



**International Centre for Education and Research (ICER)  
VIT-Bangalore**

**BLOCKCHAIN BASED SECURE DONATION PLATFORM**

**CS7610 – CAPSTONE PROJECT**

**REPORT**

*Submitted by*

**SHANTANU VARTAK – 24MSP3045**

*In partial fulfilment for the award of the degree of*

**POST GRADUATE PROGRAMME**

**INTERNATIONAL CENTRE FOR HIGHER EDUCATION AND RESEARCH**

**VIT BANGALORE**

**June 2025**



**International Centre for Education and Research (ICER)  
VIT-Bangalore**

**BONAFIDE CERTIFICATE**

Certified that this project report “**BLOCKCHAIN BASED SECURE DONATION PLATFORM**” is the bonafide record of work done by “**SHANTANU VARTAK – 24MSP3045**” who carried out the project work under my supervision.

**Signature of the Supervisor**

**Signature of Director**

**Dr. Shiyamala Gowri**

**Prof. Prema M**

**Associate Professor,**

**Director,**

ICER

ICER

VIT Bangalore

VIT Bangalore.

**Evaluation Date: 20 | 06 | 2025**



**International Centre for Education and Research (ICER)  
VIT-Bangalore**

## **ACKNOWLEDGEMENT**

I express my sincere gratitude to our director of ICER **Prof. Prema M.** for their support and for providing the required facilities for carrying out this study.

I wish to thank my faculty supervisor, **Dr. Shiyamala Gowri, Professor**, ICER for extending help and encouragement throughout the project. Without his/her continuous guidance and persistent help, this project would not have been a success for me.

I am grateful to all the members of ICER, my beloved parents, and friends for extending the support, who helped us to overcome obstacles in the study.

## TABLE OF CONTENTS

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	<b>5</b>
	<b>LIST OF FIGURES</b>	<b>6</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>7</b>
<b>2</b>	<b>LITERATURE REVIEW</b>	<b>8</b>
<b>3</b>	<b>OBJECTIVE</b>	<b>9</b>
<b>4</b>	<b>PROPOSED METHODOLOGY</b>	<b>10</b>
<b>5</b>	<b>TOOLS AND TECHNIQUES</b>	<b>13</b>
<b>6</b>	<b>IMPLEMENTATION</b>	<b>14</b>
<b>7</b>	<b>RESULTS AND DISCUSSIONS</b>	<b>17</b>
<b>8</b>	<b>CONCLUSION</b>	<b>18</b>
<b>9</b>	<b>FUTURE ENHANCEMENT</b>	<b>19</b>
<b>10</b>	<b>APPENDICES</b>	<b>20</b>
	Appendix-1: Code – Full Code	20
	Appendix-2: Plagiarism Report	22
<b>11</b>	<b>REFERENCES</b>	<b>23</b>
<b>12</b>	<b>WORKLOG</b>	<b>24</b>

# **ABSTRACT**

Donate Now is a decentralized donation platform leveraging blockchain technology to ensure secure, transparent, and tamper-proof charitable contributions. Built using Solidity smart contracts and integrated with MetaMask for Ethereum transactions, the platform allows donors to fund verified campaigns directly from their wallets. An admin verifies campaigns and organizers, while smart contracts automate fund collection and release, enhancing trust and accountability. The application features real-time progress tracking, JWT-based authentication, and robust role-based access control. By combining blockchain transparency with user-friendly design, Donate Now aims to revolutionize digital philanthropy through innovation, integrity, and decentralized trust.

## LIST OF FIGURES

<b>Figure No.</b>	<b>Figure Name</b>	<b>Pg. No.</b>
Fig. 4.1	Methodology	10
Fig. 6.1	Donor Dashboard	14
Fig. 6.2	Organizer Dashboard	15
Fig. 6.3	Admin Dashboard	16

# **1. CHAPTER - 1**

## **INTRODUCTION**

Charity and donation platforms have long served as vital channels for delivering aid to individuals and communities in need. However, the traditional mechanisms behind these platforms often face significant challenges including lack of transparency, misuse of funds, delayed disbursements, and trust issues among donors. In the modern digital age, where transparency, security, and accountability are increasingly demanded by contributors, traditional systems fall short in delivering verifiable assurances that funds reach the intended recipients.

Blockchain technology, with its decentralized, immutable, and transparent nature, presents a promising alternative to the legacy systems that dominate the nonprofit and fundraising sectors. It allows transactions to be recorded on a public ledger that is secure, auditable, and tamper-proof, fostering a new era of trustless trust. This has led to the emergence of blockchain-based solutions for charitable giving and donation tracking, aiming to create a more ethical, efficient, and accountable ecosystem.

This project, titled "Donate Now", is a secure, transparent, and decentralized donation platform that leverages Ethereum blockchain and smart contracts (written in Solidity) to streamline the fundraising process for charitable campaigns. The platform enables organizers to create campaigns, receive donations in Ether (ETH), and ensures that fund disbursements occur only upon successful campaign completion. The donor interaction is secured via integration with MetaMask, which ensures every donation is authenticated, traceable, and on-chain.

Additionally, the project provides a modern full-stack web application developed using Next.js (frontend), Node.js/Express.js (backend API), Prisma ORM (MongoDB integration), and Ethers.js (for blockchain interaction). JWT-based authentication and role-based access controls ensure that organizers, donors, and administrators have clearly defined permissions, while campaign verification ensures platform integrity.

In essence, "Donate Now" not only demonstrates how blockchain can modernize traditional philanthropic models but also provides a practical solution that addresses the shortcomings of current digital donation systems.

## **2. CHAPTER - 2**

### **LITERATURE REVIEW**

#### **[1] Blockchain-Based Trusted Secure Philanthropy Platform: Crypto-GoCharity**

Crypto-GoCharity introduces a blockchain-based framework for secure and traceable philanthropic donations. The paper emphasizes the elimination of intermediaries and introduces smart contracts to ensure that donations are released only under pre-defined conditions. Similar to this project, Donate Now also incorporates conditional fund release and donor transparency. Crypto-GoCharity's use of Ethereum and MetaMask integration supports the feasibility and technical stack chosen for "Donate Now".

#### **[2] Blockchain-Based Decentralized Application: A Survey**

This survey paper outlines architectural designs, consensus mechanisms, and DApp use-cases across various sectors. It identifies donation platforms as a high-impact area where DApps can mitigate fraud and inefficiency. Our project aligns with these findings by using a DApp model to directly connect donors and beneficiaries, thereby minimizing operational overheads and enhancing accountability through smart contracts.

#### **[3] Blockchain-Based System for End-to-End Donations Monitoring**

This system proposes a blockchain solution for the end-to-end monitoring of donations — from contribution to final disbursement. It emphasizes the need for tracking and auditing each donation, which Donate Now achieves through transaction logs and immutable smart contracts. Furthermore, the system in the paper employs role-based transparency (donor, admin, beneficiary), which matches our implementation of role-driven access.

#### **[4] Aid, Charity and Donation Tracking System Using Blockchain**

This work addresses transparency and misuse issues in traditional charity systems using blockchain's immutable ledger. The solution demonstrates how blockchain can ensure every penny is traceable and only disbursed for verified purposes. "Donate Now" draws upon this by implementing campaign verification via an admin role and maintaining complete donation traceability using Ethereum's native ledger.



### **3. CHAPTER - 3**

## **OBJECTIVE**

To build an efficient charity system which would operate using decentralized technologies to transform the traditional ones into future-proof solutions to reduce Lack of Transparency, Security Concerns, Verification Issues, and Delayed Fund Access.

- To develop a secure and decentralized platform for donations using Ethereum smart contracts.
- To allow users (donors and organizers) to interact with the platform via MetaMask.
- To integrate role-based authentication using JWT tokens.
- To enable real-time donation tracking and campaign progress monitoring.
- To ensure admin-level moderation for verifying and approving campaigns before they go live.
- To implement secure, conditional fund releases upon meeting campaign goals.

## 4. CHAPTER - 4

# PROPOSED METHODOLOGY

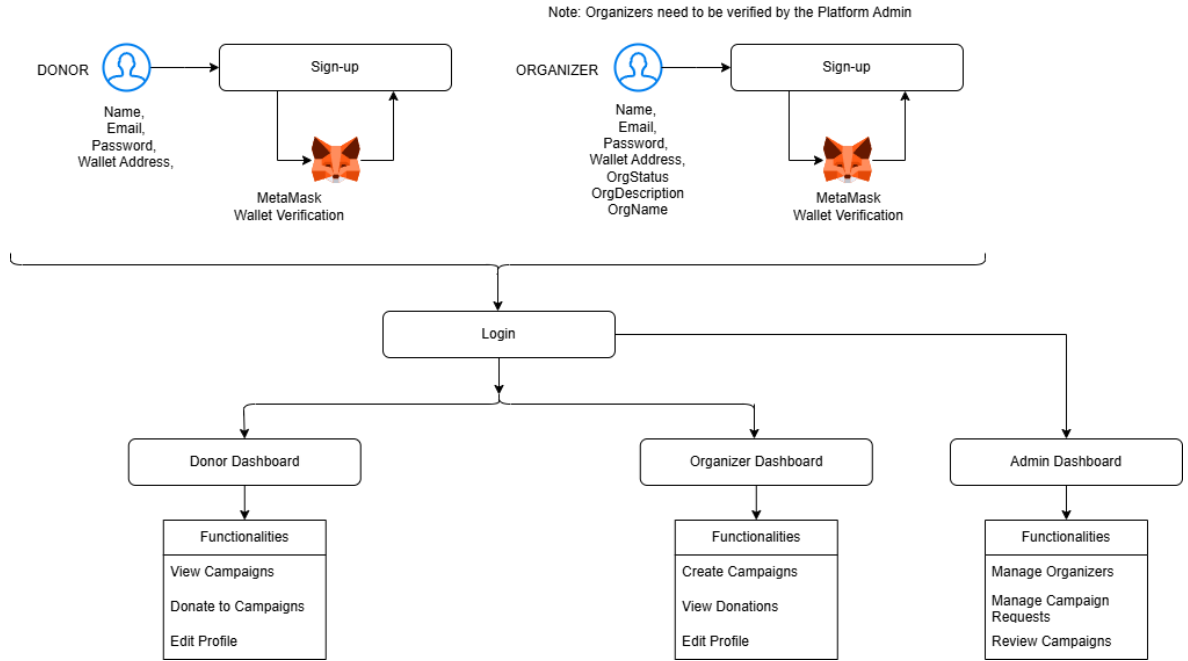


Fig 4.1 Methodology

The "Donate Now" platform is built upon a role-based blockchain-enabled web system that ensures secure, transparent, and efficient donation handling. The methodology encompasses three primary user roles: Donor, Organizer, and Admin, each following a distinct yet interconnected workflow. The platform combines user registration, MetaMask wallet integration, smart contract interaction, and dashboard-specific functionalities to deliver a seamless and accountable donation ecosystem.

### 4.1. Sign-Up Phase (Donor & Organizer)

- Both donors and organizers begin their journey with a registration (sign-up) process.
- During sign-up, users are required to link their MetaMask wallet. This Ethereum wallet is essential for any blockchain transaction, including donations and fund management.

- The user's wallet address is stored in the backend, and in the case of organizers, they are flagged as "unverified" until approved by an admin.

#### **4.2. MetaMask Integration**

- MetaMask is integrated as the authentication and transaction gateway for all blockchain operations.
- Donors use MetaMask to send Ether (ETH) to campaign contracts.
- Organizers use it to receive funds upon successful campaign completion.

#### **4.3. Login and Role Identification**

- After signing up, users log in using credentials.
- On successful authentication, the system decodes the JWT token to identify the role (donor, organizer, or admin) and routes the user to the respective dashboard.

#### **4.4. Dashboard Redirection**

Based on the decoded role, users are directed to one of the three dashboards:

##### **a) Donor Dashboard:**

- View all active campaigns.
- Donate to verified campaigns using MetaMask.
- View donation history and receipts.
- Track funding progress and campaign status.

##### **b) Organizer Dashboard:**

- Create new fundraising campaigns.
- Provide details like title, description, goal, timeline, and images.
- Await admin approval before campaigns go live.
- Monitor funds raised and receive them upon campaign success.

##### **c) Admin Dashboard:**

- View all user sign-up requests (especially organizers).

- Verify and approve or reject organizer accounts.
- Review submitted campaigns and approve or reject based on authenticity.
- Handle system-level audit and security monitoring.

#### **4.5. Smart Contract Operations**

The smart contract ensures:

- Donations are recorded immutably.
- Campaign funds are only released when the funding goal is met and marked complete by the admin.
- All transactions are traceable on the Ethereum blockchain.

#### **4.6. Functionalities**

Each dashboard presents functionalities relevant to the user's role:

- Donors can donate, track donations, and export receipts.
- Organizers can manage campaigns and monitor fund inflow.
- Admins serve as gatekeepers and verifiers, ensuring system integrity.

## 5. CHAPTER - 5

### TOOLS AND TECHNIQUES

There are several open-source tools used in this automation to get optimal results in a time efficient manner.

- **Solidity:** Solidity is the primary programming language for writing smart contracts on the Ethereum blockchain. It is used to implement campaign creation, donation tracking, fund release logic, and ensuring immutable records.
- **Remix IDE:** A browser-based IDE that allows developers to write Solidity code, compile, deploy, and test smart contracts directly on the Ethereum Virtual Machine (EVM) or using injected providers like MetaMask.
- **MetaMask:** A browser extension and mobile app that enables users to manage Ethereum-based wallets. It's used in this project to connect users (donors and organizers) to the blockchain for making and receiving ETH transactions.
- **Ethers.js:** A JavaScript library that interacts with Ethereum smart contracts from the frontend. It is used to call functions like `donate()` and `createCampaign()` and to read/write data from/to the blockchain.
- **Next.js:** A React-based full-stack framework used for building the frontend of the application. It offers server-side rendering, routing, and API support which makes it ideal for building dynamic web apps.
- **Node.js + Express.js:** Node.js is a runtime for executing JavaScript on the server, and Express.js is a minimalist framework built on top of it. Used for building RESTful APIs to handle user authentication, campaign management, and donation logging in the database.
- **Prisma ORM:** Prisma is an Object-Relational Mapping tool used to interact with MongoDB in a type-safe way. It abstracts complex queries and provides a clean interface for managing users, campaigns, and donations.
- **MongoDB:** A NoSQL document-oriented database used to store user profiles, campaign metadata, donation logs, and wallet associations. It integrates seamlessly with Prisma.
- **JWT (JSON Web Tokens):** Used to issue secure tokens on login. These tokens help identify users (donor, organizer, admin) and manage protected routes and role-based access in the app.
- **Tailwind CSS:** A utility-first CSS framework used for building responsive and custom-styled components quickly, ensuring a modern and clean UI.
- **ShadCN UI Components:** Pre-styled UI components built on top of Tailwind CSS. Used for rapidly building reusable UI features like modals, buttons, inputs, and badges.

## 6. CHAPTER - 6 IMPLEMENTATION

### 6.1. DONOR DASHBOARD:

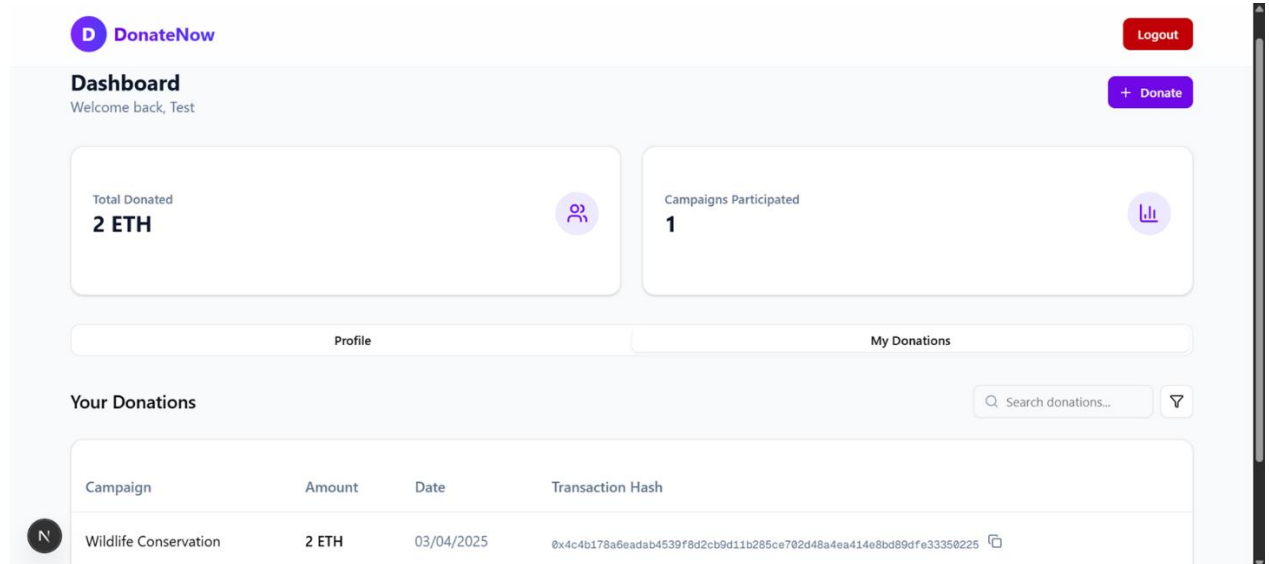


Fig 6.1. Donor Dashboard

In the above image, Donor can have interactive UI which also displays relevant details like Donation History, Total Amount Donated, Number of Donations, etc,

## 6.2. ORGANIZER DASHBOARD:

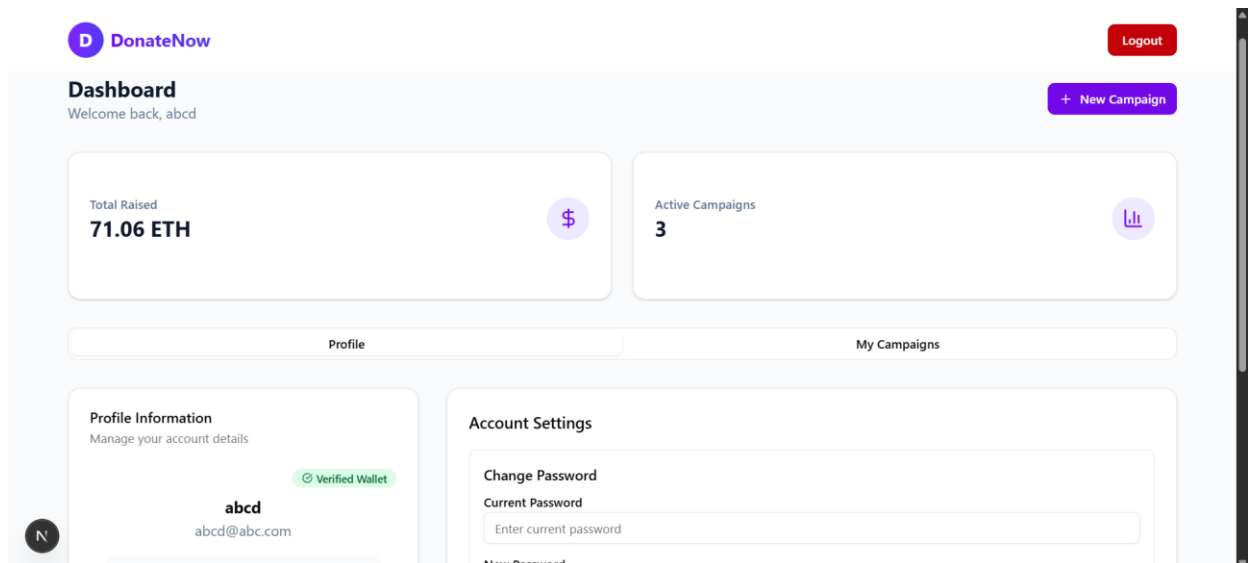


Fig 6.5. Organizer Dashboard

Similar to the Donor Dashboard, the Organizer Dashboard shows details related to the organizer and the organization which publishes the campaigns such as, Campaign History, Total Donations Collected, Active Campaigns, etc.

## 6.3. ADMIN DASHBOARD

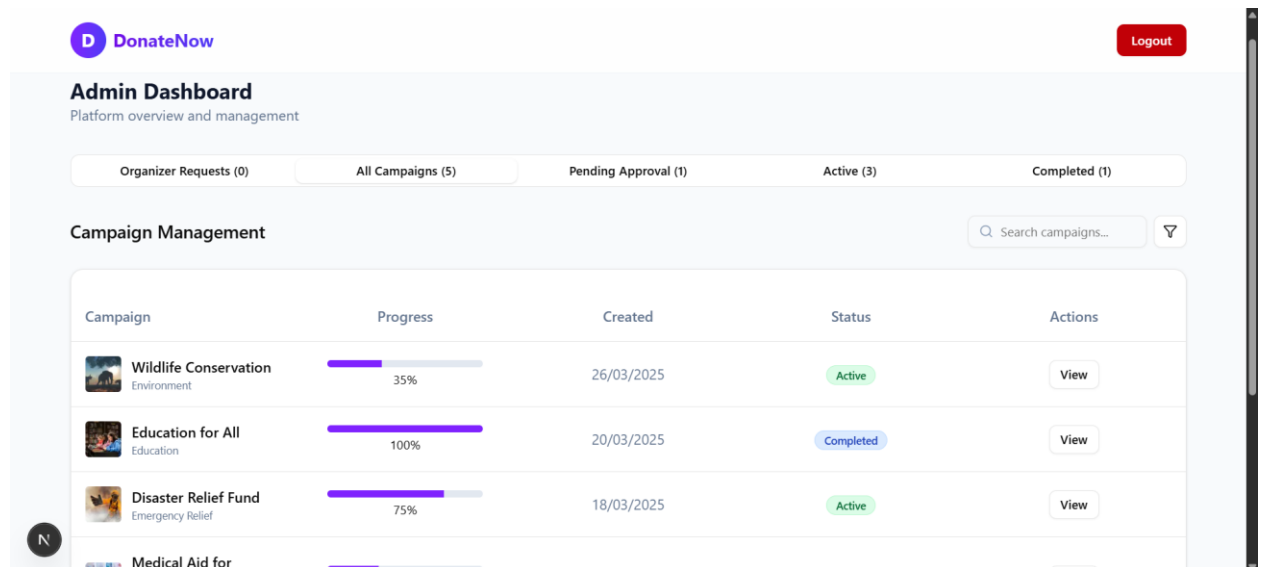


Fig 6.6. Admin Dashboard

For the Admin Dashboard, the image displays a managerial console which shows Organizer's Joining Requests, Full Campaign List, Pending/Active/Completed Campaigns, etc.



## **7. CHAPTER - 7**

### **RESULTS AND DISCUSSIONS**

The Donate Now platform successfully implements a decentralized and transparent donation system. The following functionalities were validated through systematic testing and simulation:

#### **7.1. User Role-Based Access Control**

- Donor: Can browse verified campaigns and donate ETH through MetaMask.
- Organizer: Can create campaigns, view donations, and track funds.
- Admin: Verifies organizers, approves campaigns, and monitors overall donations.

#### **7.2. Blockchain-Backed Donations**

Donations are processed via smart contracts ensuring:

- Immutable transaction records
- Gas-efficient ETH transfers
- Real-time on-chain tracking

#### **7.3. End-to-End Campaign Lifecycle**

- Campaigns move from creation → admin verification → listing → funding → completion.
- Smart contracts ensure funds are only released when campaign goals are met.

#### **7.4. Security Measures Implemented**

- JWT-based authentication secures API endpoints.
- Role-based routing protects unauthorized access to dashboards.
- Refresh token logic with popup reminders strengthens session security.
- Sensitive actions (fund release) restricted to verified admin roles.

#### **7.5. MetaMask Integration**

- Seamless integration for ETH payments using the Ethers.js library.
- Gas estimation and transaction confirmation handled via MetaMask popups.
- Transaction receipts, hashes, and status are recorded both on-chain and in the back end.

## **8. CHAPTER - 8**

### **CONCLUSION**

The Donate Now platform delivers a powerful and secure alternative to traditional donation systems by integrating blockchain technology, smart contracts, and role-based access control into one cohesive application. With the use of Solidity for on-chain logic, MetaMask for wallet integration, and Next.js + Express.js for a seamless user experience, the system ensures:

- Security: No centralized fund handling.
- Transparency: Every donation is recorded on-chain.
- Integrity: Funds are released only upon achieving goals.

This project demonstrates the viability of decentralized philanthropy and has the potential to disrupt conventional fundraising systems. Future enhancements such as multi-chain support, AI-based fraud detection, and automatic tax receipts generation can further improve the platform and align it with real-world humanitarian needs.

## **9. CHAPTER - 9**

### **FUTURE ENHANCEMENT**

- Multi-Currency Support: Add support for other cryptocurrencies like BTC, USDT.
- Zero-Knowledge Proofs (ZKPs): For privacy-preserving donations.
- Mobile App Integration: Build an Android/iOS version for greater accessibility.
- Advanced Analytics Dashboard: For both donors and organizers to track impact.
- Globalization: Add support for multiple languages and local fiat conversions.

# 10.CHAPTER - 10

## APPENDICIES

### 10.1 FULL CODE

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

contract SecureDonation {
    address public owner;
    uint256 public totalDonations;

    struct Donation {
        address donor;
        uint256 amount;
        uint256 timestamp;
    }

    struct Campaign {
        string name;
        address payable recipient;
        uint256 goal;
        uint256 fundsRaised;
        bool isCompleted;
    }

    mapping(uint256 => Campaign) public campaigns;
    mapping(uint256 => Donation[]) public donations;
    uint256 public campaignCount;

    event DonationReceived(address indexed donor, uint256 indexed
campaignId, uint256 amount);
    event FundsReleased(uint256 indexed campaignId, uint256 amount);

    modifier onlyOwner() {
        require(msg.sender == owner, "Not authorized");
        _;
    }

    constructor() {
        owner = msg.sender;
    }

    function createCampaign(string memory _name, address payable
_recipient, uint256 _goal) external onlyOwner {
        campaignCount++;
    }
}
```

```

        campaigns[campaignCount] = Campaign(_name, _recipient, _goal,
0, false);
    }

    function donate(uint256 _campaignId) external payable {
        require(msg.value > 0, "Donation must be greater than 0");
        require(_campaignId > 0 && _campaignId <= campaignCount,
"Invalid campaign ID");

        Campaign storage campaign = campaigns[_campaignId];
        require(!campaign.isCompleted, "Campaign is already
completed");

        campaign.fundsRaised += msg.value;
        totalDonations += msg.value;

        donations[_campaignId].push(Donation(msg.sender, msg.value,
block.timestamp));
        emit DonationReceived(msg.sender, _campaignId, msg.value);
    }

    function releaseFunds(uint256 _campaignId) external onlyOwner {
        Campaign storage campaign = campaigns[_campaignId];
        require(campaign.fundsRaised >= campaign.goal, "Goal not yet
reached");
        require(!campaign.isCompleted, "Funds already released");

        campaign.isCompleted = true;
        campaign.recipient.transfer(campaign.fundsRaised);
        emit FundsReleased(_campaignId, campaign.fundsRaised);
    }

    function getCampaignDetails(uint256 _campaignId) external view
returns (string memory, uint256, uint256, bool) {
        Campaign storage campaign = campaigns[_campaignId];
        return (campaign.name, campaign.goal, campaign.fundsRaised,
campaign.isCompleted);
    }
}

```

## **10.2 PLAGIARISM REPORT**

## 11. CHAPTER - 11

### REFERENCES

#### Journal References:



- [1] Swati, J., Nitin, P., Saurabh, P., Parikshit, D., Gitesh, P., & Rahul, S. (2022). *Blockchain based Trusted Secure Philanthropy Platform: Crypto-GoCharity*. 2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA), 1–8. <https://doi.org/10.1109/ICCUBEA54992.2022.10011026>
- [2] (PDF) *Blockchain-Based Decentralized Application: A Survey*. (2024). ResearchGate. <https://doi.org/10.1109/OJCS.2023.3251854>
- [3] (PDF) *Blockchain-Based System for End-to-End Donations Monitoring*. (n.d.). ResearchGate. <https://doi.org/10.1109/ACIT58888.2023.10453793>
- [4] Singh, A., Rajak, R., Mistry, H., & Raut, P. (2020). *Aid, Charity and Donation Tracking System Using Blockchain*. 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 457–462. <https://doi.org/10.1109/ICOEI48184.2020.9143001>











International Centre for Education and Research (ICER), VIT-Bangalore

**CAPSTONE PROJECT  
STUDENT'S WORKLOG SHEET**











Student Name: Shantanu Vartak  
Register Number: 24MSP3045  
Project Title: Blockchain Based Secure Donation Platform  
Domain: Cyber Security  
Tools Learned / used: Solidity, Ganache, Remix IDE



Day	Date	Task	Student's Remarks	Signature of Student with Date	Signature of the Project Mentor
Day 1	02/06/2025	Project domain discussion	Discussed Project Domain and Finalized Project Title	 02/06/25	



Day 2	03/06/2025	Zeroth Review	Successfully Completed Zeroth Review	 03/06/25		
Day 3	04/06/2025	Domain Research	Researched about Blockchain Technologies	 04/06/25		
Day 4	04/06/2025	Domain Research	Researched about Donation Campaigns and Schemes	 04/06/25		
Day 5	05/06/2025	Outlining Architecture	Drawing a Workflow of the Automation	 05/06/25		

Day 6	06/06/2025	First Review	Successfully Completed First Review	Dr. 06/06/25	Sg
Day 7	09/06/2025	Implementation	Implemented the working of Solidity Smart Contract	Dr. 07/06/25	Sg
Day 8	10/06/2025	Implementation	Integrated Ganache with Remix IDE for local access	Dr. 10/06/25	Sg
Day 9	11/06/2025	Implementation	Creating Backend and APIs for the Webapp	Dr. 11/06/25	Sg
Day 10	12/06/2025	Implementation	Integrating the Backend+APIs with the Smart Contract	Dr. 12/06/25	Sg

Day 11	13/06/2025	Second Review	Successfully Completed Second Review	 13/06/25	
Day 12	14/06/2025	Implementation	API Testing using Postman	 14/06/25	
Day 13	16/06/2025	Implementation	Implementing Frontend and connecting it to the Backend	 16/06/25	
Day 14	17/06/2025	Implementation	Testing the final Application	 17/06/25	
Day 15	18/06/2025	Third Review	Successfully Completed Third Review	 18/06/25	 18/6/25

Day 16	19/06/2025	Implementation	Preparing Final Capstone Project Report	<div> <div>  </div> <div> 19/06/25 </div> </div>
Day 17	20/06/2025	Final Review	Successfully Completed Final Review	<div> <div>  </div> <div> 20/06/25 </div> </div>