International Scientific Journal of Engineering and Management
Volume: 02 Issue: 05 | May – 2023
DOI: 10.55041/ISJEM00748
ISSN: 2583-6129
www.isjem.com
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Transparent Charity Application using Blockchain

Bhagyashree, Kishan Kumar, Sanjay Umesh Bangera, Shreyas
A J Institute of Engineering and Technology, Mangaluru

Prof. Prabhakara B. K
(Project Guide)
Associate Professor, Department of Information Science and Engineering,
A J Institute of Engineering and Technology, Mangaluru
Email: prabhakarabk@ajiet.edu.in

Abstract- People are growing more eager than ever to give back to society. Many people wish to give freely to the causes they like, but they frequently fail due to their lack of confidence in the system and they end up doing nothing. There are numerous NGOs and philanthropic organizations that try to improve society and occasionally need money. There are countless potential uses for blockchain technology. The distributed ledger technology known as blockchain makes it possible for decentralized peer-to-peer networks to process digital assets. All transactions are gathered into a block. Excellent characteristics of the blockchain include immutability, decentralization, security, transparency, and anonymity. We have implemented Blockchain Technology in our system to provide extra security and also increase the donors trust to donate to the Charities/NGOs. There are four different categories of users, including the Government, Finance Officer, Charities/NGOs, and Users (Donors). Our goal is to make the donors to donate to the Charities/NGOs. Donations details are visible to everyone in this world and this provides transparency. With this information anyone in this world can question the receiver on how did they utilize the money that was donated by "abc" donator. For a charitable donation, we are employing blockchain to increase transparency. This helps resolve the trust issues, as people already know what they are paying for and the system will help to solve the problem.

Index Terms- Blockchain, User, Government, Charity/NGO, Finance Officer.

I. INTRODUCTION

Donors have every reason to fear that charitable funds will not reach people who really need them. According to the same HSE survey in 2017, 68% of citizens are willing to donate more if there is evidence of where and what they are going. By law, foundations are required to maintain public records (in particular, to publish reports on their websites), and now all reports are prepared by employees of a foundation manually, so it is very difficult to maintain the records. It is also very difficult to find the activities. It is a Long- time process. It takes more time to prepare various events within a short time. The biggest disadvantage of most NGOs, that is they are notable to scale up their success. NGOs have many workers, and the effort they put in is considerable.
But, when they succeed, it is often in a limited area. And, they cannot easily scale up. Talking about the current system, even if the donors donate to the NGOs that information is not made available publicly unless it's a large amount. Donors should be provided complete right to ask the NGOs on how did they spend the donated money. For that the donation details must be made publicly available. In a centralized system there will be one person who will control the system and can also access the donation details of the users. This does not provide privacy for the users. Also, it is easier to tamper the database. Hackers can hack the database and can access the complete details of the user and might also misuse it. Because of this most of the donors do not donate in large amount. This will directly affect the growth of the NGOs. The problem of mistrust of donors and overloading of funds can be solved by organizing an external database nothing but Blockchain, where all transactions are recorded in the blockchain. Since the transaction details will be recorded in the Blockchain, donors will be provided more trust to donate to the system. This will also push them to donate in larger amount. Therefore, it is important to develop a social platform based on blockchain technology that can help non-profit organizations, foundations, volunteers, and social entrepreneurs in their work and make donation processes transparent and understandable for all parties. Blockchain will allow all users of the platform to see their account in their dashboard. Also, the technology of distributed ledger will guarantee a donor that the amount will reach the goal, and without any intermediaries.

A. Objective:

Following points are some of the objectives of our system:

- This system provides privacy for the users who created their account as only they have access to their account.
- This system provides security as the transaction data is securely stored in Blockchain.
- All transactions are recorded on the blockchain for higher security purpose.
- This system increases the public's trust as the transparency in government
- Activities could be solved technically with this blockchain charity system.

II. LITERATURE SURVEY

These days money fraud and inefficient use of money that has been donated by various donors made us think of creating a system that can provide transparency and security and this made us choose Blockchain technology. Blockchain technology has been gaining popularity in recent years due to its ability to create decentralized and secure networks. One area where this technology can be particularly useful is in charity applications. By using blockchain technology, a transparent and secure system can be created for donors to track their donations and ensure that their money is being used for the intended purpose. This can also help to reduce administrative costs and increase the speed of transactions, as blockchain technology can facilitate instant transactions without the need for intermediaries. Hence the use of blockchain technology in charity applications has the potential to increase trust and accountability, leading to greater support for charitable causes. In this literature survey, we will explore the various ways in which blockchain technology and other technology is being used in the charity sector and the potential benefits and challenges of implementing such systems.

Aashutosh Singh et al. [1] in "Aid, Charity and Donation Tracking System Using Blockchain" have briefed about the problems related to charity, where people have trust issues towards the NGO and then proposes of making use of blockchain to provide transparency in donation. This system uses smart contract-based incentives which helps in confirming and verifying the transaction without the use of third-party application and is also accessible to everyone. Hence increases the trust towards the NGOs. This system is a decentralized system running on a Ethereum blockchain. This system consists of 3 entities that is the NGO, User, Government official. Users will be having a unique 160bits account address and are the account holders. Author briefs about the different smart contract functions used in the system. They are also provided with a 256 bits private key and with the help of this they can perform transactions. The system also consists of a EVM (Ethereum Virtual Machine) and smart contractor. This allows carrying out the transaction in a transparent way. This paper also briefs us about the usage of EVM, transactions (details), signing Ethereum transactions and Ethereum networks. Front-end is in the form of a website and makes use of HTML, CSS & ReactJS and makes use of Ethers.js which is a JavaScript library used for interfacing the application with the Ethereum blockchain. Back-end is developed using solidity smart contracts that consists of Ethereum nodes connected to the network. One of the disadvantages of this paper is that the contractor entity is missing which might cause inefficient donation from the Donors.

Ajay Manohar Ingle et al. [2] in "E-Charity Platform for Social Welfare" tells that this application comprises of using left over item or food that can be donated to or used by the poor. System contains private individuals, hospitals, eatery who wants to help individuals like seniority home, halfway house. Person needs to enroll in order to give any leftovers. According to this they will send a vehicle to collect the stack and send it to the individuals or donors who can visit the donation site for a particular NGO and serve for advancement of society. Site also contain NGOs that will appreciate or acknowledge our donor's donation. Users can donate anything by choosing the necessary classification. Objective of this application is to connect donors and NGOs in their locality to spread awareness, to boost up the user's anticipation towards the awareness and reduce the to-handle users. The application here contains 2 modules namely NGO and Portal. He can like, comment on the photos of donation. User will donate in the form of money, food or materials to the cause. On the NGO portal, NGO must sign-in and upload all the events and verify the cause. Then it will be visible to the user. NGO can see how much donation is flowing and can withdraw the money. Some of the disadvantage of this system is that there is no trust that the NGOs are verified, donors will not have confidence that NGOs are not making any illicit utilization of their donation and since blockchain is not implemented for the storing of the data there are chances of it getting tampered.

Hadi Saleh et al. [3] in "Platform for Tracking Donations of Charitable Foundations based on Blockchain Technology" have briefed that the donations are made in person to the needy (via family and friends, the workplace or school, or a civil society project), and fundraising is not formally planned, permanent, or accountable. Even whether the money was provided via a bank account, the internet, or mobile communication (via SMS), the donors typically have no idea how their money was used but this system work offers best practices for platform design, REST API implementation, and architecture for blockchain initiatives for social reasons. The objective of this project is mainly to create the framework for integrating philanthropic organizations. In this system the other goals are like to facilitate the work of charity foundations by providing reporting documents and by developing a shared platform based on blockchain technology, philanthropic organizations can become more transparent. Here the project utilizes the test network for Ethereum and also uses the Web3.js standard library to communicate with the blockchain server portion. It uses Solidity-based smart contract implementation language. JavaScript-based platform Node.js and framework Express were used to develop the server component (REST API). MySQL is utilized as an off-chain data storage solution. Functions and processes have been designed for rapid database interaction. One of the disadvantages of this system is that the Contractor entity is missing which might cause inefficient donation from the Donors.

Chit Su Mon et al. [4] in "Mobile application: donate day" have briefed that the android mobile application described in this paper enables users to give their goods to charity organizations in Malaysia. The study used quantitative research to examine people's philanthropic giving behaviour and to get opinions on the mobile giving app. This project consists of 2 entities and a system. In this system the users are portrayed as donors who must enter their information into the system. Following that, the user could log in to the system and carry out their tasks in accordance. Then the user can, for

instance, read the details of any charitable events that are mentioned in the database. In order to ensure the charitable organization could readily collect the products, the donation order is been placed in the system with a specific location selected. Additionally, individuals were able to edit their profile details, including their password, nickname, phone number, and address. The 2 main functions of the user in this system is as follows that is the users who click on the event URL provided by the system are led to the official event website where they may learn more about the event. In addition, the system saved the user's donation order to the database. Following that, the charity had gathered the goods based on the database. After the task was finished, the non-profit updated the database, and the system informed the users of the outcome and reward point. There is a register page where users fill their data and register themselves. Some of the features are like users can view donation history, Charitable organizations can view their orders and also their completion donation order record. Some of the disadvantages of this system are like. The absence of the Contractor entity may result in ineffective donations from the Donors. Since blockchain is not used to store data, there is a danger that it will be altered.

Chang Liu et al. [5] in "A Privacy-Preserving and Overhead-Free Protocol for Direct Donations to People Impacted by COVID-19 Lockdowns" says that there were numerous large-scale, protracted lockdowns brought on by COVID-19 throughout the world and folks who require financial or other assistance, Governments, non-profit organization, and localities sprang into action to help. To directly support those who may not be able to get enough or timely aid the author created a "Fireside "which is a privacy- preserving, overhead-free protocol which helps for bringing together donors and those in need. In this system Donations to Fireside Help are exchanged directly between contributors and those in need. There are no recurring applications from other parties, including the Fireside Help system itself (code is recorded, but real information if not). In this system the donor is provided with the donor recipient's payment QR code, and update chosen donor name in the system, and both the parties keep the information as a secret. In this system there is a verifier who publicizes the contact details so donors and recipients can contact him/her and a code to prevent double-dipping. The code is created using recipient's name, school, and class, to permit precise, need-matching, and targeted donations. Some of the disadvantages of the system is that the Efficient donation from users cannot be achieved, in this system after each donation is completed the relevant transaction will be deleted and no Government officials involved so the NGOs are not verified.

Ashutosh Ashish Khanolkar et al. [6] in "Blockchain based Trusted Charity Fund- Raising" have briefed about how charity has evolved day by day. Author says that charity is a rapidly expanding industry today that has moved away from traditional organizational models and toward a decentralized crypto-currency-based one. Author also tells us about the fake charities that take advantage of our generosity. Author also talks about fake charities; he says false charities aim to profit from our compassion and altruism toward those in need. In order to get your money, scammers will pose as legitimate charities and these frauds steal funds from legitimate charities and causes, which is in addition to costing you money. In this paper the three key actors in the proposed system are the donor, the charity organization, and the needy person. A needy person can ask charitable organizations for the necessary quantity of money and donors can view all the fund requests made by charitable organizations. Donors can then select a certain cause and a donation amount from a list. These transactions are recorded in a blockchain ledger. The status of the charity events can be checked in this system. In this system every time a transaction takes place the transaction will be sent to minor node. After this the verification will take place. So, at the end the donors and the charity are the clients who basically do the transaction on the blockchain so that each transaction will provide transparency between clients.

III. REQUIREMENTS AND SPECIFICATIONS

A software requirement Specification (SRS) is a finished portrayal of the conduct of the framework to be created. It incorporates a lot of utilization cases that depicts all the collaborations that clients will have with the product.

A. Blockchain Technology

Blockchain is a method of recording information that makes it impossible or difficult for the system to be changed, hacked, or manipulated. A blockchain is a distributed ledger that duplicates and distributes transactions across the network of computers participating in the blockchain. Blockchain technology is a structure that stores transactional records, also known as the block, of the public in several databases, known as the "chain," in a network connected through peer-to-peer nodes. Blockchain supports immutability, which means that data once is written cannot be erased or replaced. Immutability means that no data tampering is possible within the network. Blockchain helps in the verification and traceability of multistep transactions needing verification and traceability. It can provide secure transactions, reduce compliance costs, and speed up data transfer processing. Blockchain technology can help contract management and audit the origin of a product. It also can be used in voting platforms and managing titles and deeds. Typically, this storage is referred to as a 'digital ledger.' Every transaction in this ledger is authorized by the digital signature of the owner, which authenticates the transaction and safeguards it from tampering. Hence, the information the digital ledger contains is highly secure. In simpler words, the digital ledger is like a Google spreadsheet shared among numerous computers in a network, in which, the transactional records are stored based on actual purchases. Blockchain could be a data structure that could be a growing list of information blocks. The knowledge blocks area unit coupled along, such recent blocks cannot be removed or altered. The blockchain is a distributed database of records of all transactions or digital event that have been executed and shared among participating parties. Each transaction verified by most participants of the system. It contains every single record of each transaction. Bitcoin is the most popular cryptocurrency

International Scientific Journal of Engineering and Management ISSN: 2583-6129
Volume: 02 Issue: 05 | May – 2023 DOI: 10.55041/ISJEM00748 www.isjem.com
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

an example of the blockchain. Blockchain is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a business network. An asset can be tangible (a house, car, cash, land) or intangible (intellectual property, patents, copyrights, and branding). Virtually anything of value can be tracked and traded on a blockchain network, reducing risk, and cutting costs for all involved.

B. Smart Contract

Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. A smart contract is a computer program or a transaction protocol which is intended to automatically execute, control or document legally relevant events and actions according to the terms of a contract or an agreement. The objectives of smart contracts are the reduction of need in trusted intermediators, arbitrations and enforcement costs, fraud losses, as well as the reduction of malicious and accidental exceptions. Users will not hesitate to donate as the steps were verified by the Government and security will be provided during payment.

C. Solidity

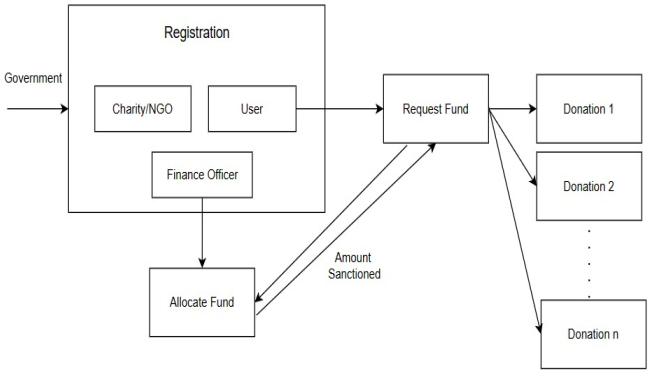
Solidity is an object-oriented, high-level language for implementing smart contracts on the Ethereum blockchain. Smart contracts are programs which govern the behavior of accounts within the Ethereum state. Solidity is a curly-bracket language. It is influenced by C++, Python and JavaScript, and is designed to target EVM known as Ethereum Virtual Machine. For Solidity, a dedicated IDE (Integrated Development Environment) known as Remix is used. On Remix, Smart contracts can be implemented and deployed at the same time. Solidity is statically typed, supports inheritance, libraries, and complex user-defined types among other features. With Solidity, contracts for use cases such as voting, crowdfunding, blind auctions, and multi-signature wallets. It's used to create smart contracts that implement business logic and generate a chain of transaction records in the blockchain system.

D. Ethereum

Ethereum is a flexible Blockchain platform which is open to use by everyone. This platform has a high level of security from different kinds of attacks. The users can create the Smart contracts and decentralized applications. The Ethereum network is made up of thousands of computers connected to each other through the Internet. Ethereum allows users to mine the cryptocurrency named Ether by creating smart contracts that verify every transaction and are recorded in a public blockchain. The remuneration in cryptocurrency compensates for the material and energy costs linked to the provision of this computing power.

E. Hardware Requirements

Following are the different hardware requirements with



respect to installation:

- Processor: Intel and above
- RAM: 4GB and Above
- Hard Disk: 10 GB of free space

Following are the different hardware requirements with respect to executing the system:

- Processor: Intel and above
- RAM: 4GB and Above
- Input Device: Standard keyboard and Mouse
- Output Device: Monitor
- Internet: Required

F. Software Requirements

Following are the different software requirements with respect to installation:

- Libraries: Node js, Web3 js
- Server: Eclipse
- Database: MySQL Workbench
- Local Blockchain: Ganache

Following are the different software requirements with respect to executing the system:

- Programming Language: Solidity, JavaScript
- Platform: VS Code
- Local Blockchain: Ganache
- Extension: MetaMask
- Server: Eclipse
- Database: MySQL Workbench
- Front end: Angular

IV. SYSTEM DESIGN

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. The purpose of the system design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural

International Scientific Journal of Engineering and Management
Volume: 02 Issue: 05 | May – 2023 DOI: 10.55041/ISJEM00748
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
www.isjem.com

entities as defined in models and views of the system architecture.

A. Architectural Design

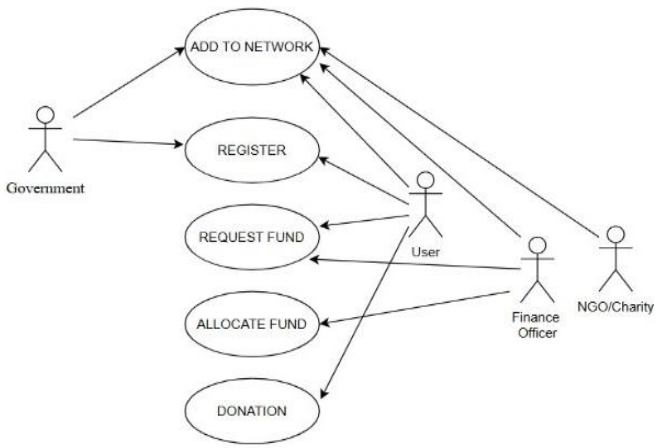
Architectural design is a creative process where you try to establish a system organization that will satisfy the functional and non-functional system requirements. Because it is a creative process the activities within the process differ radically depending on the type of system being developed, the background and experience of the system architect, and the system requirements for the system. An architecture diagram is a network map used to describe the general structure of a software program as well as the interactions, restrictions, and limits between elements.
The architectural design of this project has 4 entities, they are:

- Government
- Charity/NGO
- Finance Officer
- User

Here Government does the registration of Charity/NGO, Finance Officer, and the Users. Finance officer allocates the fund to the system as well as to the user's wallet. Users can donate to Charity/NGO using the money from their wallet hence the users have to request for a certain amount for their wallet which will be added by the Finance Officer. After that Users can donate to any number of Charity/NGO's.

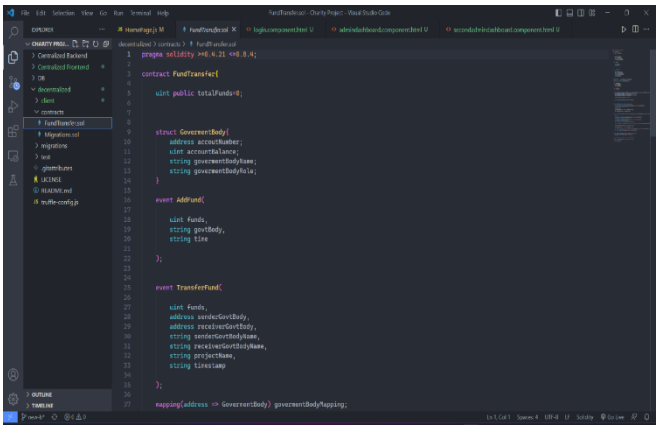
B. Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and tells how the user handles a system. A system involves a set of use cases and set of actors. The whole use case diagram is designed in a user-friendly manner. The set of use cases show the complete functionality of the system at some level of detail. Similarly, each actor represents one kind of object for which the system can perform the behavior. The set of actors represents the complete set of objects that the system can serve. Figure 4.2 shows the Use case diagram.



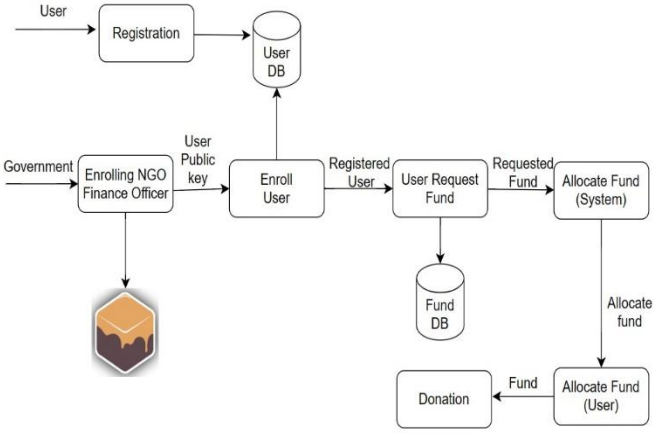
Following points talks about the work of each user:

- Government is mainly used to add the individuals to the network (User, Finance Officer, NGO/Charity)
- User is the one who will donate. He/she can register



themselves, request for fund to their wallet and do the donation.

- Finance Officer is the who will add fund to the system and to the user's wallet.
- NGO/Charity will receive the donations.



C. Data Flow Diagram

Data flow diagram is the graphical representation of the flow of the data through the information system. A data flow diagram can also be used for visualization of data processing. It is common practice for a designer to draw context-level DFD first which shows the interaction between the system and outer world entities. The context-level DFD is then exploded to show more details of the system being modelled. A DFD represents flow of data in the system. It views the system as a function that transforms the input to desired output. Data flow diagrams can be used to provide the end users with a physical idea of where the data they input ultimately influences the structure of the whole system. DFD shows the detailed flow of each sequence of operation that takes place in every step of the program. Figure 4.3 shows us the data flow between different process.

Initially Users will register by creating a new account where they need to add their public address (taken from ganache). The registration details of the user will be stored in the database. Next the Government will add the Charity/NGO and Finance Officer to the network. Using the public address of user taken from his/her registration, the Government will add the user to the network. Next, the registered users will log in to the system by using their credentials and request money for their wallet. Requested data by the user will be stored in the database. Depending upon the requested money the Finance Officer will first the fund to the system and after that to the user wallet. Now the user will receive the requested fund and using this user can donate to any number of Charities/NGOs.

V. IMPLEMENTATION

This system contains four entities government, finance officer, donor, NGO. Each of these entities needs and account with wallet address and some ether by linking accounts to metamask. To code our application Ethereum blockchain allows us to execute code with Ethereum Virtual machine (EVM) on blockchain with smart contract. In our application Smart contracts are responsible of reading and writing data to the blockchain as well as executing the logic. Smart contracts are written in programming language called Solidity. If the public ledger represents database layer of the blockchain, then smart contracts are where all the business logic that transacts with that data lives.This system contains four entities government, finance officer, user and Charity/NGO. Each of these entities needs an account with wallet address and some ether by linking accounts to metamask. To code our application Ethereum blockchain allows us to execute code with Ethereum Virtual machine (EVM) on blockchain with smart contract. In our application Smart contracts are responsible of reading and writing data to the blockchain as well as executing the logic. Smart contracts are written in programming language called Solidity. If the public ledger represents database layer of the blockchain, then smart contracts are where all the business logic that transacts with that data lives.

A smart contract is a self-executing contract with the terms of the agreement between the buyer and seller being directly written into lines of code. In the case of Solidity, the smart contract language used on the Ethereum blockchain, a typical smart contract has the following structure.

• Pragma statement - This specifies the version of Solidity compiler to use.

• Contract declaration - This defines the contract and its name.

• State variables - These are variables that hold and maintain the state of the contract. They can be public or private.

• Structs - These are custom data types defined by the programmer that hold a collection of related data.

• Events - These are triggered by the contract to notify the external world of certain events or state changes within the contract.

• Mappings - These are key-value pairs that allow for efficient storage and retrieval of data in the contract.

• Functions - These are the operations that can be performed on the contract. They can be public or private, and can have input parameters and return values.

• Modifiers - These are used to modify the behavior of a function.

• Constructor - This is a special function that is executed only once when the contract is first deployed.

The structure of a smart contract can vary depending on its specific use case and requirements. Step first is to build our application is installing all the dependencies and then writing our contract and deploying it to the blockchain successfully. This is a Solidity smart contract that defines a FundTransfer system. The contract contains functions that allow government bodies to add funds to the system, allows Donors to transfer funds to any NGO they wish, and see their wallet balance simultaneously. To create the contract, declare the smart contract with the "fundtransfer" keyword.

pragma solidity >=0.4.21 <=0.8.4;

It specifies the version of Solidity compiler that the code should be compiled with. In this case, the code is written to be compatible with Solidity versions equal to or greater than 0.4.21, but less than or equal to 0.8.4. This means that the code should be able to compile using Solidity compilers in the specified range of versions. If the code is compiled with a different version of the Solidity compiler outside of this range, it may not work as expected.

contract FundTransfer{…}

This is the beginning of a Solidity smart contract named FundTransfer. The contract keyword is used to define a new contract in Solidity, and FundTransfer is the name of this particular contract.

uint public totalFunds=0;

The contract has a single state variable named totalFunds, which is of type uint (unsigned integer). The public keyword means that this variable can be read from outside the contract, but not modified. In this case, totalFunds is initialized with a value of 0, which means that there are no funds available in the contract at the start. This variable is used to keep track of the total amount of funds that have been deposited into the contract by the government bodies.

```
struct GovermentBody
{address accoutNumber;
 uint accountBalance;
 string govermentBodyName;
string govermentBodyRole;}
```

This is a struct declaration in Solidity. It defines a custom data type named GovermentBody that has four properties.
• address accoutNumber - This is an Ethereum address representing the government body's account number. Here the accountNumber is nothing but the public address of the GovernmentBody.

• uint accountBalance - This is an unsigned integer representing the current balance in the government body's account.

• string govermentBodyName - This is a string representing the name of the government body.

• string govermentBodyRole - This is a string representing the role of the government body (e.g. "Central", "State", etc.).

Together, these properties define a single entity or object in the smart contract that represents a government body, with its account number, balance, name, and role. The GovermentBody struct can be used in the contract to create instances of this custom data type for each government body that interacts with the contract.

Events-Events in smart contracts are a way to emit a signal or a message to the external world about the occurrence of a particular event in the contract. They are essentially log entries that are recorded on the blockchain and can be accessed by external applications to obtain information about the contract's behaviour.

Events are declared using the event keyword and are typically emitted within the body of a function using the emit keyword. These are four Solidity events that have been defined within the Fund Transfer contract.

```
AddFund event
event AddFund (
uint funds,
string govtBody,
string time );
```

In this smart contract, the AddFund event is emitted whenever funds are added to the contract. It has three parameters.

• funds - an unsigned integer indicating the amount of funds added.

• govtBody - a string indicating the name of the government body that added the funds.

• time - a string indicating the time when the funds were added

```
TransferFund event
event TransferFund(
uint funds,
address senderGovtBody,
address receiverGovtBody,
string senderGovtBodyName,
string receiverGovtBodyName,
string projectName,
string timestamp );
```

The TransferFund event is triggered when a transfer of funds occurs between two government bodies in the Fund Transfer contract. It includes the following information.

• funds - the amount of funds being transferred

• senderGovtBody - the Ethereum address of the government body that initiated the transfer

• receiverGovtBody - the Ethereum address of the government body that received the transfer

• senderGovtBodyName - the name of the government body that initiated the transfer

• receiverGovtBodyName - the name of the government body that received the transfer

• projectName - the name of the project for which the funds are being transferred

• timestamp - the timestamp of the transfer

```
AddedGovtBody event
event AddedGovtBody(
string govtBodyName,
address accNumber,
string govtBodyRole );
```

The AddedGovtBody event is emitted when a new government body is added to the smart contract. It includes the following parameters.

• govtBodyName - a string representing the name of the government body.

• accNumber - the Ethereum address of the government body's account.

• govtBodyRole - a string representing the role of the government body.

This event is useful for tracking the addition of government bodies to the smart contract.

```
YourBalance event
event YourBalance(
uint balance );
```

The Your Balance event is used to emit the current balance of a government body account in the Fund Transfer contract. It takes a single argument balance, which is the current balance of the account being queried. This event can be used to keep track of the balance of an account.

Mapping-In Solidity, a mapping is a key-value data structure, similar to a hash table or dictionary in other programming languages. Mappings are declared using the mapping keyword followed by the type of the key and the type of the value in parentheses. These are two mappings used in the Fund Transfer smart contract.

```
mapping(address => GovermentBody) govermentBodyMapping;
```

The first mapping govermentBodyMapping is used to store information about government organizations. The keys of this mapping are the Ethereum addresses of the government organizations, and the values are instances of the GovermentBody struct which stores information about the account number, account balance, name, and role of the government organization.

```
mapping(string => address) govtBodyNameMapping;
```

The second mapping govtBodyNameMapping is used to map government organization names to their Ethereum addresses. The keys of this mapping are the names of the government organizations, and the values are their corresponding Ethereum addresses. These two mappings provide a way to efficiently retrieve information about a government organization given either its Ethereum address or its name.

Functions -Function is a piece of code that can be called to perform a specific task or operation. It can take input arguments and can also return a value. The contract consists of several functions that are used to manage the transfer of funds between different government bodies, they are

Function addGovernmentBody- This function is used to add a new government body to the system.

```
function addGovernmentBody(address accNumber,string memory govtBodyName,string memory govtBodyRole) public {
    govermentBodyMapping[accNumber].accoutNumber =accNumber;

govermentBodyMapping[accNumber].govermentBodyName = govtBodyName;

govermentBodyMapping[accNumber].accountBalance=0;

govermentBodyMapping[accNumber].govermentBodyRole = govtBodyRole;
    govtBodyNameMapping[govtBodyName] = accNumber;
    emit AddedGovtBody(govtBodyName,accNumber,govtBodyRole);
  }
```

It takes three arguments:

• accNumber - The unique account number for the new government body.

• govtBodyName - The name of the new government body.

• govtBodyRole - The role of the new government body, which can be "Central" or "State".

Inside the function, the information about the new government body is stored in the govermentBodyMapping mapping. The accoutNumber, govermentBodyName, accountBalance, and govermentBodyRole fields are set for the given accNumber. The account balance is set to 0. It also sets the govtBodyNameMapping mapping with the government body name and its corresponding account number.

The function emits an event AddedGovtBody with the govtBodyName, accNumber, and govtBodyRole as parameters to indicate that a government body has been added.

Function addFunds - This function allows the central government to add funds to total funds available in the system. In other words, this function is used by Finance officer to allocate funds to the system.

```
function addFunds(uint funds,string memory time) public {

require((keccak256(abi.encodePacked(govermentBodyMapping[msg.sender].govermentBodyRole)) == keccak256(abi.encodePacked("Central"))));
    totalFunds+=funds;

govermentBodyMapping[msg.sender].accountBalance+=funds;
    emit AddFund(funds,"Central Government",time);
  }
```

It takes in two arguments: funds, which is the amount of funds being added, and time, which is a string indicating the time the funds are being added.

The function first uses a require statement to check that the govermentBodyRole of the msg.sender (i.e., the government

International Scientific Journal of Engineering and Management ISSN: 2583-6129
Volume: 02 Issue: 05 | May – 2023 DOI: 10.55041/ISJEM00748 www.isjem.com
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

body invoking the function) is equal to "Central". In this system "Central" governmentBodyRole is associated with Finance Officer who adds the funds to the system. If this condition is not met, the function will revert and the funds will not be added.

If the condition is met, the function adds the funds to the totalFunds variable and the accountBalance of the msg.sender in the govermentBodyMapping. It then emits an event AddFund with the details of the funds added, the name of the government body, and the time of the transaction.

Function transferFunds- This function transfers funds from the account of the government body calling the function to another government body's account.

```
function transferFunds(uint funds,string memory projectName,string memory govtBodyName,string memory timestamp) public {
require((
keccak256(abi.encodePacked(govermentBodyMapping[msg.sender].govermentBodyRole)) == keccak256(abi.encodePacked("Central"))) ||
      (
(abi.encodePacked(govermentBodyMapping[msg.sender].govermentBodyRole))
      == keccak256(abi.encodePacked("State"))));
      address fundRecevier = govtBodyNameMapping[govtBodyName];
      require((keccak256(abi.encodePacked(msg.sender)))!=(keccak256(abi.encodePacked(govtBodyNameMapping[govtBodyName]))));

govermentBodyMapping[fundRecevier].accountBalance+=funds;
      govermentBodyMapping[msg.sender].accountBalance-=funds;
 emit TransferFund(funds,msg.sender,fundRecevier,govermentBodyMapping[msg.sender].govermentBodyName,govermentBodyMapping[fundRecevier].govermentBodyName,projectName,timestamp);

      emit YourBalance(govermentBodyMapping[msg.sender].accountBalance);

 }
```

The function takes four parameters:

• funds - The amount of funds to transfer.

• projectName - A string representing the name of the project for which the funds are being transferred.

• govtBodyName - A string representing the name of the government body to which the funds are being transferred.

• Timestamp - A string representing the timestamp of the transfer.

The function first checks if the calling government body is either the "Central" or "State" government body using a require statement. It then retrieves the address of the government body to which funds are being transferred using the govtBodyNameMapping mapping. It checks that the sender and the receiver of funds are not the same entity using another require statement. The function then updates the account balances of the sender and receiver government bodies by adding and subtracting the transferred funds, respectively.

It emits an event to log the transfer details, including the sender and receiver government body names, the project name, and the timestamp. Finally, it emits another event to log the balance of the sender government body's account after the transfer.

Function getFundBalance -This function is used to get the account balance of a government body by providing its account address as input. It returns the account balance as an unsigned integer.

```
function getFundBalance(address fok) public payable returns (uint) {
uint mbal = govermentBodyMapping[fok].accountBalance;
return mbal; }
```

The function takes a single parameter, which is the account address of the government body. It retrieves the account balance of the government body from the govermentBodyMapping mapping and returns the account balance as an unsigned integer.

Function getTotalFundBalance -This function that returns the total amount of funds available in the system.

```
function  getTotalFundBalance() public view returns (uint){
return totalFunds; }
```

When called, this function will return the value of the totalFunds variable which keeps track of the total amount of funds in the system.

Setting Up Ganache- Ganache is a local development blockchain that can be used to mimic the behavior of a public blockchain. It will allow us to deploy smart contracts, develop applications and run tests. When you open Ganache for the first time, you'll see the home screen. On this screen you're prompted to load an existing workspace (if any exist), create a new custom workspace, or QuickStart a one-click blockchain with default options. For now, let's go with a

International Scientific Journal of Engineering and Management
Volume: 02 Issue: 05 | May – 2023 DOI: 10.55041/ISJEM00748
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
www.isjem.com

QuickStart workspace. Select the desired blockchain from the QUICKSTART drop down; you can choose to start an Ethereum node or Corda network, then click the QUICKSTART button.

Installing Node package manager- Once the local blockchain is running, we need to configure our environment for developing smart contract. npm is a package manager for the JavaScript programming language maintained by npm, Inc. npm is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry.

Setting up Truffle Framework - This provides a suite of tools for developing Ethereum smart contracts with Solidity programming language. Truffle also requires that you have a running Ethereum client which supports the standard JSON RPC API (which is nearly all of them). There are many to choose from, and some better than others for development.

Setting up Metamask Ethereum wallet- Most web browsers do not currently connect to blockchain networks, so we should install metamask to do so. Metamask will allow us to manage our personal account when we connect to the blockchain, as well as manage our Ether funds that we use to pay for transactions. After installing, click on the MetaMask icon on the top right corner of the chrome browser. It will open up MediMask UI, scroll all the way down and click Accept to agree MetaMask's terms of use. Then enter a password, confirm the password, and then click Create for a new Ethereum account.

Project Setup- Download the starter truffle project directory from the truffle. We can create smart contracts, test them and build front end application using web3.js, react.js and bootstrap all inside this project. Identify a use case and develop a technology plan. Develop a proof of concept. Administer a field trial, which involves a limited-production run with customer-facing data, and is stepped up to involve more customer-facing products and data volumes.

Front End- Write a code which creates a Home page, login page which allows user/government/finance officer to enter their username and password and also contains a register form where the users can register themselves. It also contains a dashboard for user, government and finance officer to proceed with their respective works. enters all the candidate details including the phase changes in the voting procedure.

In our application Smart contracts are responsible of reading and writing data to the blockchain as well as executing the logic. Smart contracts are written in programming language called Solidity. If the public ledger represents database layer of the blockchain, then smart contracts are where all the business logic that transacts with that data lives.

VI. RESULTS

This chapter shows the screenshots of our results. It contains the screenshots of all the pages in our project along with the screenshot of our local blockchain.

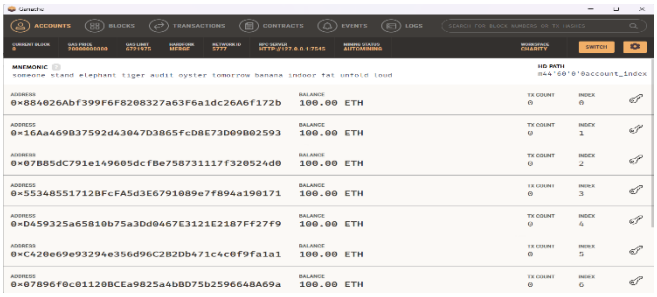
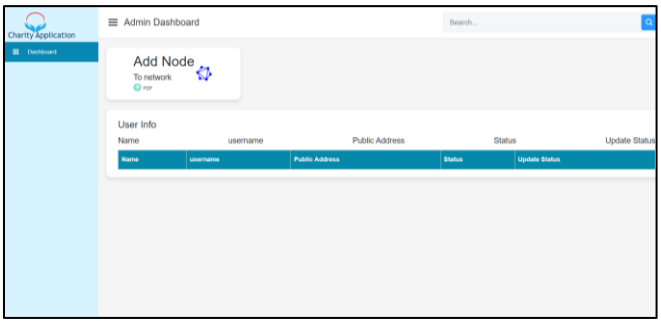


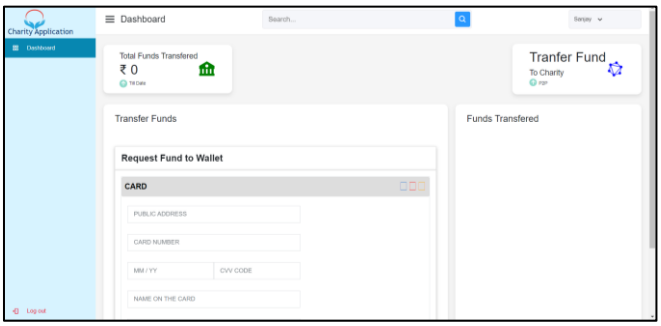
Figure shows the local blockchain that is used to store the data and enables us to access the private key of the Ethereum accounts. This provides 10 accounts each having 100 ethers.

This Figure shows the details of deploying the 2 contracts 'Migrations.sol' and 'FundTransfer.sol' onto the blockchain.

This Figure depicts the Homepage of our application. This page contains a table where all the detail of the transaction will be displayed.

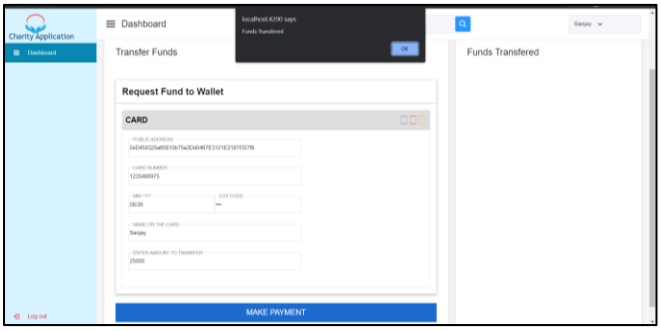


This Figure shows the Government dashboard. He is responsible for adding the users to the system. Here he can see all the registered users with their name, username, public address, status (added/not added) and an update status button.

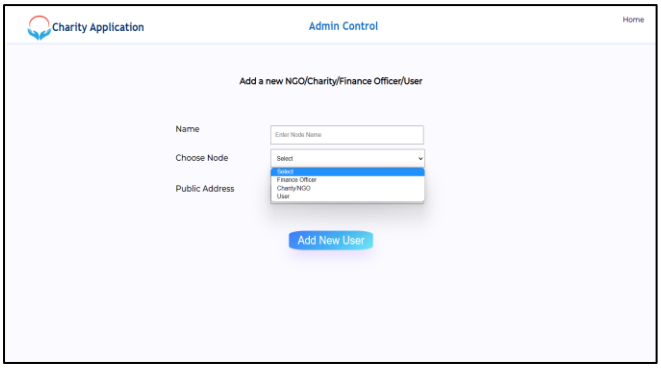


This Figure shows the User/Donor dashboard, he/she must first register as a User before logging in.

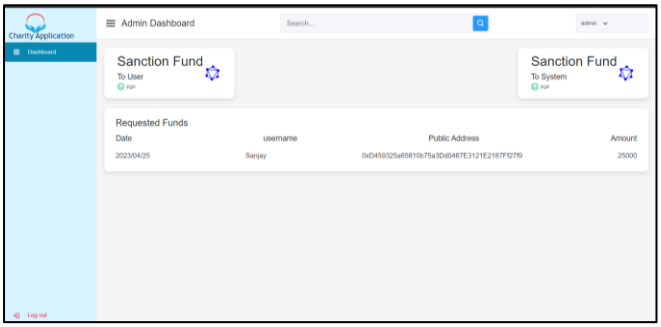
International Scientific Journal of Engineering and Management
Volume: 02 Issue: 05 | May – 2023 DOI: 10.55041/ISJEM00748
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
www.isjem.com



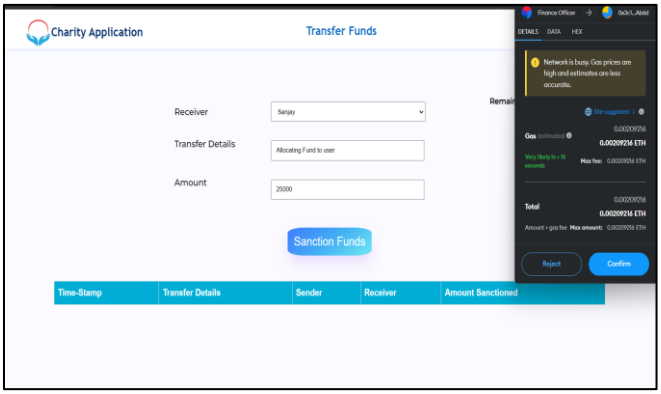
Then, the user must request for funds by filling in the details and sending a request as shown in above figure. The transfer details for his convenience will be displayed on the right side. He/she can also access the decentralized dashboard where he/she can donate to different Charities/NGOs.



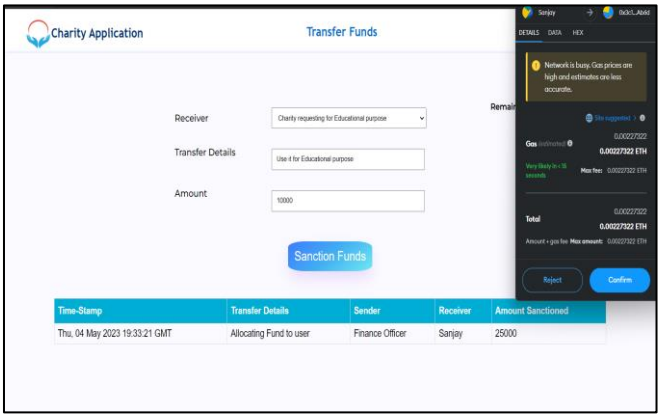
This Figure shows that Government adding nodes to the system. Here, the government is responsible for adding the nodes (Finance Officer, Charity/NGO and user) to Blockchain network. These nodes can be added to the system by utilizing public keys.



Finance officer is responsible for allocating funds to the system as well as to the user's wallet. Here, the Finance Officer will first allocate the funds to the system as shown in above figure.



Finance Officer can see the balance as shown in below figure and also allocate the fund as per the request sent by the user. The balance is shown according to the connected MetaMask account.



Above Figure shows the Users donation page where he/she can see the balance. Now the user can donate to different Charities/NGOs.

Above Figure shows the homepage that has been updated after each transaction. In this way it provides transparency to the public.

VII. CONCLUSION

Transferring money to charity using blockchain technology can provide a more transparent, secure, and efficient way of ensuring that donations reach their intended recipients. The use of blockchain technology can also enable more efficient and faster distribution of funds, reducing administrative costs and ensuring that donations are put to good use as quickly as possible. Overall, the use of blockchain technology in

charitable giving has the potential to revolutionize the way we donate, making it more accountable and impactful.

Charity chain uses Smart contracts to perform the process of donations and track them. This will provide transparency in the donations will ultimately motivate the donor to contribute more to such flexible yet efficient charities. Blockchain technology has the potential to transform the way we donate to charity. By providing a decentralized, transparent, and secure platform for fund transfers, blockchain can help increase donor trust, reduce fraud and corruption, and ensure that funds are used for their intended purpose. The use of blockchain technology for fund transfer to charities has several potential benefits, such as transparency, immutability, security, and traceability. These benefits can help increase donor trust, reduce fraud and corruption, and ensure that funds are used for their intended purpose.

By leveraging smart contracts, blockchain-based platforms can automate the verification and distribution of funds, reducing administrative costs and increasing efficiency. Additionally, blockchain can enable micropayments and facilitate cross-border transactions, making it easier for people around the world to donate to charitable causes.

VIII. FUTURE SCOPE

The future scope of charity applications to transfer funds using blockchain is significant. As the adoption of blockchain technology continues to grow, more charities and donors are likely to recognize the potential benefits of using blockchain for fund transfers. Below are some potential areas of growth and innovation in the future of charity applications using blockchain:

- Increased Adoption: As blockchain technology becomes more mainstream, more charities and donors are likely to adopt blockchain-based platforms for fund transfers. This increased adoption could lead to greater transparency, efficiency, and security in the charity sector, ultimately benefiting the organizations and the beneficiaries they serve.
- Tokenization: One area of potential growth in the future of charity applications using blockchain is tokenization. Tokenization refers to the process of creating digital tokens that represent a particular asset or value. In the context of charity applications, tokens could represent donations, enabling donors to easily track their contributions and charities to efficiently distribute funds to beneficiaries.
- Decentralized Autonomous Organizations (DAOs): Decentralized Autonomous Organizations (DAOs) are organizations that operate autonomously on a blockchain-based platform, using smart contracts to automate decision-making and governance processes.
- Cross-border Transactions: Blockchain technology has the potential to enable more efficient and cost-effective cross-border transactions, making it easier for people around the world to donate to charitable causes. As blockchain technology becomes more scalable and interoperable, it is likely that more charities will be able to leverage this technology to reach a global audience.
- The future scope of charity applications is to transfer funds using blockchain is significant. As blockchain technology continues to evolve and become more mainstream, it is likely that more charities and donors will recognize the potential benefits of using this technology for fund transfers. From tokenization to DAOs to cross-border transactions, there are many areas of growth and innovation in the future of charity applications using blockchain.

REFERENCES

[1] Singh, A., Rajak, R., Mistry, H. and Raut, P., 2020, June. Aid, charity and donation tracking system using blockchain. In 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184) (pp. 457-462). IEEE.

[2] Ajay Manohar Ingle, Ankur Singh, Pooja Gopal Rathod, Deepali Shrikhande in "E-Charity Platform for Social Welfare". (2021)

[3] Saleh, H., Avdoshin, S. and Dzhonov, A., 2019, November. Platform for tracking donations of charitable foundations based on blockchain technology. In 2019 Actual Problems of Systems and Software Engineering (APSSE) (pp. 182-187). IEEE.

[4] Mon, C.S., Cheng, K.Y. and Shibghatullah, A.S., 2020, April. Mobile application: donate day. In Journal of Physics: Conference Series (Vol.1529,No. 3, p. 032022). IOP Publishing.

[5] Liu, C. and Liu, L., 2020. A privacy-preserving and overhead-free protocol for direct donations to people impacted by COVID-19 lockdowns. In 2020 IEEE Global Humanitarian Technology Conference (GHTC) (pp. 1-2). IEEE.

[6] Ashutosh Ashish Khanolkar, Ashish Rajendra Gokhale, Amrish Sanjay Tembe, Vinayak Bharadi "Blockchain based Trusted Charity Fund-Raising". (July 2020)

[7] Dema, C., Mongar, V., Zangmo, S., Dema, K. and Wangchuk, T., 2017, October. Feasibility study to eradicate poverty in Bhutan through ICT. In 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON) (pp. 345-350). IEEE.

[8] Abhijeet Khamkar, Prathamesh Kotwal, Anand Khatri in "Charity Chain – A Blockchain Based Charity Application". (Dec 2021)

[9] Francesco Restuccia, Alvatore D'Oro, Salil S. Kanhere, Tommaso Melodia and Sajal K. Das in "Blockchain for the Internet of Things: Present and Future" (Jan 2018)

[10] Amir a. Khwaja1, Adnan nadeem, Muhammad Mumtaz bhutta, hafiz Farooq ahmad, Muhammad khurram khan, Moatiz a. Hanif in "A Survey on Blockchain Technology: Evolution, Architecture and Security" (April 2021)

[11] Nofer, Michael et al. "Blockchain. Business & Information Systems Engineering" 2020.

[12] Muhammad Nasir et al "A Survey on Blockchain Technology 2019.

[13] Thippa Reddy Gadekallu et al. in "Blockchain Technology And Decentralized Governance: Is The StateStill Necessary. 2020

[14] Hang, L et al in "Design and implementation of an integrated iot blockchain platform". 2021

[15] Prashanth P. Bungale at al. in "A Comparitive Analysis on E-Voting System Using Blockchain." 2019

[16] Mukhopadhyay, U. et al. in "A brief survey of cryptocurrency system" 2020

[17] Suma, V et al. [17] in "security and privacy mechanism using blockchain " 2019

[18] Lu, Q., & Xu, X et al . [18] in "Adaptable blockchain-based systems: A case study for product traceability". 2021

[19] S. Nakamoto, et al. [19] in "Bitcoin: A peer-to-peer electronic cash system" 2020.

[20] Xu, X et al. [20]. in "A taxonomy of blockchain-based systems for architecture design". 2021