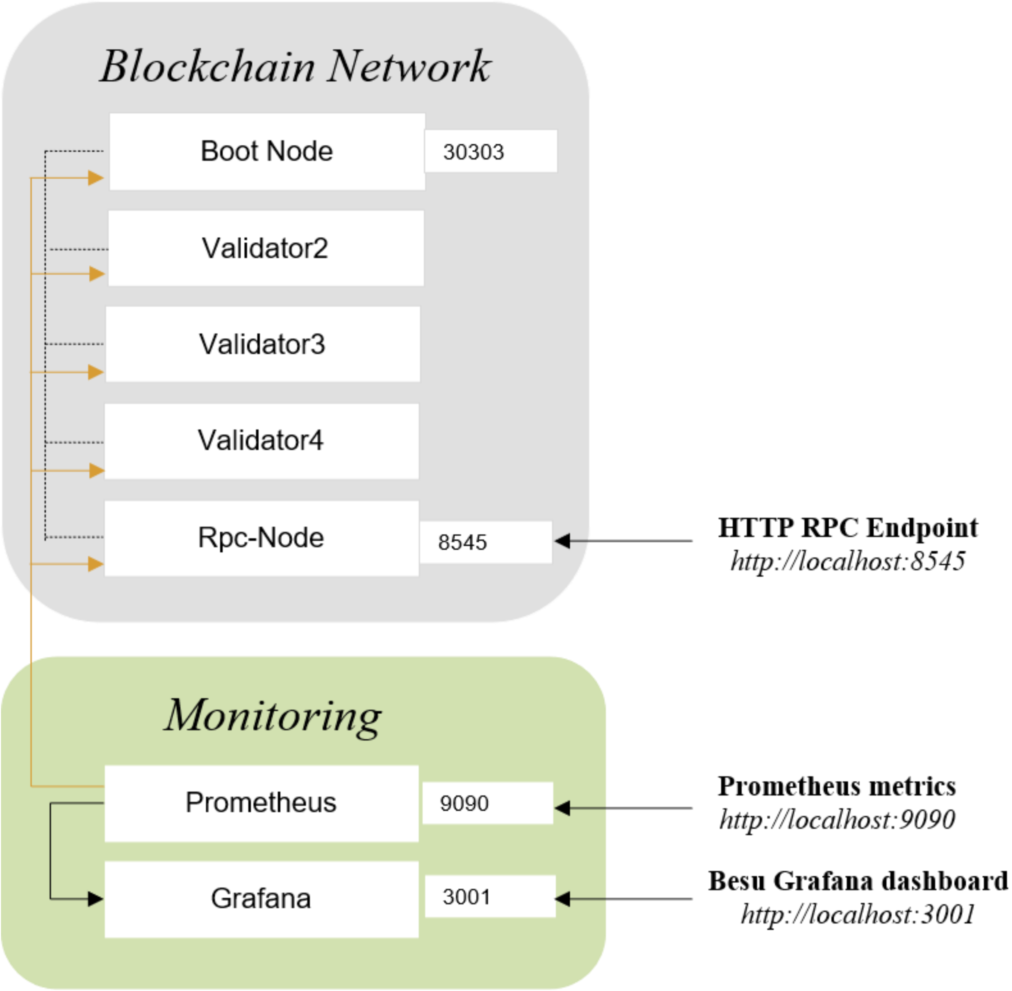


Fig. 3.3.2 Configured blockchain network for development

- Configured consensus algorithm: IBFT 2.0

Table 3.3.1 Comparison of consensus algorithms

Name	Compute required	Trusted environment	*Finality?
Ethash (PoW)	HIGH	NO	NO
Clique (PoA)	LOW	YES	NO
IBFT2 (PoA)	LOW	YES	YES

4 PROPOSED SYSTEM ANALYSIS AND DESIGN

4.1 Introduction

The proposed system consists of two backend modules. Firstly, the credential manager module has contracts that can add new beneficiaries and government authorities to Ethereum's local storage. It also helps authorize government officials when they initiate any transaction of resources or the beneficiaries at the distribution points when they receive their allotted resources. Secondly, the inventories module maintains a record of all authorized transactions. During the authorization process, the inventory module communicates with the credential manager module to authenticate beneficiaries or authorities. Apart from these functions, both the modules emit events that facilitate the maintenance of records. These records track unauthorized transactions.

4.2 Requirement analysis

4.2.1 Functional requirements

4.2.1.1 Product perspective

The current architecture's flaws allow the occurrence of malicious transactions that cannot be tracked. This shortcoming has enabled various corrupt individuals and entities to unscrupulously steal resources. The proposed architecture successfully prevents unauthorized transactions and maintains a record of any attempts at the same. It also makes the data available for everyone thus removing the problem of obscurity.

4.2.1.2 Product features

The features which add value to the proposed system are:

- *Architecture*:

  which coalesces multiple layers of operations into an immaculate framework.
- *Blockchain technology*:

  which impedes unsanctioned transaction of resources.
- *Smart contracts*:

  which ensures the imposition of terms when any transactions are made.

4.2.1.3 User characteristics

There are two types of users that will use the functionalities of the product:

- *Beneficiaries:*

These users can interact with the system in two ways. Firstly, by registering to the scheme. Secondly, for authorization when they receive resources.

- *Scheme authorities:*

  There can be many types of scheme authorities. For instances, authorities at central government level, state government level, district government level and authorities at distribution points. They will be using the functionality of initiating transfer of resources and acknowledging the receipt of resources.

4.2.1.4 Assumptions and Dependencies

The following assumptions were made while developing the prototype:

- Since access to the Aadhar database is granted only to Authentication User Agencies (AUA) or to entities with business relationships with AUAs (sub-AUA), randomly generated data was used instead.
- The data of authorities and scheme workers was similarly randomly generated due to the lack of accessibility.

4.2.1.5 Domain requirements

Domain specific data:

- Data of beneficiaries registered to the scheme and their biometric data.
- Data of scheme workers and officials who are employed for the scheme's operation.

4.2.1.6 User requirements

- **Registration portal:**

  A portal for registration of users and scheme workers.
- **Beneficiary authentication:**

  A module for biometric authentication of beneficiaries.
- **Authority authorization:**

  A module authorizing scheme workers.
- **Resource dispatch facilitation:**

  A module for authorized scheme workers to enable transfer of resources.
- **Resource receipt acknowledgement:**

A module for authorized scheme workers to sign off on the receipt of resources using QR Code technology.

4.2.2 Non-functional requirements

4.2.2.1 Product requirements

4.2.2.1.1 Efficiency

In such a system, efficient use of space is essential. The designed architecture ensures the non-existence of redundancy in data. All algorithms used are either in linear time or lower, thus ensuring fast responses from the backend.

4.2.2.1.2 Reliability

The proposed architecture is exceptionally reliable. All transactions of resources are available to all the users registered in the credential management module, whichever level the user might be in.

4.2.2.1.3 Usability

The designed interface is minimalistic in nature and quite easy to understand. Beneficiaries can just as easily access the system as any other scheme authority.

4.2.2.2 Organizational requirements

4.2.2.2.1 Implementation requirements

The implementation of the prototype requires quite a bit of RAM since running 4 mining nodes on private network being hosted locally is a bit expensive but not impossible. Running the nodes along with other essential applications will considerably slow down the system. The prototype developed is being run on system with 8 GB RAM. Running the network on cloud, for instance, Azure, reduces the load.

4.2.2.2.2 Engineering standard requirements

The developed prototype uses REST Architecture. All calls made to the backend are in the form of REST methods [get, post, put, delete and patch] and the resultant replies from the backend contain the correct status codes.

4.2.2.3 Operational requirements

4.2.2.3.1 Economic

The application is very viable economically. It can potentially save crores of rupees and tons of resources which could have been lost to corrupt officials.

4.2.2.3.2 Social

This application helps ensures social justice by facilitating the secure flow of resources to the intended people rather than unprincipled individuals who have intentions to steal the resources otherwise.

4.2.2.3.3 Inspectability

Since all transactions of resources are logged and available for inspection by anyone participating in the scheme, the architecture is open for inspections at all times.

4.2.3 System requirements

4.2.3.1 Software requirements

- Hyperledger besu [13]:

  A private block chain network with miner nodes which are hosted locally using Hyperledger besu.
- React JS [12]:

  Open-source front end development JavaScript library.
- Web3 JS [11]:

  A collection of libraries that allow interaction with Ethereum code using HTTP, IPC or WebSocket
- Ethereum-Solidity [10]:

  Allows implementation of smart contracts, which govern the behavior of accounts within the Ethereum state.

5 RESULTS AND DISCUSSIONS

In this paper, a generic architecture is discussed that leverages the potential benefits of blockchain to eliminate possible mis happenings in a set of government aided schemes. Major focus has been laid to digitize the process and reduces human intervention from the process to the maximum extent possible.

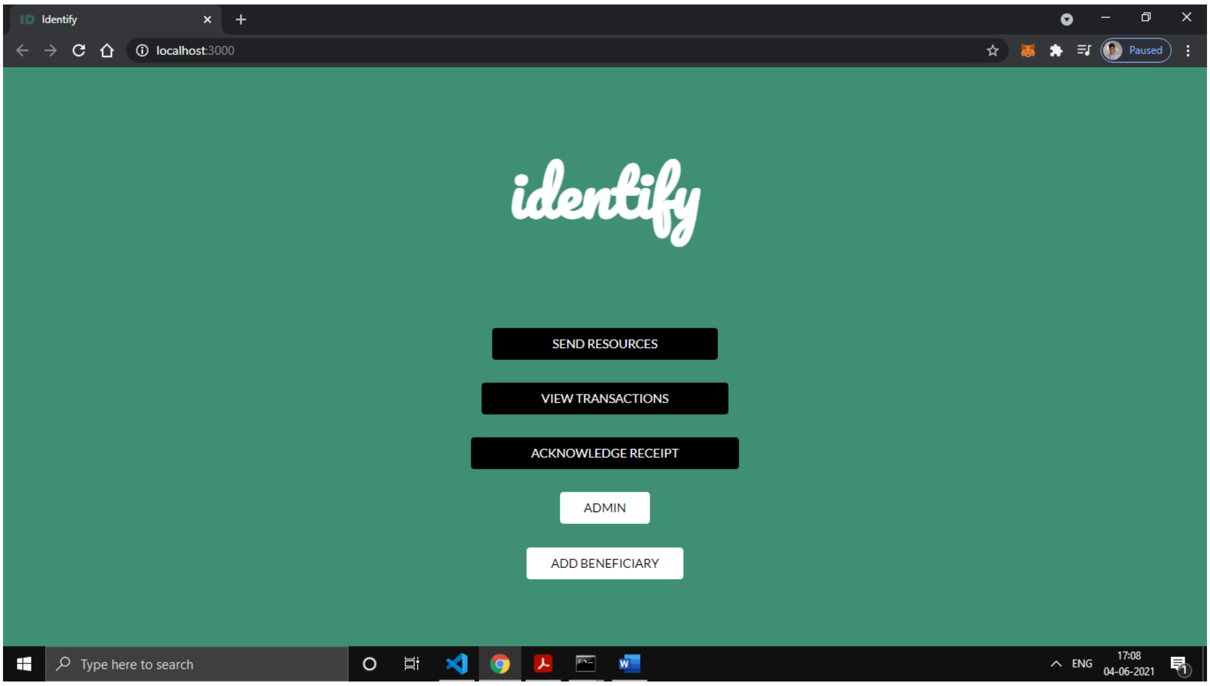


Fig. 5.1 Landing Page of the portal

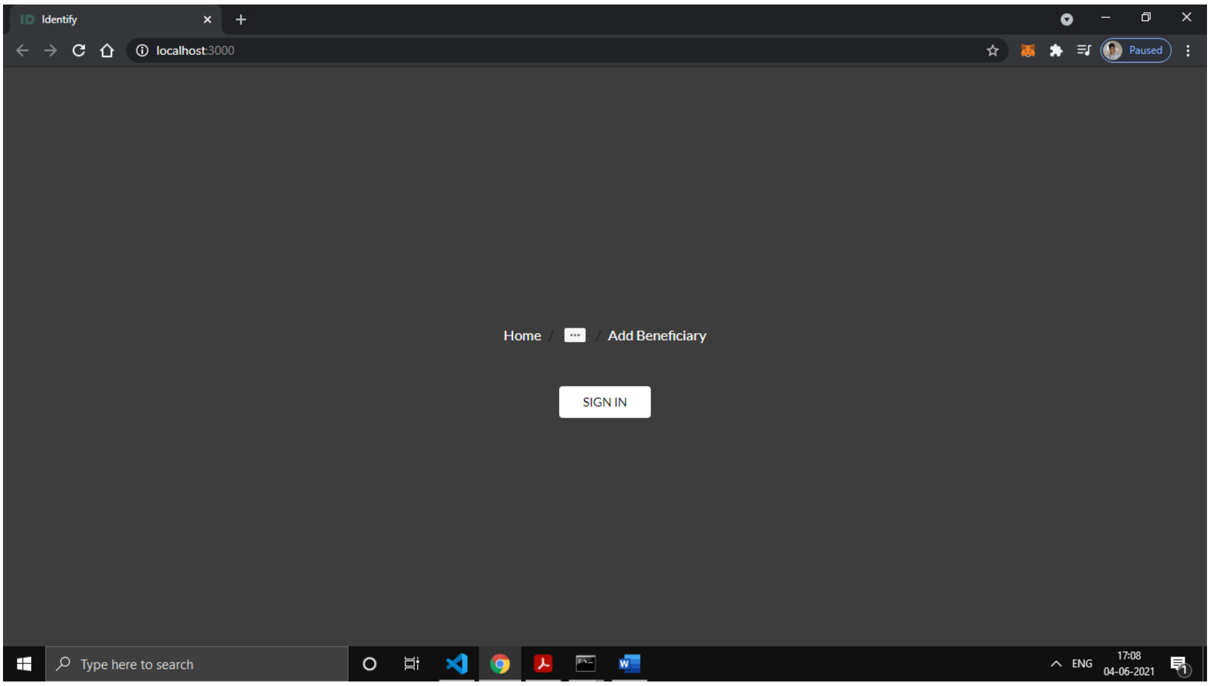
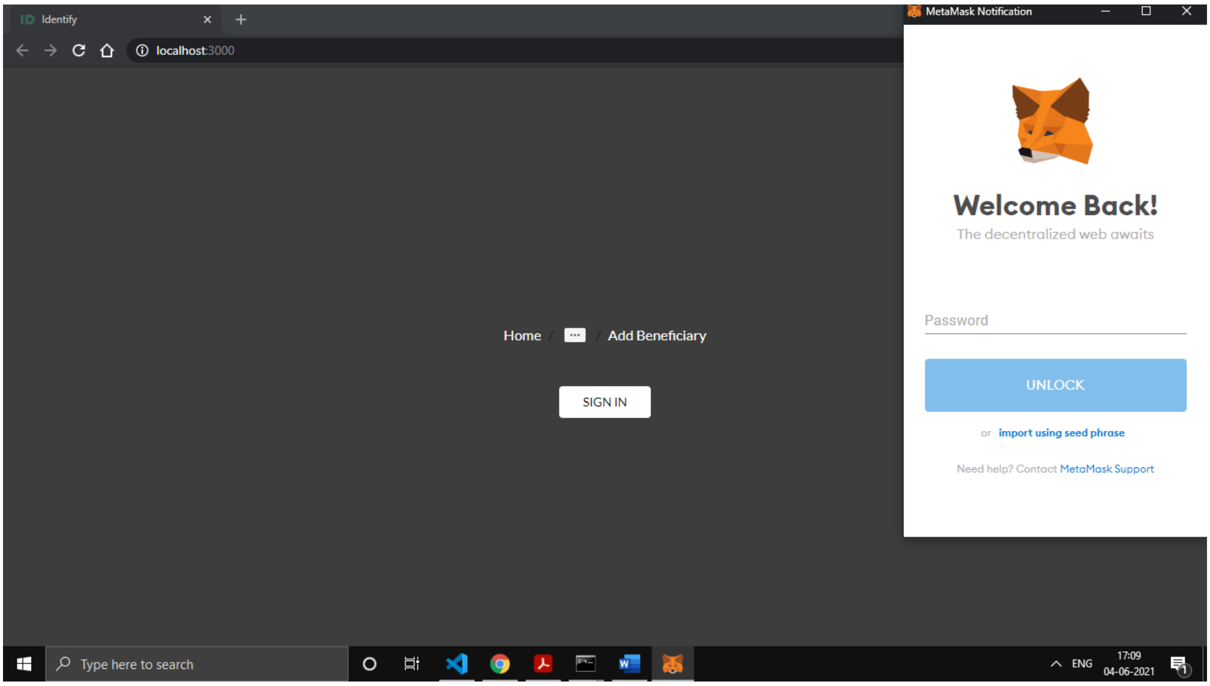


Fig. 5.2 Admin login screen

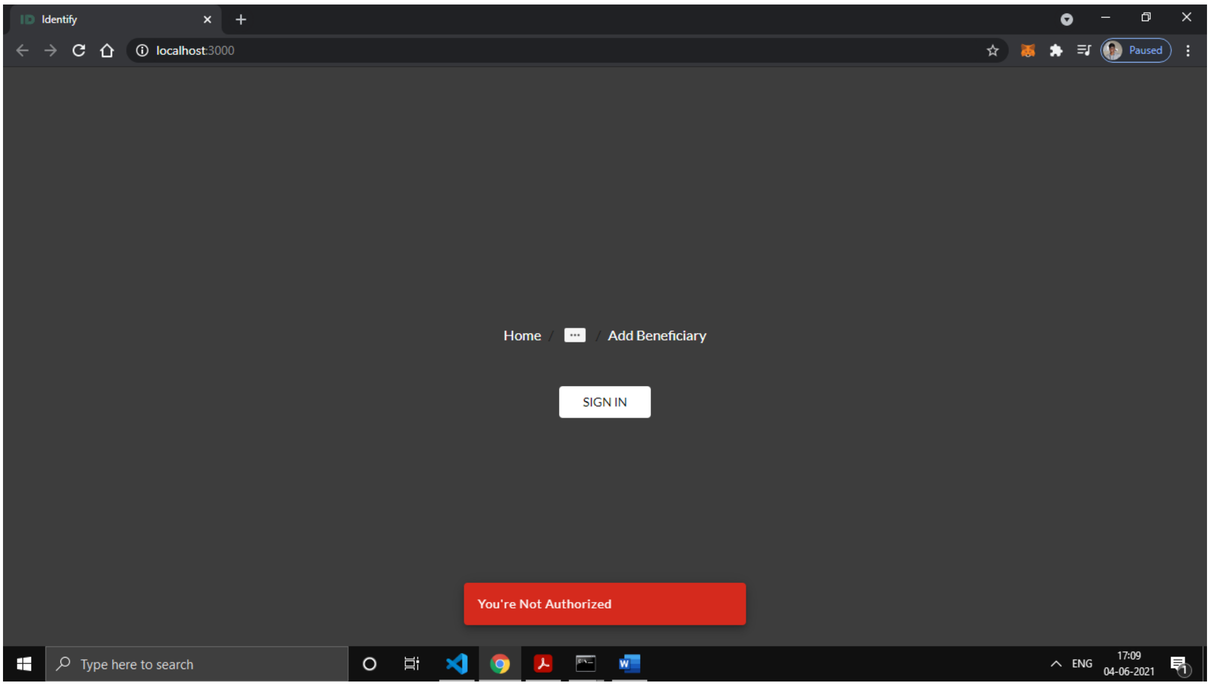


Welcome Back!

The decentralized web awaits

Need help? Contact MetaMask Support

Fig. 5.3 Admin login prompt (Uses metamask along with smart contracts (Credential Manager smart contract) to authorize Authorities & here admin)



You're Not Authorized

Fig 5.4 Result: Unauthorized login

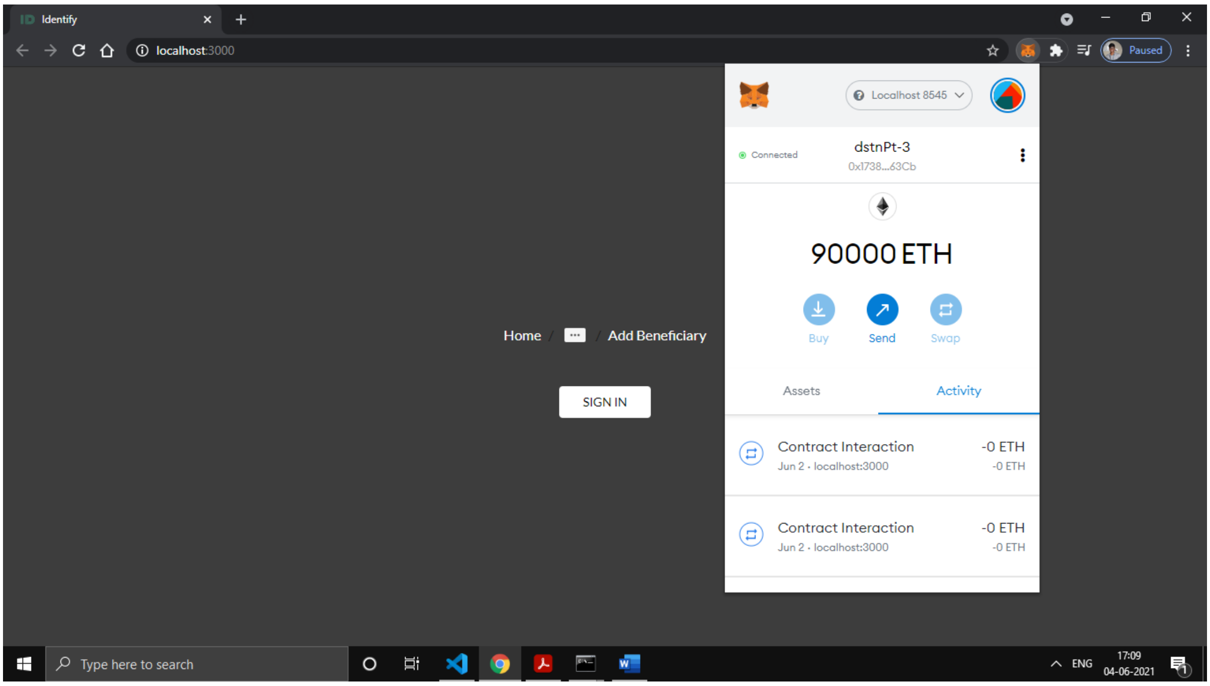


Fig. 5.5 Selected account for login that resulted in an unauthorized login:0x173897136202ef44a2fe1FCFabE701BBcf2e63Cb

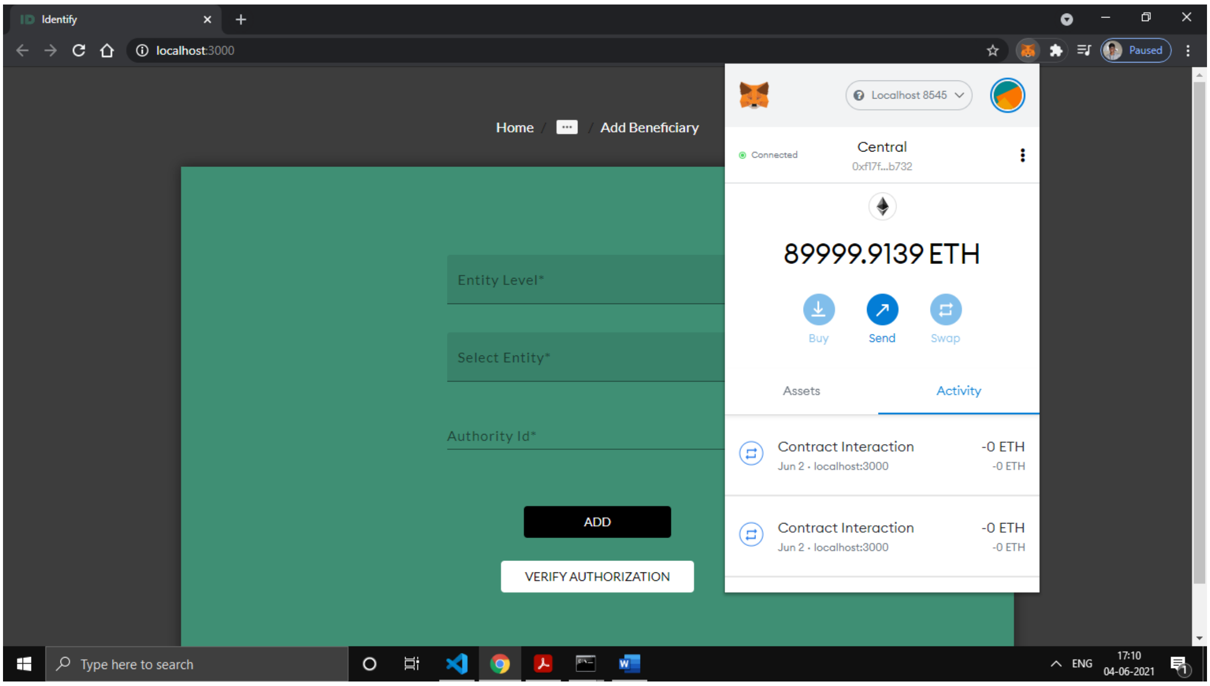
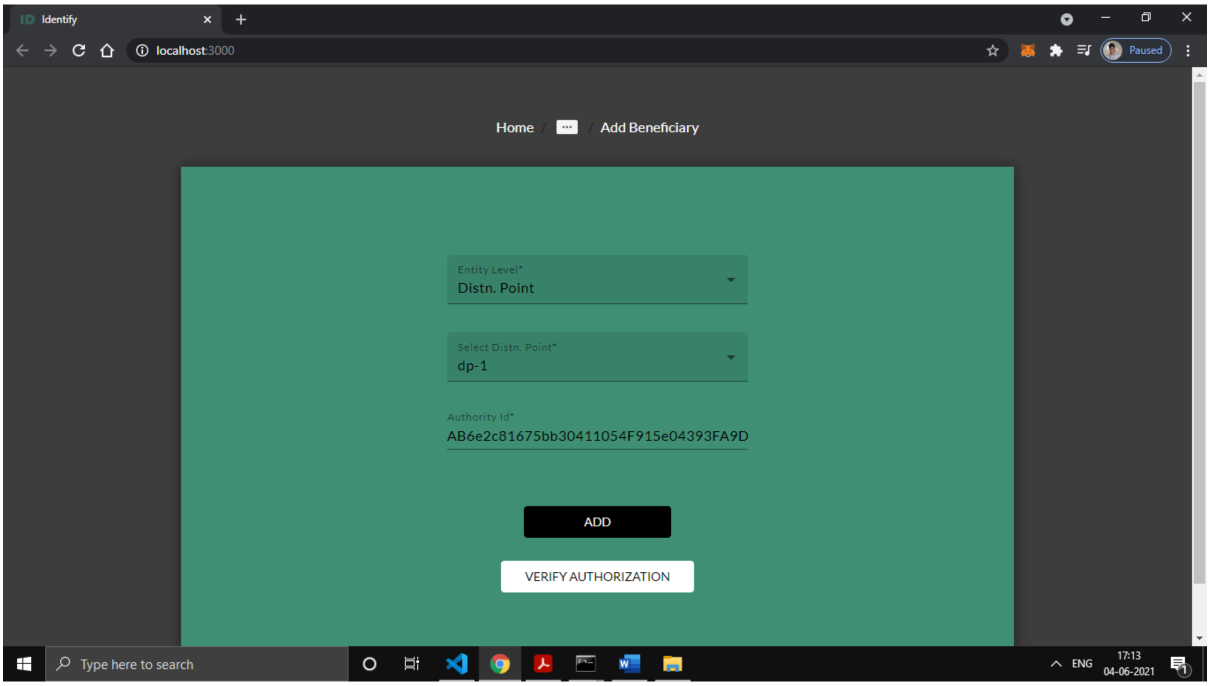


Fig. 5.6 Admin Screen/portal on successful login. Account selected:0xf17f52151EbEF6C7334FAD080c5704D77216b732

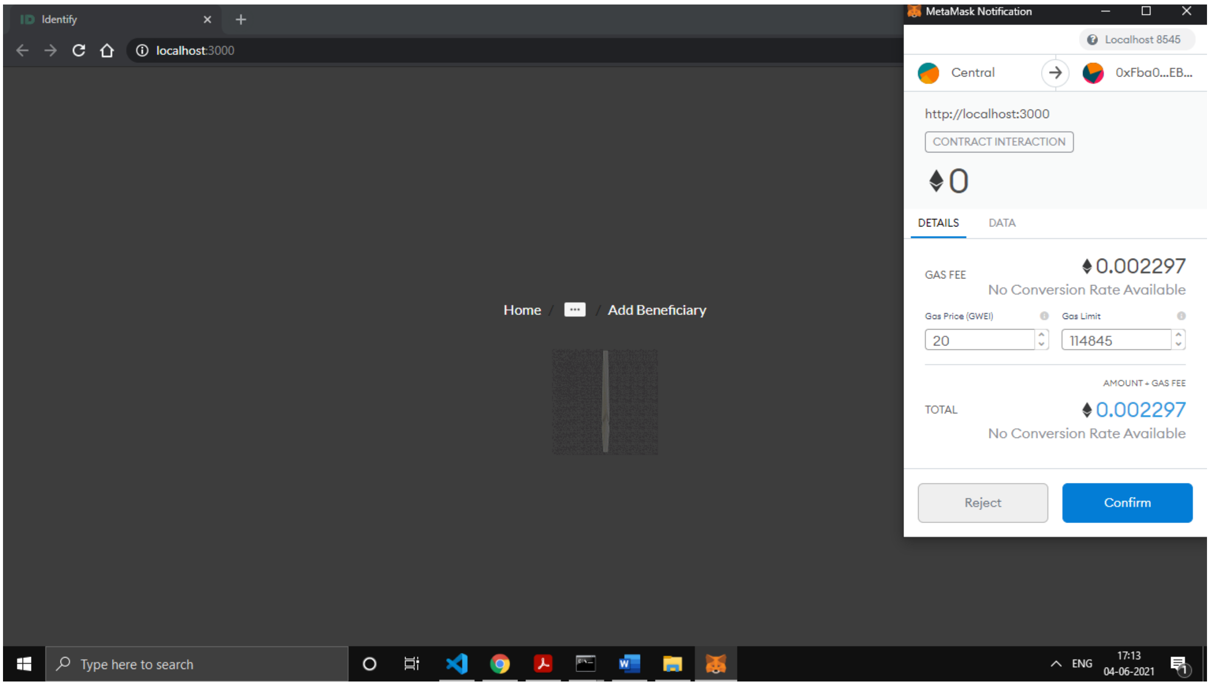


Entity Level*
Distn. Point

Select Distn. Point*
dp-1

Authority Id*
AB6e2c81675bb30411054F915e04393FA9D

Fig. 5.7 Assign authority to distribution point 1. Authority id/address: 0xAB6e2c81675bb30411054F915e04393FA9D20441



Central → 0xFba0...EB...

http://localhost:3000

CONTRACT INTERACTION

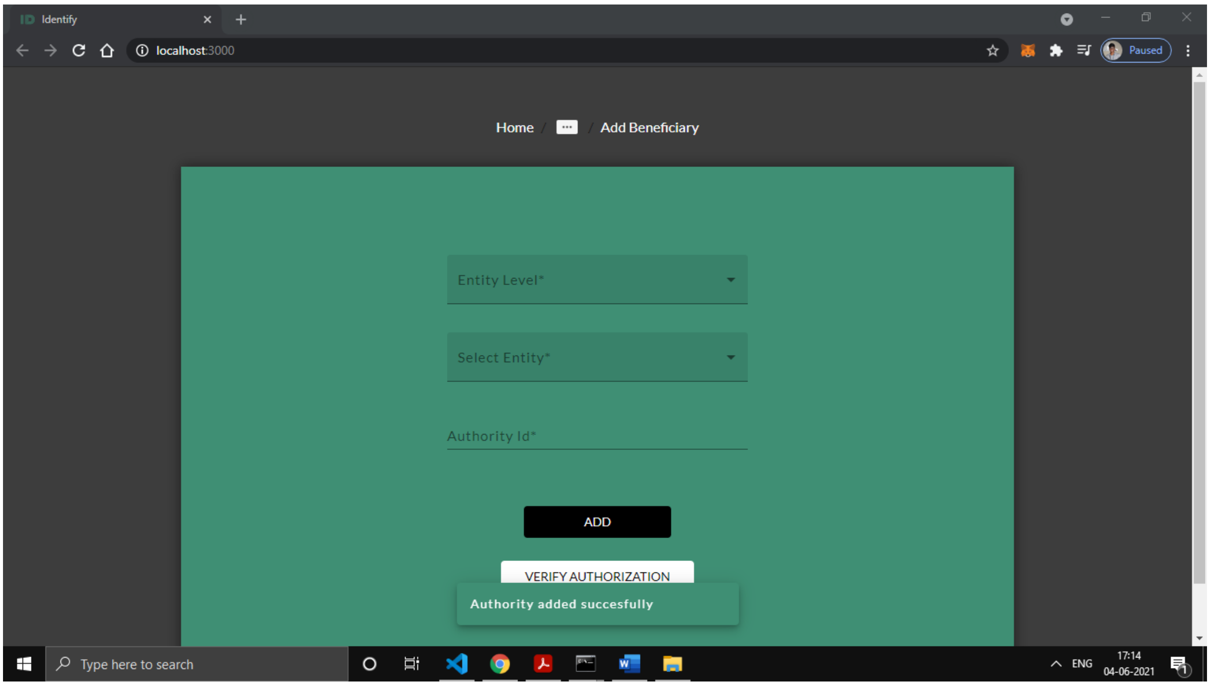
0

GAS FEE 0.002297
No Conversion Rate Available

Gas Price (GWEI) 20
Gas Limit 114845

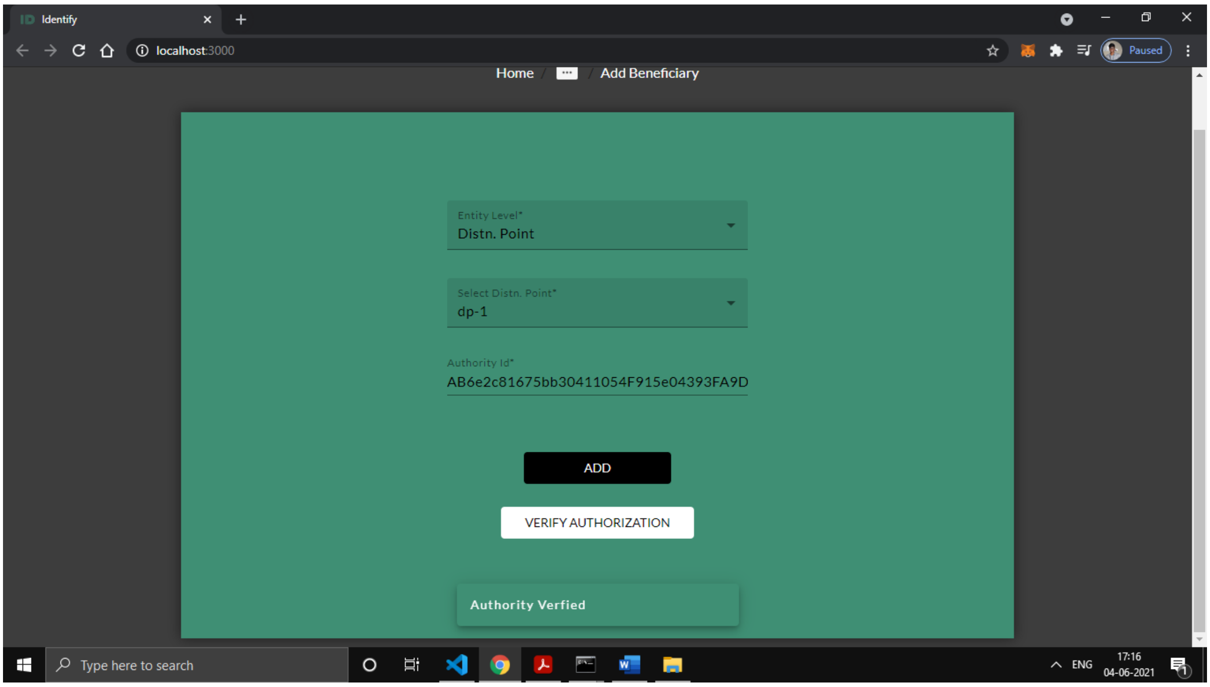
AMOUNT + GAS FEE
TOTAL 0.002297
No Conversion Rate Available

Fig. 5.8 Metamask prompt for signing the (add authority) transaction



Authority added succesfully

Fig 5.9 Result: Successful assignment of authority to distribution point 1



Entity Level*
Distn. Point

Select Distn. Point*
dp-1

Authority Id*
AB6e2c81675bb30411054F915e04393FA9D

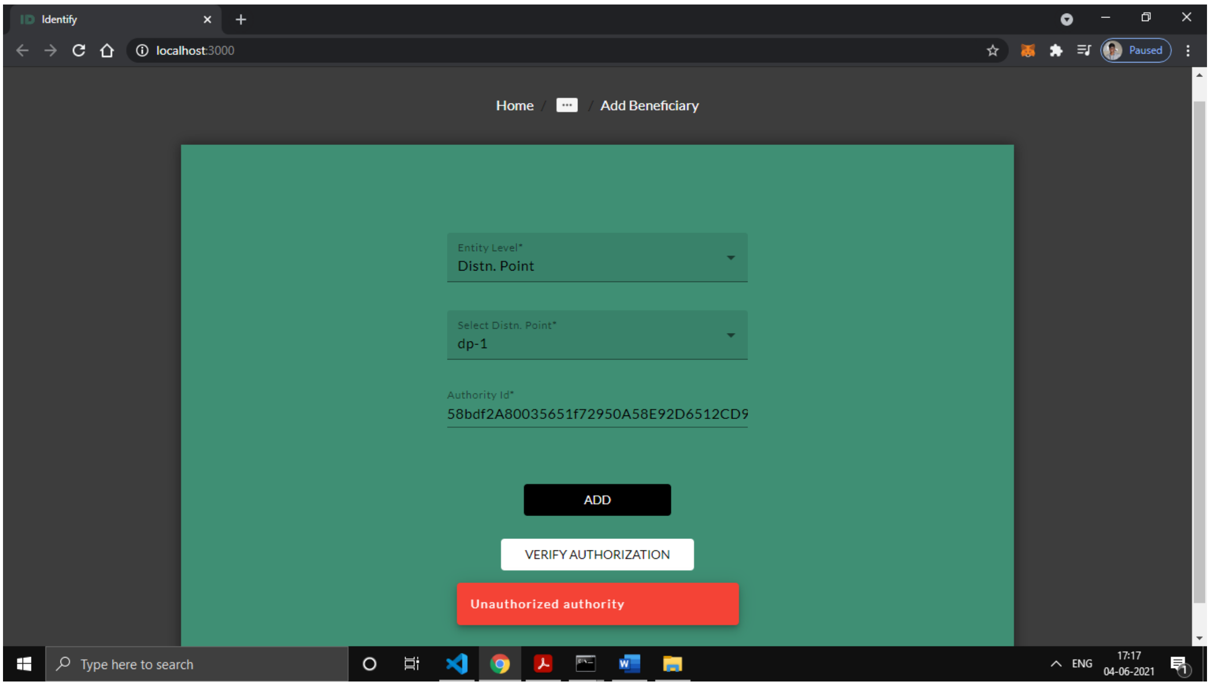
Authority Verfied

Fig 5.10 Result: Authority verification success

Similarly, two more authorities were assigned to distribution point-1.

Authority ids of distribution point-1:

- AB6e2c81675bb30411054F915e04393FA9D20441 (Authority-1)
- 3C28961eeB55d82b212B848DD779f7467D77a1A4 (Authority-2)
- E8C26fC27209066C4d2F0ca9703B2c4B604904B5 (Authority-3)



Entity Level*
Distn. Point

Select Distn. Point*
dp-1

Authority Id*
58bdf2A80035651f72950A58E92D6512CD9

Unauthorized authority

Fig. 5.11 Result: Verification for unauthorized authority

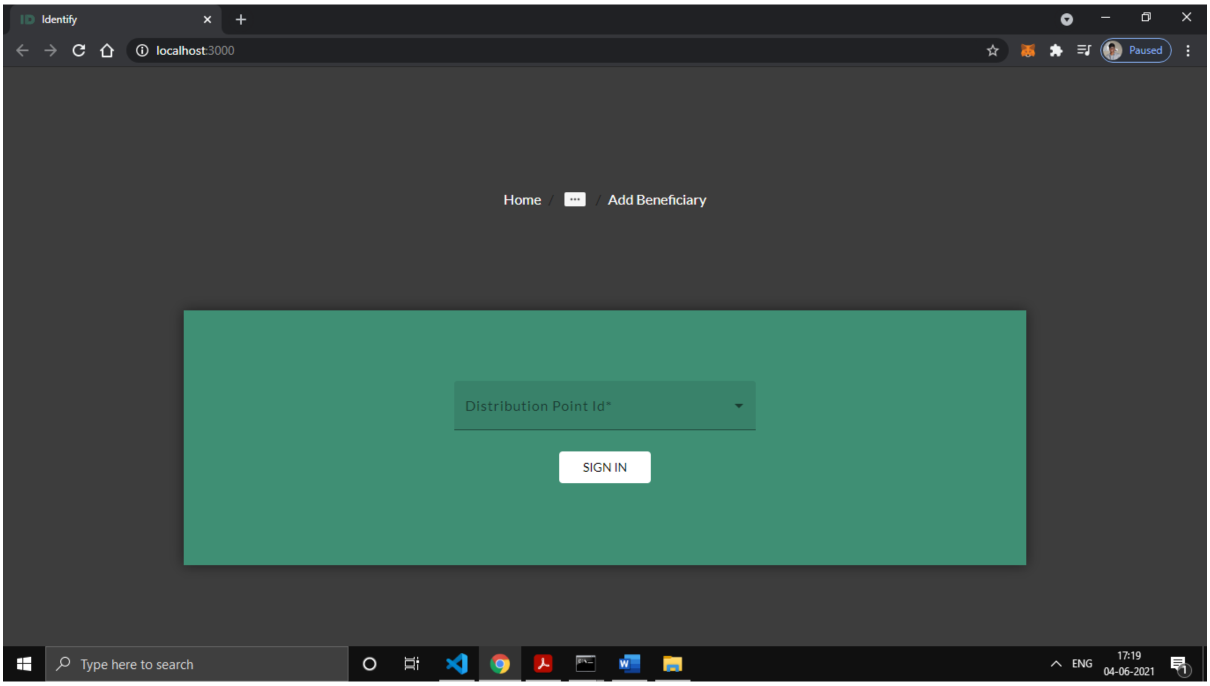
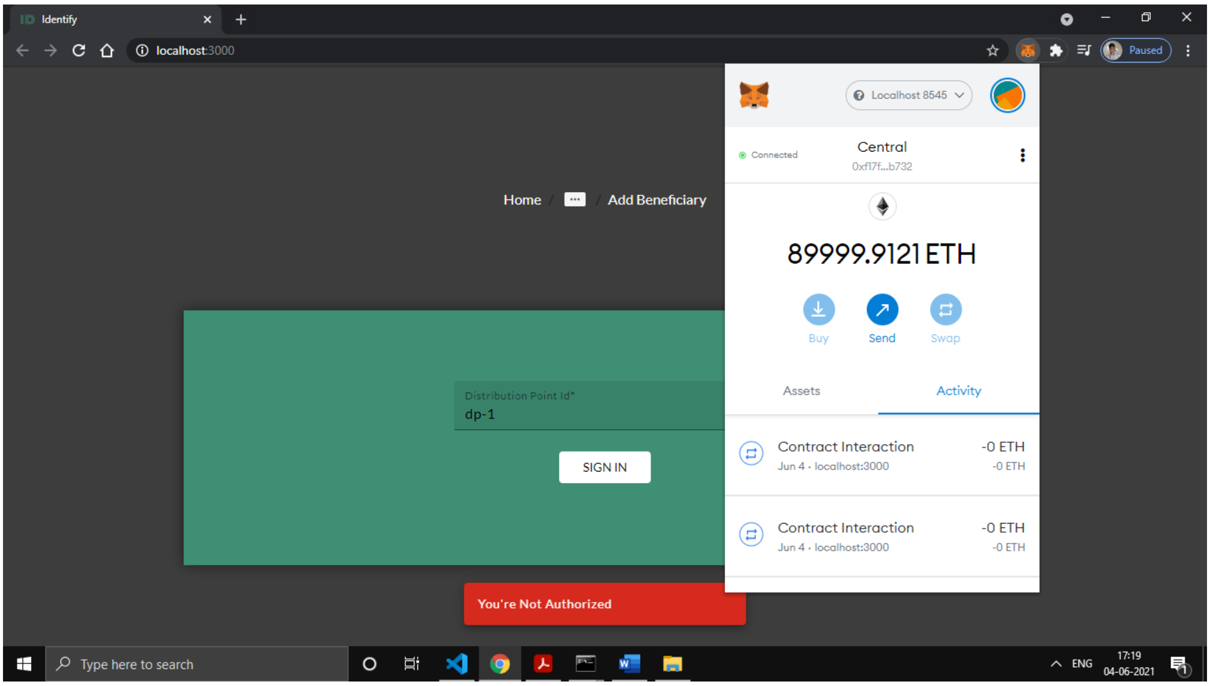


Fig. 5.12 Login screen for distribution points



Distribution Point Id*
dp-1

You're Not Authorized

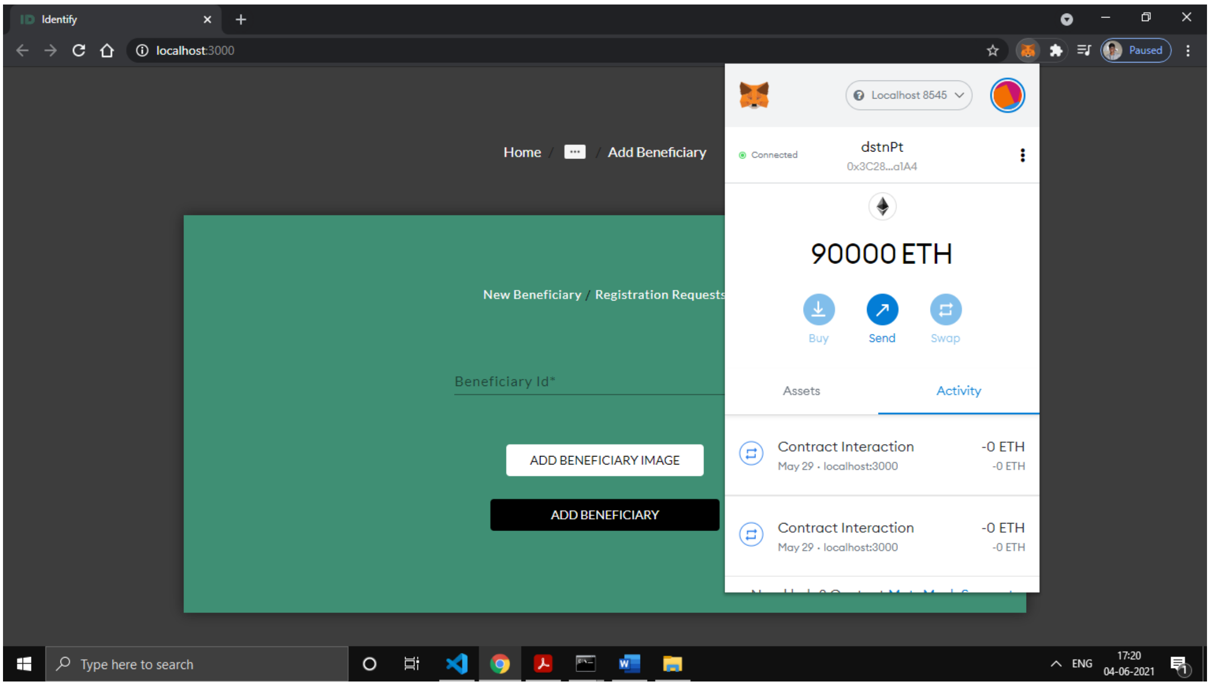
Connected
Central
0xf17f...b732

89999.9121 ETH

Contract Interaction -0 ETH
Jun 4 · localhost:3000 -0 ETH

Contract Interaction -0 ETH
Jun 4 · localhost:3000 -0 ETH

Fig. 5.13 Unauthorized login attempt for distribution point -1



Beneficiary Id*

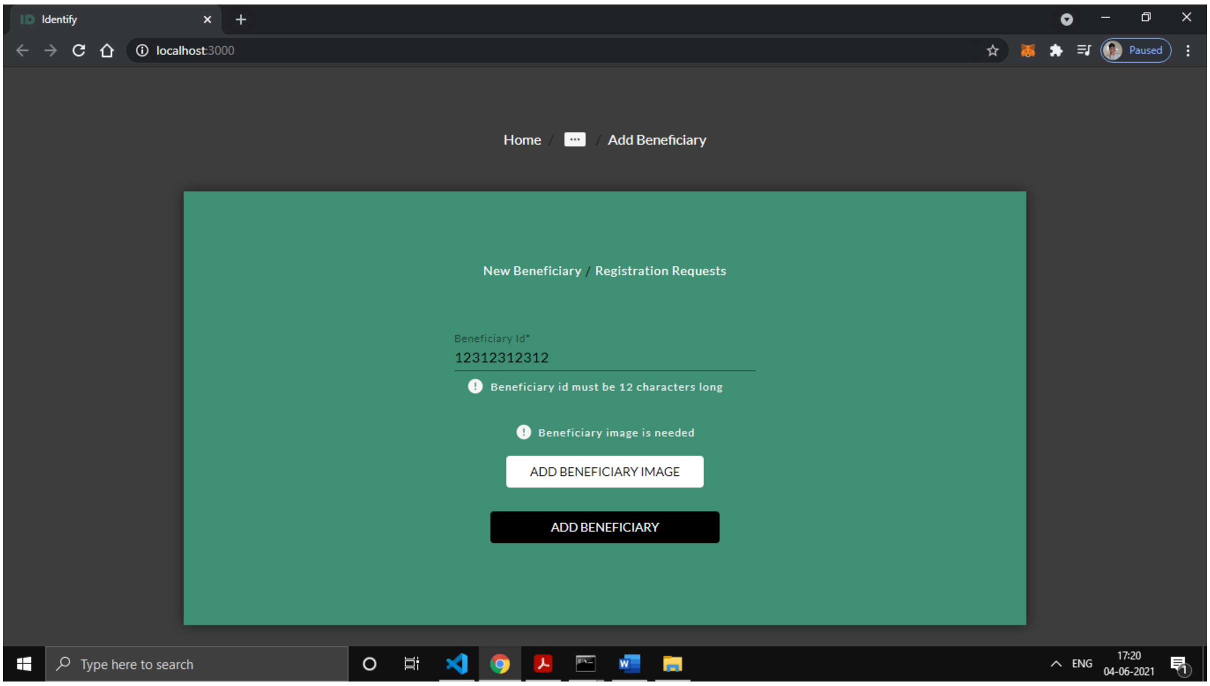
Connected
dstnPt
0x3C28...a1A4

90000 ETH

Contract Interaction -0 ETH
May 29 · localhost:3000 -0 ETH

Contract Interaction -0 ETH
May 29 · localhost:3000 -0 ETH

Fig. 5.14 Portal for distribution points after successful login (Currently logged in authority: 3C28961eeB55d82b212B848DD779f7467D77a1A4)

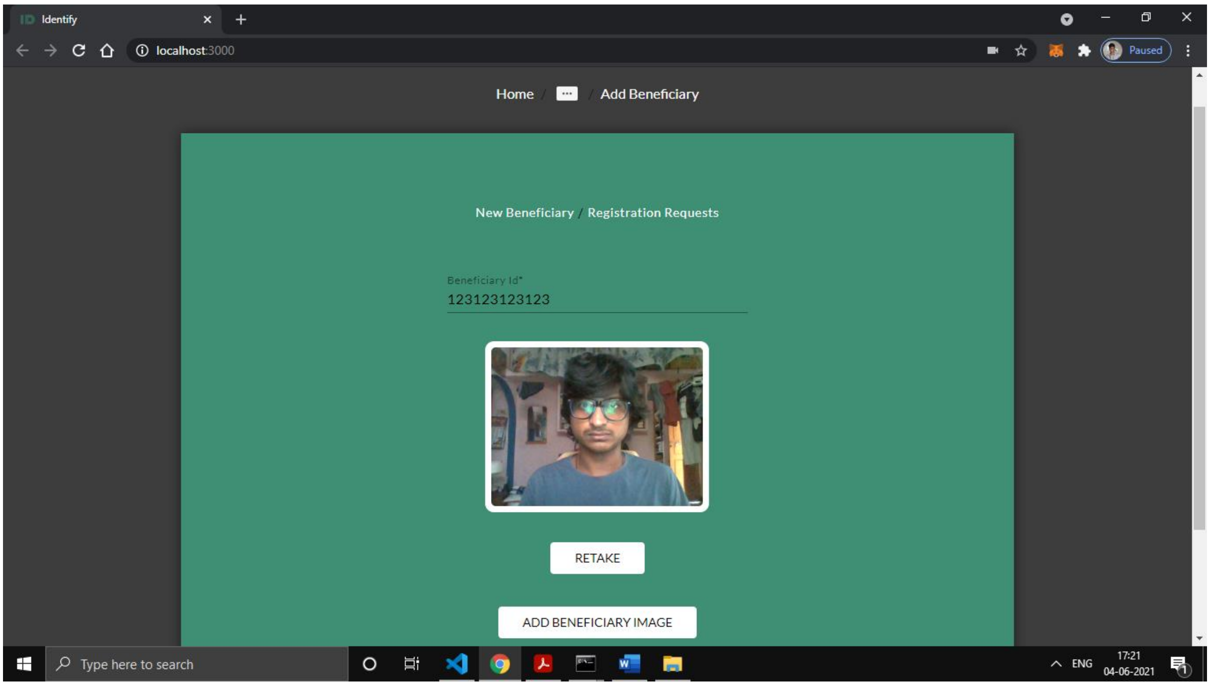


Beneficiary Id*
12312312312

Beneficiary id must be 12 characters long

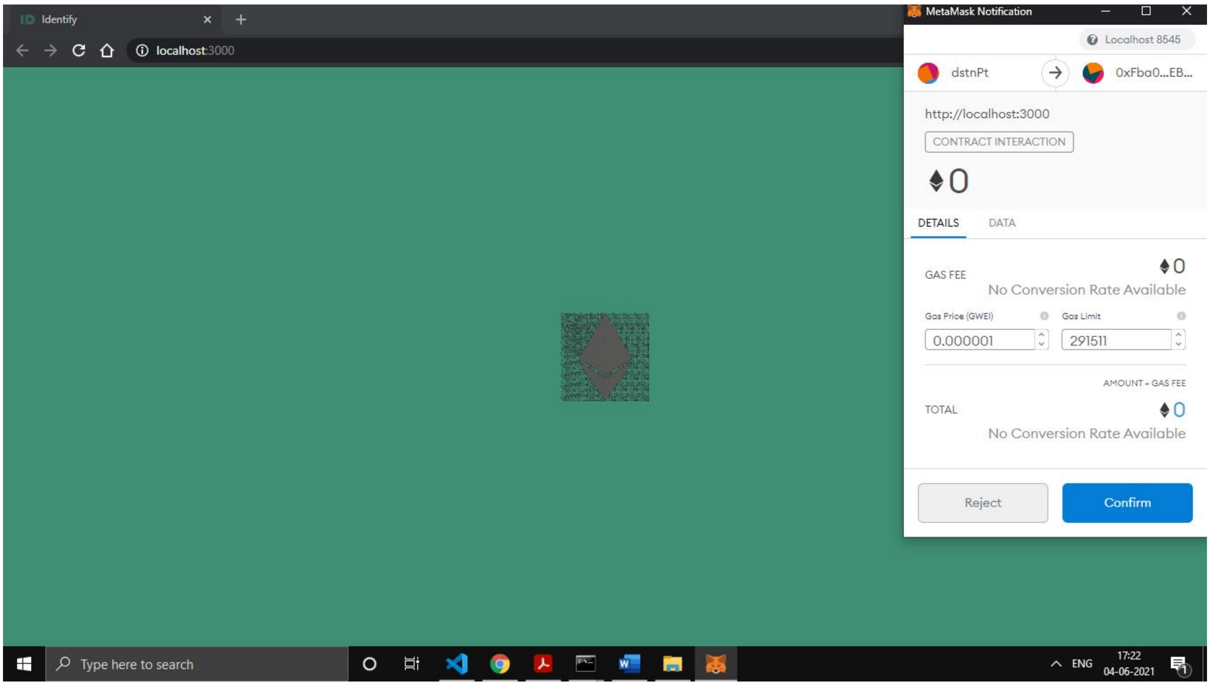
Beneficiary image is needed

Fig. 5.15 Result: Invalid data entry for beneficiary registration



Beneficiary Id*
123123123123

Fig. 5.16 Captured beneficiary image for registration



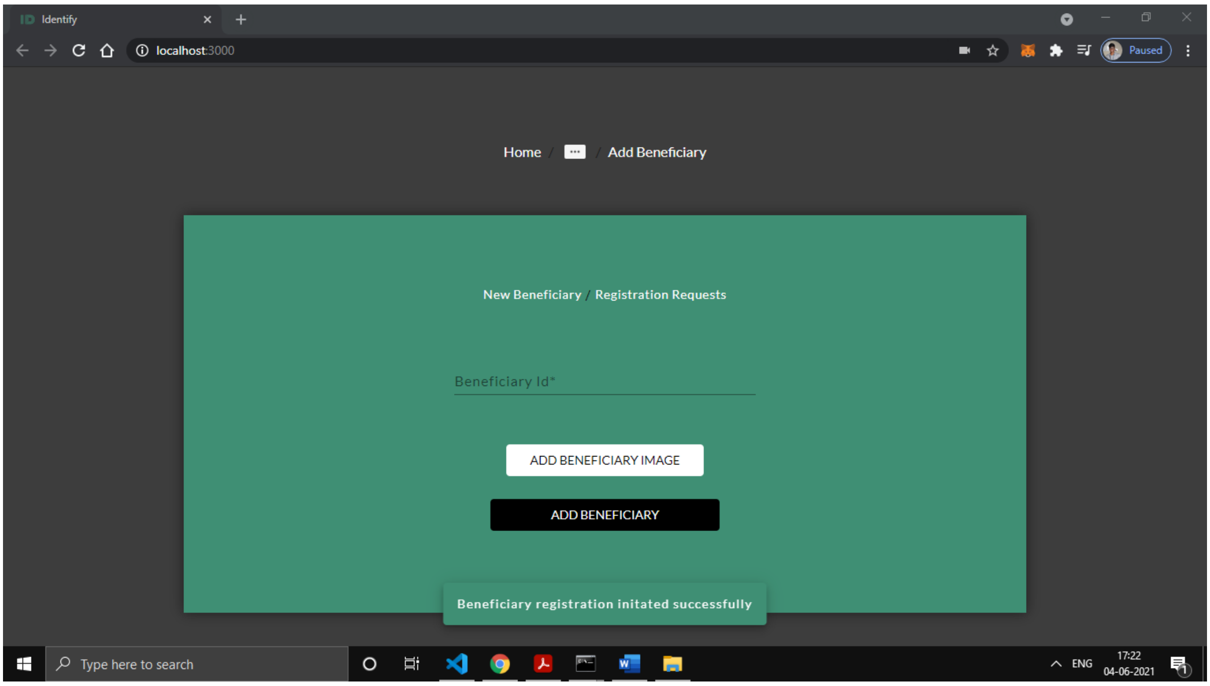
0

GAS FEE 0
No Conversion Rate Available

Gas Price (GWEI): 0.000001
Gas Limit: 291511

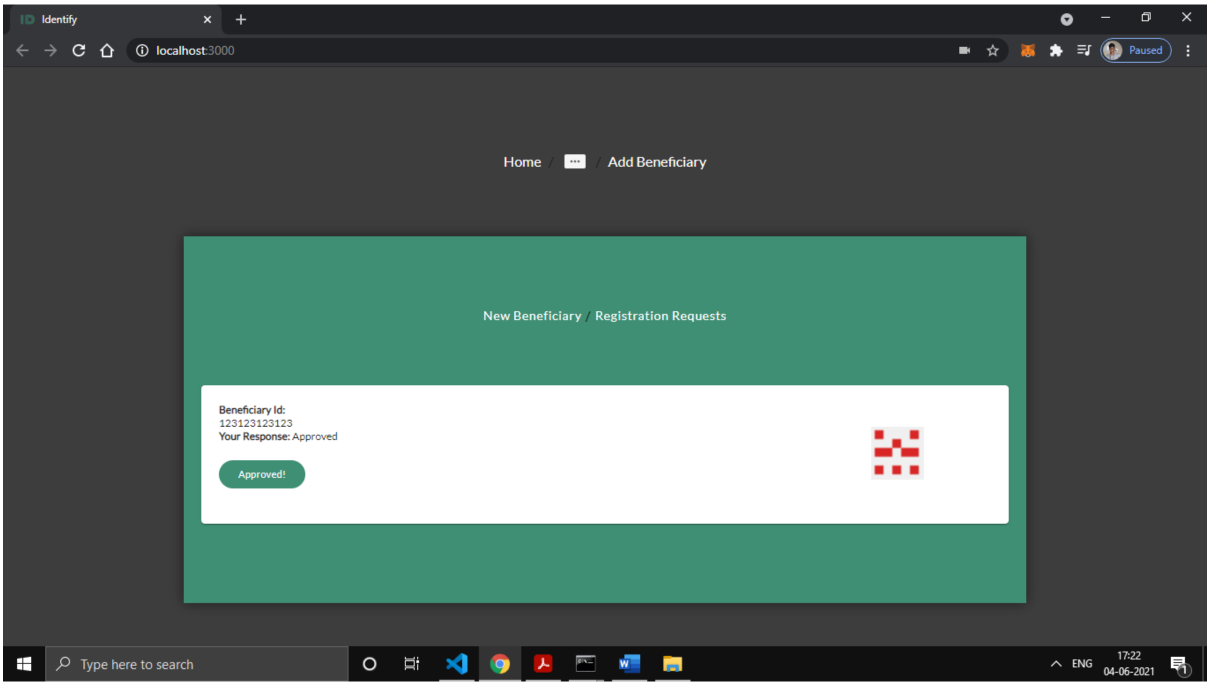
AMOUNT + GAS FEE
TOTAL 0
No Conversion Rate Available

Fig. 5.17 Metamask Prompt to sign Beneficiary registration transaction



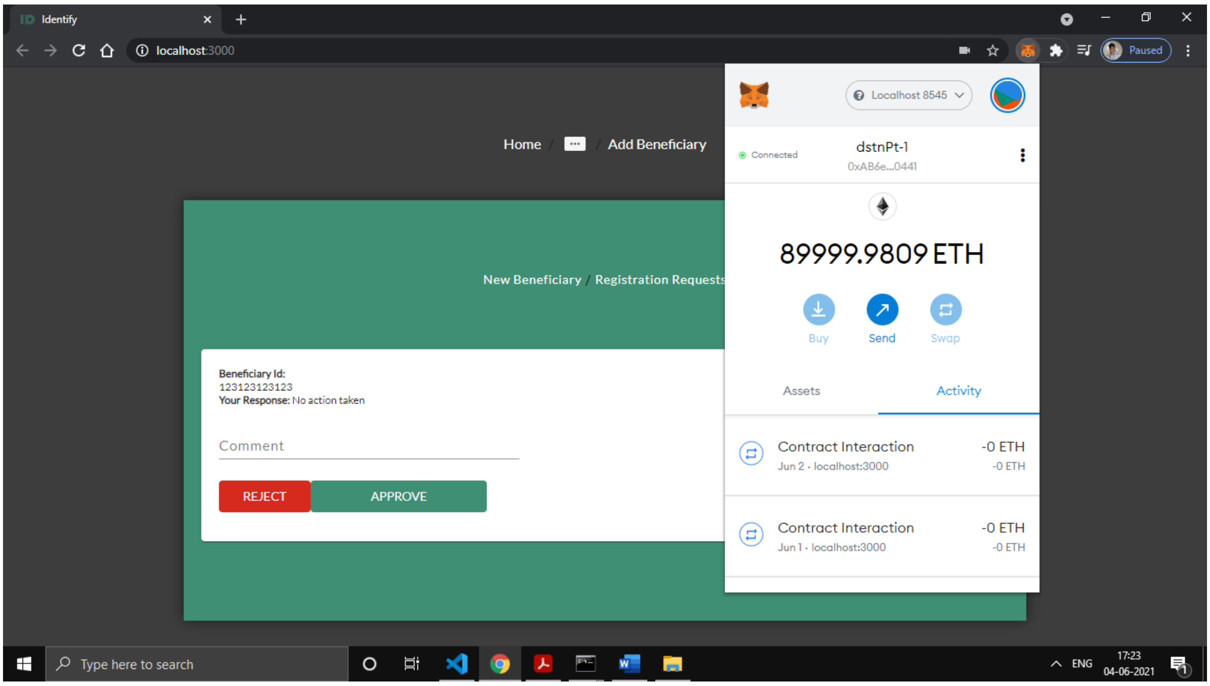
Beneficiary registration initated successfully

Fig. 5.18 Result: Successful Beneficiary registration initiation



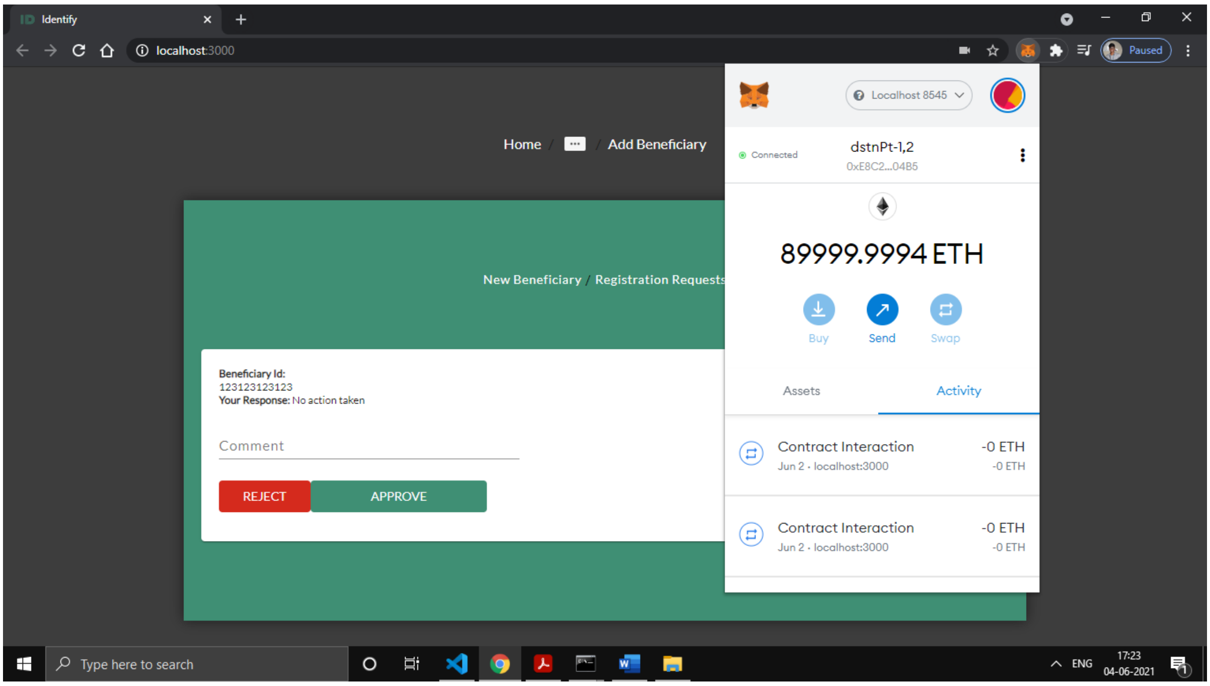
Beneficiary Id: 123123123123
Your Response: Approved

Fig. 5.19 Registration requests for distribution point -1 authority. (Currently logged in authority: 3C28961eeB55d82b212B848DD779f7467D77a1A4*)*



Beneficiary Id: 123123123123
Your Response: No action taken

Fig. 5.20 Registration requests for distribution point -1 authority. (Currently logged in authority: AB6e2c81675bb30411054F915e04393FA9D20441*)*



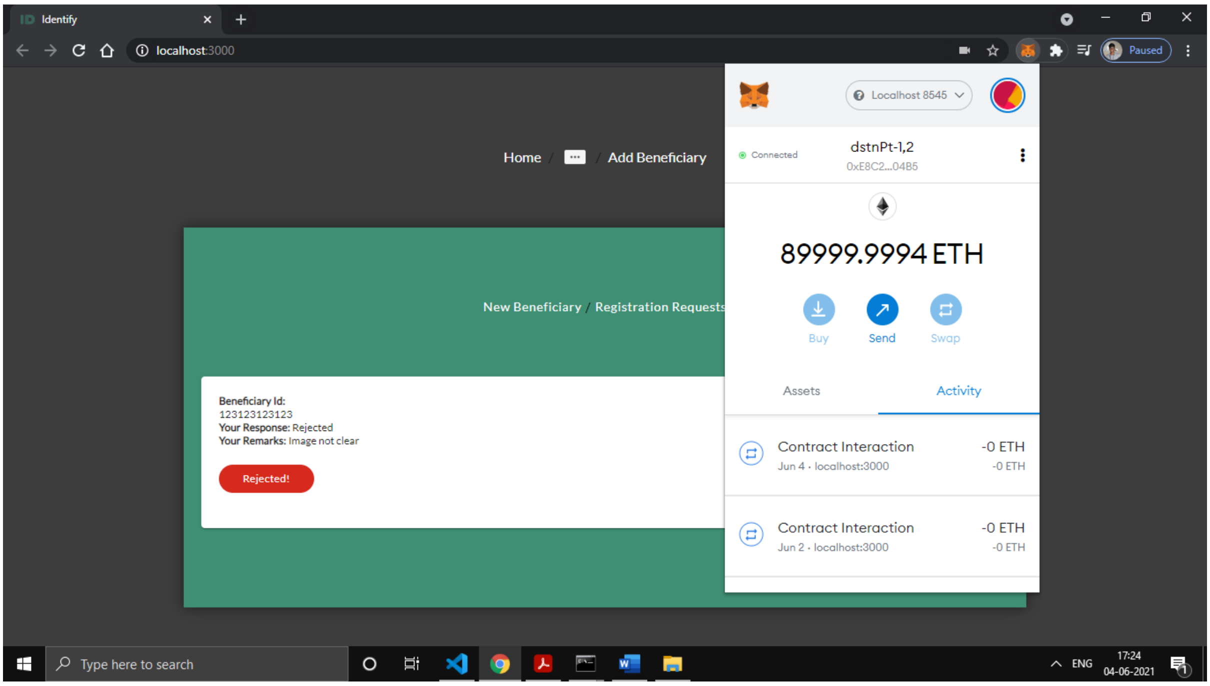
Beneficiary Id:
123123123123
Your Response: No action taken

Contract Interaction -0 ETH
Jun 2 · localhost:3000 -0 ETH

Contract Interaction -0 ETH
Jun 2 · localhost:3000 -0 ETH

Fig. 5.21 Registration requests for distribution point -1 authority. (Currently logged in authority: E8C26fC27209066C4d2F0ca9703B2c4B604904B5*)*

Image 5.20 shows the registration request for authority AB6e2c81675bb30411054F915e04393FA9D20441 of distribution point-1. As a beneficiary registration has been initiated by the authority-2, option to either reject/approve the beneficiary (id: 123123123123) is being show to authority-1 and authority -3 (Image 5.21).



Beneficiary Id:
123123123123
Your Response: Rejected
Your Remarks: Image not clear

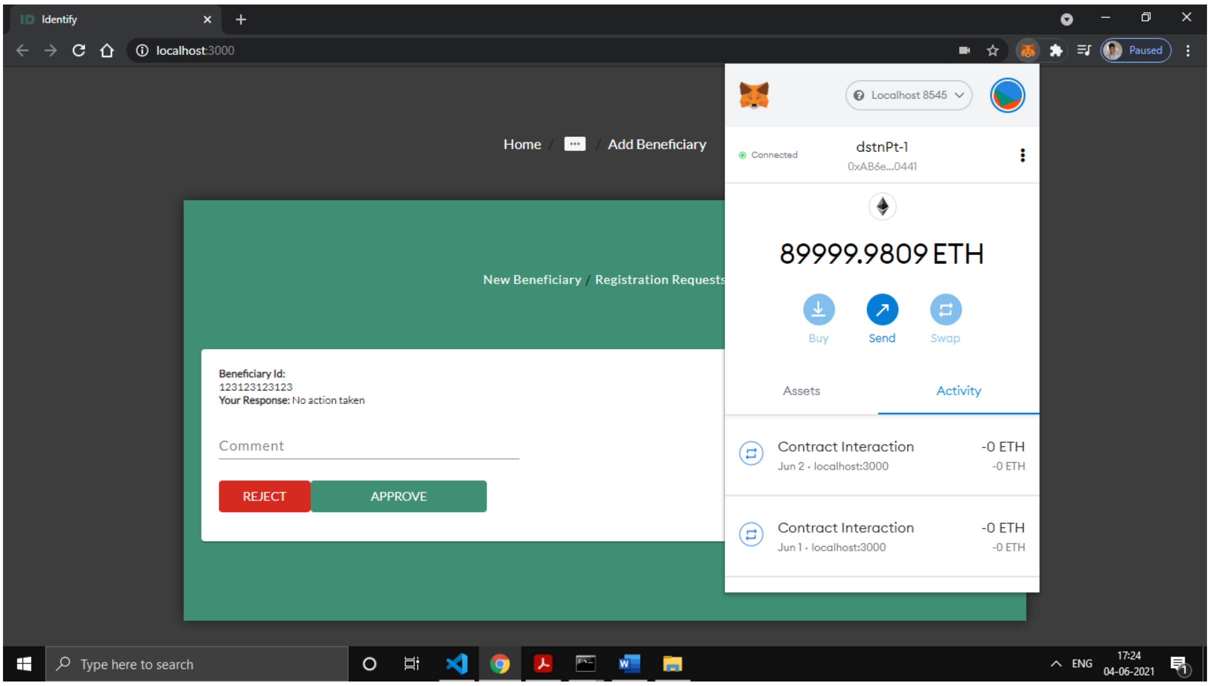
Contract Interaction -0 ETH
Jun 4 · localhost:3000 -0 ETH

Contract Interaction -0 ETH
Jun 2 · localhost:3000 -0 ETH

Fig. 5.22 Beneficiary (id:123123123123) rejected by authority-3

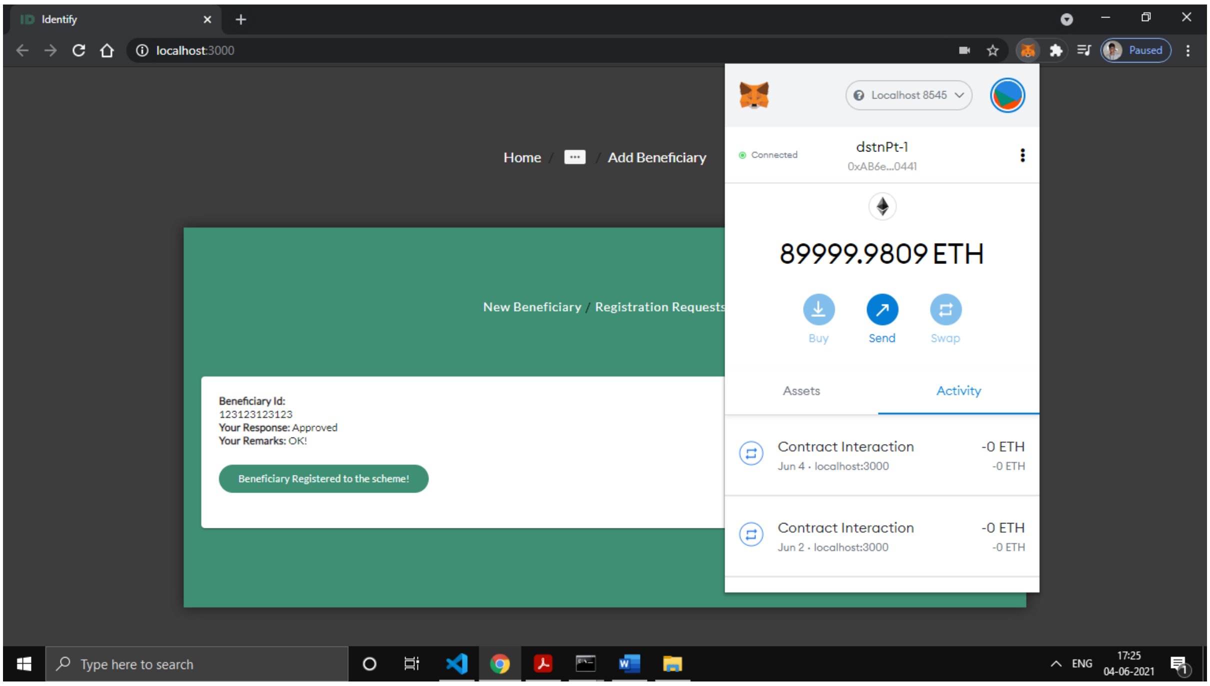
Table 5.1 Table showing beneficiary (id:123123123123) status after authority-3's rejection

Authority-id	**Action taken**	**Beneficiary Status (id: 123123123123)**
3C28961eeB55d82b212B848DD779f7467D77a1A4	Approved	Pending
E8C26fC27209066C4d2F0ca9703B2c4B604904B5	Rejected	Pending



Beneficiary Id:
123123123123
Your Response: No action taken

Fig. 5.23 Registration requests for distribution point -1 authority. (Currently logged in authority: AB6e2c81675bb30411054F915e04393FA9D20441*)*



Beneficiary Id:
123123123123
Your Response: Approved
Your Remarks: OK!

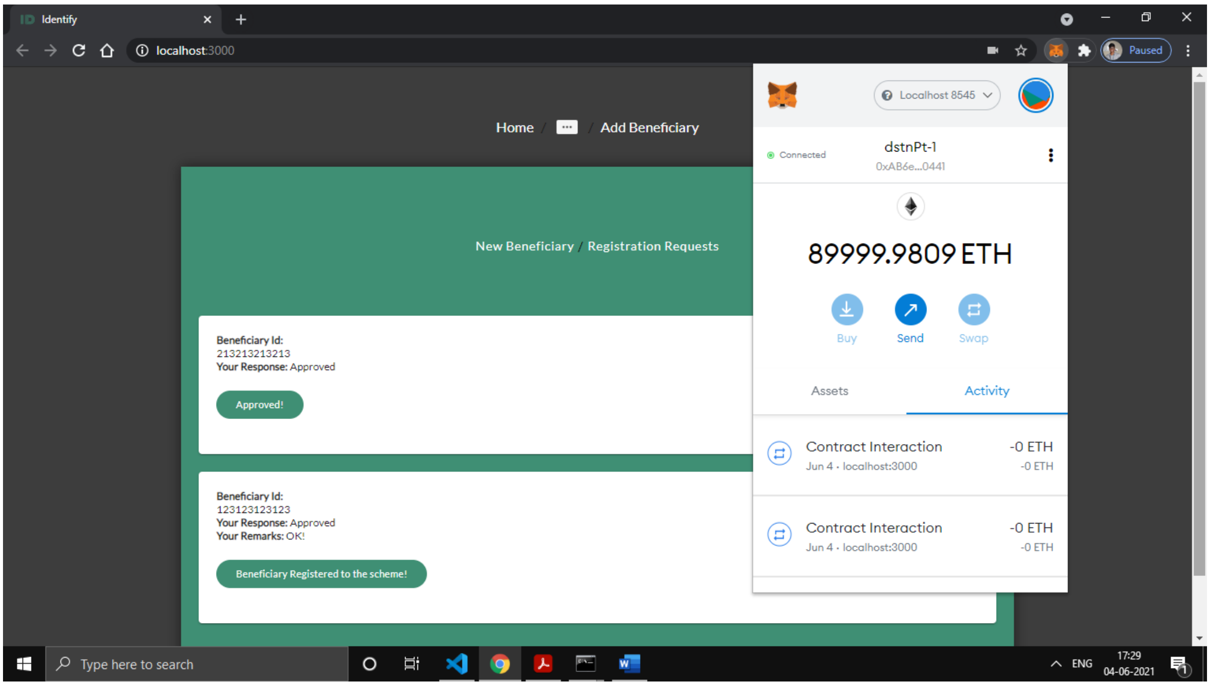
Beneficiary Registered to the scheme!

Fig. 5.24 Result: Registration status of beneficiary 123123123123 after approval by authority-1

Table 5.2 Table showing beneficiary (id:123123123123) status after authority-1's approval

Authority-id	**Action taken**	**Beneficiary Status (id: 123123123123)**
3C28961eeB55d82b212B848DD779f7467D77a1A4	Approved	Pending
E8C26fC27209066C4d2F0ca9703B2c4B604904B5	Rejected	Pending
AB6e2c81675bb30411054F915e04393FA9D20441	Approved	Registered successfully to the scheme

As a super majority of authorities have approved the beneficiary(id:123123123123), the beneficiary has been successfully added to the scheme. Here, responses of all the 3 authorities of distribution point-1 have been recorded to determine the beneficiary registration.



Beneficiary Id:
213213213213
Your Response: Approved

Approved!

Beneficiary Id:
123123123123
Your Response: Approved
Your Remarks: OK!

Beneficiary Registered to the scheme!

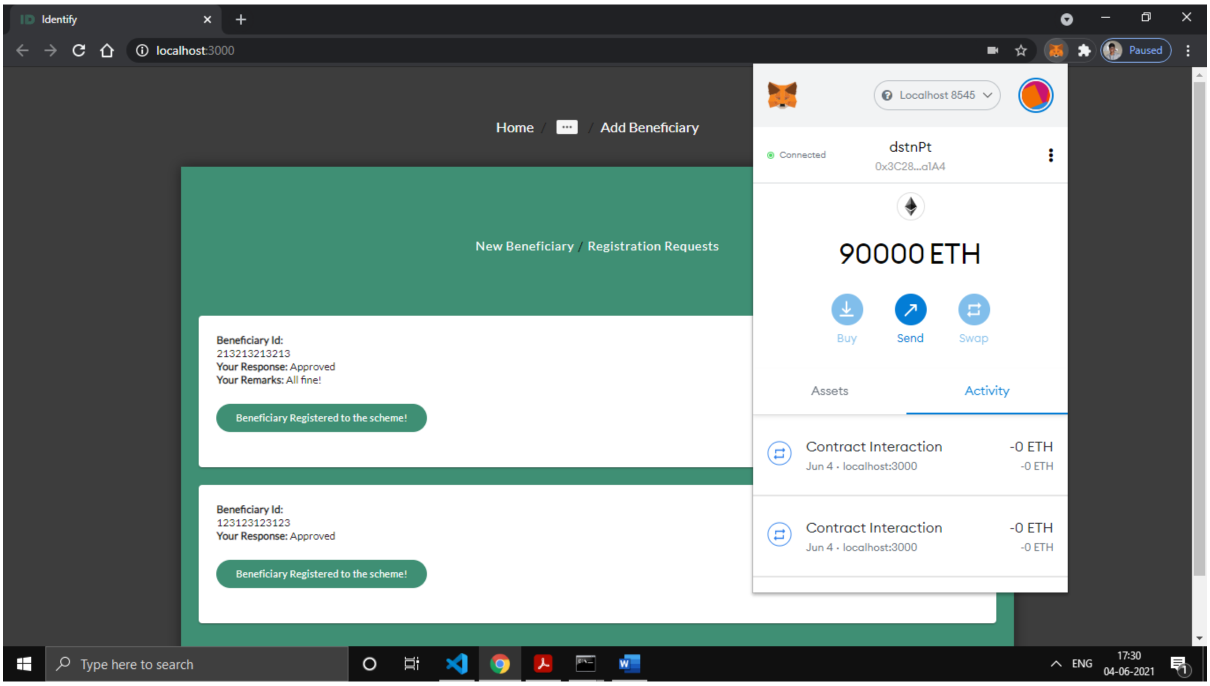
Contract Interaction -0 ETH
Jun 4 · localhost:3000 -0 ETH

Contract Interaction -0 ETH
Jun 4 · localhost:3000 -0 ETH

Fig. 5.25 New Beneficiary registration (id:2123213213213) initiated by authority-1

Table 5.3 Table showing beneficiary status (id:213213213213) after authority-1's registration initiation

Authority-id	**Action taken**	**Beneficiary Status (id: 123123123123)**
AB6e2c81675bb30411054F915e04393FA9D20441	Approved (Implicit as initiator)	Pending
3C28961eeB55d82b212B848DD779f7467D77a1A4	-	-
E8C26fC27209066C4d2F0ca9703B2c4B604904B5	-	-



Beneficiary Id:
213213213213
Your Response: Approved
Your Remarks: All fine!

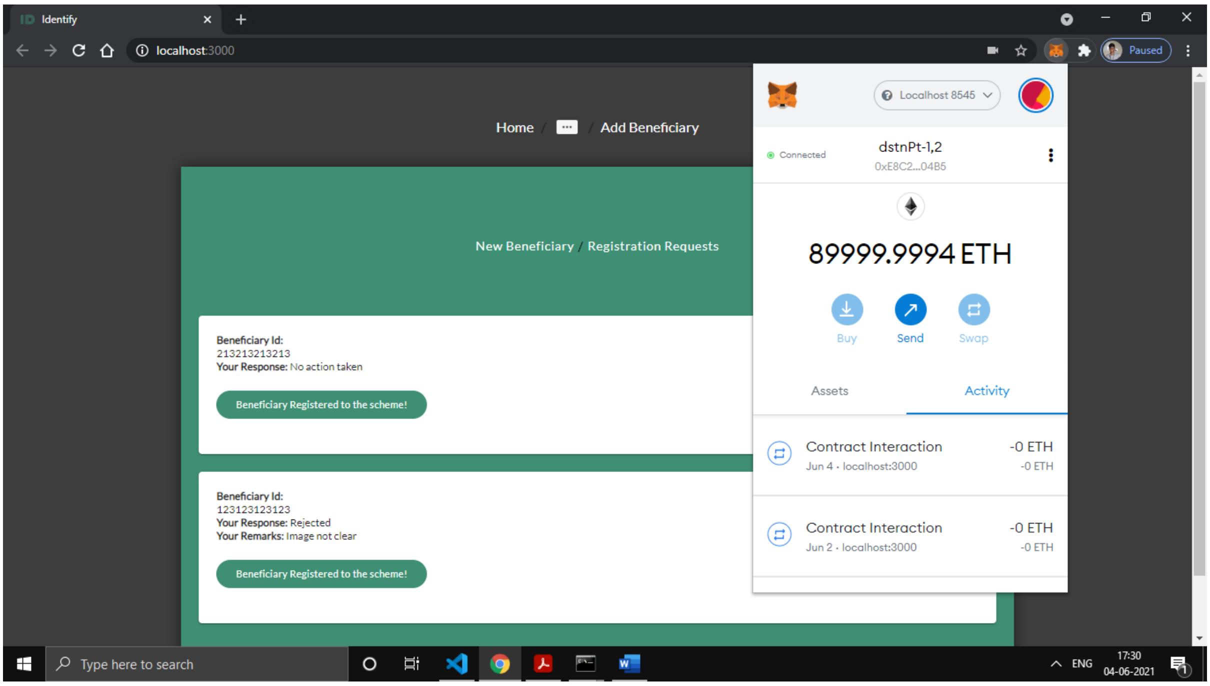
Beneficiary Id:
123123123123
Your Response: Approved

Fig. 5.26 Beneficiary registered (id:2123213213213) to the scheme after approval by initiated by authority-2

Table 5.4 Table showing beneficiary status (id:213213213213) after authority-2's approval

Authority-id	**Action taken**	**Beneficiary Status (id: 213213213213)**
AB6e2c81675bb30411054F915e04393FA9D20441	Approved (Implicit as initiator)	Pending
3C28961eeB55d82b212B848DD779f7467D77a1A4	Approved	Registered successfully to the scheme
E8C26fC27209066C4d2F0ca9703B2c4B604904B5	-	-

As a super majority of authorities have approved the beneficiary (id:213213213213), even before authority-3 (E8C26fC27209066C4d2F0ca9703B2c4B604904B5) has taken any action, the beneficiary has been successfully added to the scheme. Here, response of authority-3 isn't recorded since approval of more than 50% of authorities has been done.



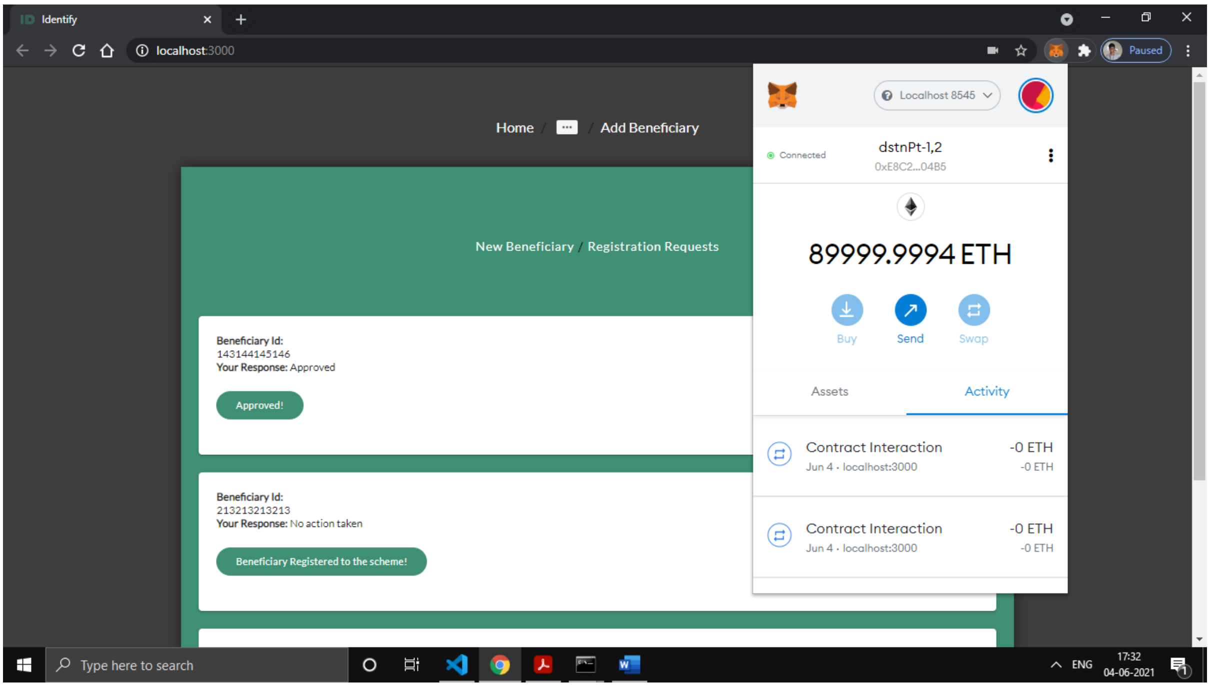
Beneficiary Id:
213213213213
Your Response: No action taken

Beneficiary Registered to the scheme!

Beneficiary Id:
123123123123
Your Response: Rejected
Your Remarks: Image not clear

Beneficiary Registered to the scheme!

Fig. 5.27 Registration requests for distribution point -1 authority after the registration of beneficiary (id:213213213213) (Currently logged in authority: E8C26fC27209066C4d2F0ca9703B2c4B604904B5)



Beneficiary Id:
143144145146
Your Response: Approved

Approved!

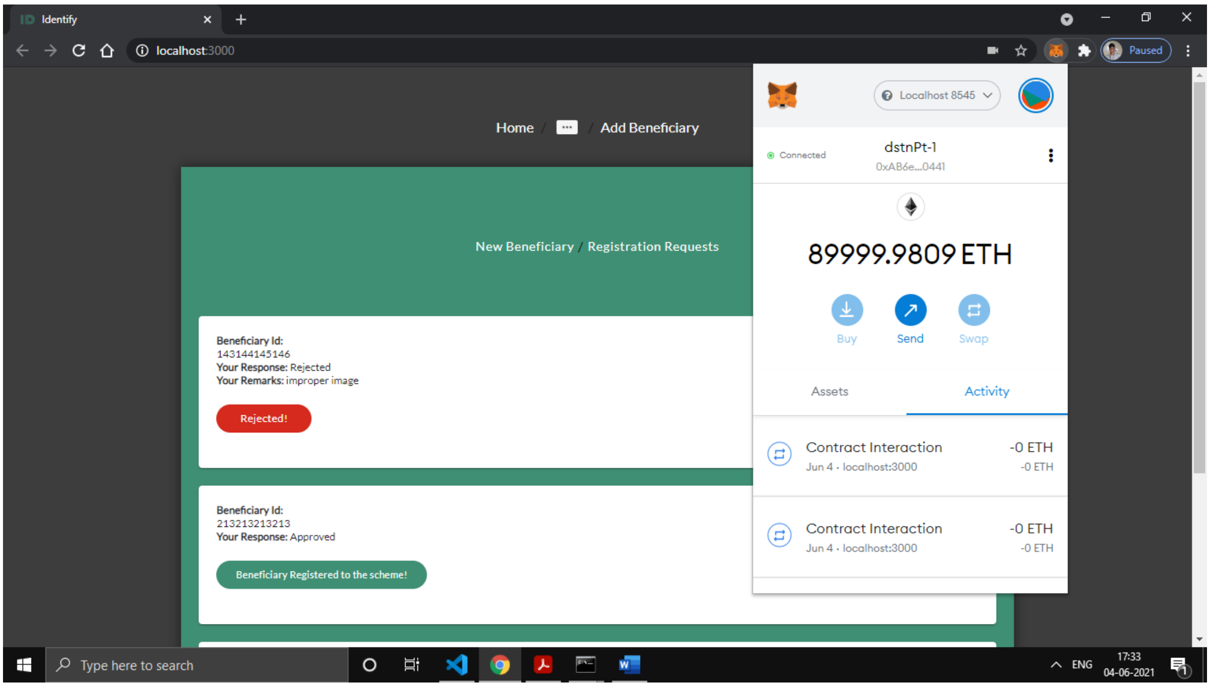
Beneficiary Id:
213213213213
Your Response: No action taken

Beneficiary Registered to the scheme!

Fig. 5.28 New Beneficiary registration (id:143144145146) initiated by authority-1

Table 5.5 Table showing beneficiary status (id:143144145146) after authority-3's registration initiation

Authority-id	**Action taken**	**Beneficiary Status (id: 143144145146)**
E8C26fC27209066C4d2F0ca9703B2c4B604904B5	Approved (Implicit as initiator)	Pending
3C28961eeB55d82b212B848DD779f7467D77a1A4	-	-
AB6e2c81675bb30411054F915e04393FA9D20441	-	-



Beneficiary Id:
143144145146
Your Response: Rejected
Your Remarks: improper image

Rejected!

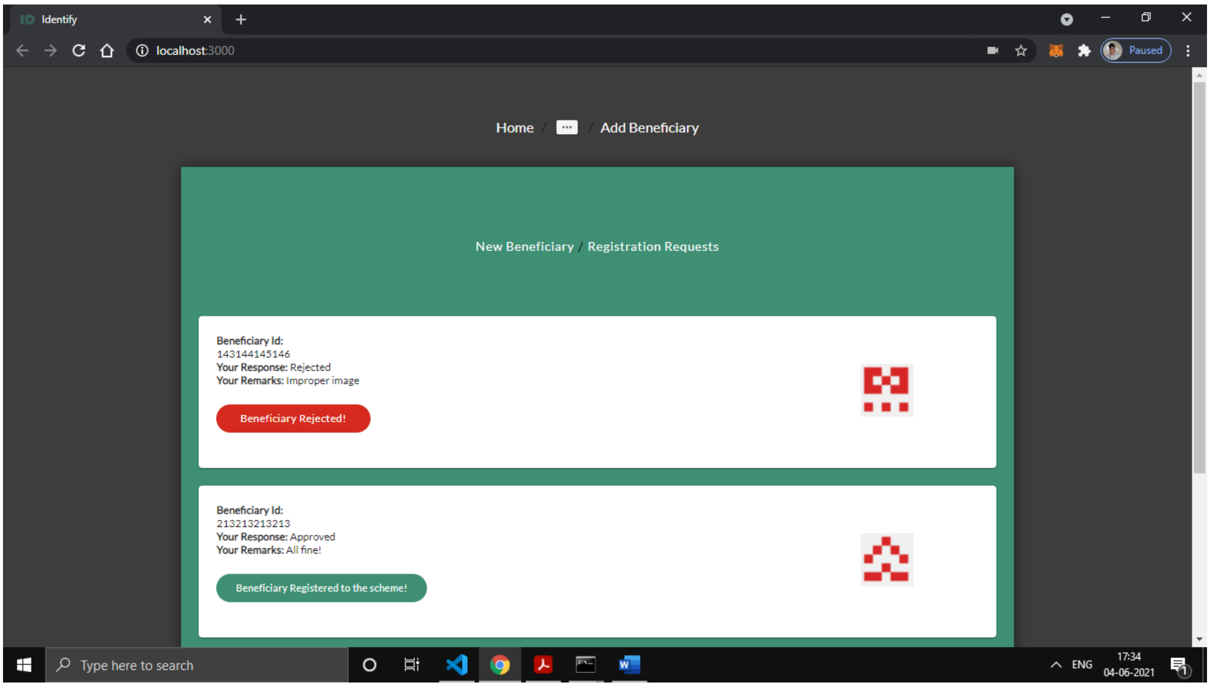
Beneficiary Id:
213213213213
Your Response: Approved

Beneficiary Registered to the scheme!

Fig. 5.29 Result: Registration status of beneficiary 143144145146 after rejection by authority-1

Table 5.6 Table showing beneficiary status (id:143144145146) after authority-1's rejection

Authority-id	Action taken	Beneficiary Status (id: 143144145146)
E8C26fC27209066C4d2F0ca9703B2c4B604904B5	Approved (Implicit as initiator)	Pending
3C28961eeB55d82b212B848DD779f7467D77a1A4	-	-
AB6e2c81675bb30411054F915e04393FA9D20441	Rejected	Pending


Identify
localhost:3000
Paused
Home / ··· / Add Beneficiary
New Beneficiary / Registration Requests
Type here to search
ENG
17:34
04-06-2021

Beneficiary Id:
143144145146
Your Response: Rejected
Your Remarks: Improper image

Beneficiary Rejected!

Beneficiary Id:
213213213213
Your Response: Approved
Your Remarks: All fine!

Beneficiary Registered to the scheme!

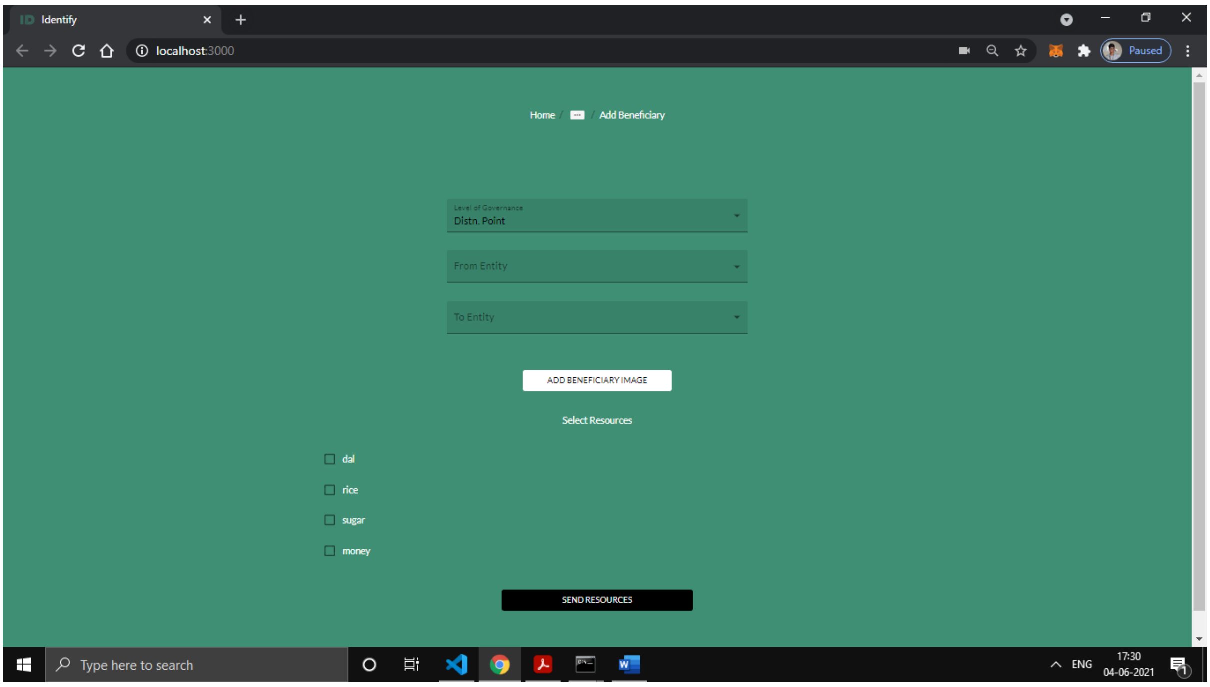
Fig. 5.30 Result: Registration status of beneficiary 143144145146 after rejection by authority-2

Table 5.6 Table showing beneficiary status (id:143144145146) after authority-1's rejection

Authority-id	Action taken	Beneficiary Status (id: 143144145146)
E8C26fC27209066C4d2F0ca9703B2c4B604904B5	Approved (Implicit as initiator)	Pending

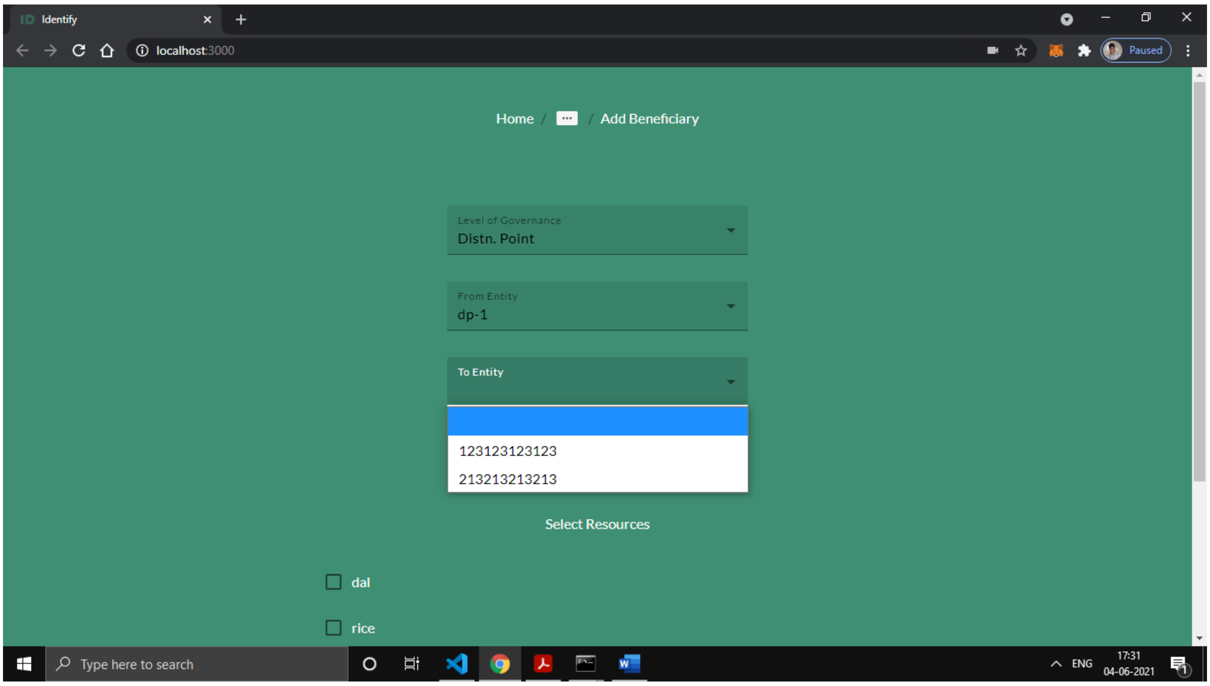
3C28961eeB55d82b212B848DD779f7467D77a1A4	Rejected	Rejected by the distribution point. Not registered to the scheme.
AB6e2c81675bb30411054F915e04393FA9D20441	Rejected	Pending

As a super majority of authorities have rejected the beneficiary(id:143144145146), the beneficiary has not been added to the scheme. Here, responses of all the 3 authorities of distribution point-1 have been recorded to determine the beneficiary registration.



Level of Governance: Distn. Point

Fig. 5.31 Resource transfer initiation page with from Entity selected: ‘Distribution point’.



Level of Governance
Distn. Point

From Entity
dp-1

To Entity

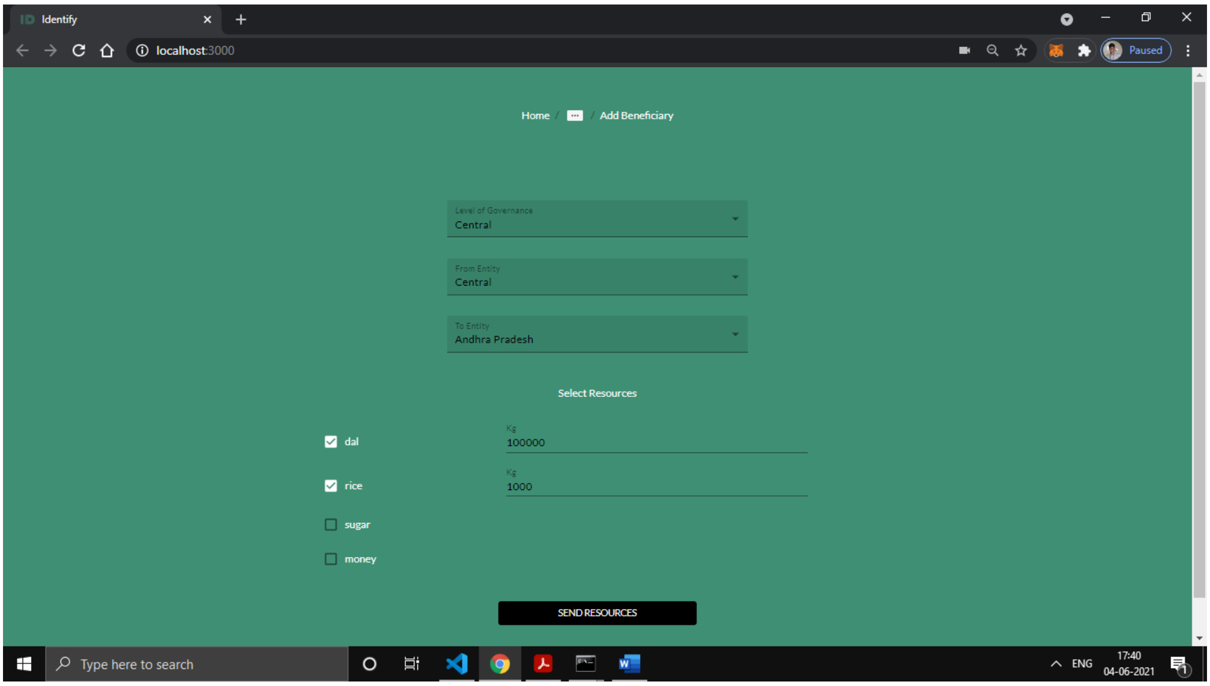
123123123123
213213213213

Select Resources

□ dal

□ rice

Fig. 5.32 Image showing registered beneficiaries under distribution point-1



Level of Governance
Central

From Entity
Central

To Entity
Andhra Pradesh

Select Resources

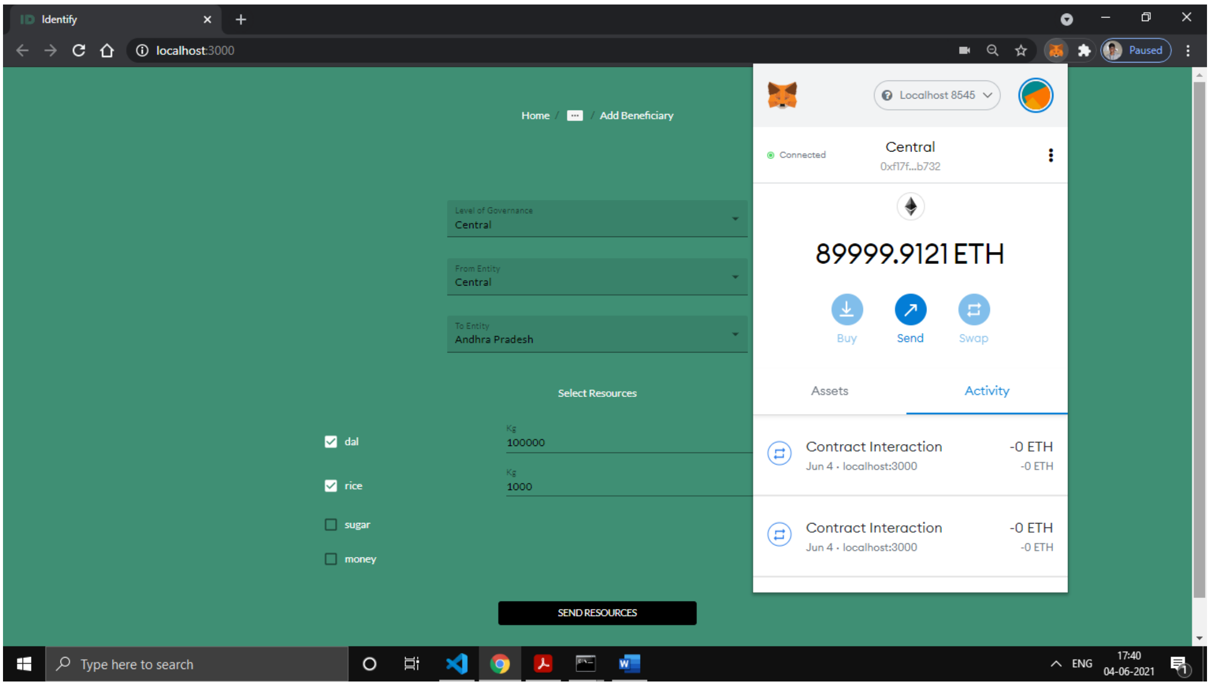
☑ dal Kg 100000

☑ rice Kg 1000

□ sugar

□ money

Fig. 5.33 Image showing resource-transfer-initiation from 'Central' to 'Andhra Pradesh'



Home / ... / Add Beneficiary

Level of Governance
Central

From Entity
Central

To Entity
Andhra Pradesh

Select Resources

☑ dal — Kg 100000

☑ rice — Kg 1000

□ sugar

□ money

SEND RESOURCES

Localhost 8545

Connected

Central
0xf17f...b732

89999.9121 ETH

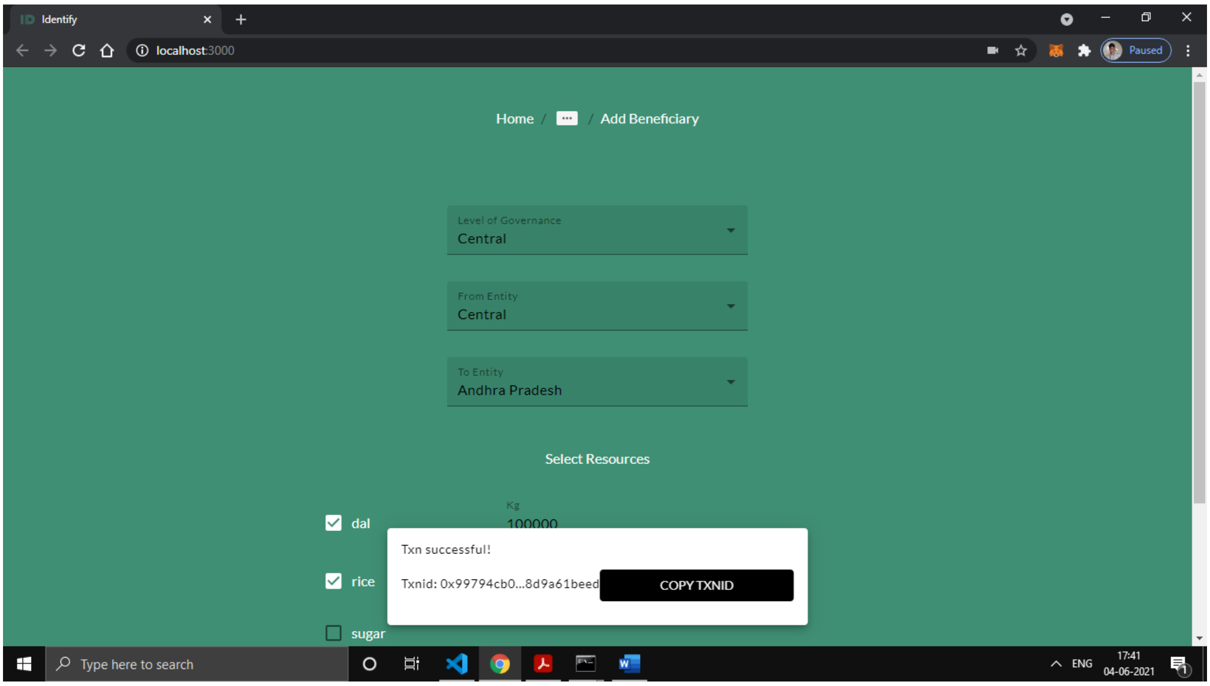
Buy Send Swap

Assets Activity

Contract Interaction -0 ETH
Jun 4 · localhost:3000 -0 ETH

Contract Interaction -0 ETH
Jun 4 · localhost:3000 -0 ETH

Fig. 5.34 Image showing metamask prompt to sign the resource-transfer-transaction (Currently logged account: f17f52151EbEF6C7334FAD080c5704D77216b732 [Authorized as central authority in the smart contract])



Home / ... / Add Beneficiary

Level of Governance
Central

From Entity
Central

To Entity
Andhra Pradesh

Select Resources

☑ dal — Kg 100000

☑ rice

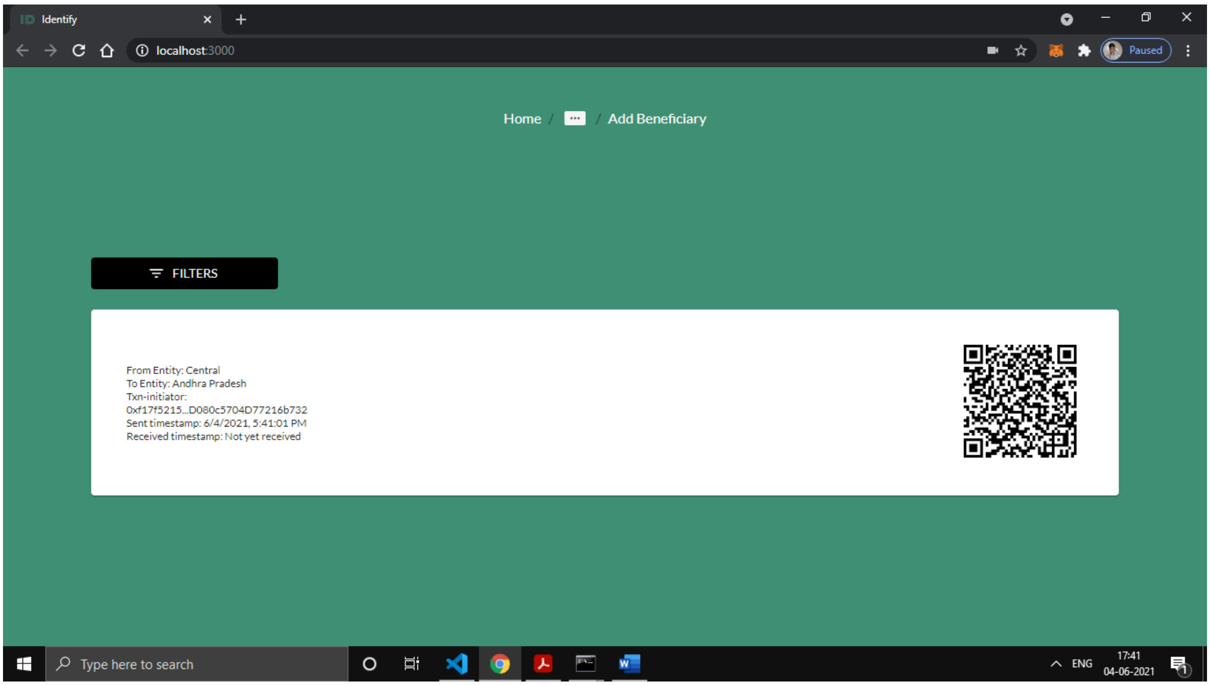
□ sugar

Txn successful!

Txnid: 0x99794cb0...8d9a61beed COPY TXNID

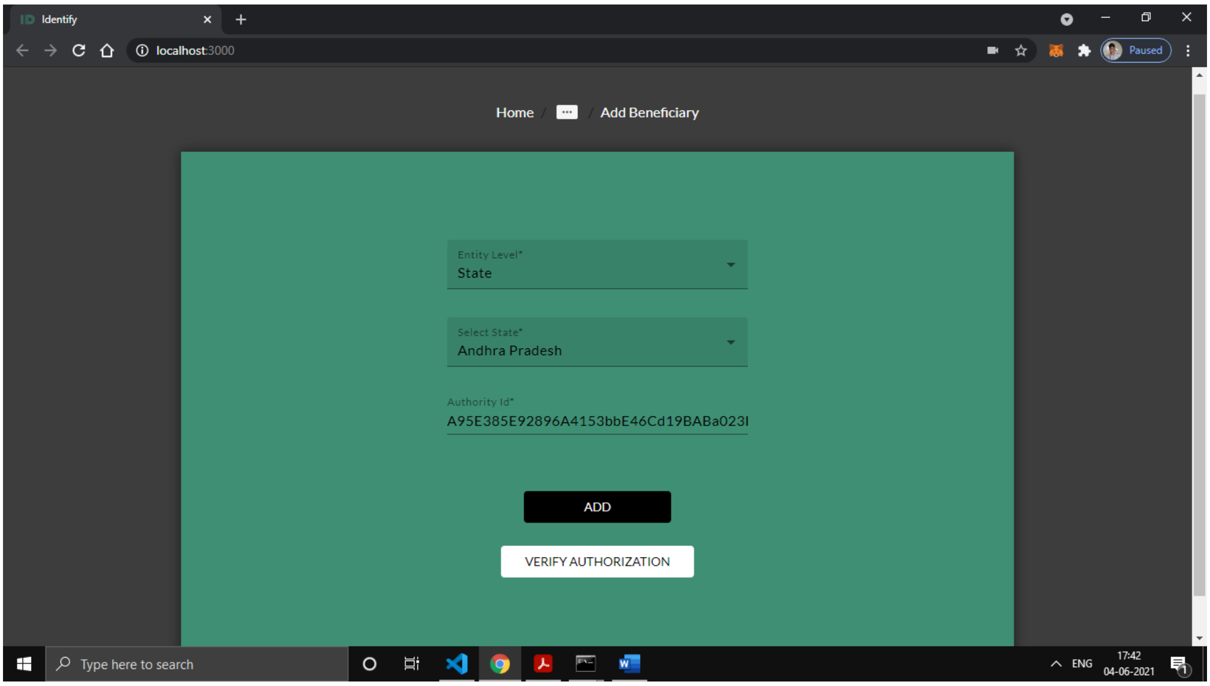
Fig. 5.35 Image showing successful resource-transfer-initiation transaction

txnId: 0x99794cb0d697ab249b6d0feabff2e200bde97b059f13c77f53a5f28d9a61beed



From Entity: Central
To Entity: Andhra Pradesh
Txn-initiator:
0xf17f5215...D080c5704D77216b732
Sent timestamp: 6/4/2021, 5:41:01 PM
Received timestamp: Not yet received

Fig. 5.36 Result: transaction list with the transaction b/n central and Andhra Pradesh

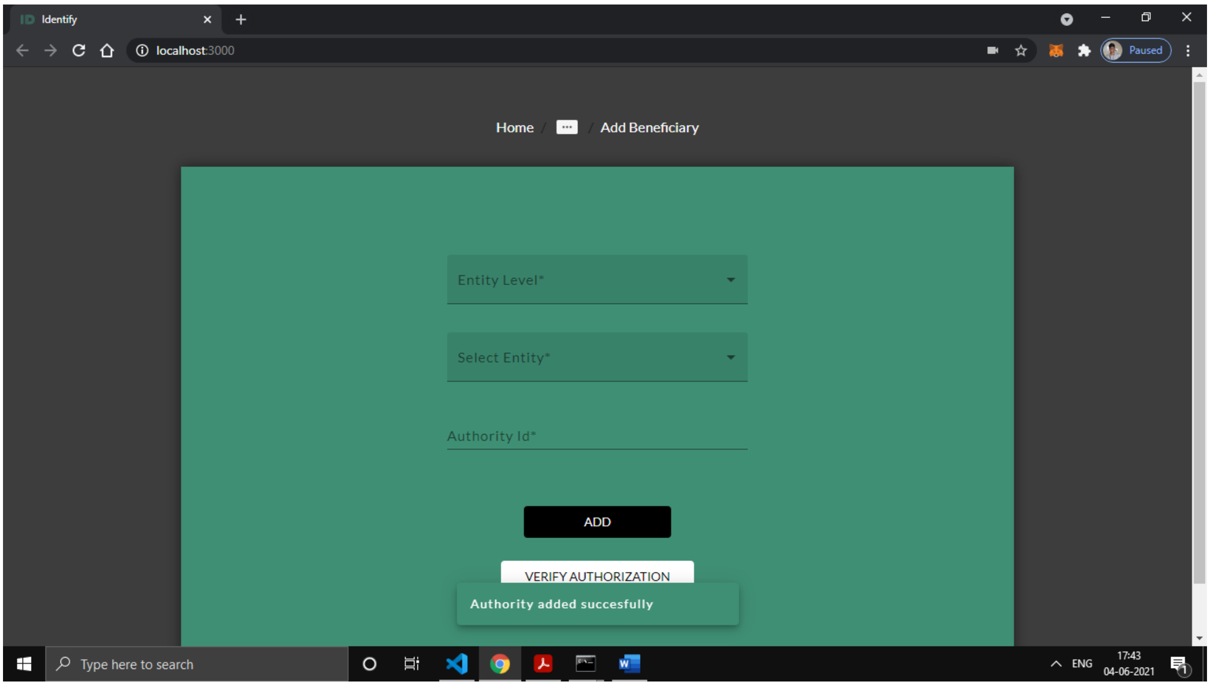


Entity Level*
State

Select State*
Andhra Pradesh

Authority Id*
A95E385E92896A4153bbE46Cd19BABa023I

Fig. 5.37 Assigning authority (0xA95E385E92896A4153bbE46Cd19BABa023Bd9A99) to Andhra Pradesh



Authority added succesfully

Fig. 5.38 Authority assignment successful for Andhra Pradesh

(0xA95E385E92896A4153bbE46Cd19BABa023Bd9A99) – Authority id for Andhra Pradesh

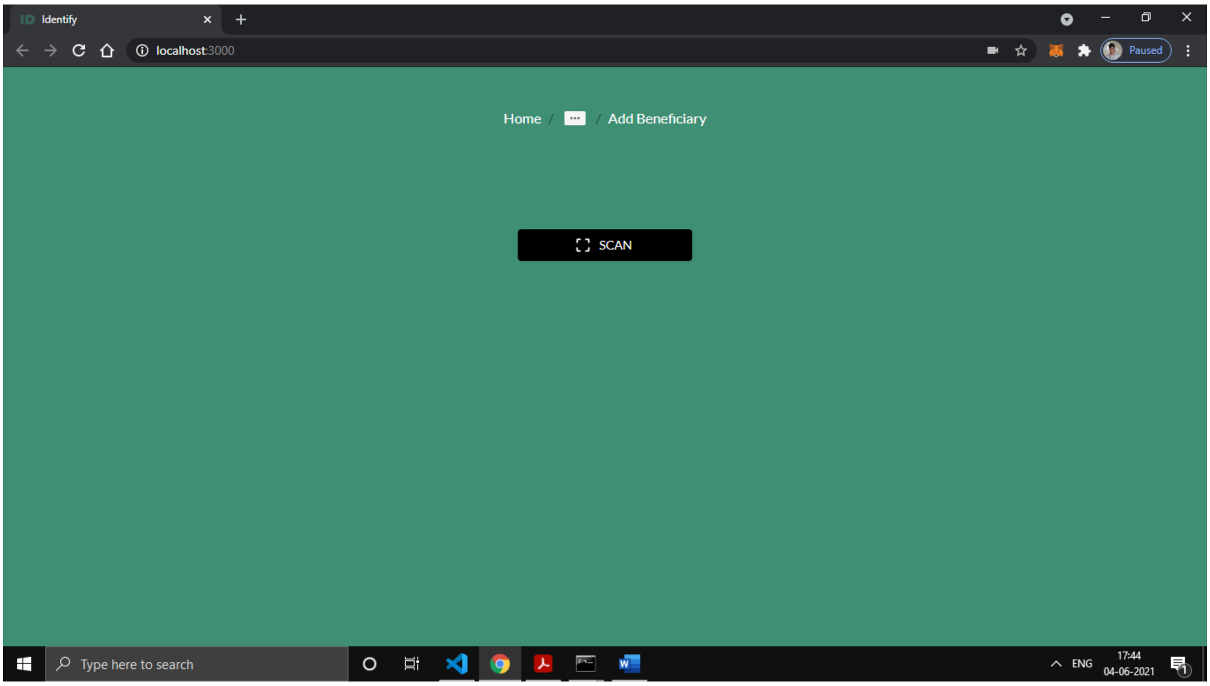


Fig. 5.39 Interface used to acknowledge the receipt of resources using QR codes.

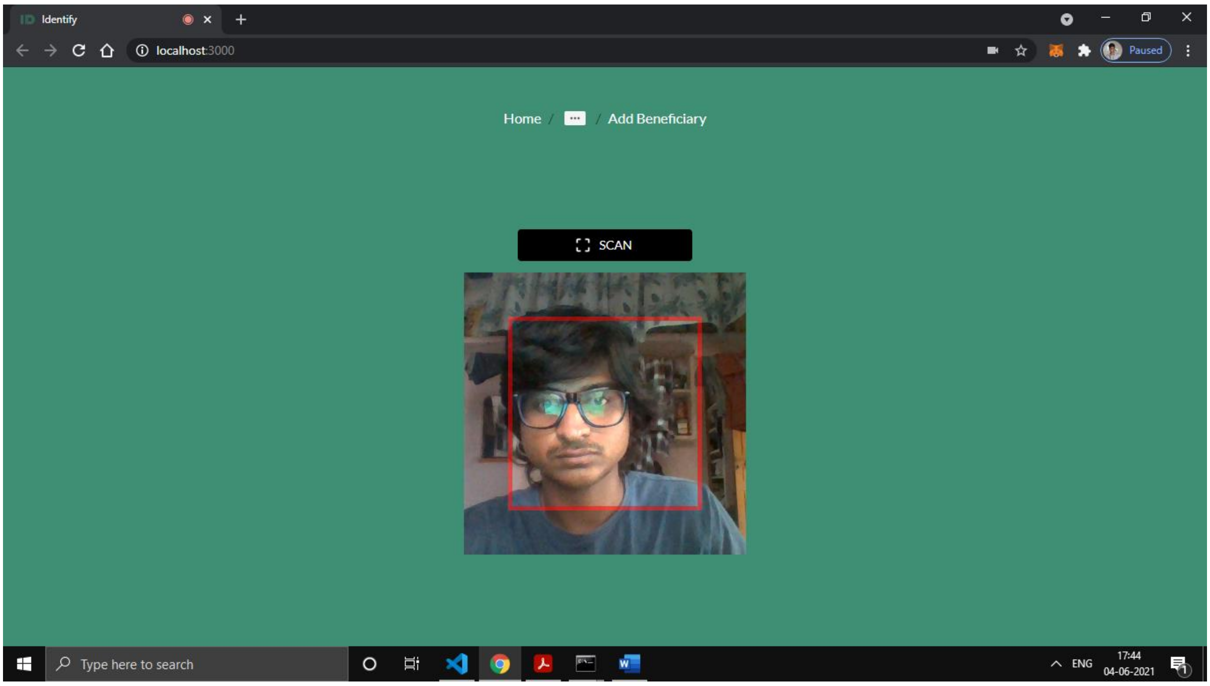
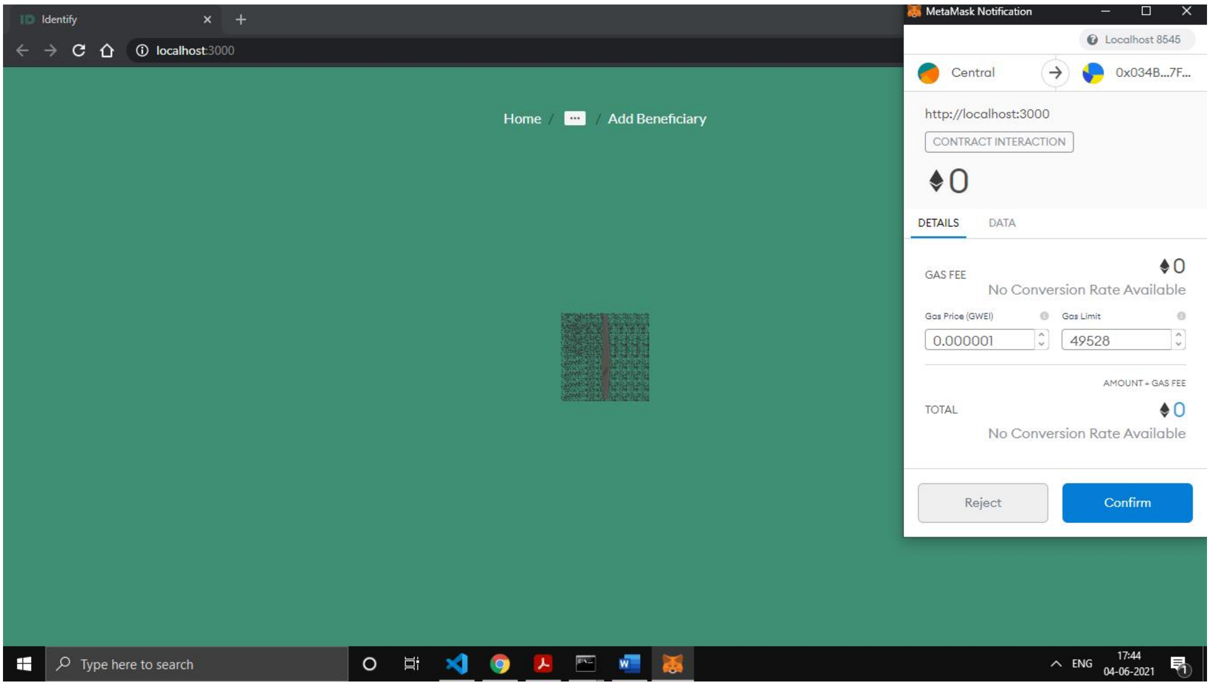


Fig. 5.40 Image showing an active QR reader



Localhost 8545

Central → 0x034B...7F...

http://localhost:3000

CONTRACT INTERACTION

0

DETAILS DATA

GAS FEE 0

No Conversion Rate Available

Gas Price (GWEI) 0.000001

Gas Limit 49528

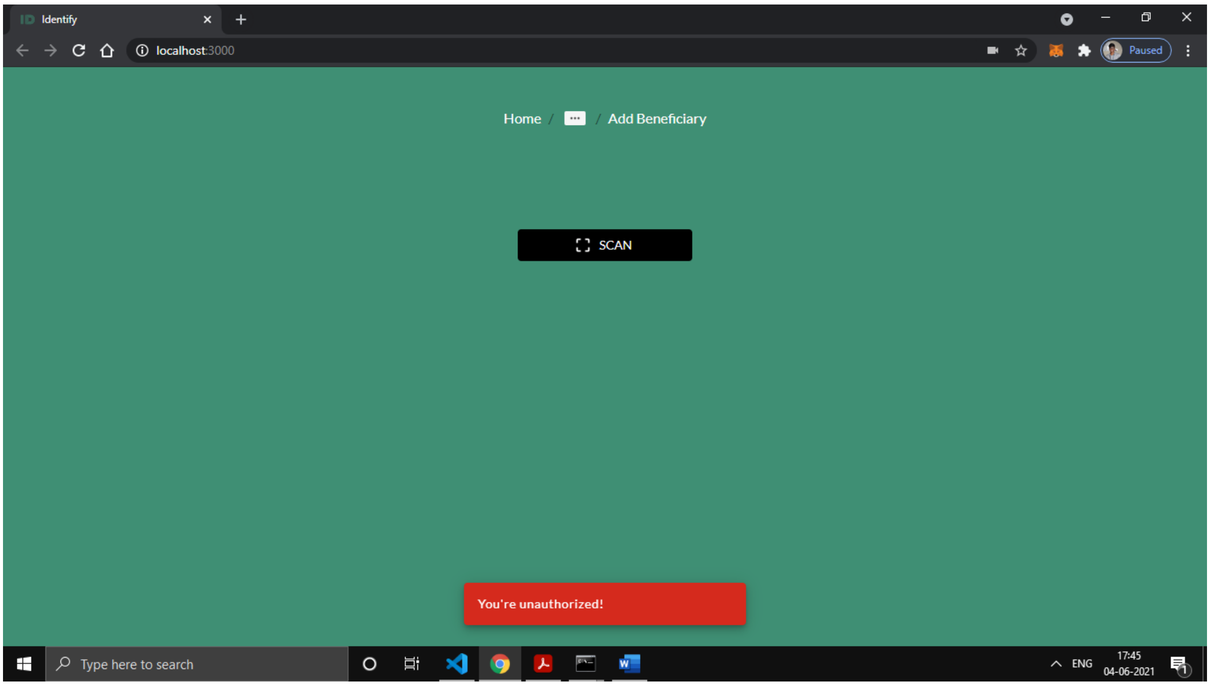
AMOUNT + GAS FEE

TOTAL 0

No Conversion Rate Available

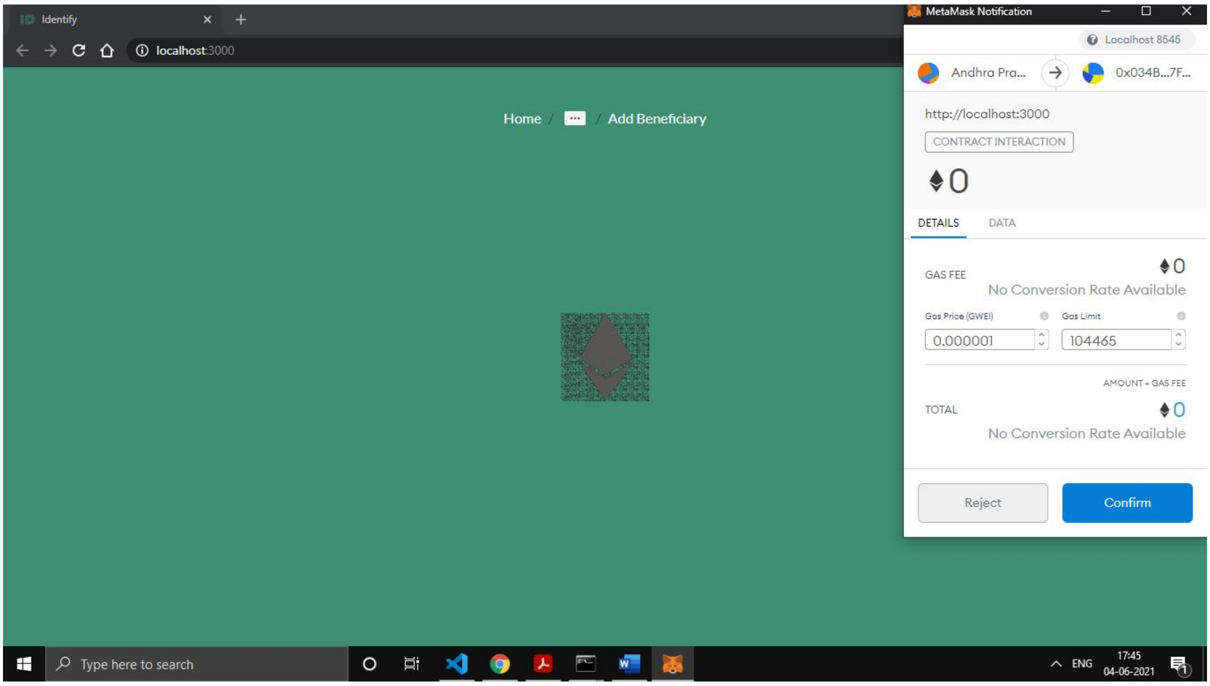
Reject Confirm

Fig. 5.41 Metamask prompt to sign an acknowledgment txn (Currently selected account: f17f52151EbEF6C7334FAD080c5704D77216b732 {Unauthorized for acknowledging since the receipt is supposed to be Andhra Pradesh})



You're unauthorized!

Fig. 5.42 Image showing response of an unauthorized transaction



http://localhost:3000

CONTRACT INTERACTION

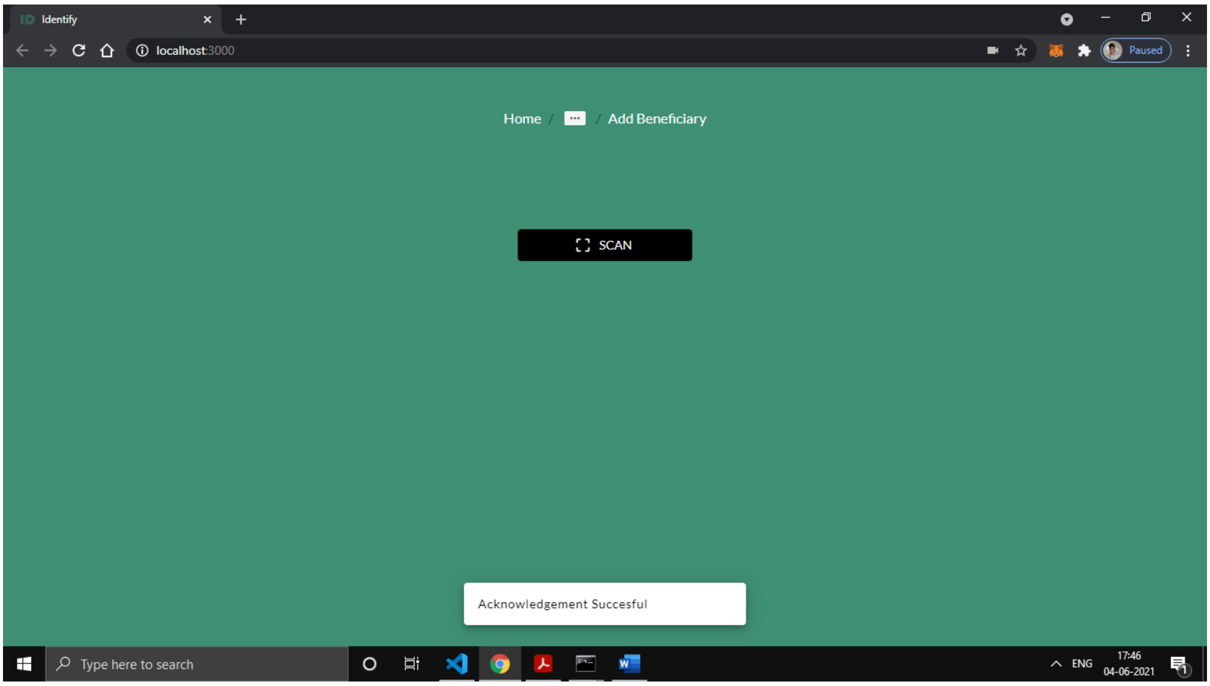
♦0

GAS FEE ♦0
No Conversion Rate Available

Gas Price (GWEI): 0.000001
Gas Limit: 104465

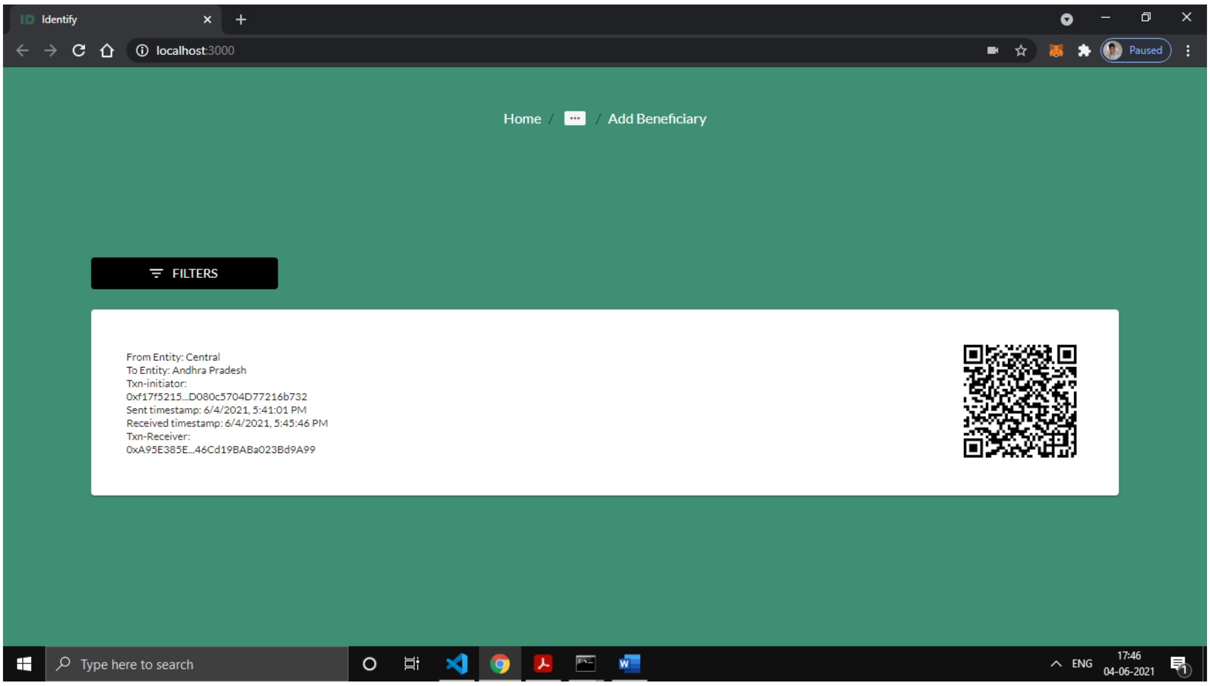
AMOUNT + GAS FEE
TOTAL ♦0
No Conversion Rate Available

Fig 5.43 Metamask prompt to sign the receipt acknowledgement (Currently selected account: 0xA95E385E92896A4153bbE46Cd19BABa023Bd9A99 {Authorized})



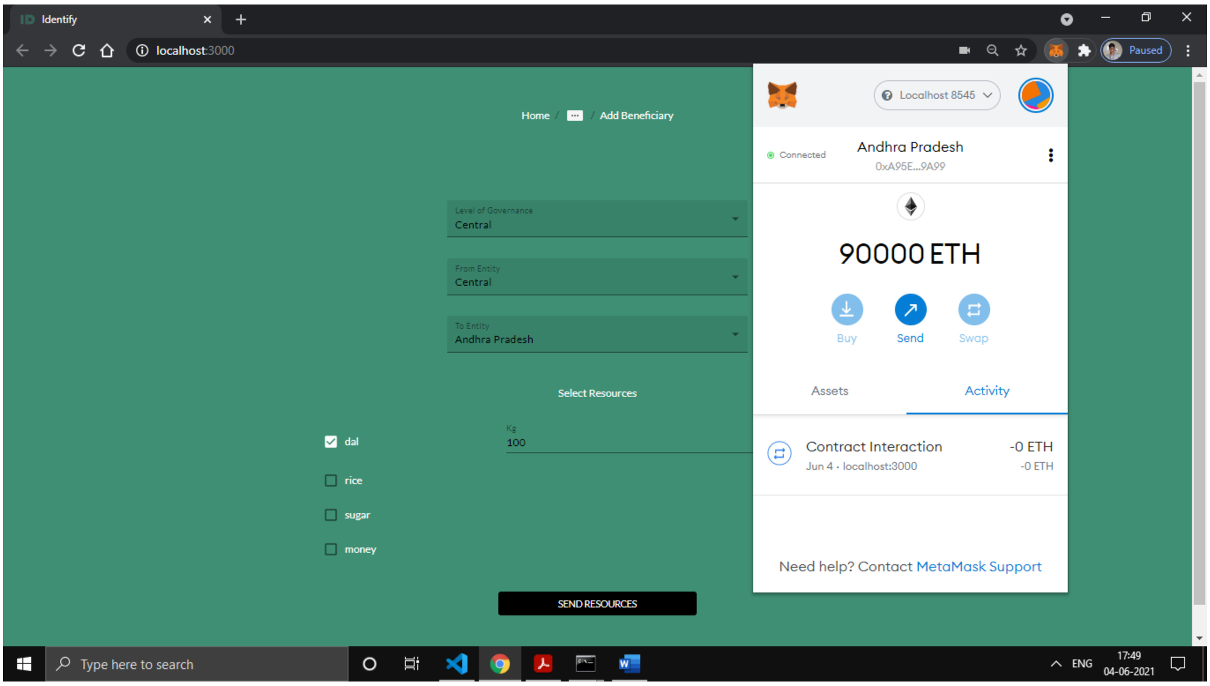
Acknowledgement Succesful

Fig 5.44 Image showing response of a successful receipt acknowledgment



From Entity: Central
To Entity: Andhra Pradesh
Txn-initiator:
0xf17f5215...D080c5704D77216b732
Sent timestamp: 6/4/2021, 5:41:01 PM
Received timestamp: 6/4/2021, 5:45:46 PM
Txn-Receiver:
0xA95E385E...46Cd19BABa023Bd9A99

Fig. 5.45 Transaction list with updated receiver details for transaction b/n Central & Andhra Pradesh



Level of Governance
Central

From Entity
Central

To Entity
Andhra Pradesh

Select Resources

☑ dal Kg 100
□ rice
□ sugar
□ money

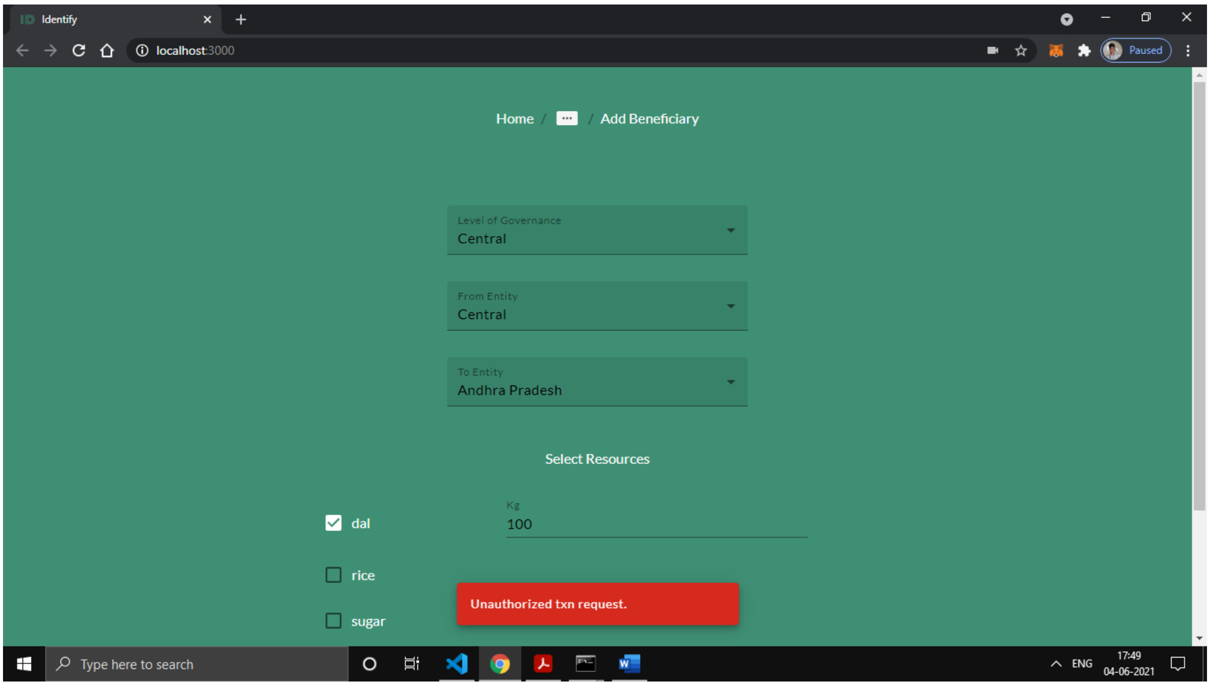
Connected
Andhra Pradesh
0xA95E...9A99

90000 ETH

Contract Interaction -0 ETH
Jun 4 · localhost:3000 -0 ETH

Need help? Contact MetaMask Support

Fig. 5.46 Image showing resource transfer transaction from Central to Andhra Pradesh with selected account: 0xA95E385E92896A4153bbE46Cd19BABa023Bd9A99 (Unauthorized to perform the transaction as the authorized authority should be of Central entity)



Level of Governance
Central

From Entity
Central

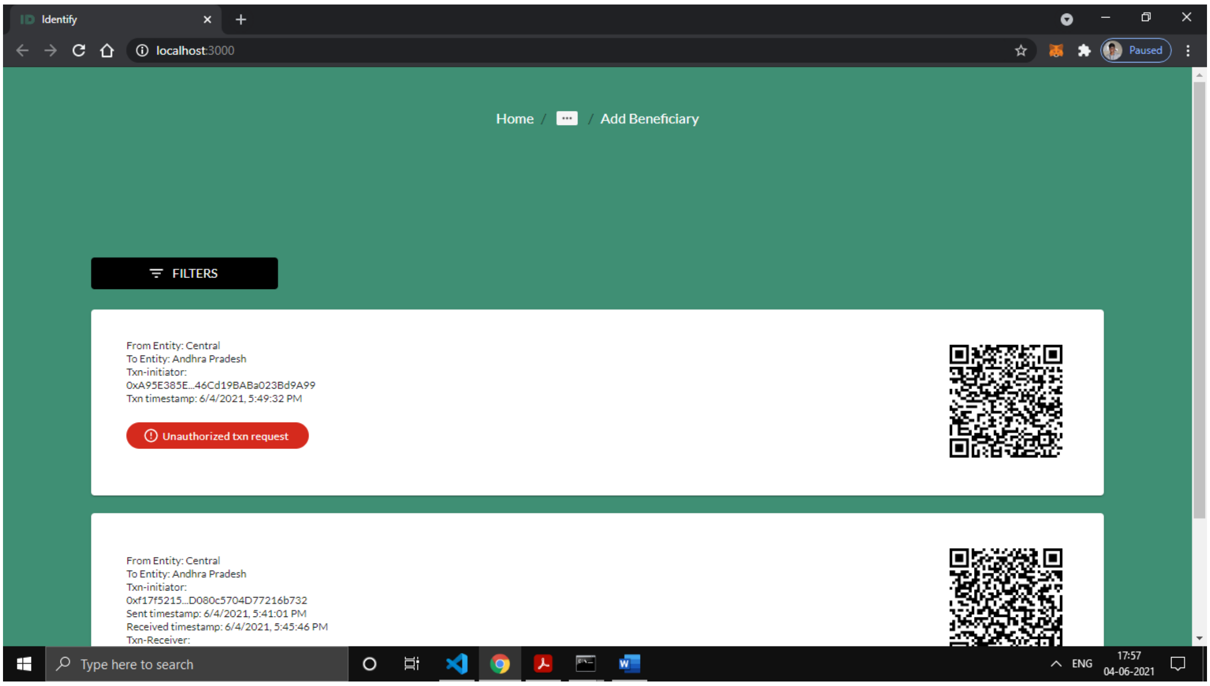
To Entity
Andhra Pradesh

Select Resources

☑ dal Kg 100
□ rice
□ sugar

Unauthorized txn request.

Fig. 5.47 Response of an unauthorized resource transaction request

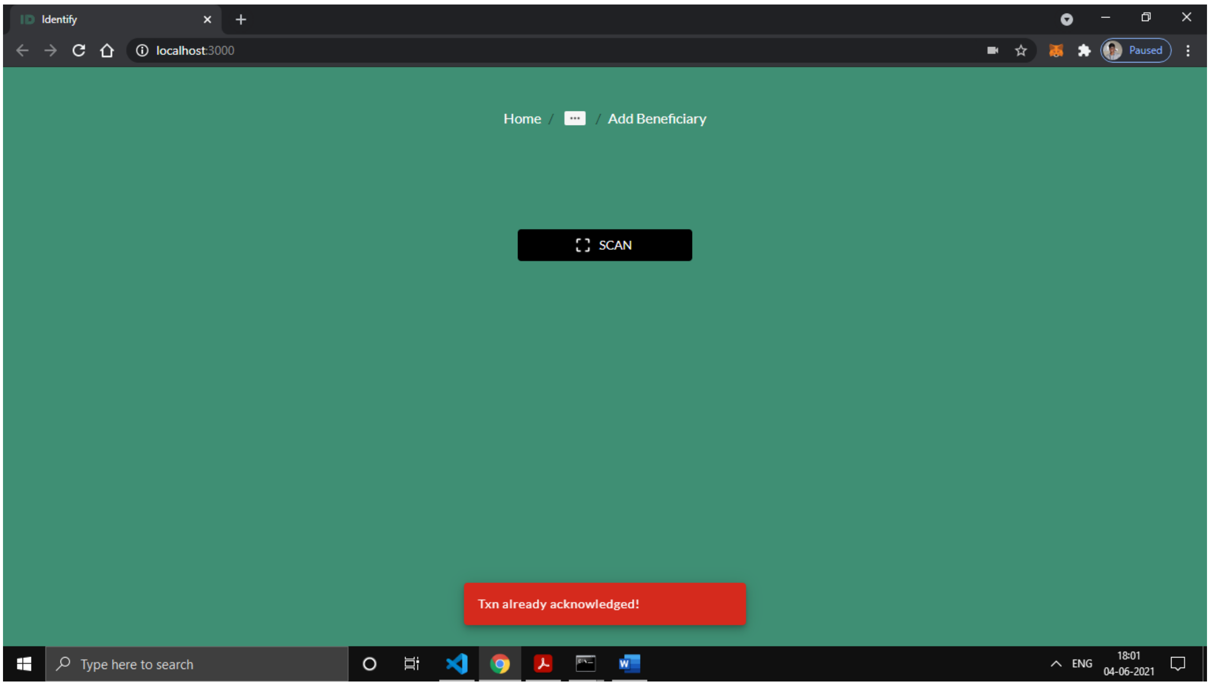


From Entity: Central
To Entity: Andhra Pradesh
Txn-initiator:
0xA95E385E...46Cd19BABa023Bd9A99
Txn timestamp: 6/4/2021, 5:49:32 PM

Unauthorized txn request

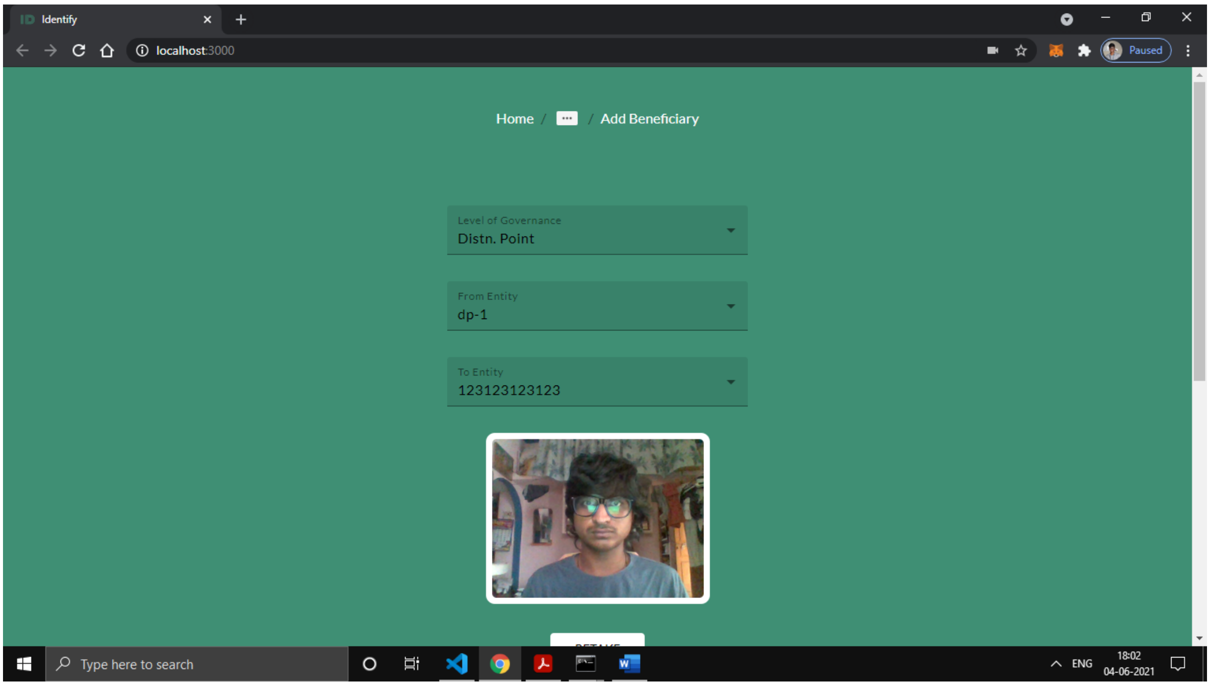
From Entity: Central
To Entity: Andhra Pradesh
Txn-initiator:
0xf17f5215...D080c5704D77216b732
Sent timestamp: 6/4/2021, 5:41:01 PM
Received timestamp: 6/4/2021, 5:45:46 PM
Txn-Receiver:

Fig. 5.48 Transaction list showing Unauthorized transaction request



Txn already acknowledged!

Fig. 5.49 Result: Response of acknowledging an already acknowledged transaction

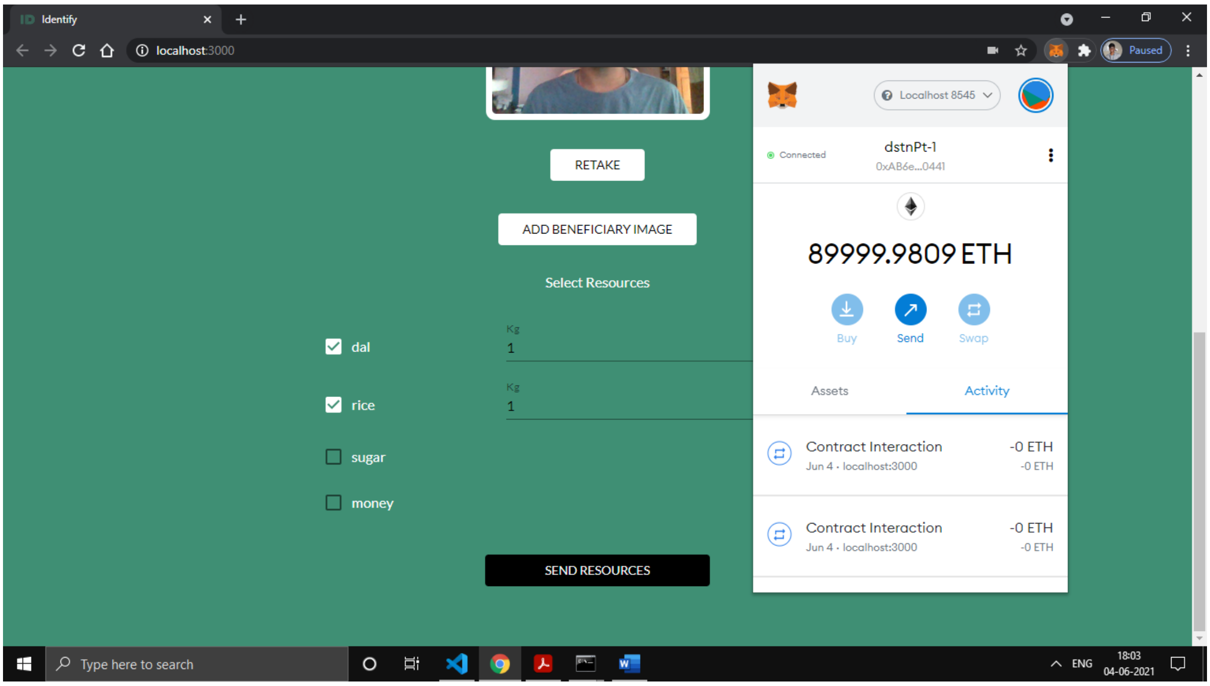


Level of Governance
Distn. Point

From Entity
dp-1

To Entity
123123123123

Fig. 5.50 Resource transfer at distribution point-1 with beneficiary image (id:123123123123) captured



☑ dal Kg 1

☑ rice Kg 1

□ sugar

□ money

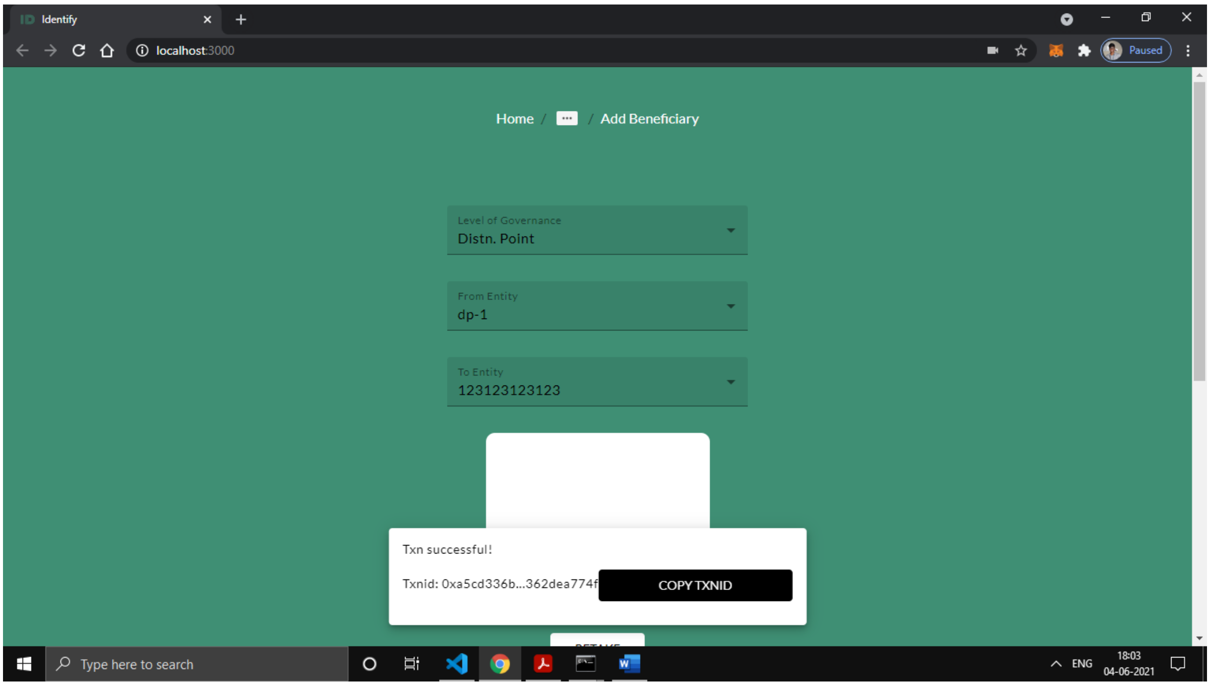
Connected
dstnPt-1
0xAB6e...0441

89999.9809 ETH

Contract Interaction -0 ETH
Jun 4 · localhost:3000 -0 ETH

Contract Interaction -0 ETH
Jun 4 · localhost:3000 -0 ETH

Fig. 5.51 Resource transfer at distribution point with beneficiary image captured (Selected account: 0xAB6e2c81675bb30411054F915e04393FA9D20441)



Level of Governance
Distn. Point

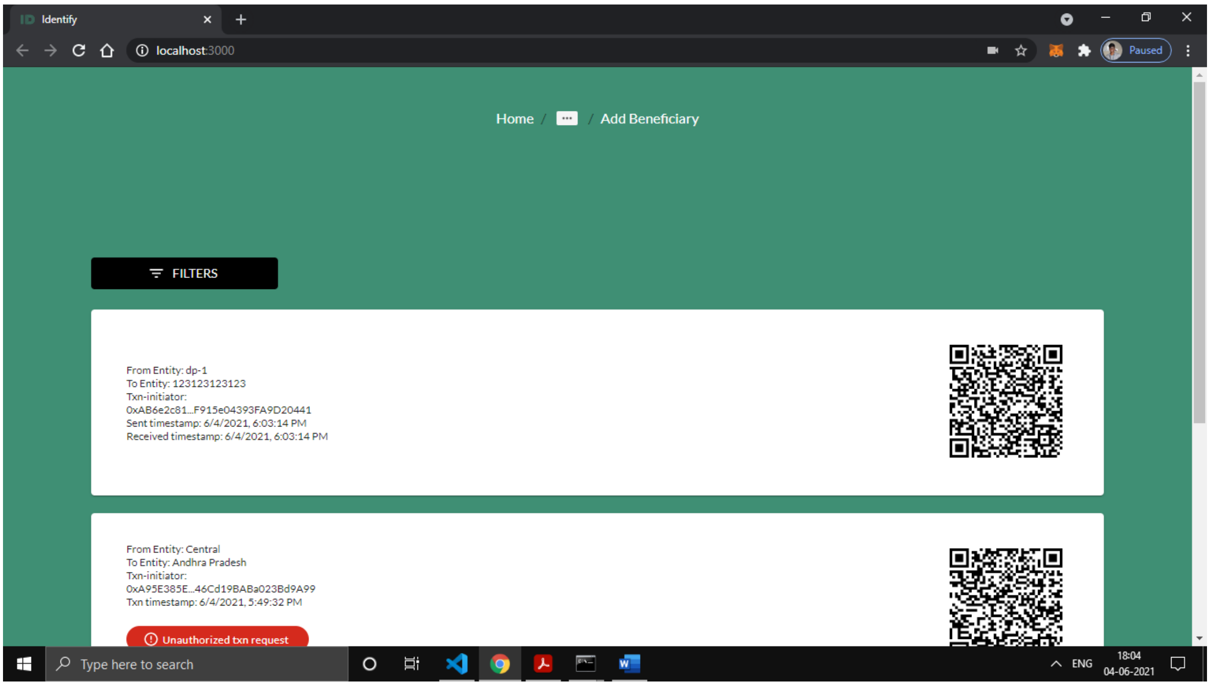
From Entity
dp-1

To Entity
123123123123

Txn successful!

Txnid: 0xa5cd336b...362dea774f

Fig. 5.52 Result: Successful transaction at distribution point-1 implying a successful beneficiary authentication (Facial-image-recognition-based authentication). Transaction Id: 0xa5cd336b5b532c68c603b2047c3683a149aedef58b14348b823f31362dea774f

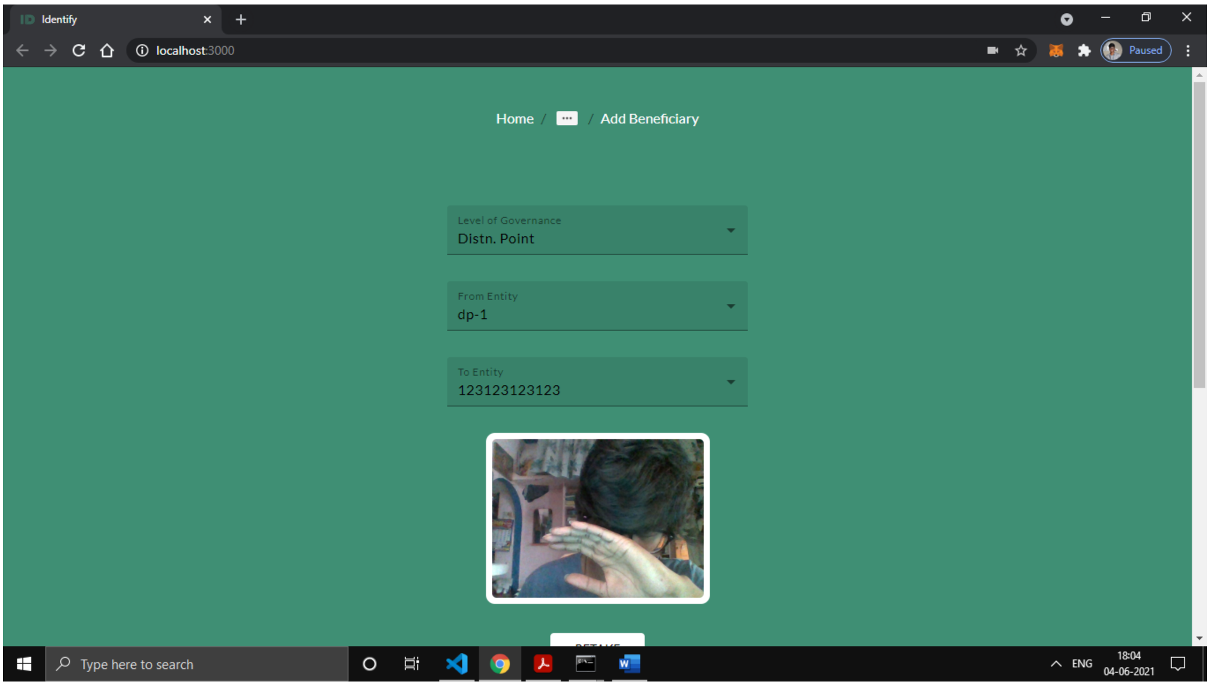


From Entity: dp-1
To Entity: 123123123123
Txn-initiator:
0xAB6e2c81...F915e04393FA9D20441
Sent timestamp: 6/4/2021, 6:03:14 PM
Received timestamp: 6/4/2021, 6:03:14 PM

From Entity: Central
To Entity: Andhra Pradesh
Txn-initiator:
0xA95E385E...46Cd19BABa023Bd9A99
Txn timestamp: 6/4/2021, 5:49:32 PM

Unauthorized txn request

Fig 5.53 Transaction list with transaction b/n distribution point-1 and beneficiary (id:123123123123)

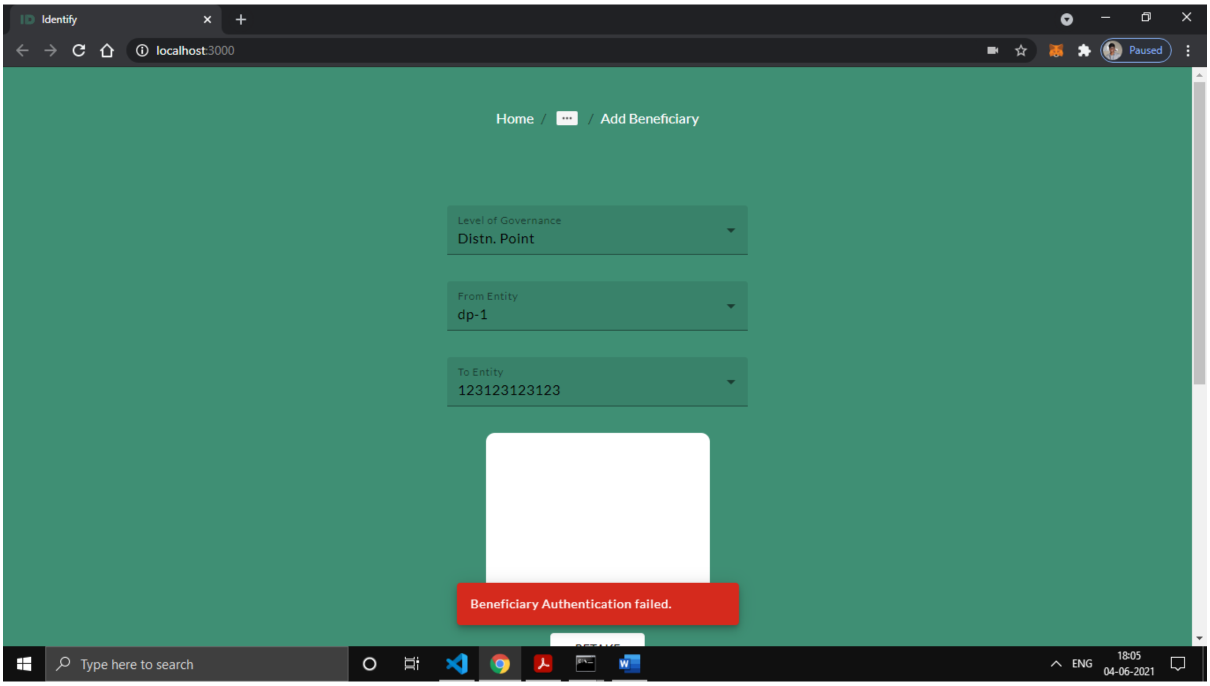


Level of Governance
Distn. Point

From Entity
dp-1

To Entity
123123123123

Fig. 5.54 Distribution point transaction with invalid beneficiary image



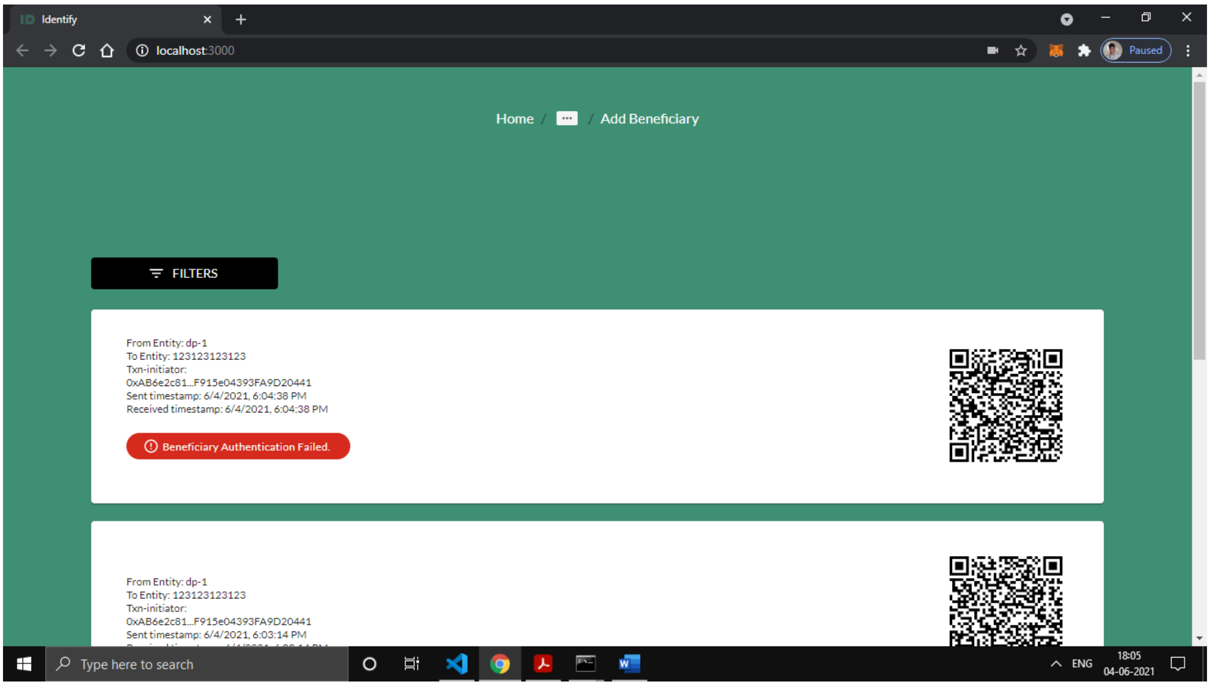
Level of Governance
Distn. Point

From Entity
dp-1

To Entity
123123123123

Beneficiary Authentication failed.

Fig. 5.55 Result: Image showing failed beneficiary authentication

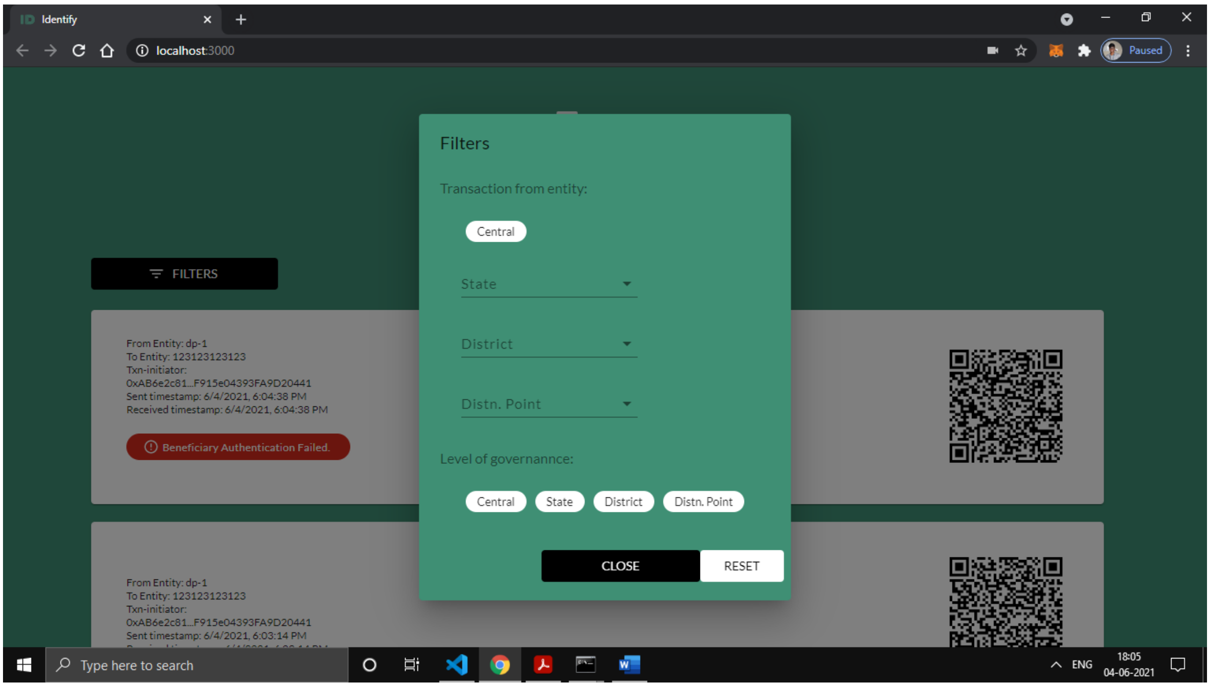


From Entity: dp-1
To Entity: 123123123123
Txn-initiator:
0xAB6e2c81...F915e04393FA9D20441
Sent timestamp: 6/4/2021, 6:04:38 PM
Received timestamp: 6/4/2021, 6:04:38 PM

Beneficiary Authentication Failed.

From Entity: dp-1
To Entity: 123123123123
Txn-initiator:
0xAB6e2c81...F915e04393FA9D20441
Sent timestamp: 6/4/2021, 6:03:14 PM

Fig. 5.56 Transaction result showing unauthenticated beneficiary transaction



Filters

Transaction from entity:

Central

State

District

Distn. Point

Level of governannce:

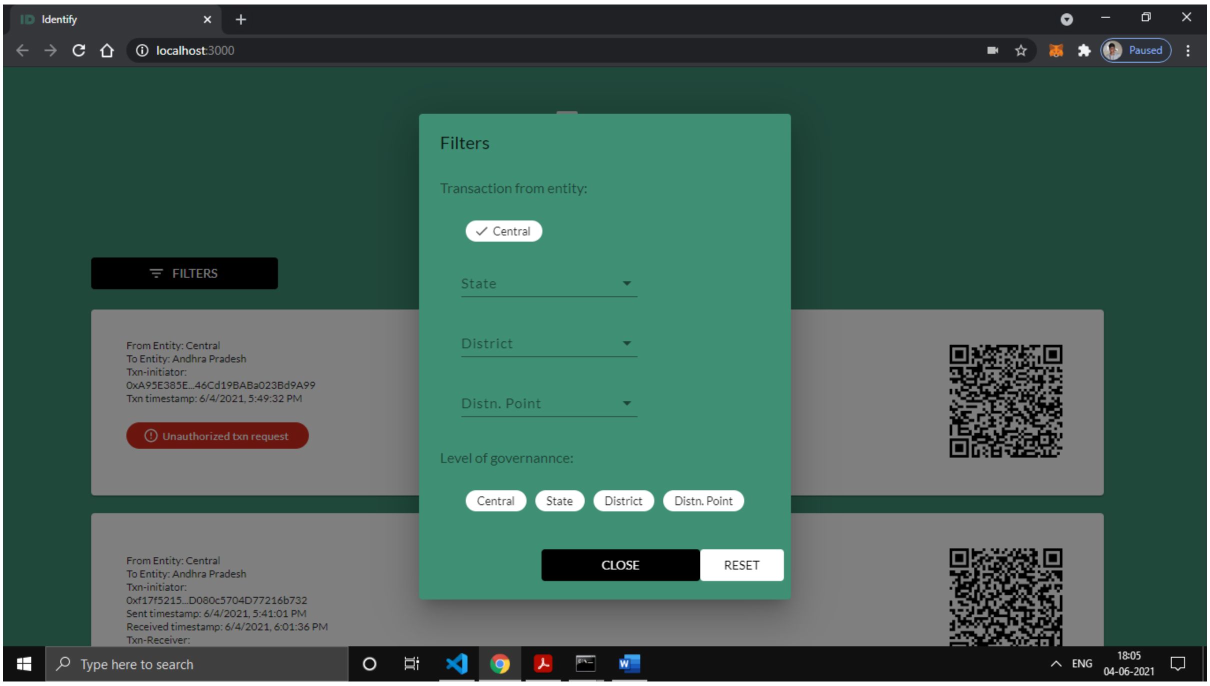
Central State District Distn. Point

From Entity: dp-1
To Entity: 123123123123
Txn-initiator:
0xAB6e2c81...F915e04393FA9D20441
Sent timestamp: 6/4/2021, 6:04:38 PM
Received timestamp: 6/4/2021, 6:04:38 PM

Beneficiary Authentication Failed.

From Entity: dp-1
To Entity: 123123123123
Txn-initiator:
0xAB6e2c81...F915e04393FA9D20441
Sent timestamp: 6/4/2021, 6:03:14 PM

Fig. 5.57 Transaction filter interface



Filters

Transaction from entity:

✓ Central

State

District

Distn. Point

Level of governannce:

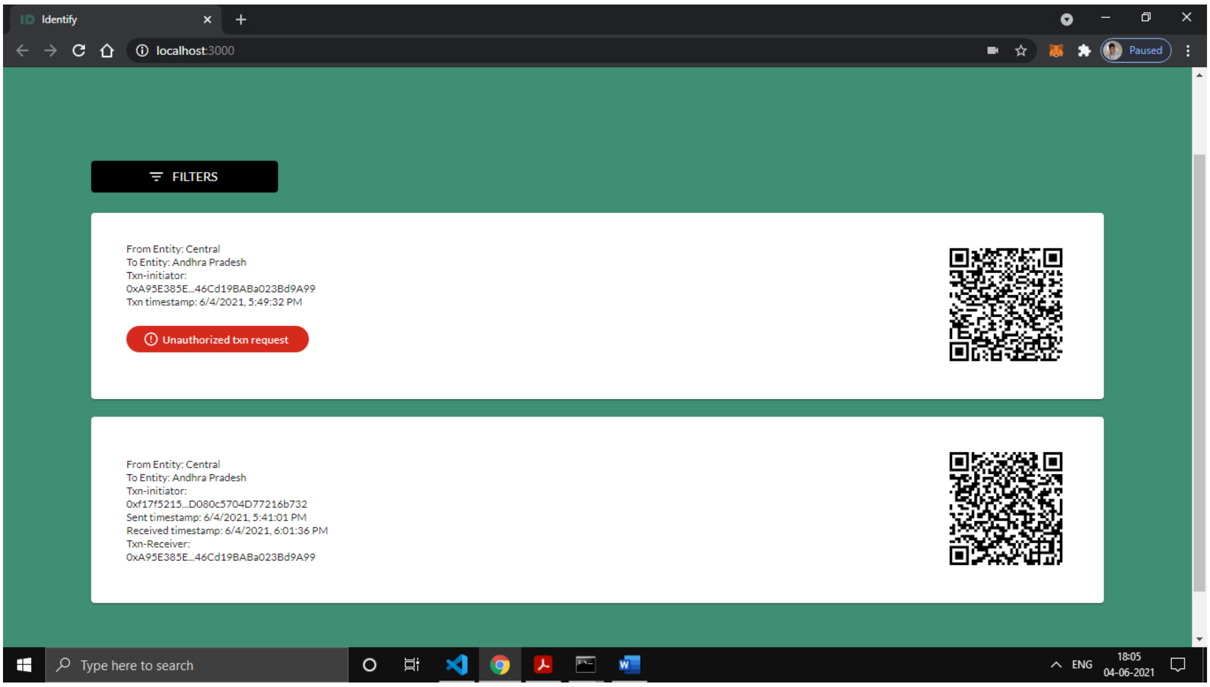
Central State District Distn. Point

From Entity: Central
To Entity: Andhra Pradesh
Txn-initiator:
0xA95E385E...46Cd19BABa023Bd9A99
Txn timestamp: 6/4/2021, 5:49:32 PM

Unauthorized txn request

From Entity: Central
To Entity: Andhra Pradesh
Txn-initiator:
0xf17f5215...D080c5704D77216b732
Sent timestamp: 6/4/2021, 5:41:01 PM
Received timestamp: 6/4/2021, 6:01:36 PM
Txn-Receiver:

Fig. 5.58 Transaction filter applied: {From Entity: central}

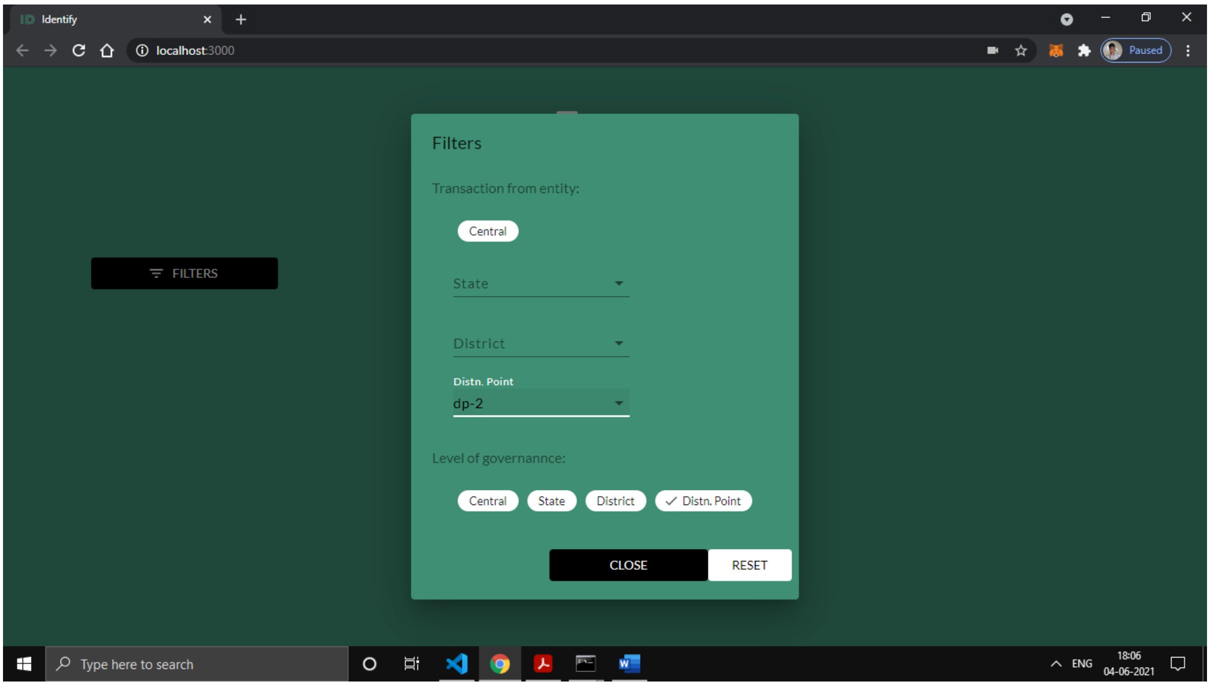


From Entity: Central
To Entity: Andhra Pradesh
Txn-initiator:
0xA95E385E...46Cd19BABa023Bd9A99
Txn timestamp: 6/4/2021, 5:49:32 PM

Unauthorized txn request

From Entity: Central
To Entity: Andhra Pradesh
Txn-initiator:
0xf17f5215...D080c5704D77216b732
Sent timestamp: 6/4/2021, 5:41:01 PM
Received timestamp: 6/4/2021, 6:01:36 PM
Txn-Receiver:
0xA95E385E...46Cd19BABa023Bd9A99

Fig.5.59 Transactions from Central entity



Filters

Transaction from entity:

Central

State

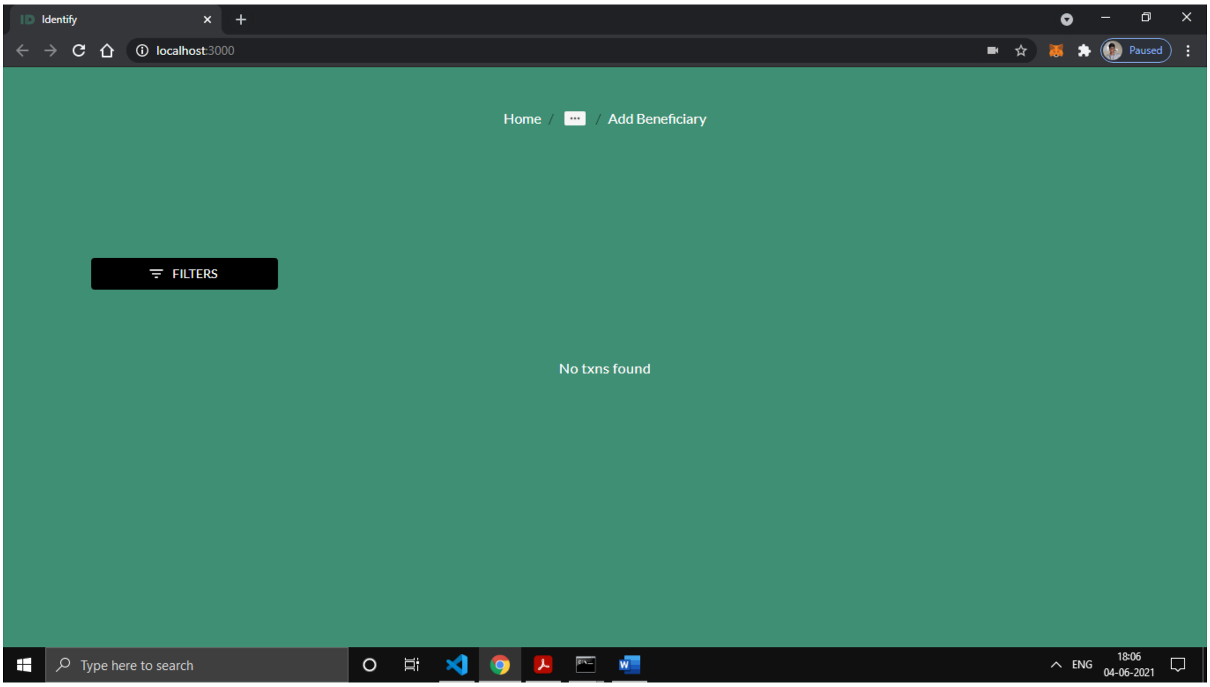
District

Distn. Point
dp-2

Level of governannce:

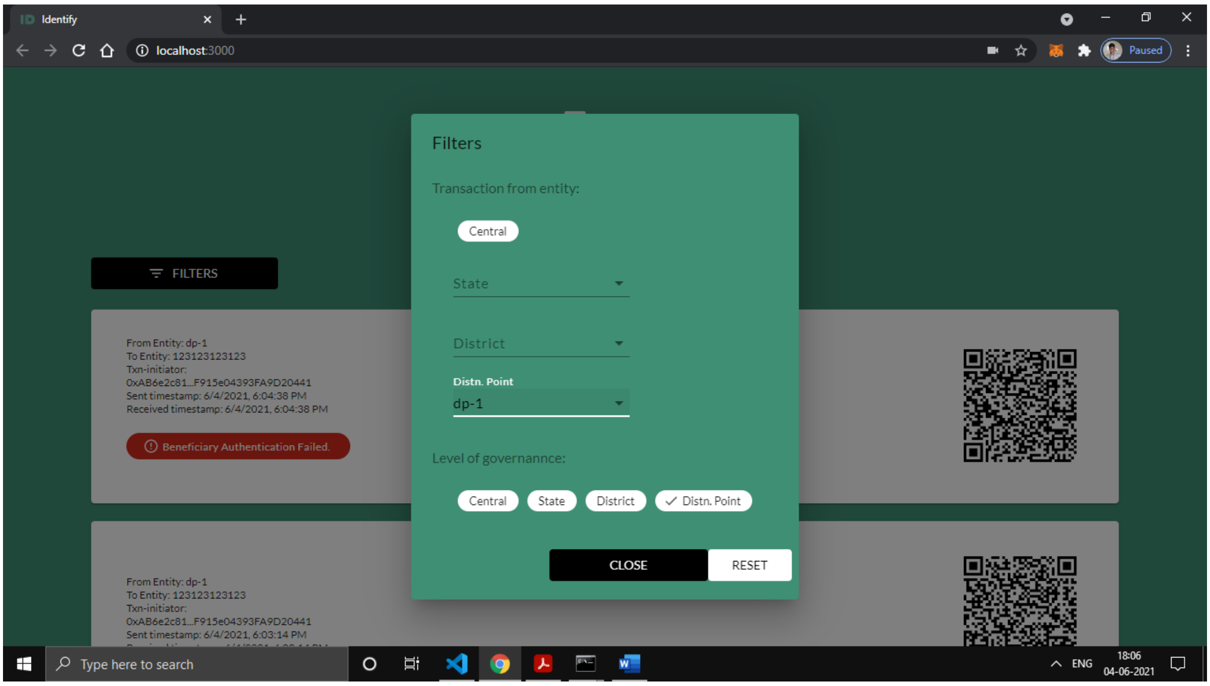
Central State District ✓ Distn. Point

Fig. 5.60 Transaction filter applied {From entity: dp-2 (distribution point-2)}



No txns found

Fig. 5.61 Transactions by distribution point-2



Filters

Transaction from entity:

Central

State

District

Distn. Point
dp-1

Level of governannce:

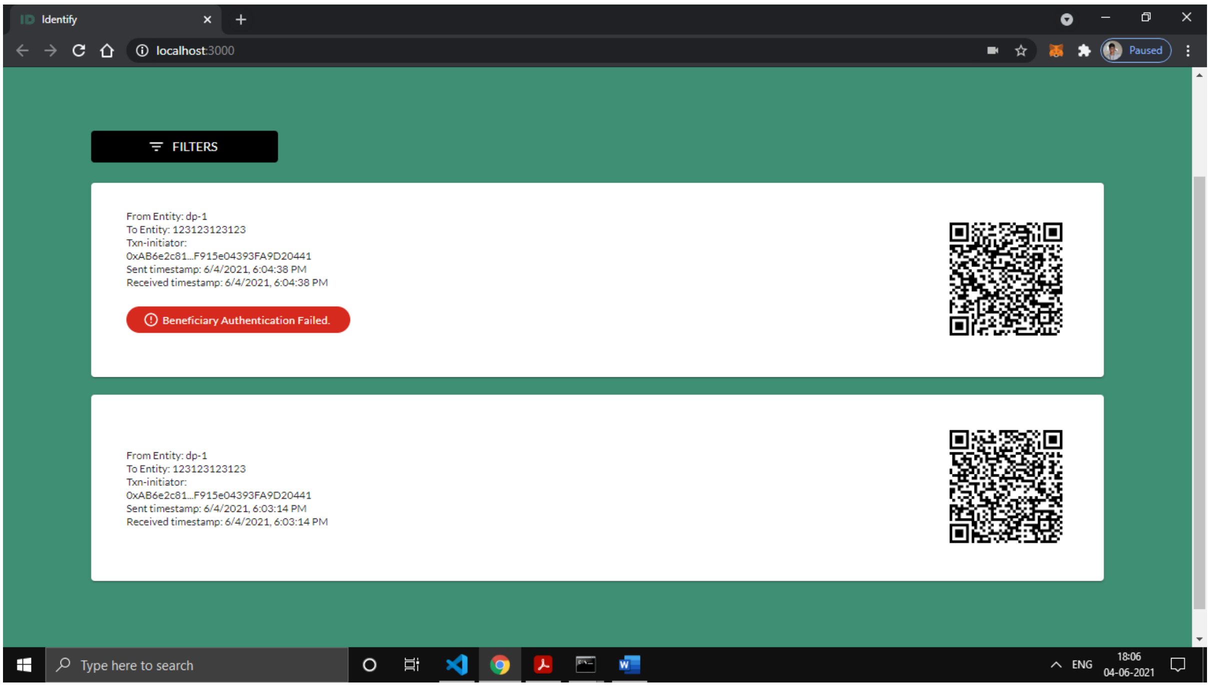
Central State District ✓ Distn. Point

From Entity: dp-1
To Entity: 123123123123
Txn-initiator:
0xAB6e2c81...F915e04393FA9D20441
Sent timestamp: 6/4/2021, 6:04:38 PM
Received timestamp: 6/4/2021, 6:04:38 PM

Beneficiary Authentication Failed.

From Entity: dp-1
To Entity: 123123123123
Txn-initiator:
0xAB6e2c81...F915e04393FA9D20441
Sent timestamp: 6/4/2021, 6:03:14 PM

Fig. 5.62 Transaction filter applied {From entity: dp-1 (distribution point-1)} and level selected as Distribution point level

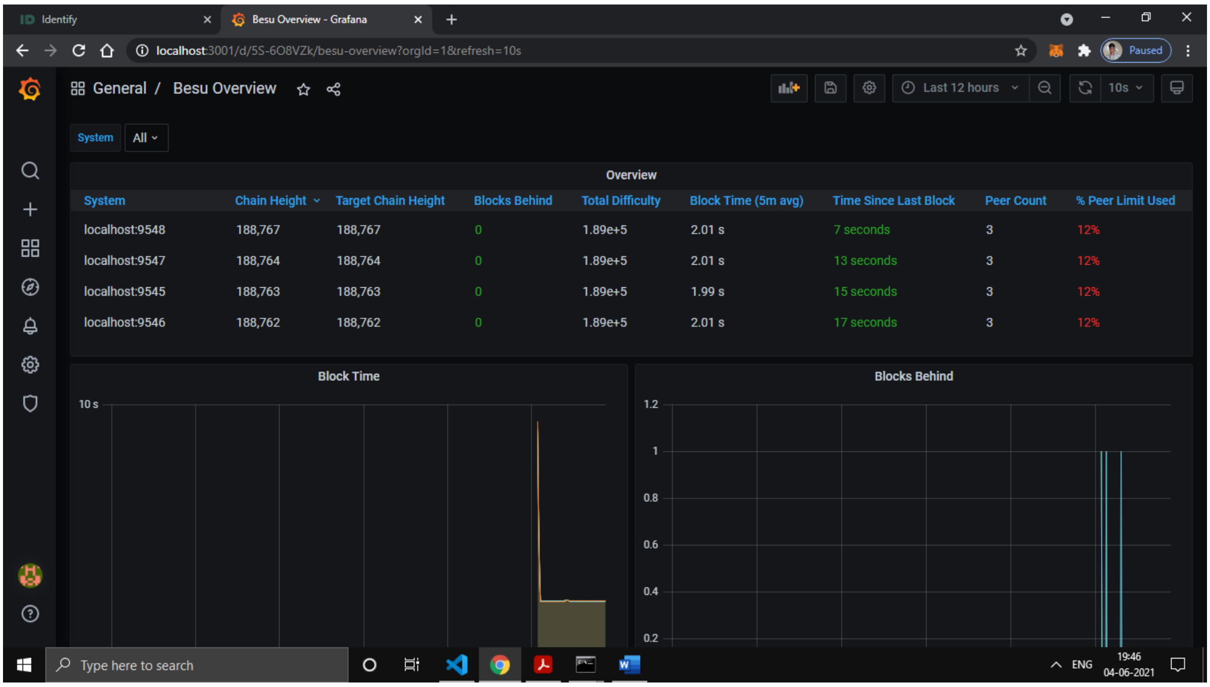


From Entity: dp-1
To Entity: 123123123123
Txn-initiator:
0xAB6e2c81...F915e04393FA9D20441
Sent timestamp: 6/4/2021, 6:04:38 PM
Received timestamp: 6/4/2021, 6:04:38 PM

Beneficiary Authentication Failed.

From Entity: dp-1
To Entity: 123123123123
Txn-initiator:
0xAB6e2c81...F915e04393FA9D20441
Sent timestamp: 6/4/2021, 6:03:14 PM
Received timestamp: 6/4/2021, 6:03:14 PM

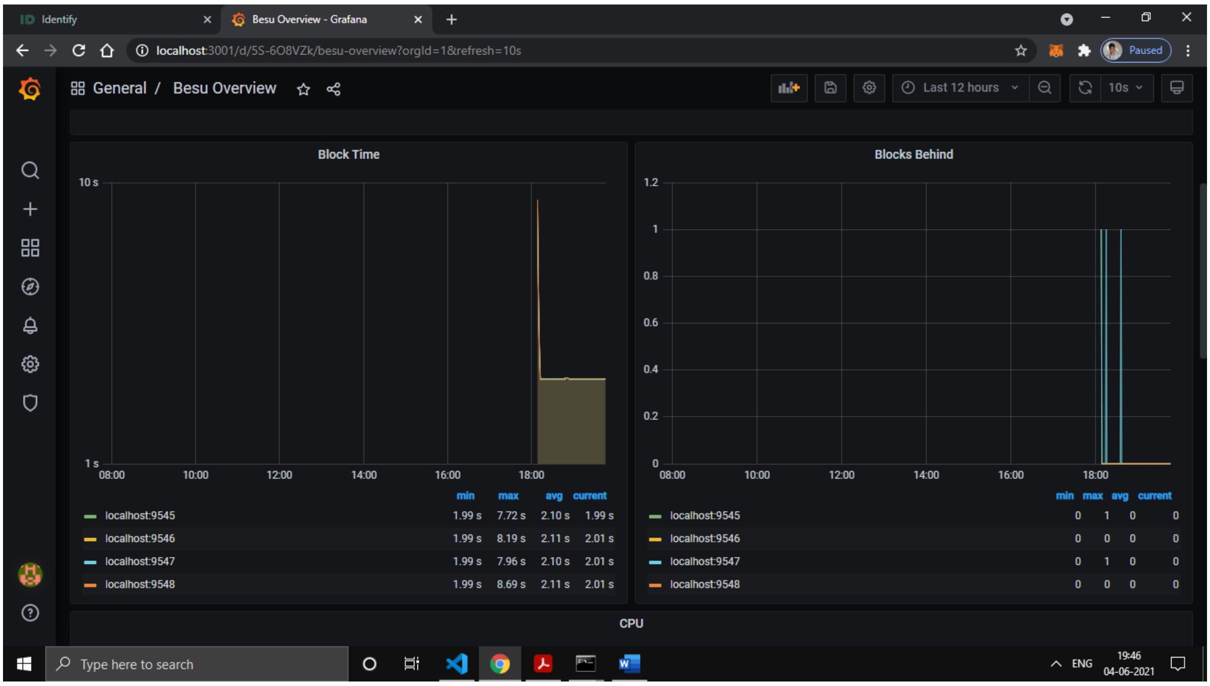
Fig. 5.63 Transactions at distribution level



Overview

System	Chain Height	Target Chain Height	Blocks Behind	Total Difficulty	Block Time (5m avg)	Time Since Last Block	Peer Count	% Peer Limit Used
localhost:9548	188,767	188,767	0	1.89e+5	2.01 s	7 seconds	3	12%
localhost:9547	188,764	188,764	0	1.89e+5	2.01 s	13 seconds	3	12%
localhost:9545	188,763	188,763	0	1.89e+5	1.99 s	15 seconds	3	12%
localhost:9546	188,762	188,762	0	1.89e+5	2.01 s	17 seconds	3	12%

Fig. 5.64 Grafana Dashboard visualizing node metrics polled by data source Prometheus



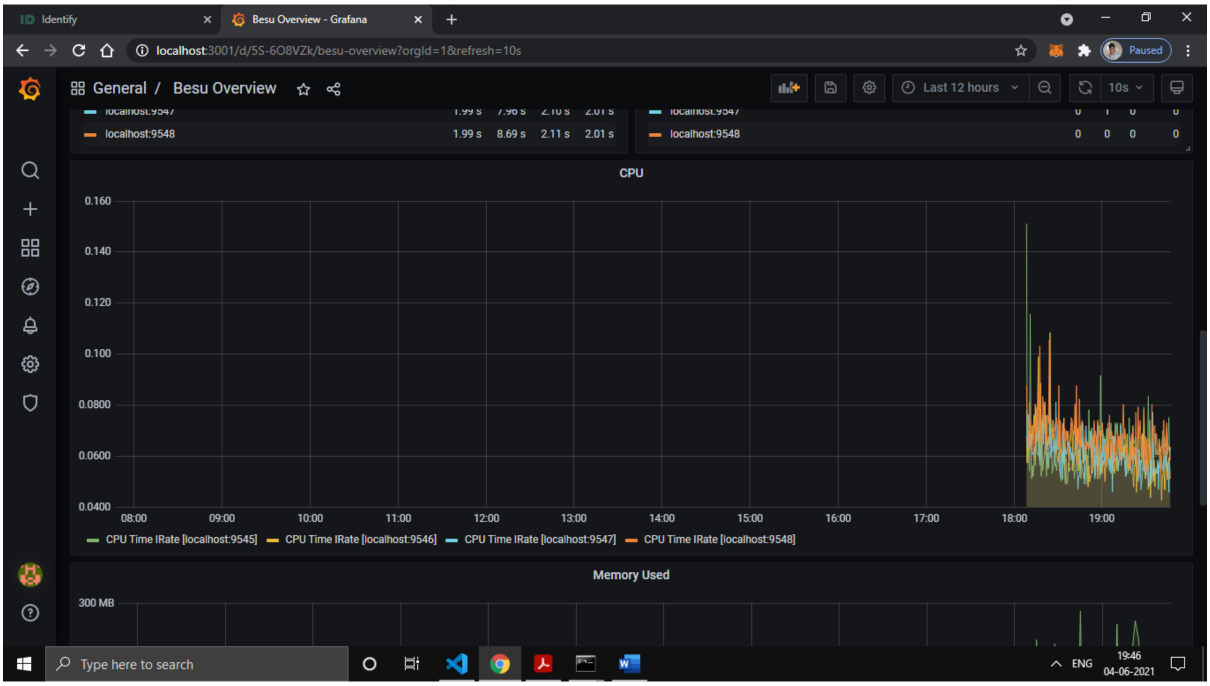
Block Time

	min	max	avg	current
localhost:9545	1.99 s	7.72 s	2.10 s	1.99 s
localhost:9546	1.99 s	8.19 s	2.11 s	2.01 s
localhost:9547	1.99 s	7.96 s	2.10 s	2.01 s
localhost:9548	1.99 s	8.69 s	2.11 s	2.01 s

Blocks Behind

	min	max	avg	current
localhost:9545	0	1	0	0
localhost:9546	0	0	0	0
localhost:9547	0	1	0	0
localhost:9548	0	0	0	0

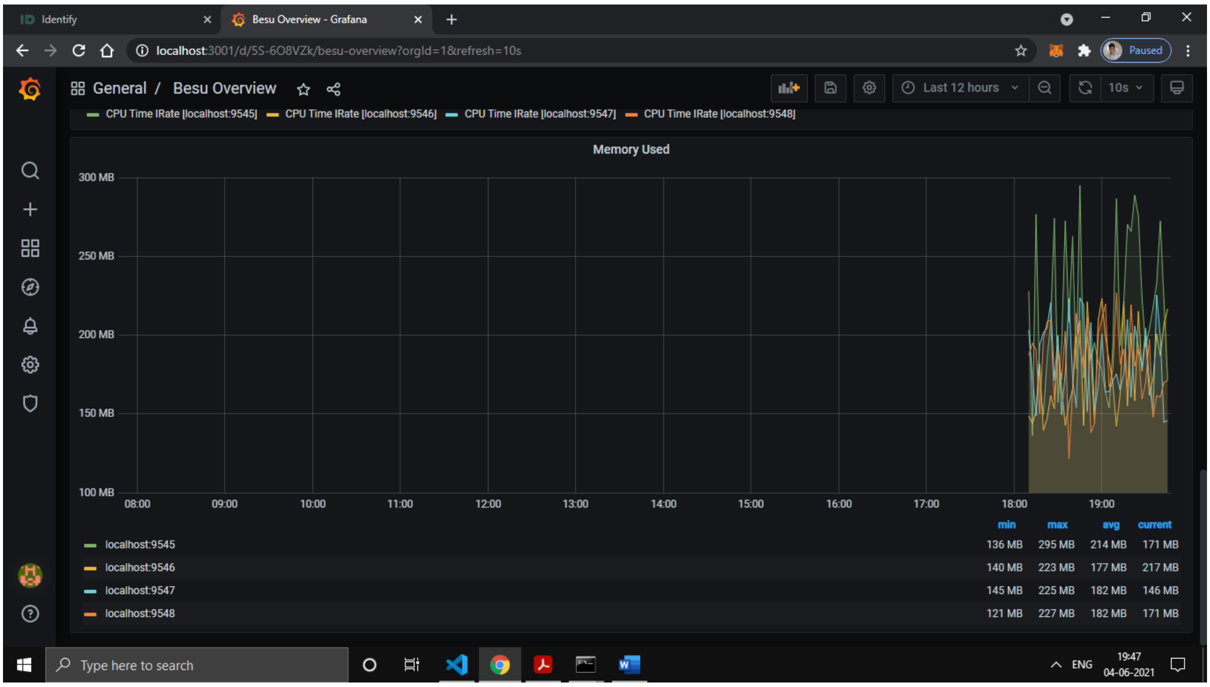
Fig 5.65 Graphical data showing the block time of each node and the number of blocks each Besu node is behind the main chain



localhost:9548	1.99 s	8.69 s	2.11 s	2.01 s

localhost:9548	0	0	0	0

Fig. 5.66 Graphical representation of CPU utilization by each Besu node



	min	max	avg	current
localhost:9545	136 MB	295 MB	214 MB	171 MB
localhost:9546	140 MB	223 MB	177 MB	217 MB
localhost:9547	145 MB	225 MB	182 MB	146 MB
localhost:9548	121 MB	227 MB	182 MB	171 MB

Fig. 5.67 Graphical representation showing RAM utilization by each Besu node

6 REFERENCES

[1] Joshi, P., Kumar, S., Kumar, D. and Singh, A.K., 2019, September. A blockchain based framework for fraud detection. In *2019 Conference on Next Generation Computing Applications (NextComp)* (pp. 1-5). IEEE. https://doi.org/10.1109/NEXTCOMP.2019.8883647

[2] Parameswari, C.D. and Mandadi, V., 2021. Public Distribution System Based on Blockchain Using Solidity. In *Innovative Data Communication Technologies and Application* (pp. 175-183). Springer, Singapore. https://link.springer.com/chapter/10.1007/978-981-15-9651-3_15

[3] Casino, F., Kanakaris, V., Dasaklis, T.K., Moschuris, S. and Rachaniotis, N.P., 2019. Modeling food supply chain traceability based on blockchain technology. *Ifac-Papersonline*, *52*(13), pp.2728-2733. https://doi.org/10.1016/j.ifacol.2019.11.620

[4] B. Saiharsha, R. S. Sanjay Tharagesh, S. V. Nikish Kumar and N. Harini, "Cloud Based Data Integrity model for Minimal Human Intervention using Smart Contracts," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 910-916. https://doi.org/10.1109/ICCES48766.2020.9137939

[5] Sachan, Karan. Aadhaar & Blockchain: opportunities and challenges for India. Diss. Massachusetts Institute of Technology, 2018. https://dspace.mit.edu/bitstream/handle/1721.1/118523/1054910734-MIT.pdf?sequence=1

[6] Singh, Sandeep Kumar, et al. "A conceptual model for Indian public distribution system using consortium blockchain with on-chain and off-chain trusted data." Information Technology for Development (2020): 1-25.

[7] R. Chatterjee and R. Chatterjee, "An Overview of the Emerging Technology: Blockchain," 2017 3rd International Conference on Computational Intelligence and Networks (CINE), Odisha, India, 2017, pp. 126-127, doi: 10.1109/CINE.2017.33.

[8] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han and F. Wang, "Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 11, pp. 2266-2277, Nov. 2019, doi: 10.1109/TSMC.2019.2895123.

Weblinks:

[9] https://www.indiatoday.in/india/story/up-midday-meal-scam-exposed-workers-were-selling-children-s-food-for-cash-1600288-2019-09-18

[10] https://docs.soliditylang.org/en/v0.8.3/

[11] https://web3js.readthedocs.io/en/v1.3.4/

[12] https://reactjs.org/

[13] https://www.hyperledger.org/use/besu