# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"JnanaSangama", Belgaum -590014, Karnataka.**

**BLOCKCHAIN**
**AAT REPORT**
**on**

# Money Donation DApp

***Submitted by***

**Kanjika Singh(1BM21CS086)**
**Neha Bhaskar Kamath(1BM21CS113)**

***Under the Guidance of***
**Prof. Namratha M**
**Assistant Professor, BMSCE**

***in partial fulfillment for the award of the degree of***
**BACHELOR OF ENGINEERING**
***in***
**COMPUTER SCIENCE AND ENGINEERING**

**B.M.S. COLLEGE OF ENGINEERING**
**(Autonomous Institution under VTU)**
**BENGALURU-560019**
**March 2024 to June 2024**

**B. M. S. College of Engineering,**
**Bull Temple Road, Bangalore 560019**
(Affiliated To Visvesvaraya Technological University, Belgaum)

**Department of Computer Science and Engineering**

## CERTIFICATE

This is to certify that the AAT work entitled "**Money Donation DApp**" is carried out by **Kanjika Singh (1BM21CS086)), and Neha Bhaskar Kamath (1BM21CS113))** who are bonafide students of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visveswaraya Technological University, Belgaum during the year 2023-2024. The AAT report has been approved as it satisfies the academic requirements in respect of **Blockchain (22CS6PCBLC)** work prescribed for the said degree.

Signature of the Guide
Prof. Namratha M
Assistant Professor
BMSCE, Bengaluru

Signature of the HOD
Dr. Jyothi S Nayak
Prof.& Head, Dept. of CSE
BMSCE, Bengaluru

# B.M.S. COLLEGE OF ENGINEERING

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING


### DECLARATION

We, **Kanjika Singh and Neha Bhaskar Kamath,** students of 6th Semester, B.E, Department of Computer Science and Engineering, B. M. S. College of Engineering, Bangalore, hereby declare that, this AAT entitled "Money Donation DApp" has been carried out by us under the guidance of Prof. Namratha M, Assistant Professor, Department of CSE, B. M. S. College of Engineering, Bangalore during the academic semester March 2024 to June 2024.

We also declare that to the best of our knowledge and belief, the development reported here is not from part of any other report by any other students.

# Table of contents

# 1 Introduction

## 1.1 Problem statement

Traditional donation mechanisms often lack transparency, making it challenging for donors to track the utilization of their contributions. Our algorithm utilizes blockchain's immutable ledger to create a transparent and auditable record of donations. Ethereum is chosen as the platform due to its robust smart contract capabilities and widespread adoption within the blockchain community. Moreover, Ethereum's mature ecosystem offers a wide range of development tools, libraries, and resources, facilitating rapid development and deployment of decentralized applications. Additionally, Ethereum's large and active developer community ensures ongoing innovation and support, providing confidence in the platform's long-term viability. Furthermore, Ethereum's native cryptocurrency, Ether, provides a liquid and widely accepted medium of exchange, simplifying the process of transferring funds within the ecosystem.

## 1.2 Motivation

Motivated by a desire to transform charitable giving, we embarked on this endeavor to redefine the donation landscape. By harnessing blockchain technology, we seek to eliminate intermediaries, reduce transaction costs, and foster trust between donors and beneficiaries.

## 1.3 Work highlights

Our decentralized application (DApp) serves as a streamlined platform facilitating monetary donations, ensuring transparency and accountability in the process. Leveraging cutting-edge tools like Hardhat for smart contract development, the Sepolia network for testing, and MetaMask for seamless interaction with Ethereum wallets, our DApp offers a user-friendly experience. Hardhat provides a robust environment for compiling, testing, and deploying smart contracts, ensuring the reliability and security of our platform. The Sepolia network offers a testing environment for Ethereum applications, enabling developers to test their DApps in a simulated blockchain environment. MetaMask acts as a bridge between users and the Ethereum blockchain, allowing for secure transactions and interaction with decentralized applications directly from web browsers. Together, these technologies empower our DApp to revolutionize charitable giving by providing a transparent, secure, and accessible platform for making monetary contributions.

## 2 Methodology

Our decentralized application (DApp) for facilitating monetary donations operates through a systematic process that ensures transparency, security, and user-friendliness.

**Smart Contract Implementation:**

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.5.0 <0.9.0;

contract donation {
    struct Memo {
        string name;
        string message;
        uint256 timestamp;
        address from;
    }

    Memo[] memos;
    address payable owner;

    constructor() {
        owner = payable(msg.sender);
    }

    function donate(string memory name, string memory message) public payable {
        require(msg.value > 0, "Please pay greater than 0 ether");
        owner.transfer(msg.value);
        memos.push(Memo(name, message, block.timestamp, msg.sender));
    }

    function getMemos() public view returns (Memo[] memory) {
        return memos;
    }
}
```

Fig 1.1 Smart contract code

The smart contract begins by defining a struct named "Memo." A struct is a user-defined data type in Solidity that allows for the creation of custom data structures. In this context, the struct "Memo" is created to encapsulate the details of each donation transaction. It consists of the following fields:

- name (string): This field stores the name of the sender/donor.
- message (string): This field stores any accompanying message provided by the sender along with the donation.
- timestamp (uint256): This field records the timestamp of when the donation transaction occurred. The timestamp is represented as a Unix timestamp, which counts the number of seconds that have elapsed since January 1, 1970.
- from (address): This field stores the Ethereum account address of the sender/donor.

After defining the struct "Memo," the contract declares additional state variables to manage donation transactions:

- memos (array of Memo): This array stores instances of the "Memo" struct, effectively creating a dynamic list of donation records. Each element in the array represents a

unique donation transaction, with its associated sender's name, message, timestamp, and Ethereum account address.
- owner (address payable): This variable stores the Ethereum account address of the contract owner. It is declared as "payable," indicating that the address is capable of receiving Ether transfers.

The contract includes a constructor function, which initializes the "owner" variable with the Ethereum account address of the contract deployer.

The contract contains a function named "donate," which enables users to make donations to the contract. The function accepts two parameters:

- name (string): The name of the sender/donor.
- message (string): Any accompanying message provided by the sender along with the donation. Additionally, the function is marked as "payable," indicating that it can receive Ether transfers.

The contract includes a function named "getMemos," which allows users to retrieve the list of donation memos stored in the contract. This function is marked as "view," indicating that it does not modify the contract's state. It returns the entire array of "memos," providing users with access to all donation records stored in the contract.

**Utilization of Sepolia Network:**

To facilitate efficient testing and development of our DApp, we utilize the Sepolia network. The Sepolia network serves as a testing environment for Ethereum applications, enabling developers to simulate blockchain interactions without incurring actual transaction costs. By deploying our smart contract on the Sepolia network, we can thoroughly test its functionality and ensure its robustness before deploying it on the main Ethereum network.

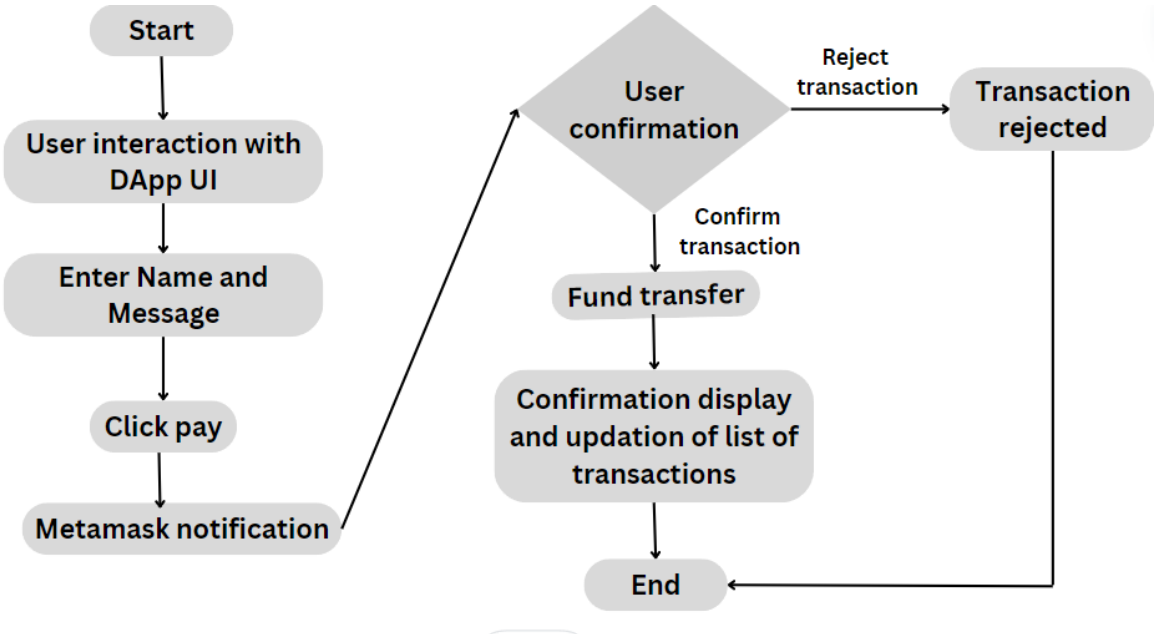
**Integration with Hardhat:**

Hardhat, a powerful Ethereum development environment, plays a crucial role in our development workflow. We utilize Hardhat to compile, deploy, and test our smart contracts. By running commands such as npx hardhat compile and npx hardhat run scripts/finalDeploy.js --network sepolia, we compile our Solidity code and deploy the contract on the Sepolia network. Hardhat streamlines the development process by providing tools for automated testing, debugging, and deployment, ensuring the reliability and efficiency of our DApp.

**Interaction with MetaMask:**

MetaMask serves as a bridge between our DApp and users' Ethereum wallets, enabling seamless interaction with the Ethereum blockchain. When users access our DApp's user interface, they can enter their name and a message and initiate a donation by clicking on the "Pay" button. Upon initiating a donation, MetaMask displays a notification, estimating the transaction's gas fees and allowing users to confirm the transaction. Once confirmed, MetaMask facilitates the secure transfer of funds, and users receive a confirmation of the completed transaction. Subsequently, users can reload the page to view their donation entry in the list of messages, along with the sender's name, timestamp, and message.

**Flowchart**


Start
User interaction with DApp UI
Enter Name and Message
Click pay
Metamask notification
User confirmation
Reject transaction
Transaction rejected
Confirm transaction
Fund transfer
Confirmation display and updation of list of transactions
End

## 3 Results and discussion

The effectiveness of our decentralized application (DApp) is clearly illustrated through a series of screenshots that capture key elements of the user experience and transaction process. These images highlight the user-friendly interface where individuals can input their name and message before proceeding with a donation. Next, the MetaMask notification screen displays transaction details, including gas fees and confirmation options, ensuring transparency and allowing users to maintain control. Upon confirmation, the screenshots show the successful transfer of funds and the immediate logging of the donation in the smart contract. This is further confirmed by the updated donation list visible upon refreshing the page, which shows the donor's name, message, timestamp, and Ethereum account address. These visual results demonstrate the effectiveness and accessibility of our DApp, showcasing the seamless integration of blockchain technology to enable secure and transparent donations.

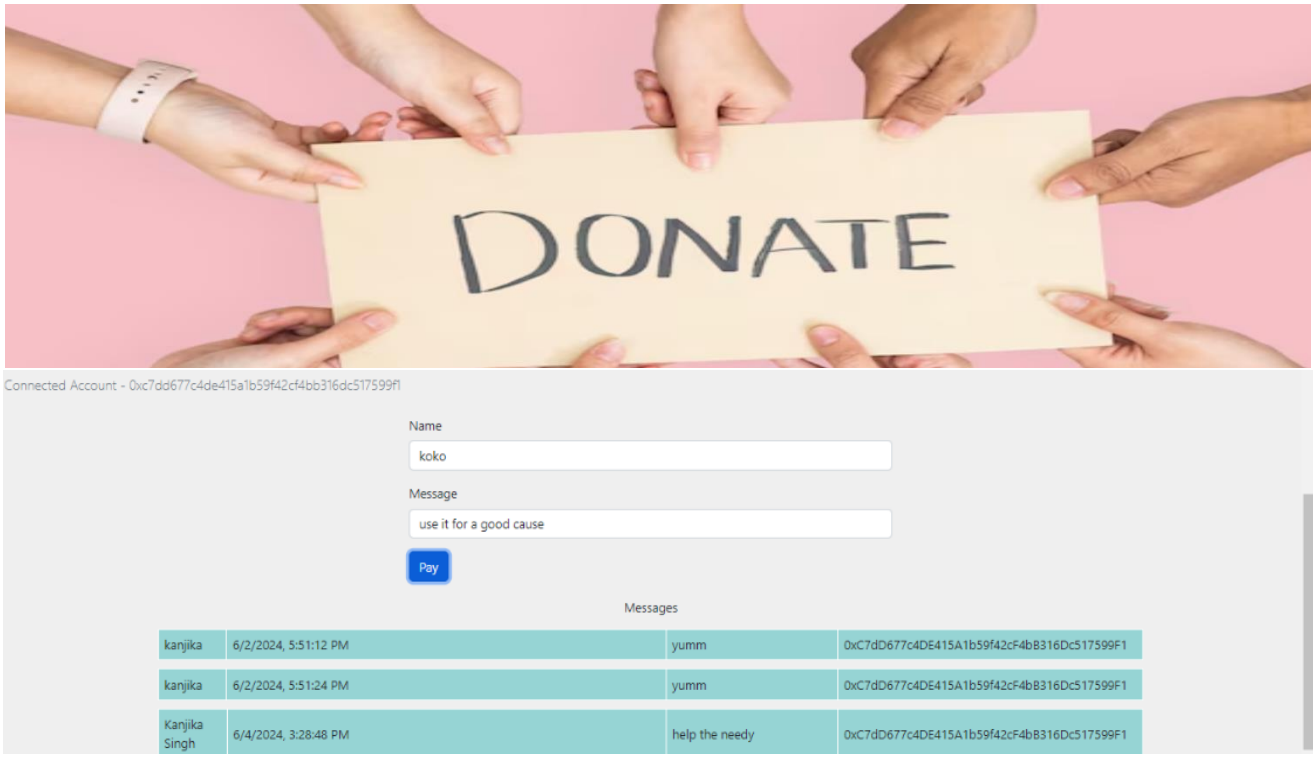



Fig 1.2 User interface

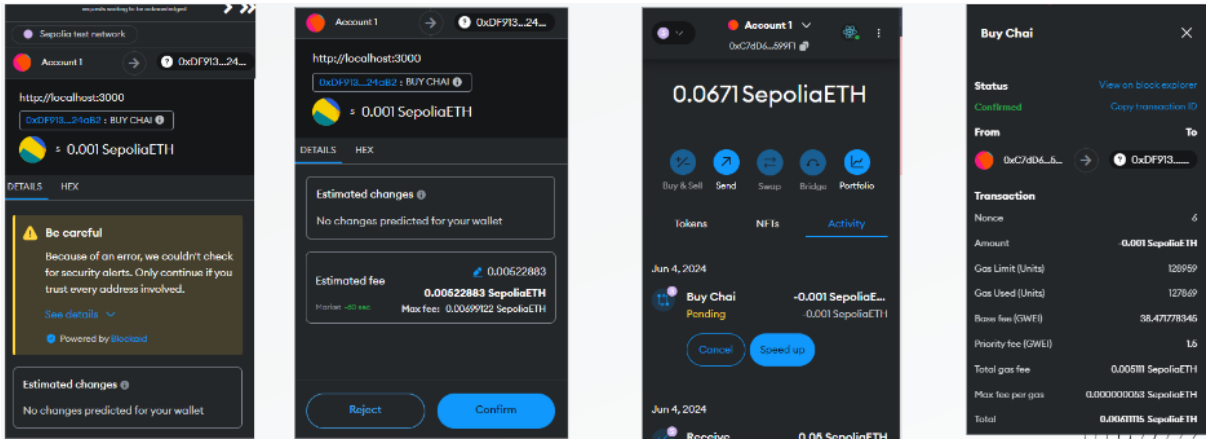



Fig 1.3 Transaction confirmation

| | | | |
|---|---|---|---|
| kanjika | 6/2/2024, 5:51:12 PM | yumm | 0xC7dD677c4DE415A1b59f42cF4bB316Dc517599F1 |
| kanjika | 6/2/2024, 5:51:24 PM | yumm | 0xC7dD677c4DE415A1b59f42cF4bB316Dc517599F1 |
| Kanjika Singh | 6/4/2024, 3:28:48 PM | help the needy | 0xC7dD677c4DE415A1b59f42cF4bB316Dc517599F1 |
| koko | 6/4/2024, 4:59:24 PM | use it for a good cause | 0xC7dD677c4DE415A1b59f42cF4bB316Dc517599F1 |

Fig 1.4 Updated messages

# 4 Conclusion

In conclusion, our decentralized application (DApp) represents a significant advancement in the realm of charitable donations, addressing many of the challenges faced by traditional donation systems. By leveraging blockchain technology, we have developed a platform that prioritizes transparency, security, and user-friendliness, fundamentally transforming how monetary contributions are made and tracked.

The core of our solution lies in the implementation of an efficient smart contract that meticulously records each donation, ensuring that every transaction is transparent and auditable. This approach not only fosters trust among donors but also provides recipients with a clear and immutable record of funds received.

The seamless integration with tools like Hardhat and MetaMask enhances the functionality and usability of our DApp. Hardhat, with its comprehensive suite of tools for smart contract development and testing, has allowed us to create a reliable and efficient backend infrastructure. MetaMask, on the other hand, facilitates a smooth and secure interaction between users and the Ethereum blockchain, enabling donors to contribute with ease and confidence.

Utilizing the Sepolia network for testing has been instrumental in ensuring that our platform operates flawlessly under various conditions. Our platform is designed to be intuitive, allowing users to make donations by simply entering their name and a message, and then confirming the transaction through MetaMask.

By combining the strengths of blockchain technology with user-centric design principles, our DApp not only enhances the donation process but also sets a new standard for transparency and accountability

## 5 References

https://cloud.google.com/application/web3/faucet/ethereum

https://www.youtube.com/watch?v=CdUDHj2i01Y

https://github.com/kshitijofficial